DE GRUYTER OPEN

Journal of ELECTRICAL ENGINEERING, VOL. 66, NO. 3, 2015, 159–163

IMPROVING CPA ATTACK AGAINST DSA AND ECDSA

Marek Repka [*] — **Michal Varchola** [**] — **Miloš Drutarovský** [**]

In this work, we improved Correlation Power Analysis (CPA) attack against Digital Signature Algorithm (DSA) and its various derivations, such as Elliptic Curve Digital Signature Algorithm (ECDSA). The attack is aimed against integer multiplication with constant secret operand. We demonstrate this improvement on 16-bit integer multiplier in FPGA. The improvement makes it possible to guess more blocks of key, and the improvement also eliminates errors of simulated attacks what is very important when approximating attack success rate and complexity based on simulated attacks. We also discus a possible efficient countermeasure.

K e y w o r d s: side-channel-attacks, correlation power analysis, Hamming distance power model, DSA, ECDSA, PKI

1 INTRODUCTION

Many techniques exploit dependency of the power consumption or electromagnetic emanation on data processing operations performed within a cryptographic hardware. For example, attacks like differential power analysis (DPA) [1], correlation power analysis (CPA) [2], differential electromagnetic emanation analysis (DEMA) [3], or correlation electromagnetic emanation analysis (CEMA) [4] are common, and not so difficult to perform, side channel attacks (SCA). All these attacks require an appropriate description of the data-dependent power consumption or electromagnetic emanation using information leakage models, such as Hamming weight (HW) or Hamming distance (HD) power models. Construction of the HW power model (HWPM) is less complex than the construction of the HD power model (HDPM), but also less efficient. Power models are usually made considering the architecture of the cryptographic algorithm, or rather register transfer level (RTL) description of the algorithm that is implemented in the attacked device. More about power-analysis attacks can be found, *eg* in [5].

Side-channel-leakage arises during processing sensitive intermediate values by data-dependent operations causing data-dependent power consumption or another physical behavior. We can further distinguish between data, and operation dependences, respectively. Examples of these operations are data registering, multiplexing and addressing, but also data transferring, and any combinational logic operations on data (*eg* AND, OR, XOR). Note that any high level function can be decomposed to these basic operations. The side-channel- leakage depends on the technical realization of these basic elements. For example, registers created in programmable logic blocks in FPGA cause higher side-channel-leakage than registers in embedded memories because the programmable logic blocks are more complex due to their programmability features while registers in embedded memory are hard-wired, optimized and small.

1.1 Other SCAs

The Correlation or Differential family of attacks is a very generic method to attack when only limited information is known about the implementation, and only limited access is possible to the device. They are dangerous and can reveal the secret in many cases, but there are more powerful attacks called Template or Profiling attacks [6]. Such attacks use more sophisticated description of the sensitive leakage [7], like stochastic methods [8], multivariate Gaussian distribution [9], multivariate regression, and conditional entropy (mutual information analysis MIA [10]). These attacks, however, need to have access to the same device (or another instance of the device) before the attacks are performed, in other to make the statistical profile of the leakage (the templates). There are also some works using evolution and genetic algorithms [11]. Very powerful attacks are also active side-channel-attacks, namely Fault Injection Attacks (FIA) [12, 13], and hardware trojan horses [14].

1.2 Related work & our contribution

Work [15] deals with CPA against integer multiplication with constant secret operand. In that work, attack against ECDSA implementation in passive RFID is performed. The ECDSA implementation is based on 163-bit elliptic curve, and the sensitive multiplication is performed using a 16-bit integer multiplier. They demonstrate revealing of the first 2 16-bit blocks of the one chosen secret constant operand k (private key).

$$s = n^{-1}(\mathrm{Hash}(m) + kr) \pmod{q}. \tag{1}$$

[*] Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology, Ilkovičova 3, Bratislava, SK-812 19, Slovak Republic, marek.repka@stuba.sk; [**] Department of Electronics and Multimedia Communications, Faculty of Electrical Engineering and Informatics, Letná 9, 04120, Košice, Slovak Republic, michal@varchola.sk

DOI: 10.2478/jee-2015-0025, Print ISSN 1335-3632, On-line ISSN 1339-309X

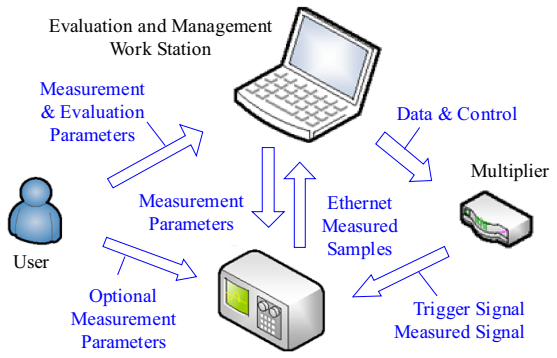

Fig. 1. Top-level measurement & attack setup

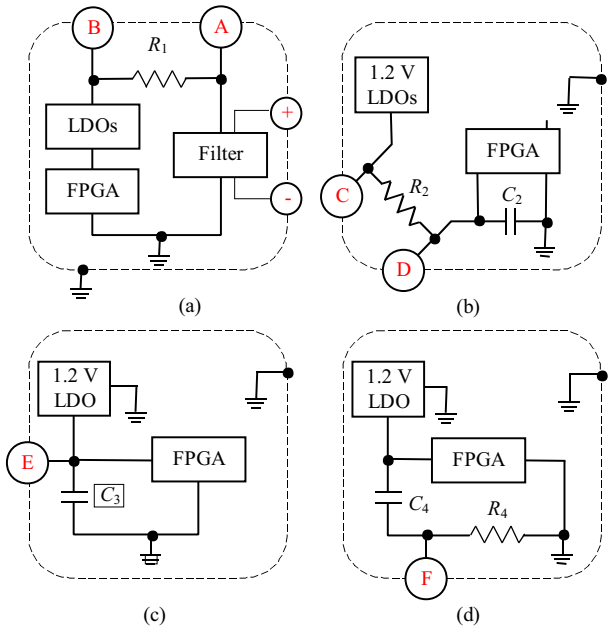

Fig. 2. Schematic diagrams of measurements points in the DISIPA FPGA board. (a) — current flow from a linear regulator to the FPGA, (b) — current flow from the power supply to a linear regulator; c) the voltage on the decoupling capacitor, (d) — current flow from a decoupling capacitor to the FPGA

The sensitive integer multiplication is the multiplication kr, where r is known and k is the private key. This sensitive multiplication is performed in DSA and its variations, such as in ECDSA. The pair (s, r) creates the digital signature of the message m. The n is a per-message random nonce.

In this work, we randomly and uniformly generate 665 16-bit keys and try to reveal them. We used measured as well as simulated power traces using HDPM. Based on these results, we estimated success rate and complexity of the attack against 16-bit blocks of k, and we demonstrate the improvement on this results.

Finally, possible efficient countermeasure is discussed, and the work is concluded.

2 PRELIMINARIES

We can expect more than one key hypothesis remaining after the simulated correlation analysis. It is because multiplication by a constant is a linear function, and furthermore we use HDPM, while for example, if attacking AES S-Box, there is only one key hypothesis (only one correlation peak) because it is nonlinear function at all. In this work, we show how it is possible to repress the impact of the linearity to achieve better success rate considering complexity.

The measurement and attack setup used is depicted in Fig. 1. A 16-bit integer multiplier is implemented in FPGA. The FPGA has further implemented only the necessary functionality for our experiments. Goal of this work is to demonstrate the improvement for guessing a constant operand of a 16-bit multiplier from generic point of view. This constant operand (noted as key or k) has been multiplied by known ordered set of second operands. In order to distinguish between possible hypotheses about the value of the constant operand, correlation coefficient is used. There are not special analyses or preprocessing techniques, nor special side-channel-leakage models, used. There is only the classical correlation power analysis employed. Our goal is not to adjust the analysis of the multiplier implementation to gain the best success rate, and make it appropriate for the one implementation instance, but rather see such generic attack possibilities. The CPA uses HDPM of the multiplication result

$$P_{m,k'} = HD(h_{m,k'}, h_{m+1,k'}) \tag{2}$$

where $h_{m,k'}$ is a hypothesis to the m-th multiplication result as a consequence of the hypothesis k' to the real k. The $P_{m,k'}$ is then hypothesis to the power consumption of registering $(m+1)$-th multiplication result.

The CPA analysis aim is to exploit power consumption caused by registers that register results of multiplications. It is generally accepted that the power consumption of registers is linearly dependent on number of $1 \to 0$ and $0 \to 1$ transitions. Thus, the power consumption can be simulated by HD which is better fitting than the HW. However, measured power consumption will be noised by other functionality of the FPGA, which runs parallel, and also by the environment. Consider now Signal to noise ratio. In our case of analysis, signal consists of dynamic power consumption caused by the 32-bit registers for multiplication results. The noise signal consists of dynamic power consumption caused by LFSR (used to generate the known ordered set of second operands), state machine (used to control dataflow), UART (for communication), and signal added by environment and measurement.

The FPGA (Altera Cyclone III) and measurement points (Fig. 2) circuitry have their own chamber in the shield. All: linear regulators and filters, configuration circuitry, input/output circuitry, and the main Murata filter have separate chambers as well. Described improvements enhance signal to-noise ratio of the leakage, or in other words will reduce the number of traces needed for a successful CPA attack. We want to get as clean leakage signal as possible in order to assess the strength of particular

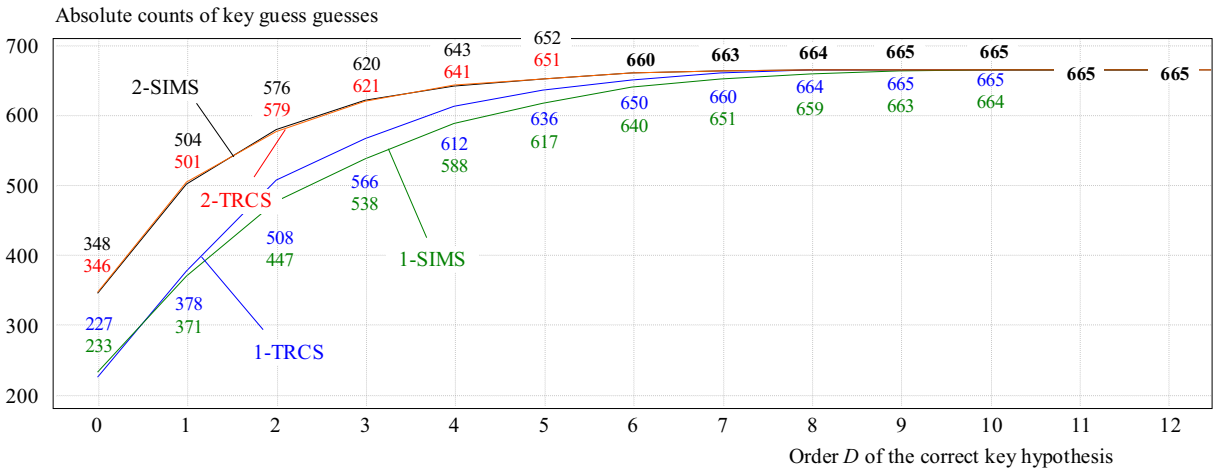

Fig. 3. Demonstration of the improvement on results of guessing 665 randomly and uniformly generated 16-bit keys: After 1^{st} CPA using trcs — "1-TRCS", using sims — "1-SIMS", after 2^{nd} CPA using trcs — "2-TRCS" and using sims — "2-SIMS"

Table 1. Difference between probability and complexity of the attack after 1st and 2nd CPA, data in this table is mentioned only for the most complex attack, for more information about the difference, see Fig. 4, note that the complexity was bounded by 2^{60}

Key size (bits)	After 1st CPA		After 2nd CPA	
	Probability	Complexity	Probability	Complexity
368	NA	NA	0.613	$2^{59.45}$
352	NA	NA	0.626	$2^{56.89}$
336	0.619	$2^{58.95}$	0.853	$2^{56.95}$
320	0.856	2^{60}	0.941	2^{60}
304	0.866	2^{57}	0.944	2^{57}
288	0.973	$2^{57.06}$	0.973	$2^{57.06}$

countermeasures. We are curious, if simple (but efficient) EMI shielding, or the usage of another measurement point causes otherwise secure CPA countermeasure to be inadequate. Up to now, we have found that the selection of measurement points matters. The voltage drop on a series measurement resistor is definitely not the best choice. We found out that the voltage on the decoupling capacitor (Fig. 2(c)) gives us the best results. Therefore power traces were measured using this measurement point, and they were averaged by 128 traces.

The oscilloscope used has 8-bit AD converter, and 20G samples per second rate of signal sampling. The FPGA used has frequency 131,072MHZ. We recorded the power traces exactly in the clock when results of multiplication are registered. We considered only key hypotheses with negative correlations.

3 THE IMPROVEMENT

CPA is used to order key hypotheses from the most fitting one to the worst fitting one. The hypotheses are ordered based on the correlation coefficient in that way that the lower one is the most fitting, and the closed to 0 one is the worst fitting, and we throw all the hypotheses with positive correlation coefficient. Afterwards the hypotheses are ordered, the correct key hypothesis is between the first D of them with some probability. Guessing of 665 randomly and uniformly generated 16-bit keys can be seen in the Fig. 3. From this figure, we can see that if we take 10 first key hypotheses ($D = 9$) after the 1st CPA, the attack will succeed in 100 % for measured power traces. For simulated power traces, we must take $D = 11$. The 1st CPA uses HDPM of all 32-bit registers for multiplication result (2).

In order to improve the attack, we took the first 10 key hypotheses ordered according to the correlation coefficient after the 1st CPA in both cases, and performed 2nd CPA attack in order to reorder the first 10 key hypotheses. In the 2nd CPA attack, we made HDPM only to the vector of the 16 least significant bits of the possible result of multiplication

$$P_{m,k'} = HD\big(LSB_{0\ldots15}(h_{m,k'}\,,\; LSB_{0\ldots15}(h_{m+1,k'})\big). \quad (3)$$

The new order of the 10 first key hypotheses brings improvement as can be seen in Fig. 3. In this figure, counts for CPA using measured as well as simulated power traces are depicted. The success rate for simulated CPA is negligible different of the real CPA after the 2nd CPA. The improvement in the case of the simulated CPA is crucial in estimation of success rate and complexity for guessing of N 16-bit blocks of key.

Estimations of success rate and complexity for guessing of N 16-bit blocks of key based on measured power traces can be found in Fig. 4. In this graph the improvement is demonstrated on difference in success rate and complexity after 1st and 2nd CPA respectively. The estimations are bounded for maximal complexity 2^{60} as this is a boundary of our computation power, and for minimal probability of success which must by greater than 0.5 as there must be probability of success more than 50 %. These are boundaries for our demonstration of our improvement in this work.

When we look at the guessing of N 16-bit blocks after the second CPA attack (Fig. 4), we can see the brought

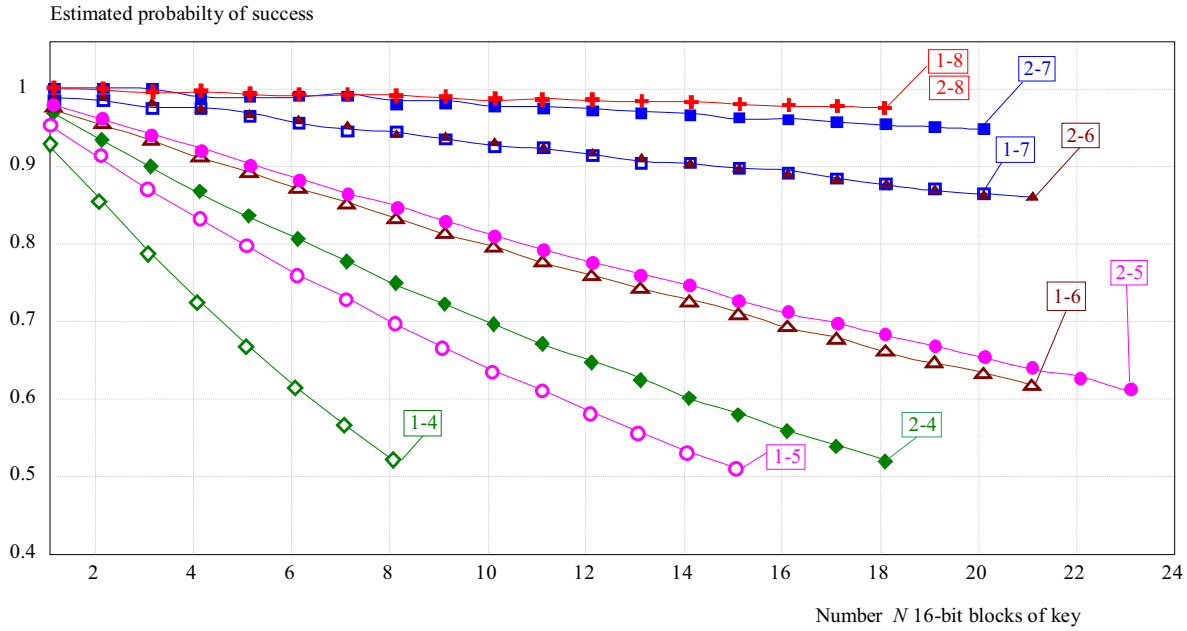

Fig. 4. Estimatied attack probability after the first or second CPA (first number in rectangle tag appearing in the graphic) for different values of order D (second number in graphic's tags); Estimatied attack complexity: $2^{57.06}$ for $\boxed{1-8}$ and $\boxed{2-8}$, 2^{60} for $\boxed{1-7}$ and $\boxed{2-7}$, $2^{58.95}$ for $\boxed{1-6}$, and $\boxed{2-6}$ and further $2^{59.45}$ for $\boxed{2-5}$, $2^{41.79}$ for $\boxed{2-4}$, $2^{38.77}$ for $\boxed{1-5}$, and $2^{18.58}$ for $\boxed{1-4}$. Note the improvements: $\boxed{1-4}$ to $\boxed{2-4}$, further $\boxed{1-5}$ to $\boxed{2-5}$ and $\boxed{1-6}$ to $\boxed{2-6}$ as well as $\boxed{1-7}$ to $\boxed{2-7}$.

improvement since, now, 368-bit ($N = 23$ 16-bit blocks) of the key can be guessed with approximated probability 0.613 and complexity $2^{59.45}$ ($D = 5$), while after the first CPA, only 336-bit ($N = 21$ 16-bit blocks) could be guessed with probability 0.62 and complexity $2^{58.95}$ ($D = 6$). We can see an improvement in success rate or complexity also for other cases. For instance, when $D = 4$, 8 16-bit blocks has been improved to 18 blocks (288-bit) of the key with complexity $2^{41.79}$. Further, when $D = 6$, the probability of success was improved from cca 0.61 to 0.86. For $D = 7$ the success rate was improved from cca 0.86 to 0.94. The success rate for $D = 18$ remains the same. The comparison of probability and complexity for the most complex attack after 1st and 2nd CPA can be found in Tab. I.

4 POSSIBLE COUNTERMEASURE AGAINST THIS CPA

This CPA needs to have the secret operand constant and to know some second operands of the multiplication. One possible countermeasure to thwart this attack, which does not need special countermeasures, such as hiding (dummy cycles, noise generator, dual-rail-logic [16]) or masking (boolean, multiplicative [17]), is to use the nonce n (the per message randomly and uniformly generated number) to mask the key as

$$s = n^{-1}\,\mathrm{Hash}(m) + kn^{-1}r \pmod{q}. \qquad (4)$$

Before the private key is multiplied by known r, it is multiplied by the inversion of an unknown nonce. In order to make this attacks impossible, (1) must be replaced by (4).

The cost of this countermeasure is one more multiplication by the inversion of the nonce. This countermeasure is effective since the nonce is random and not public, thus it is not know to the adversary, and this countermeasure is efficient because it costs only one more multiplication and not other special logic, such as in case of an additional masking and hiding.

If there would be the countermeasure made in the way that the key instead of r would be multiplied by the n^{-1}, the countermeasure would not be effective enough, because a next leakage would be produced. Power consumption of multiplication $n^{-1}\,\mathrm{Hash}(m)$ and multiplication $n^{-1}k$ must correlate for processing blocks of $\mathrm{Hash}(m)$ having values equal to values of the corresponding blocks of k.

5 CONCLUSION

We improved the CPA attack and we eliminated the error of the simulated CPA attack. The improvement is in performing the second CPA but only on the first 10 key hypotheses ordered according to the correlation coefficient from the first CPA. As the power model for the second CPA, we used again generic HDPM, however, this time, only for the first half of the least significant bits instead of all the bits of the multiplication result. This brought improvement since after the second CPA, it is possible to guess more blocks of the key with approximated probability 0.613 and complexity $2^{59.45}$. Also for the other choices of D, there is an improvement in success rate or complexity. For more details, consult Tab. I and the graph depicted in Fig. 4. After the second CPA, simulated attacks achieve negligible difference in success

rate and complexity in comparison with attacks using measured power traces. It is very important for approximation of attack success rate and complexity based on simulated attacks.

Finally, we discussed possible effective and efficient countermeasure. In order to thwart this correlation power attack by the discussed countermeasure no further masking or hiding must be employed in case of DSA and ECDSA.

Acknowledgment

This work has been Supported in part by grant APVV- 0586-11 (Digital Signature Power Analysis Attacks DISIPA Project); the OP Research and Development for project: Establishment, Development and Scientific Management of a Research Center for the Analysis and Protection of Data, ITMS: 26240120037, co-funded by the EU; and the NATOs Public Diplomacy Division in the framework of Science for Peace, SPS Project 98452.

References

[1] KOCHER, P. C.—JAFFE, J.—JUN, B.: Differential Power Analysis, Proc. of the 19th Annual International Cryptology Conference on Advances in Cryptology in CRYPTO '99, Springer-Verlag, London, UK, 1999, pp. 388–397.

[2] BRIER, E.—CLAVIER, C.—OLIVIER, F.: Correlation Power Analysis with a Leakage Model, CHES, Handbook, Mill Valley, CA: University Science, 2004, pp. 16-29.

[3] QUISQUATER, J.-J.—SAMYDE, D.: Electro-Magnetic Analysis (EMA), Measures and Counter-Measures for Smart Cards Proceedings of E-SMART '01, Springer-Verlag, London, UK, 2001, pp. 200–210.

[4] DING, G. L.—CHU, J.—YUAN, L. ZHAO, Q.: Correlation Electromagnetic Analysis for Cryptographic Device, Proc. of the 2009 Pacific-Asia Conference on Circuits, Communications and Systems, IEEE Computer Society, Washington, DC, USA, 2009, pp. 388–391.

[5] MANGARD, S.—OSWALD, E.—POPP, T.: Power Analysis Attacks – Revealing the Secrets of Smart Cards, Advances in Information Security, Handbook, Springer-Verlag, New York, 2007.

[6] MEDWED, M.—OSWALD, M. E.: Template Attacks on ECDSA, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23–25, 2008, Revised Selected Papers in Lecture Notes in Computer Science, Springer, 2009, pp. 14–27.

[7] STANDAERT, F.-X.—MALKIN, T.—YUNG. M.: A Unified Framework for the Analysis of Side-Channel Key Recovery Attacks, In Advances in Cryptology EUROCRYPT 2009 (A. Joux,, ed.), LNCS 5479, Springer, Berlin, pp. 443–461.

[8] SCHINDLER, W.—LEMKE, K.—PAAR, C.: A Stochastic Model for Differential Side Channel Cryptanalysis, In Cryptographic Hardware and Embedded Systems CHES 2005, LNCS 3659, Springer, pp. 30–46.

[9] RIVAIN, M.: On the Exact Success Rate of Side Channel Analysis in the Gaussian Model, In Selected Areas in Cryptography (R. Avanzi, L. Keliher, and F. Sica, eds.), LNCS 5381, Springer, Berlin, pp. 165–183.

[10] BATINA, L.—GIERLICHS, B.—PROUFF, E.—RIVAIN, M.—STANDAERT, F.-X.—VEYRAT-CHARVILLON, N.: Mutual Information Analysis: a Comprehensive Study, Journal of Cryptology **24** No. 2 (2011), 269–291.

[11] HEUSER, A.—ZOHNER, M.: Intelligent Machine Homicide – Breaking Cryptographic Devices using Support Vector Machines, In Constructive Side-Channel Analysis and Secure Design 3th International Workshop, COSADE 2012, Proceedings (Schindler and Huss, eds.), LNCS 7275, Springer, Darmstadt, Germany, 2012, pp. 249–264.

[12] KARPOVSKY, M. G.—KULIKOWSKI, K. J.—TAUBIN, A.: Robust Protection against Fault Injection Attacks on Smart Cards Implementing the Advanced Encryption Standard, In: DSN, IEEE Computer Society, Florence, Italy, 2004, pp. 93–101.

[13] GUILLEY, S.—SAUVAGE, L.—DANGER, J. L.—SELMANE, N.: Fault Injection Resilience, In FDTC, IEEE Computer Society, Santa Barbara, CA, USA, Aug 2010, pp. 51–65.

[14] CLAVIER, C.—GAJ, K.: Trojan Side-Channels: Lightweight Hardware Trojans through Side-Channel Engineering, Proceedings 11th International Workshop Lausanne, Switzerland, September 6-9, 2009 CHES, Springer, Berlin Heidelberg, 2009, pp. 382–395.

[15] HUTTER, M.—MEDWED, M.—HEIN, D.—WOLKERSTORFER, J.: Attacking ECDSA-Enabled RFID Devices, ACNS 2009, LNCS 5536, Springer-Verlag, Berlin Heidelberg, 2009, pp. 519–534.

[16] DANGER, J. L.—GUILLEY, S.—BHASIN, S.—NASSAR, M.: Overview of Dual Rail with Precharge Logic Styles to Thwart Implementation-Level Attacks on Hardware Cryptoprocessors, New Attacks and Improved CounterMeasures, In: SCS (November 6-8, 2009), IEEE, Jerba Tunisia, p. 18.

[17] AKKAR, M.-L.—GIRUAD, C.: An Implementation of DES and AES, Secure Against some Attacks, In Cryptographic Hardware and Embedded Systems – CHESS 2001, Proceedings Third Interational Workshop, Paris, France, May 14–16, Springer, 2001, pp. 309–318.

Received 6 February 2015

Marek Repka was born in Czech Republic in 1985. In 2010, he achieved master degree in the applied informatics focused on security of information systems at the Institute of Computer Science and Mathematics FEI STU in Bratislava, Slovakia. He is a PhD student specializing on side-channel-cryptanalysis. He is working with TEMPEST, a.s. company in Bratislava, Slovakia, focused on application security, and implementation and integration of security controls.

Michal Varchola was born in Slovakia in 1984. He received PhD degree in info-electronics, from Technical University of Košice, Slovakia in 2010. He works as young researcher at Technical University of Košice from 2010. His main fields of interests are: side channel analysis of cryptographic devices, true random number generators, implementation and integration of FPGA and MCU embedded systems and digital signal processing. He is a member of International Association for Cryptologic Research and has received research project dean's award in 2014.

Miloš Drutarovský was born in Prešov in Slovakia, in 1965. He received his Ing (MSc) degree and PhD degree in Radioelectronics from the Faculty of Electrical Engineering, Technical University of Košice, in 1988 and 1995, respectively. He is currently working as an associate professor at the Department of Electronics and Multimedia Communications of the Faculty of Electrical Engineering and Informatics, Technical University of Košice. His current research focuses on embedded electronics, applied cryptography, algorithms and architectures for embedded cryptographic architectures, digital signal processing, digital signal processors, field programmable devices and soft microcontrollers embedded into FPGA circuits.