Implementation of AES Algorithm Resistant to Differential Power Analysis

Marek Strachacki
Graphics Development Group
Intel Technology Poland
ul. Slowackiego 173, 80-298 Gdansk, Poland
marek.strachacki@intel.com

Stanisław Szczepański
Department of Microelectronic Systems
Gdansk University of Technology
ul. Narutowicza 11/12, 80-952 Gdansk, Poland
stanisla@eti.pg.gda.pl

***Abstract*—This paper[1] describes differential power analysis (DPA) of encryption algorithms hardware implementations. Proposed DPA-resistant design method combines power equalization for synchronous and combinatorial circuits. AES algorithm has been implemented in Xilinx Spartan II-E field programmable gate array (FPGA) device using the standard and DPA-resistant methods. XPower tool has been introduced to collect power traces for DPA. Results show that the standard AES implementation can be broken using DPA in N=2000 encryption operations. At the same time DPA of modified AES implementation for N=2000 encryption operations does not show any correlation between power consumption and the cipher key.**

I. INTRODUCTION

The advance in cryptography led to development of strong encryption algorithms, which ensure high level of security. At the same time research on cryptoanalysis gave methods for finding the cipher key by using known plaintext and the corresponding ciphertext. Among various methods most often used are: interpolation attacks, linear cryptoanalysis and differential cryptoanalysis. To increase the level of security, encryption algorithms are often implemented in a field programmable gate array (FPGA) or an application specific integrated circuit (ASIC) that prevents algorithm modification and cipher key readout.

But even a secure encryption algorithm does not guarantee that its implementation is equally secure. Attacks on hardware implementations use side channel information, what is neither plaintext nor ciphertext. Underlying hardware provides data on processing time, power consumption, electromagnetic radiation and enables tampering with clock signal and power supply. These kinds of attacks can be easily done in short time using inexpensive hardware [1][2]. Attacks can be different: active non-invasive (e.g. tampering), active invasive (e.g. fault injection) and passive (e.g. timing attack and power analysis).

This paper describes an FPGA implementation of encryption algorithm, designed to be resistant to power analysis. Results of power analysis using XPower simulator for standard and proposed design methods are presented.

[1] This work was founded by Intel Technology Poland and the Polish Ministry of Science and Higher Education under R&D grant no. R02 014 01 from the Science Budget 2007-2008.

II. POWER ANALYSIS

The attack bases on power consumption analysis during performing operations by underlying hardware (e.g. ASIC, smartcard, microprocessor) and was introduced in [2].

Simple power analysis (SPA) is the immediate correlation between power consumption and various algorithm phases. If an execution path of the algorithm depends on processed data, then SPA detects the sequence of operation and obtains information on the cipher key. The example is an attack on DES algorithm, where the difference in power consumption is observed for permutation, rotation and comparison instructions, especially in software. Another example is an attack on RSA algorithm, where power consumption during multiplication and squaring is different and strongly correlated with Hamming weight of operands [1][2][3].

Differential power analysis (DPA) is the extension of SPA and is based on fundamental assumption that the power profile is dependent on the cipher key. DPA runs in two stages. In the first stage for each encryption the power consumption of determined operation is collected. In the second stage correlation analysis is performed [4][5] or noise is filtered out using a distance of the mean test [2][3][6]. Both tests verify hypotheses on particular bytes of the cipher key.

In correlation analysis encryption using hypothetical key is performed and the correlation between the power consumption for the unknown and hypothetical keys is computed. If a byte of the cipher key was determined correctly, correlation reaches its maximum. DPA correlation attack on 8 most significant bits of the cipher key in AES algorithm is described in [4][5].

In the distance of the mean test the average power consumption is computed and then encryption using hypothetical key is performed. Power consumption values are assigned to one of two subsets by the selection function. If there is a correlation between mean value of power consumption for whole set and one of the subsets, then a given subset determines the value of one bit [3][6]. DPA distance of the mean attack on DES is described in [2][3].

978-1-4244-2182-4/08/$25.00 ©2008 IEEE.

III. PROTECTION AGAINST POWER ANALYSIS

General recommendation for software implementations of encryption algorithms is to use basic logical operations and to avoid operations which power consumption is data-dependent [3]. Power analysis protection bases on its decorrelation with input data or hiding its variations to increase the complexity of an attack [5]. Basic methods:

- power equalization – introducing additional registers and complementary logic (known as balancing in [1]),
- power randomization – introducing digital noise.

Power equalization requires compensation of register and combinatorial logic switching. This causes an increase in power consumption, but makes it cipher key independent, thus preventing DPA. Special libraries ensure power equalization on the gate or the transistor levels [5][7].

Power randomization [8] by pseudorandom digital noise generation requires additional computation, increasing power consumption or introducing the additional delay [5]. This does not prevent DPA attack, but makes it inefficient because data have to be collected in a very long period [3][8].

Another randomization method masks linear operations and modifies non-linear operations [9]. In DES mask is added before initial permutation and after final permutation. In AES mask is added in the Galois field GF(2^8). The only information disclosed by SPA is the Hamming weight of masked value. This method requires considerable changes in the algorithm and slows it down [5].

IV. PROPOSED DESIGN METHOD

Proposed design method ensures DPA resistance of hardware encryption algorithms implementations. Power randomization is difficult to apply due to iterative structure of block ciphers and frequent modules activation. In this paper combined power equalization techniques are proposed.

To equalize the power consumption in synchronous circuits, it should be ensured that the same number of flip-flops switches on every clock cycle. To the original flip-flop Q_1, flip-flop Q_2 is added in parallel, that:

$$Q_1(t\text{-}1) \oplus Q_1(t) \oplus Q_2(t\text{-}1) \oplus Q_2(t) = \text{‘1’}. \quad (1)$$

This ensures that exactly one flip-flop switches at a time and leads to equation:

$$Q_2(t) = Q_1(t\text{-}1) \oplus Q_1(t) \oplus Q_2(t\text{-}1) \oplus \text{‘1’}. \quad (2)$$

where: Q_1, Q_2 – output of the original and additional flip-flops; t, t-1 – time of current and previous clock cycles.

This method is proposed independently of [6].

To equalize the power consumption in combinatorial logic, it should be ensured that the same number of outputs switches on every change of the input vector on any single position. To the original logic function f(x), complementary logic function g(x) is added in parallel, that:

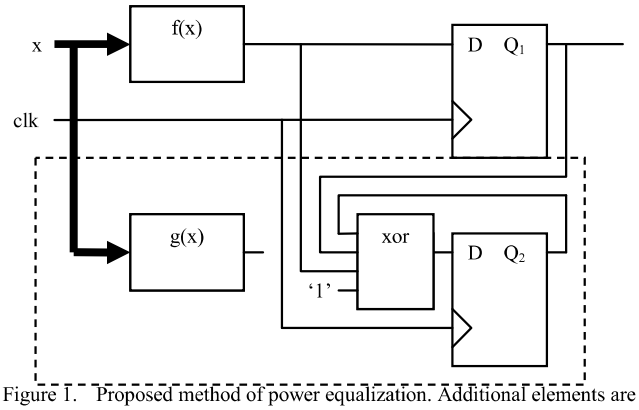

Figure 1. Proposed method of power equalization. Additional elements are outlined by dotted line.

$$f(x) \oplus g(x) = h(x). \quad (3)$$

h(x) is the XOR function of input vector ensuring that exactly one output switches at a time and leads to equation:

$$g(x) = f(x) \oplus h(x). \quad (4)$$

where: x – input vector; f(x) – the original logic function; g(x) – complementary logic function.

Function f(x) is by no means restricted neither to number of operands nor kinds of operations, what is a generalization of methods proposed in [7].

Combined application of both techniques leads to the schematic shown in Fig. 1. Since it is inconvenient to insert additional blocks in the behavioral source code, the circuit should be modified automatically after logic synthesis.

V. DESIGN REQUIREMENTS

For an implementation in FPGA device AES algorithm was chose [10]. There are many published results of DPA for AES, what enables to carry out comparative study [4][5].

Power consumption is most often estimated by counting flip-flop switching directly in VHDL code [4][6] or physically measured in a real environment [3][4]. In this paper, apart from counting flip-flop switching in software, power simulator XPower was used. XPower enables timed simulation of power consumption based on input value changed dump (VCD) file. According to our best knowledge, this is the first usage of XPower for DPA.

Cipher architecture was designed to simplify DPA. There are independent reset signals for encryption and key expansion modules. Only one round was implemented with a full internal pipeline by adding registers after each basic operation of the round. Encryption requires 10 iterations and 40 clock cycles.

Hardware implementation was done in FPGA Xilinx Spartan-II E. Design process included functional simulation of VHDL code, logical synthesis, physical synthesis, functional simulation after physical synthesis and power simulation.

During functional simulation round keys were generated. Then for each of N=2000 operations encryption module reset was carried out and first four cycles (one encryption round) were performed.

Two versions of AES were implemented: standard without power equalization and modified with power equalization. In a modified implementation complementary blocks were added after logical synthesis.

DPA was performed using correlation analysis for N=2000 encryption operations as described in [4][5]. Both software-counted number of flip-flop switching and data generated by XPower were collected and then correlated with software-counted number of flip-flop switching for a hypothetical key.

VI. Experiments and Results

Experiments carried out in a simulated environment showed that DPA of AES is possible in the first encryption round for KeyAdd, ByteSub or ShiftRow operations. The next operation MixColumn causes an inter-byte interaction, thus introducing decorrelation during partial key guessing. To determine the cipher key for KeyAdd operation it was enough to perform 500 encryptions as shown in Fig. 2. For ByteSub operation it was enough to do 200 encryptions, because its non-linearity causes faster decorrelation for wrong hypothesis as shown in Fig. 3.

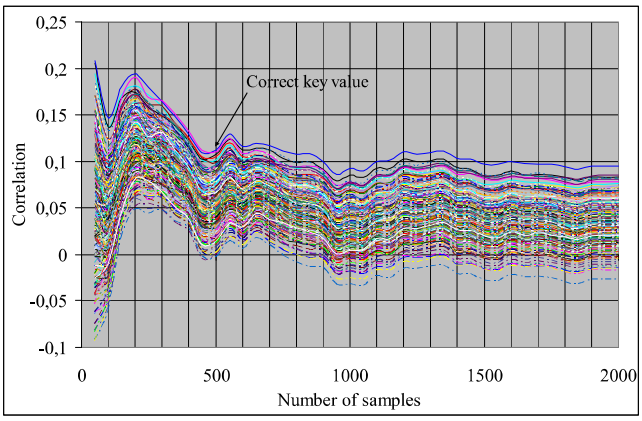

Figure 2. Correlation for KeyAdd operation in simulated environment.

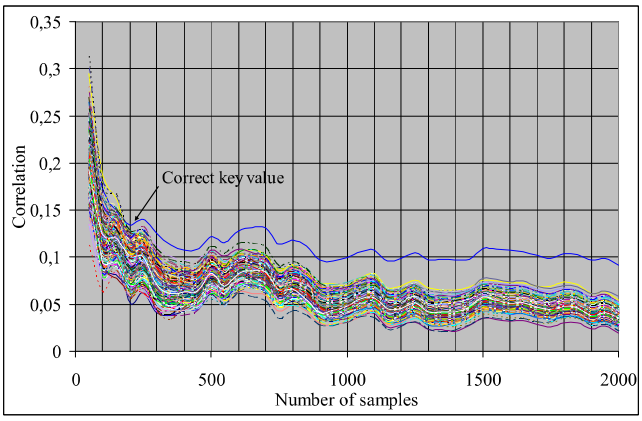

Figure 3. Correlation for ByteSub operation in simulated environment.

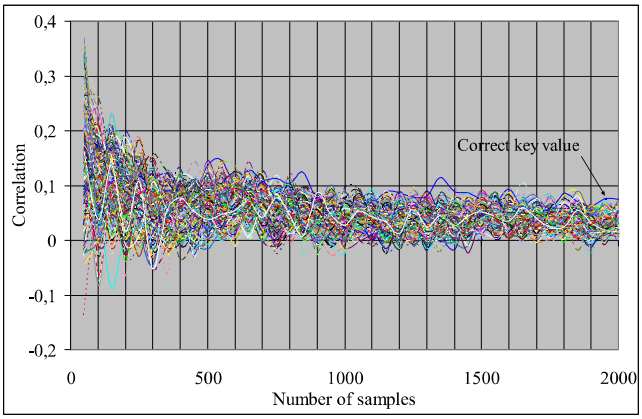

Figure 4. Correlation for ByteSub operation with random noise SNR=8.

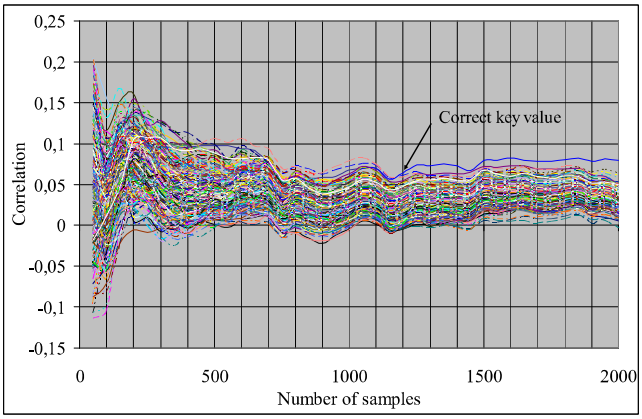

Figure 5. Correlation for ByteSub operation with quantitive noise SNR=4.

The results show that register stage should be applied only after MixColumn operation. Operations KeyAdd, ByteSub and ShiftRow should be implemented by combinatorial logic, what ensures better power dispersion in time.

Next the impact of measure precision on DPA results was investigated. There were performed 2000 encryptions with random and quantitive noise model introduced. Random noise characterizes real measure environment while quantitive noise exists during analog to digital conversion. Fig. 4 and Fig. 5 illustrates that DPA for ByteSub operation enables random noise filtering from signal to noise ratio SNR=8 and quantitive noise from SNR=4.

The results confirm that DPA is a very strong analysis tool, able to filter out the noise even for small values of SNR and does not require high resolution measures.

Two versions of AES algorithm were implemented. Standard version uses 3931 configurable logic block (CLB) slices and minimal clock period is 15.5 ns. Modified version requires 6831 CLB slices and minimal clock period is 18 ns. Almost twofold higher resource usage is the effect of complementary logic generation for encryption module (for key expansion module complementary logic was not generated). Slightly lower maximal frequency is caused by using more reprogrammable connections in FPGA.

XPower simulator generates power consumption results with resolution determined by the user. Too low resolution cumulates power values of several operations to one period. Too high resolution disperses power values of one operation to several periods. After experiments simulation resolution was set to flip-flop delay (about 1 ns), what ensures that power values of all flip-flop switching are collected in one period.

During XPower simulated data analysis, precise points of time should be determined for all operations. In this paper two kinds of power simulations were carried out. The first one was performed after mapping, considering time of flip-flop switching and the second one was performed after place and route process, additionally considering signals propagation delay. Points of time calculated this way were used for getting power values from XPower simulator. In this case corresponding delays are 1.2 ns for simulation after mapping and 1.8 ns for simulation after place and route. The example of timing is shown in Fig. 6.

Experiments also showed limitations of XPower simulator. Maximal VCD file size is limited to 2 GB and maximal simulation time should be less than 2.147 ms, what makes it impossible to perform a large number of encryptions.

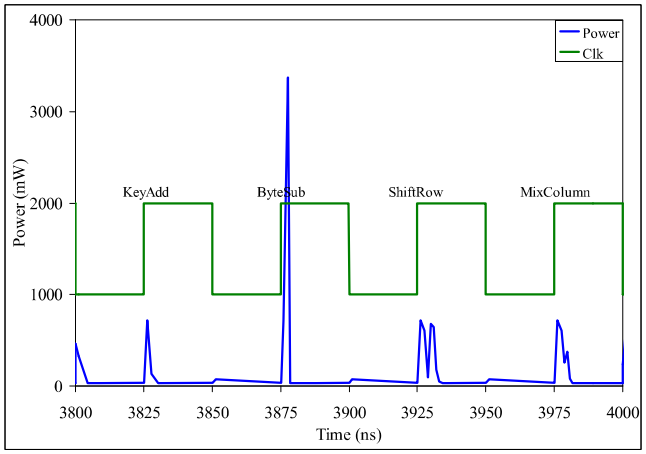

Figure 6. Power consumption of standard implementation during the 1st round of AES encryption.

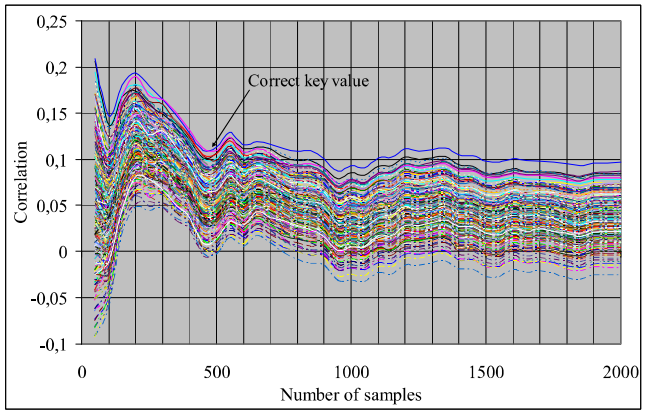

Figure 7. Correlation for KeyAdd operation of standard implementation.

DPA was performed for both AES implementations. For the standard implementation, DPA showed that the cipher key can be easily determined by performing N=2000 encryptions and analyzing power consumption of KeyAdd, ByteSub or ShiftRow operations in the first encryption round, as shown in Fig. 7. For the modified implementation, power consumption during flip-flop switching was not constant, but oscillated within 5% range. This oscillation could be caused by additional logic switching or signal propagation from flip-flop. DPA for N=2000 encryptions showed no correlation between the power consumption and a key value. This proves proposed design method to be effective. Detailed research results will be given during presentation.

VII. Conclusion

This paper describes analytic power of DPA. It was shown that successful attack can be mounted for ByteSub operation in AES algorithm using N=200 encryption operations. DPA resistant hardware implementation method was proposed, which is the first published combination of power equalization for synchronous circuits and combinatorial logic. Both standard and modified versions of AES were implemented and power consumption was simulated. XPower simulation tool was used for the first time for DPA attack. Based on XPower simulated data it was proven that the proposed design method effectively prevents DPA. Despite additional resources and higher power consumption, the method described in this paper can be used for DPA resistant hardware implementation for any encryption algorithm.

References

[1] H. Bar-El, "Introduction to side channel attacks", White Paper, Discretix Technologies Ltd., Netanya, Israel, 1999.

[2] P. Kocher, J. Jaffe, B. Jun, "Introduction to differential power analysis and related attacks", Technical Report, Cryptography Research Inc., San Francisco, California, 1998.

[3] P. Kocher, J. Jaffe, B. Jun, "Differential power analysis", Advances in Cryptology: Proceedings of CRYPTO-99, LNCS 1666, pp. 388-397, Santa Barbara, California, August 15-19, 1999.

[4] S. Ors, F. Gurkaynak, E. Oswald, B. Preneel, "Power-analysis attack on an ASIC AES implementation", International Conference on Information Technology: Coding and Computing, pp. 546-552, Las Vegas, Nevada, April 5-7, 2004.

[5] M. Pierson, B. Brady, "Low cost differential power analysis (DPA) resistant crypto-chips", EE244 Project Presentation, University of California, Berkeley, California, 2006.

[6] L. McDaniel III, "An investigation of differential power analysis on FPGA-based encryption systems", Master of Science Thesis, Virginia Polytechnic Institute, Blacksburg, Virginia, May 29, 2003.

[7] K. Tiri, I. Verbauwhede, "Synthesis of secure FPGA implementation", International Workshop on Logic and Synthesis (IWLS 2004), pp. 224-231, Temecula Creek, California, June 2-4, 2004.

[8] L. Benini, A. Macii, E. Macii, E. Omerbegovic, M. Poncino, F. Pro, „Energy-aware design techniques for differential power analysis protection", Design Automation Conference (DAC 2003), pp. 36-41, Anaheim, California, June 2-6, 2003.

[9] M. Akkar, C. Giraud, "An implementation of DES and AES, secure against some attacks", International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2001), LNCS 2162, pp. 309-318, Paris, France, May 13-16, 2001.

[10] J. Daemen, V. Rijmen, "AES Proposal: Rijndael", Proceedings of First AES Candidate Conference, Ventura, California, August 20-22, 1998.