Speeding up differential power analysis using integrated power traces

Vojtěch Miškovský, Hana Kubátová, Martin Novotný
Faculty of Information Technology
Czech Technical University in Prague
Prague, Czech Republic
{miskovoj,kubatova,novotnym}@fit.cvut.cz

Abstract—**Side-channel attacks, including differential power analysis (DPA), are still an emerging topic. To make a deep research about DPA, one needs to be able to perform it as fast as possible. There are many possible ways to decrease the time of the attack. In this paper, we propose a way to decrease the duration of the correlation computations of this kind of attack by decreasing the number of samples per a power trace using an integration based aggregation method. We comprehensively describe this idea and present the results of an experimental evaluation focusing on the time efficiency of this approach.**

Keywords-differential power analysis, side-channel attack, security, aggregation, correlation

I. Introduction

Side-channel attacks pose a security thread to any modern cryptographic device. As these attacks are based on physical properties of the device rather than cryptographic properties of the cipher, it is an issue even for the ciphers considered mathematically secure, such as AES. There are many possible side-channels (e.g. temperature [1], EM radiation [2], time [3], noise [4], etc.), but the most commonly used and the most exhaustively researched one is the power consumption.

In this paper, we focus on a common power consumption based side-channel attack called differential power analysis (DPA) [5] and specifically its correlation based variant (also called correlation power analysis (CPA)) [6]. The first step of this attack is to measure power consumption of the device during the cryptographic operation (usually encryption) using known plain text or cipher text. A large number of power consumption traces needs to be collected using the same secret (cipher key) during this step. In the next step, we need to choose a proper power model. The power model should be a function of plain or cipher text and a key and it should somehow estimate the expected power consumption. Using the power model, we estimate power consumption for each combination of previously measured plain/cipher text and possible key candidates (for this purpose, the key should be split into smaller parts, e.g. bytes or nibbles). Finally, we calculate correlation between real power consumption and estimations for each key candidate, where the most correlating key candidate is considered to be the correct one.

To research this attack, it is necessary to be able to perform it as fast as possible. There are multiple ways to achieve this. There are two highly time demanding parts of the attack: power measurement and correlation calculation. The time of the measurement mostly depends on the number of traces necessary to obtain the correct key and on the effectiveness of the trace acquisition. The number of traces needed is mostly given by the device under attack, but this number can also be decreased by some statistical methods (e.g. [7]). There are ways to increase the efficiency of the trace acquisition, as well. Some of these are proposed e.g. in [8]. In case of the correlation calculations, there are three properties influencing the computation time: number of power traces, number of samples per trace, and algorithmic efficiency. As the problem of decreasing the number of traces was discussed above and the ways to increase the efficiency of the calculations are proposed e.g. in [9], the remaining scope of improvement is in decreasing the number of samples per trace.

The main idea of this research is to somehow aggregate the traces. As the power model usually represents some intermediate value exhibited in the device during a single clock cycle, the most straightforward way is to integrate the traces by time for each clock cycle. Therefore, the number of samples per trace will be close to the number of clock cycles of one encryption (usually tens) instead of the usual hundreds or thousands. In this article, we show how this simple trace aggregation affects the effectiveness of the attack. Specifically, we will show how it affects the number of traces needed to reveal the correct key and how it affects the overall calculation time of the attack on three different FPGA platforms.

We introduce the FPGA platforms, AES implementation and the experiment setup in Section II. Detailed results are presented in Section III. We summarize and conclude the experiment in Section IV and finally we propose possible ways to continue this research in Section V.

II. Experiment setup

In this section, we comprehensively describe the process of trace aggregation. We also introduce the FPGA platforms and the AES implementation we used.

This research has been partially supported by the grant GA16-05179S of the Czech Grant Agency, "Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features" (2016-2018) and CTU project SGS17/213/OHK3/3T/18.

A. Hardware platforms

We performed the experiment on three different FPGA platforms:

- **Evariste III** with Altera Cyclone III module — an open modular cryptographic platform developed by Fischer et al. [10]
- **Sakura-G** — standard evaluation board for side-channel attacks based on Xilinx Spartan-6 FPGA [11]
- **DPAboard** — open board for differential power analysis with Xilinx Artix-7 FPGA developed by Bartik et al. [12]

The whole experiment is same for all platforms, only the clock frequencies differ. Evariste III and Sakura-G run at 5 MHz while DPAboard runs at 6.25 MHz.

PicoScope 6404D was used for the measurement. The sampling rate was set to 625 MS/s. We collected 2000 samples per one power trace.

B. AES

AES cipher [13], specifically its 128-bit variant, is used for the experiment as it is the most commonly used and researched block cipher. The 128bit variant of AES consists of ten regular rounds and one initial round. The regular rounds consist of a key addition (AddRoundKey function), a non-linear transformation (SubBytes function) and two linear transformations (ShiftRows and MixColumns functions) with an exception of the last round, where the MixColumns function is omitted. The initial round performs only the key addition. Each round uses a different round key derived from the initial one.

In our implementation, each round is processed by a combinational logic during one clock cycle. Therefore, the whole encryption takes 11 clock cycles. A diagram of our AES implementation can be seen in Fig. 1.

C. Trace aggregation

The main idea of our trace aggregation approach is based on fact, that the intermediate value used by the power model and the real power consumption are not significantly correlated in one specific power sample, but the correlation should be spread through the whole clock cycle the intermediate value is processed within. As the number of samples is the same during each clock cycle, we do not need to take it into account and the integration process can be simplified to summation. Therefore, we preprocess each power trace by summarizing the samples corresponding to each single clock cycle of the encryption. In our case, each clock cycle corresponds to 125 samples (Evariste III and Sakura-G) or 100 samples (DPAboard).

We summarized each clock cycle of the encryption (11 cycles) and one before and one after the encryption. We obtained 13 samples per trace instead of the original 2000 samples this way.

We also aggregated the traces with variable starting sample that we call *offset*. For example, with offset 0, the first sample of the integration is the first sample of the clock cycle; with offset 25, the first sample of the integration is the 26th sample of the clock cycle, therefore the last sample is the 25th sample of the next clock cycle. We used five offsets: 0, 25, 50, 75 and 100 in case of Evariste III and Sakura-G, and 0, 20, 40, 60, 80 in case of DPAboard. For simplicity, we will use relative designations of these offsets: 0, 1/5, 2/5, 3/5 and 4/5.

To summarize, we aggregated each power trace using 5 different offsets. Therefore, we gained 5 traces of 13 samples from each of the original traces of 2000 samples.

D. Evaluation

For each of hardware platforms, we measured 30 sets of power traces. Then each collected trace was aggregated using the method described above. Therefore, we obtained 5 aggregated power traces and the original ones for all the traces in each of the sets.

Having all the data collected, we performed the computational part of the attack. For each of the 3×30×(5+1) sets of power traces (3 platforms; 30 sets; 5 offsets + 1 non-aggregated), we used incremental algorithm proposed in [9] to obtain the minimal number of power traces needed to reveal the correct cipher key. These collected results are presented in Section III.

We also compared the running time for various number of traces to demonstrate the efficiency of our approach. All time measurements were done on a computer equipped with Intel i5 6440HQ CPU (quad core), 8GB DDR3 RAM, an SSD drive and an up-to-date installation of Windows 10 Enterprise.

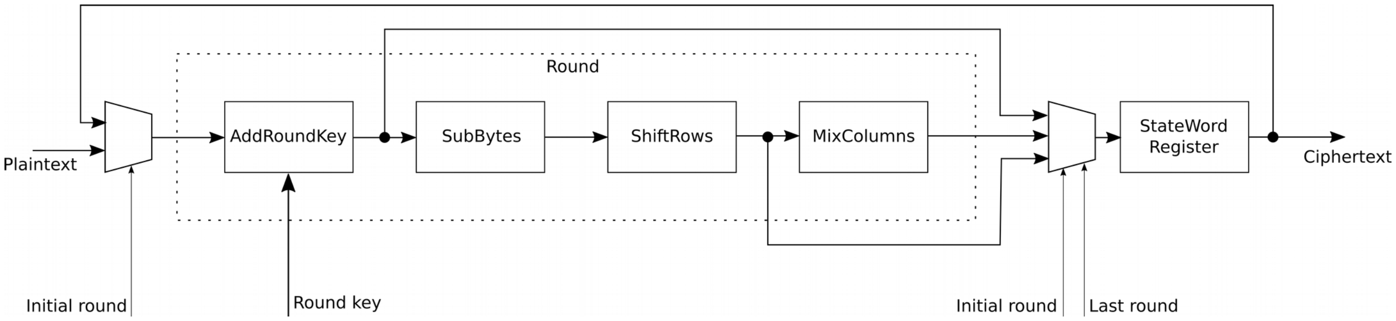


Figure 1: Diagram of AES implementation

III. RESULTS

In this section, we present results obtained from the experiment proposed in Section II. At first, we show how our methods affects the number of power traces needed to reveal the correct key. Afterwards, we present a time comparison of common methods and our new approach.

A. *Power traces needed*

We successfully performed the attack using the proposed approach with all offsets and on all platforms. We compared the minimal number of traces needed using basic correlation and our preprocessed variant. In Table I, we can see medians of minimal needed traces calculated from results of all trace sets.

In case of Evariste III and DPAboard, the number of traces needed is even lower than with the original traces for all offsets.

In case of Sakura-G, we can see significant differences between the offsets. The results are better than with the original approach when offsets 1/5 or 2/5 are used, but in case of offset 4/5, the number of traces needed is more than 9 times higher. On the other hand, the time saved by our method overcomes this increase, as is shown below.

Detailed results can be seen in Fig. 2: a) for Evariste III, b) for Sakura-G and c) for DPAboard.

B. *Time comparison*

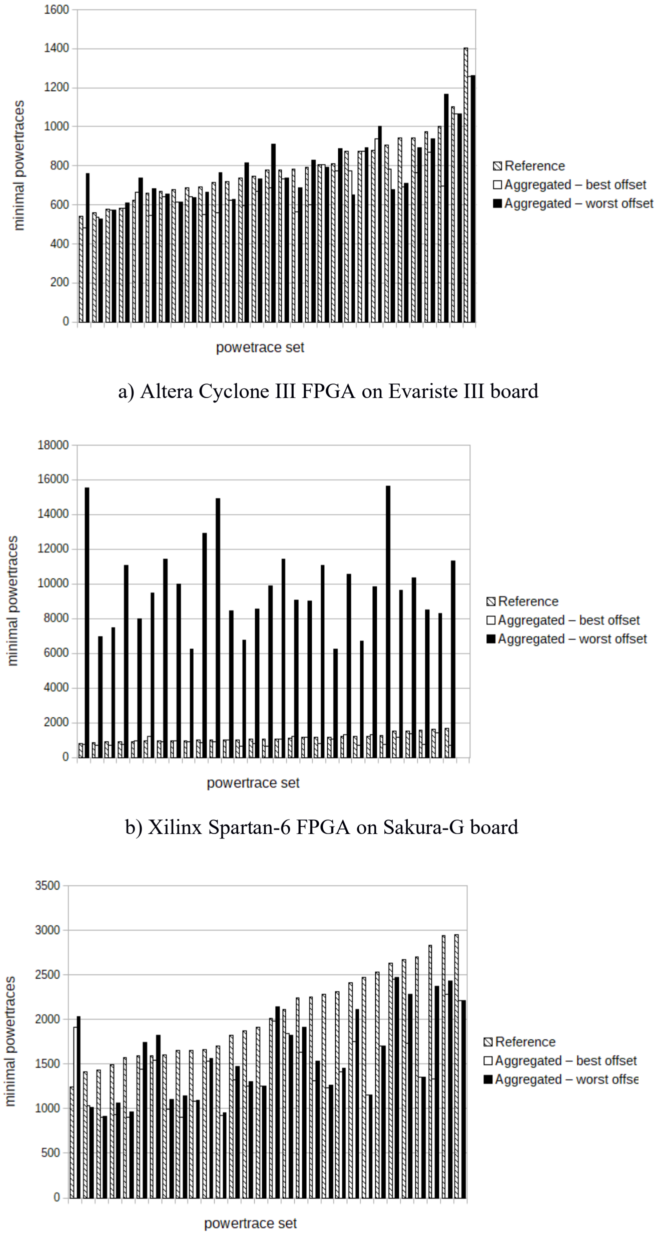
Considering the algorithm being linearly dependent on number of samples (as is shown in [9]), we expected our approach to be significantly more time efficient than the original method. As we can see in Table II, the correlation calculation using 13 samples per trace is approximately 50 times faster than the one using 2000 samples per trace. As the preprocessing integration duration is also much lower, the total time consumption using our approach is roughly 40 times lower than with the original approach. In Fig. 3, we can see that this improvement is constant with the number of power traces.

TABLE I: MEDIANS OF POWER TRACES NECESSARY TO OBTAIN THE CIPHER KEY USING VARIOUS FPGAS AND USING THE ORIGINAL TRACES OR INTEGRATED TRACES WITH VARIOUS OFFSETS

Traces		**Platform (FPGA)**		
		Evariste III (Cyclone III)	***Sakura-G (Spartan-6)***	***DPAboard (Artix-7)***
Original		779	1048	1967
Integrated	Offset 0	666	2202.5	1347.5
	Offset 1/5	714.5	885	1359
	Offset 2/5	737	993.5	1353
	Offset 3/5	692.5	1722.5	1470.5
	Offset 4/5	683.5	9718	1501.5

TABLE II: PLOT OF MINIMAL POWER TRACES NECESSARY TO REVEAL THE CIPHER KEY WITH THE ORIGINAL POWER TRACES AND POWER TRACES OF THE BEST AND THE WORST OFFSET FOR ALL POWER TRACE SETS, ALTERA CYCLONE III FPGA ON EVARISTE III BOARD

Procedure		**Number of traces**			
		1K	***10K***	***100K***	***1M***
Correlation	2000 samples	1.8	17.6	177	1850
	13 samples	0.06	0.36	3.7	36
Integration		0.03	0.12	1.11	11



a) Altera Cyclone III FPGA on Evariste III board

b) Xilinx Spartan-6 FPGA on Sakura-G board

c) Xilinx Artix-7 FPGA on DPAboard

Figure 2: Plot of minimal power traces necessary to reveal the cipher key with the original power traces and power traces of the best and the worst offset for all power trace sets

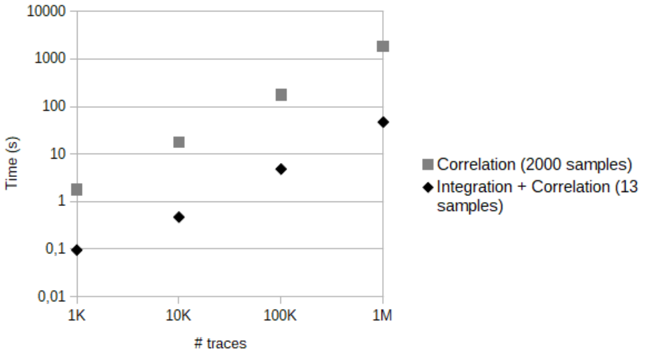


Figure 3: Plot of running time of the original method (Correlation with 2000 samples per trace) and our approach (Integration + correlation with 13 samples per trace) for various number of power traces (logarithmic scale)

IV. Conclusion

We propose a new approach to evaluate the correlation coefficients when performing differential power analysis. It is based on power trace preprocessing. In each trace, we integrate traces corresponding to each clock cycle of the encryption. This preprocessing is easy to implement and insignificantly time demanding in comparison with the correlation calculations. This approach makes the correlation calculations much less time demanding as it drastically decreases the number of samples per trace. We successfully confirmed these assumptions in previous section. We show, that the overall time of the calculation part of attack is approximately 40 times lower with our approach compared to the original method.

We also show, how the integration affects the number of power traces necessary to obtain the correct key. Three different platforms were used. On two platforms, the number of traces was even lower than without the preprocessing. Also, on the third platform our approach is more time efficient even though more power traces are required when using some particular offsets.

Considering these facts, the presented method was confirmed as an excellent way to speed up the differential power analysis.

V. Future work

Since the preprocessing appears to be a possible way to decrease the number of samples per trace, and also the number of traces needed for differential power analysis, more similar preprocessing methods could be explored. For example, choosing just one sample (e.g. the highest one) in each clock cycle could be a way.

Another way to improve the time efficiency of the attack could be the key candidate selection. So far, the key candidate with the highest absolute value of correlation coefficient is chosen as the correct one. Some more advanced evaluation of the correlation traces decreasing the number of power traces could be proposed.

References

[1] Hutter, M., & Schmidt, J. M. (2013, November). The temperature side channel and heating fault attacks. In *International Conference on Smart Card Research and Advanced Applications* (pp. 219-235). Springer, Cham.

[2] Gandolfi, K., Mourtel, C., & Olivier, F. (2001, May). Electromagnetic analysis: Concrete results. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 251-261). Springer, Berlin, Heidelberg.

[3] Bernstein, D. J. (2005). Cache-timing attacks on AES.

[4] Genkin, D., Shamir, A., & Tromer, E. (2014, August). RSA key extraction via low-bandwidth acoustic cryptanalysis. In *International Cryptology Conference* (pp. 444-461). Springer, Berlin, Heidelberg.

[5] Kocher, P., Jaffe, J., & Jun, B. (1999, August). Differential power analysis. In *Annual International Cryptology Conference* (pp. 388-397). Springer, Berlin, Heidelberg.

[6] Brier, E., Clavier, C., & Olivier, F. (2004, August). Correlation power analysis with a leakage model. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 16-29). Springer, Berlin, Heidelberg.

[7] Liu, W., Wu, L., Zhang, X., & Wang, A. (2014, November). Wavelet-based noise reduction in power analysis attack. In *Computational Intelligence and Security (CIS), 2014 Tenth International Conference on* (pp. 405-409). IEEE.

[8] Schneider, T., & Moradi, A. (2015, September). Leakage assessment methodology. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 495-513). Springer, Berlin, Heidelberg.

[9] Socha, P., Miškovský, V., Kubátová, H., & Novotný, M. (2017, April). Optimization of Pearson correlation coefficient calculation for DPA and comparison of different approaches. In *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2017 IEEE 20th International Symposium on* (pp. 184-189). IEEE.

[10] Fischer, V., Bernard, F., & Haddad, P. (2013, September). An open-source multi-FPGA modular system for fair benchmarking of true random number generators. In *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on* (pp. 1-4). IEEE.

[11] Guntur, H., Ishii, J., & Satoh, A. (2014, October). Side-channel attack user reference architecture board SAKURA-G. In *Consumer Electronics (GCCE), 2014 IEEE 3rd Global Conference on* (pp. 271-274). IEEE.

[12] Bartík, M., & Buček, J. (2016, November). A low-cost multi-purpose experimental FPGA board for cryptography applications. In *Advances in Information, Electronic and Electrical Engineering (AIEEE), 2016 IEEE 4th Workshop on* (pp. 1-4). IEEE.

[13] Pub, N. F. (2001). 197: Advanced encryption standard (AES). *Federal information processing standards publication*, *197*(441), 0311.