Combinatorial Logic Circuitry as Means to Protect Low Cost Devices against Side Channel Attacks

Frank Vater, Steffen Peter, and Peter Langendörfer

IHP
Im Technolgiepark 25, 15236 Frankfurt (Oder), Germany
{vater, peter, langend}@ihp-microelectronics.com

Abstract. In this paper we present a clock frequency watch dog that can be realized using a digital standard CMOS library. Such watch dog is required to prevent clock speed manipulations that can support side channel attacks on cryptographic hardware devices. The additional area and power consumed by the watch dog for an AES hardware accelerator are 4,200μm^2 and 2nJ per 128 bit respectively. The physical properties and the use of standard CMOS technology ensure extremely low additional production cost. Thus, our approach is very well suited to improve the security of low cost devices such as wireless sensor nodes.

1 Introduction

Wireless sensor networks (WSN) are becoming more and more popular, and the area of their application is constantly increasing. Their use in military and homeland security applications obviously demands a high level of security. This holds true also for other application areas such as vehicular scenarios [5]. The fact that wireless sensor nodes are exposed to potential attackers requires means to protect them against side channel attacks that exploit physical access to the devices, e.g. against power analysis attacks. These protection mechanisms are normally quite expensive and therefore not used in low cost devices. However, high end smart cards are already equipped with initial protection mechanisms [5].

The main contribution of this paper is the introduction of a circuit that is capable to detect manipulations of the clock frequency, which can be used to simplify differential power analysis attacks. The major benefits of this circuit are extremely small area and energy consumption i.e. 3.5 per cent more energy and approximately 1.0 per cent of the silicon area of the AES (Advanced Encryption Standard [6]) core we used for our experiments. Additionally our approach is a combinatorial logic so that it can be manufactured in a pure CMOS design, which dramatically reduces the costs on integrating this kind of a clock frequency watch dog into crypto hardware. By this, the proposed approach turns formerly unprotected devices to be partly protected devices of level 3 according to the FIPS 140-02 [7].

The rest of this paper is structured as follows. In section 2 we give a overview of the side channel issue and present previous solutions. The AES crypto device that we are evaluating and a successful side channel attack are described in section 3. The a low cost countermeasure is presented and the results are discussed before the paper concludes.

2 Related Work

Side channel attacks as means for revealing secret information of a cryptographic implementation have a long history. In the 1960s intelligence exploited sound and electromagnetic emissions to deduct secret information from cryptographic typewriter [19]. In 1985 van Eck published an analysis of electromagnetic emanations of computer devices [17]. Nowadays timing and power consumption are the major side channels that can be used against cryptographic devices. The first timing attack was published in 1996 [11]. If the processing of a '1' takes a different amount of time than for a '0', key information can be deducted. With a strict separation of data path and control path, as it is possible for modern cryptographic algorithms, that attack does not pose a serious threat anymore. In contrast, power attacks have been a major threat for cryptographic devices. Two kinds of power attacks can be distinguished: the simple power analysis (SPA) and the differential power analysis (DPA). The SPA tries to deduct information directly from the power trace. It can be applied if power consumption for different keys differs a lot. An example for an SPA attack on AES is shown in [12]. The DPA [10] analyses the power consumption of hundreds up to millions of operations and exploits smallest differences of the power trace in order to deduct information. DPA attacks on AES implementations are described in [13][14][4]. Several countermeasures have been presented in [2][3]. A common approach to prevent such analysis is randomization or masking of the performed operations so that small operational differences are covered by intentionally inserted noise. Another approach is the avoidance of any power side channel information. However, solutions for constant power dissipating logic [16, 8] require additional logic gates, additional power consumption and individual design libraries that render these approaches very expensive if not infeasible. In particular for mobile environment, smart cards or in wireless sensor networks where production costs and power consumption must be kept as low as possible such ideas are not practicable.

Additionally most approaches have weaknesses if the circuit is forced into exceptional states. Fault injection, inserted glitches or tampered clock speeds produce errors that finally reveal secret information as described in [13]. Many attacks require a tampered clock frequency in order to force faults. Countermeasures are embedded clock generator or PLLs [15], which are quite expensive and require a lot of additional energy. Thus, we are looking for a low cost mechanism that ensures that the circuit is driven at correct frequency.

3 AES chip

In this section we first describe the architecture of the evaluated cryptographic accelerator. Afterwards a DPA attack to the AES accelerator is presented.

The IHP Dual^2-Crypto-Chip is a hardware accelerator for the symmetric cipher algorithm AES (Advanced Encryption Standard) and the asymmetric cipher algorithm ECC (Elliptic Curve Cryptography). The chip was manufactured in IHP 0.25μm CMOS technology [9]. Two different interfaces for the connection of the ASIC with a PC were implemented. One is 16-bit PCCARD-Interface, and the other one is a 32-bit PCI-interface. Figure 1 shows the schematic of the interfaces and the AES block. We implemented both interfaces on the demonstration device pursuing the goal to test the device on as many computers and environments as possible. It should be mentioned that the PCI interface gets a system generated clock (33 MHz) while the clock for the PCMCIA interface has to be generated by an external quartz oscillator. That is why we perform our security analysis with the PCMCIA interface and do not consider PCI.

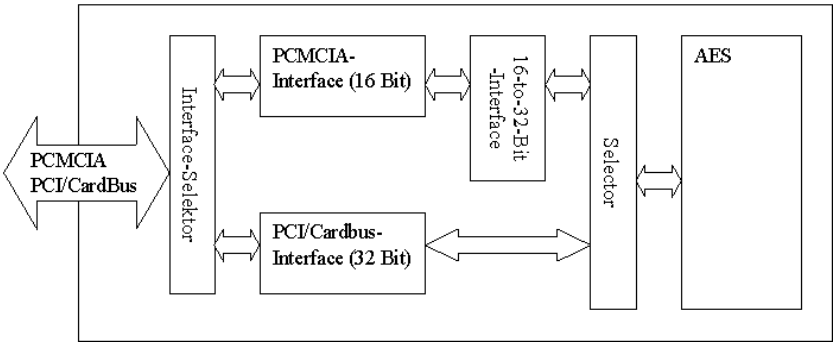

Fig. 1. Block diagram of the IHP Dual^2 crypto chip.

3.1 Description of AES implementation

From side channel security aspects the most interesting block of the IHP Dual^2-Crypto-Chip is the AES block. The AES block investigated in this paper is a similar preceding version of the AES implementation comprehensively described in [18].

The version considered in this paper needs 90 clock cycles to encrypt or decrypt a 128 bit data word using a 128 bit key. As usual the implementation consists of a key generator, an algorithm part and a controller block, which coordinates the key and the algorithm block (see Fig. 2). AES [6] is a block cipher protocol that encrypts each block (i.e. 128 bit) not only in one steps

but in several rounds (10). Based on the initial key the AES algorithm needs a new key for every round. The new round key is computed in the key generator block. Once per round the round key is applied in the algorithm block where key and data is combined to the output stream. In the algorithm block the sub-algorithms add key, shift row, substitution box and mix column are performed once per round. Since the implementation focusses very small low energy devices, the operation are executed successively and not in parallel.

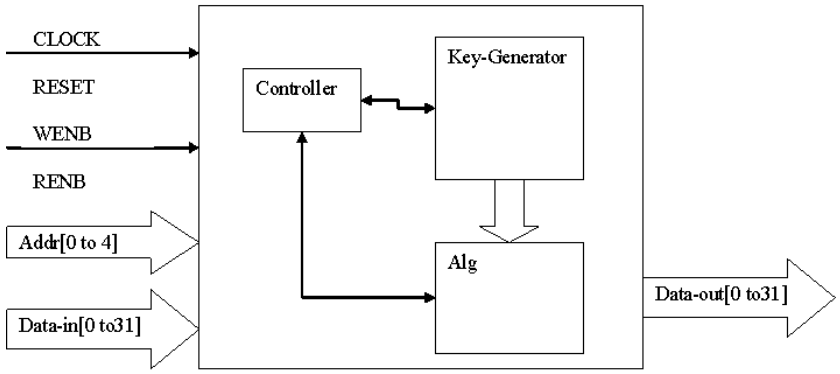

Fig. 2. Block diagram of the AES block.

3.2 Successful DPA attack on the chip

Considered the AES key is stored on the chip, protocol is known, data streams are fully accessible and the chip can be fully accessed, we want to deduct the internal key exploiting the power consumption as side channel. We also use the fact that we can set the input stream.

In order to deduct the key we observe the initial round of the AES. In other words, we are interested in what the chip is doing when our input data and the stored key are combined for the first time. We are not interested in the actual result of the encryption. In the first AES round the key is bitwise logically XORed with the input data word. The result is stored in a register. The idea is to observe the power that is required for writing the data into the register.

Our approach is to select a constant input value (e.g. all bits '0'). Then we toggle one bit. The theory is that writing a '1' requires more energy than writing a '0', because after reset all registers are set to zero [1]. The power consumption

[1] That all registers are set to zero after a reset is a property of the circuit that severely weakens the side channel resistance.

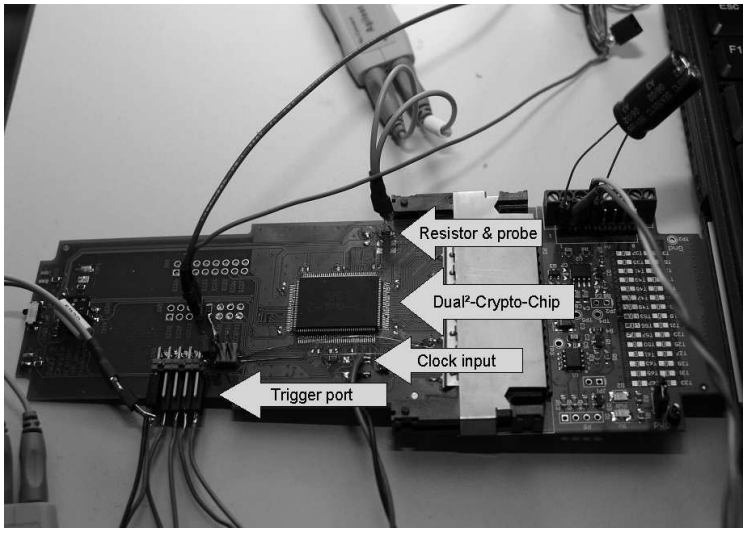

Fig. 3. The tested Dual^2 crypto card with attached measurement equipment.

can be monitored with an oscilloscope (Agilent 54854A 20GSa/s [1]). Since the power cannot be measured directly, we use a resistor in the power line that generates a voltage drop equivalent to the supply current. Based on our assumption, repeatedly executing the measurement we could filter the noise from other blocks and determine the key. The test environment is shown in Figure 3. It shows a PCMCIA card with the mounted crypto chip, resistor and a port for an external adjustable clock.

In our experiment in average we could deduct 113 bits out of the 128 bit AES key within 850 iterations of the measuring process. Please consider that even without sophisticated attack statistically the number of correctly guessed bits of the key is already 64.

Our described attack has similarities to the one presented in [14]. However, [14] described an ASIC with separated power supplies for core and I/O-pads, what significantly improves the observability of the core. Our investigated ASIC has only one power supply for both core and pads, what is a more practical scenario for very small and cheap devices as they are applied in mobile environments.

4 Low cost countermeasures

Initially we expected that higher capacities of the I/O pads would better cover the action inside the chip. But with the attack assumptions, that are indeed based on the knowledge of the chip design, we could deduct the internal key quite easily. It is a bit surprising and poses the question for efficient countermeasures. In literature several approaches have been described. Many of them require very

costly changes in design process and design libraries what is not acceptable in most cases, in particular for cheap small devices.

That is why we are looking for low cost approaches that need as little additional silicon as possible, do not imply much higher power consumption and do not require huge changes in the design chain.

4.1 Clock stabilization approach

One thing we could observe in our experiments is that lower clock frequencies improve the observability of the chip significantly. In [14] the clock frequency of the device was reduced to 2 MHz. Our results were also achieved with clock frequencies far under the intended chip specifications (i.e. 10 MHz). We assume that due to capacities on the chip the observability of the internal transitions is reduced with higher clock frequencies - a property that is strengthened by the combined power supply for pads and core.

A straightforward solution for this issue is to ensure that the device is always driven at the intended clock frequency. A standard approach is to integrate a PLL (phase looked loop) into the chip [15]. Though it indeed solves the problem, it is quite expensive. A PLL requires many analogue design elements that are not available in a low cost pure digital design process, so it implies increased costs in developing process and especially for manufacturing.

In the following we propose an approach that can be realized by standard CMOS digital libraries on chip, and is also applicable for FPGAs. The idea is that the gate delay of standard gates is known. With a chain of such gates (e.g. an inverter chain) it is possible to construct a specific delay in the circuit. If the clock edge compared to the output of the inverter chain comes too early or to late, the circuit concludes that the clock frequency has been tampered and causes an invalidation of all results. Since gates in real circuits are not ideal and are depending on temperature and voltage, it is necessary to implement an additional margin. For example if the ideal clock is 50 MHz then one could implement a tolerance of 10 per cent, so that 45 - 55 MHz are still accepted. Figure 4 shows the schematic of such a circuit. The clock is connected to an inverter chain. Two connected inverters are one inseparable buffer. From the inverter chain we fork two signals. The first corresponds to the lower time limit and the second (that is the end of the chain) is the upper threshold. The number of required buffers is computed with following equation:

$$buffers_{short} = \frac{100 - p}{200} \cdot \frac{period}{delay_{per\ buffer}} \tag{1}$$

$$buffers_{long} = \frac{100 + p}{200} \cdot \frac{period}{delay_{per\ buffer}} \tag{2}$$

Where p is the percentage of tolerance.

In our example circuit we want to ensure a clock frequency of 50 MHz with 10% tolerance while the delay per buffer element is 0.0914ns in the considered

0.25μm CMOS technology. Then the required number of buffers needed to supervise the lower boundary of the allowed clock frequency interval is

$$buffers_{short} = \frac{100 - 10}{200} \cdot \frac{20\text{ns}}{0.0914\text{ns}} = 98 \tag{3}$$

and the number of buffers for the upper threshold is 120.

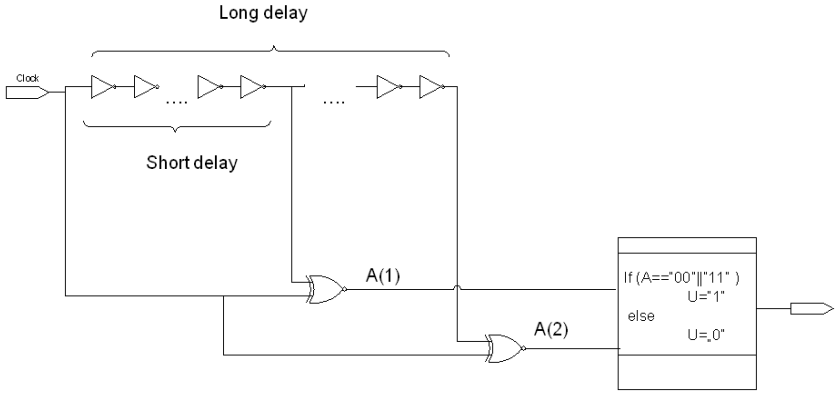

Fig. 4. Structure of the clock watch dog element: Signals A1 and A2 are forked from the delay chain corresponding to the computed threshold time. If A1 equals A2 at toggling clock, it is an indicator for a tampered clock.

As shown on Figure 4 each of the forked signals is XNORed with the actual clock signal. In the evaluation logic (box at the right) both XNOR results are compared. We want to see that in the moment the clock toggles, the first forked signal (A1) has been toggled and the second signal not yet. That is, if A1 is '0' and the A2 is '1', or vice versa, then the clock frequency is correct. If in contrast both signals are identical then the clock speed is either too slow or to high because it means that the clock came before or after both threshold signals changed. In case such condition is recognized, the 'untouched' signal is set to '1' what causes an invalidation of the cryptographic circuit. If the clock is assumed as correct the 'untouched' signal is '0'.

4.2 Results

The result of our experiment is shown in Figure 5. If the clock frequency is too slow (on the right) the untouched signal is '1' and consequently the cryptographic operation will not be executed. In the area from 45 to 55 MHz untouched_clk is '0', i.e. it is correct, and higher clock frequencies result in a '1'. However, in the figure it can be seen that higher frequencies sometimes are mistakenly recognized as correct. It happens when the applied clock frequency is a multiple of the correct frequency. That is still an open issue in our approach.

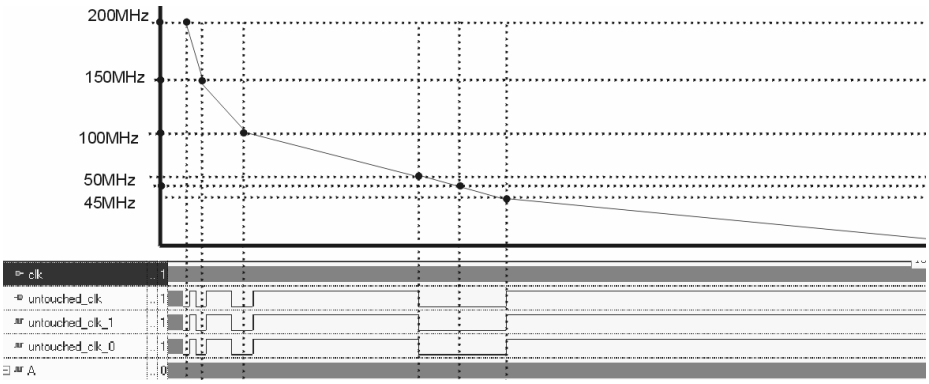

Fig. 5. Waveform of simulated watch dog circuit. On the top is the clock frequency. The row 'untouched_clk' shows the (low active) result of the watch dog. Is it '0' the clock is assumed to ok. Is it '1' it is an indication for a tampered clock.

In order to evaluate power and area consumption we integrated our approach into our in house AES implementation. For the IHP 0.25μm CMOS technology the chain of 240 inverters requires 3810μm^2 and the needed additional logic requires 374μm^2. The AES design has an silicon area of 430,000μm^2, thus the additional 4,200μm^2 are reasonable.

The AES encryption of 128 bit with the initial design requires energy consumption of 57 nJ. The same design with secured clock frequency needs 59 nJ, i.e. merely 3.5 percent more energy.

4.3 Discussion and further work

Figure 5 has already shown that high clock frequencies could not be properly detected. Another potential problem is that the production yield will be decreased if the variance of the manufacturing process leads to deviations from the ideal gate delays. It is considerable to implement a delay element that can be calibrated on time after production.

However, beside production deviations, voltage and temperature interfere gate delay times. Lower voltage and higher temperature cause slower circuits. That effect can cause false alerts but can also be exploited to tamper the clock frequency. The PVT (production, voltage, temperature) effects are a serious issue that we have to evaluate in practice after manufacturing silicon circuits with integrated clock watch dog.

Indeed, the proposed mechanism is not a one size fits all solution that prevents from all potential side channel attacks. It is merely a stand alone solution that solves the specific issue of clock frequency manipulation when DPA is used. Thus, it is a piece of a puzzle which can provide complete protection as soon as it is completed. In other words additional means that have to be invented and investigated are still essentially needed. Some potential means are inverse data

paths, an increased level of random noise and fixing of design flaws that increase the observability. Also additional capacitors in the pad path could reduce the leaked side channel information in particular if it is guaranteed that the design is driven on sufficiently high clock speeds.

5 Conclusions

In this paper we have presented a clock frequency watch dog realized using combinatorial logic. Our approach causes minimal additional power consumption and negligible area overhead. For our AES design these costs were 3.5 per cent more energy and approximately 1.0 per cent more area. Our simulation results clearly show that the proposed design works very well if a reduction of the clock frequency has to be detected. In the opposite case not all manipulation have been detected. Due to the fact that especially a clock frequency reduction bears a serious risk with respect to DPA, we think that our result is very encouraging.

With the technology presented in this paper we provide a first step towards the realization of partly protected low-cost devices. According to the FIPS 140-02 [7] classification devices using our mechanism can even be grouped into level 3: "devices, that implement tamper evidence and tamper response mechanisms".

To summarize, our approach is a suitable means to significantly improve the security of wireless sensor network for example.

Acknowledgement

This work was partially funded by the German Ministry of Education and Research under grant 01AK060B.

References

1. Agilent Technologies, http://www.home.agilent.com/USeng/nav/-35813.536882578/pd.html. *54854A Infiniium Oscilloscope and InfiniiMax 1132A Probing System*, 2006.
2. Mehdi-Laurent Akkar and Christophe Giraud. An implementation of des and aes, secure against some attacks. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop*, pages 309–318, 2001.
3. J. Blomer, J. Merchan, and V. Krummel. Provably secure masking of aes, 2004.
4. V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier. Electromagnetic side channels of an fpga implementation of aes, 2004.
5. Augusto Julio Domingues Casaca and Dirk Westhoff. Ubisec&sens d0.1 "scenario definition and initial threat analysis". Technical report, June 2006.
6. FIPS. *Advanced Encryption Standard (AES)*. National Institute for Standards and Technology (NIST), November 2001.
7. FIPS. *Security Requirements for Cryptographic Modules*. National Institute for Standards and Technology (NIST), May 2001.
8. J. Fournier, S. Moore, H. Li, R. Mullins, and G. Taylor. Security evaluation of asynchronous circuits, 2003.

9. Innovations for High Performance microelectronics, http://www.ihp-ffo.de/24.0.html. *IHP microelectronics: technology*, 2006.
10. Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.
11. Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. *Lecture Notes in Computer Science*, 1109:104–113, 1996.
12. Stefan Mangard. A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion. In *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea*, volume 2587 of *Lecture Notes in Computer Science*. Springer, 2003.
13. Stefan Mangard, Norbert Pramstaller, and Elisabeth Oswald. Successfully Attacking Masked AES Hardware Implementations. In *Cryptographic Hardware and Embedded Systems – CHES 2005, 7th International Workshop, Edinburgh, Scotland*, volume 3659 of *Lecture Notes in Computer Science*. Springer, 2005.
14. Siddika Berna Örs, Frank Gürkaynak, Elisabeth Oswald, and Bart Preneel. Power-analysis attack on an asic aes implementation. In *ITCC '04: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04) Volume 2*. IEEE Computer Society, 2004.
15. S.W. Smith and S.H. Weingart. Building a high-performance, programmable secure processor, tech. report rc 21102. Technical report, IBM T.J. Watson Research Center, 1998.
16. Kris Tiri and Ingrid Verbauwhede. Design method for constant power consumption of differential logic circuits. In *DATE '05: Proceedings of the conference on Design, Automation and Test in Europe*, pages 628–633, Washington, DC, USA, 2005. IEEE Computer Society.
17. Wim van Eck. Electromagnetic radiation from video display units: An eavesdropping risk, 1985.
18. Frank Vater and Peter Langendörfer. An area efficient realization of aes for wireless devices. *it - Information Technology*, 3, 2007.
19. Peter Wright. Spycatcher: The candid autobiography of a senior intelligence officer, 1987.