ELectronics EXpress

Countermeasure against fault sensitivity analysis based on clock check block

Jinbao Zhang[1a)], **Ning Wu**[1], **Fen Ge**[1], **Fang Zhou**[1], **and Xiaoqing Zhang**[2]

[1] *College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

[2] *College of Electrical Engineering, Anhui Polytechnic University, Wuhu 241000, China*

a) *zjb4050811@126.com*

Abstract: Fault sensitivity analysis (FSA), as a new type of fault attacks, has been proved a serious threat to the security of cryptographic circuits. It exploits the fault clock to obtain fault sensitivity so as to recover the secret keys. According to the characteristics of FSA, this letter proposes a countermeasure to thwart FSA. We first design a clock check block (CCB) to detect the fault clock which is necessary for carrying on FSA, and then design an enable signal module to change the output of the cryptographic circuit once there is any clock glitch be detected. The implementation results show the proposed CCB can detect the abnormal clock successfully, and our countermeasure can effectively resist FSA. This letter also investigates the hardware overhead of the proposed countermeasure. Compared with these existing countermeasures, although our countermeasure consumes a little more hardware resources due to the extra 495 gates, it has better versatility.

Keywords: fault sensitivity analysis, fault attack, AES, countermeasure

Classification: Integrated circuits

References

[1] J. Daemen and V. Rijmen: *The Design of Rijndael: AES—The Advanced Encryption Standard* (Springer-Verlag, Berlin, Germany, 2002).

[2] N. F. Ghalaty, *et al.*: "Differential fault intensity analysis," Proc. IEEE Int. Conf. FDTC Workshops (2014) 49 (DOI: 10.1109/FDTC.2014.15).

[3] A. Barenghi, *et al.*: "Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures," Proc. IEEE **100** (2012) 3056 (DOI: 10.1109/JPROC.2012.2188769).

[4] Y. Li, *et al.*: "Fault sensitivity analysis," Conf. CHES Workshops, Proc. LNCS **6225** (2010) 320 (DOI: 10.1007/978-3-642-15031-9_22).

[5] S. Endo, *et al.*: "A silicon-level countermeasure against fault sensitivity analysis and its evaluation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. **23** (2015) 1429 (DOI: 10.1109/TVLSI.2014.2339892).

[6] A. Moradi, *et al.*: "On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting," CHES 2011, LNCS **6917** (2011) 292 (DOI: 10.1007/978-3-642-23951-9_20).

[7] Y. Li, *et al.*: "An extension of fault sensitivity analysis based on clockwise

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018

collision," Inscrypt 2012, LNCS **7763** (2012) 46 (DOI: 10.1007/978-3-642-38519-3_4).
[8] A. Wang, *et al.*: "Fault rate analysis: Breaking masked AES hardware implementations efficiently," IEEE Trans. Circuits Syst. II, Exp. Briefs **60** (2013) 517 (DOI: 10.1109/TCSII.2013.2268379).
[9] S. Endo, *et al.*: "An efficient countermeasure against fault sensitivity analysis using configurable delay blocks," Proc. IEEE Int. Conf. FDTC Workshops (2012) 95 (DOI: 10.1109/FDTC.2012.12).
[10] N. F. Ghalaty, *et al.*: "Analyzing and eliminating the causes of fault sensitivity analysis," Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE) (2014) 1 (DOI: 10.7873/DATE.2014.217).
[11] L. Wang, *et al.*: "An implementation of fast-locking and wide-range 11-bit reversible SAR DLL," IEEE Trans. Circuits Syst. II, Exp. Briefs **57** (2010) 421 (DOI: 10.1109/TCSII.2010.2048379).

1 Introduction

With the rapid development of computer and communication technology, the requirement of security for information storage, processing and exchange becomes more and more urgent. Cryptographic chip, as one of the important carrier for secret information, its security implementation has drawn much attention. Although the cryptographic algorithm—Advance Encryption Standards (AES) [1] has been proved to withstand most conventional attacks, its hardware implementation has exhibited vulnerabilities to physical attacks. Fault attack [2, 3], as one of the well-known physical attacks, which exploits the faulty outputs to crack the key of cryptographic modules, has became a serious threaten to the security of cryptographic implementation.

Fault sensitivity analysis (FSA) is a new type of fault attacks, and it has been proved a serious threat to the security of cryptographic circuits. In 2010, Li et al. first proposed the concept of FSA and applied it to AES [4]. FSA utilizes the dependency between the secret information and fault sensitivity (FS) of target circuit to break the key. FS means the critical condition in which faulty outputs beginning to appear, and it can be measured by a temporal over clocking caused by a glitch clock. The adversary carries out several cryptographic operations while gradually increasing the frequency of glitch clock with the same plaintext, and the frequency is defined as the FS where a faulty output occurring for the first time [5].

In 2011, A. Moradi et al. extended FSA to masked AES implementation by combining FSA with collision attack [6]. Soon after that, paper [7] presented a Clockwise Collision Fault Sensitivity Analysis (CC-FSA) attack on unmasked AES. The authors stated that if two consecutive cycles' inputs were identical in an iterative AES implementation, then the second cycle's setup time would be extremely short. Later, Wang et al. presented an improved CC-FSA, and they succeeded in attacking a masked serial AES S-box implementation using the improved method [8].

In recent years, researchers have already realized the serious threat of FSA and proposed some countermeasures. In 2012, S. Endo et al., proposed a counter-

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018

IEICE Electronics Express, Vol.15, No.11, 1–6

measure using a configurable delay block (CDB) to resist FSA [9]. Later work, namely [5], the main ideas was to use the CDB to break the fixed delay of the target circuit, and then destroy the dependency between FS and secret data. In 2014, paper [10] further presented an analysis of the structure of a design and its relation to FS for the first time, the authors pointed out the arrival time of the signals and the number of logic levels of FS gates were two factors that mainly affect the FS of a design. The countermeasure proposed in [10] was based on inserting delay elements to mask the two factors so as to make FS uniform when the design dealing with different signals.

According to the characteristics of FSA, and on the basis of previous studies, we propose our countermeasure against FSA. Different from the methods mentioned in [5] and [10] which are all based on the critical path replica of serial delay elements, our countermeasure only needs to use a clock check block (CCB) module to check whether or not the clock signal is abnormal, and then judge whether the target circuit is already under the threaten of FSA.

2 Principle of the proposed countermeasure

As mentioned above, when carrying out FSA, an abnormal clock needs to be caused in the target circuit. In view of this, we design a CCB to detect whether the clock signal is abnormal and determine whether there is a FSA occurred.

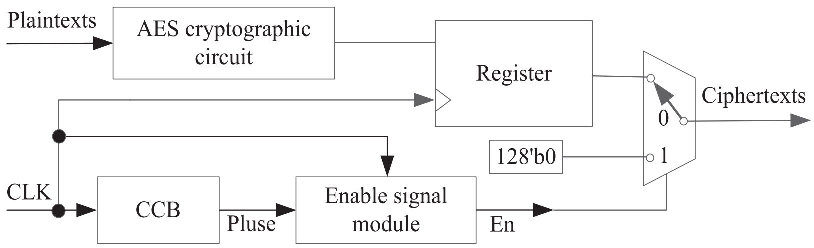

Fig. 1. Concept of countermeasure against FSA based on CCB

Fig. 1 describes the block diagram of proposed countermeasure where the cryptographic module consists of CCB, Enable signal module, AES cryptographic circuit, a register, and a multiplexer.

The clock signal CLK will be checked when having passed through the CCB module while the plaintexts of AES cryptographic circuit are processed. Once an abnormal clock has been detected, Pluse (the output signal of CCB) will not be zero. Then the Enable signal module will produce non-zero enable signal En. Finally, the cryptographic module will output a series of 128’b0. Contrary, the cryptographic module will output the real ciphertexts.

The specific structure for our CCB is as shown in Fig. 2. CCB consists of a BUFG, a Delay module and a AND gate. Because it is not allowed to process CLK directly, so CLK should pass through a BUFG first, the purpose is to reduce the transmission delay and improve the driving capability of the signal. Then CLK_{BUFG} (the output of BUFG) will be delay half cycle when having passed through the Delay module, and CLK_{delay} is defined as the output signal of Delay module.

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018

IEICE Electronics Express, Vol.15, No.11, 1–6

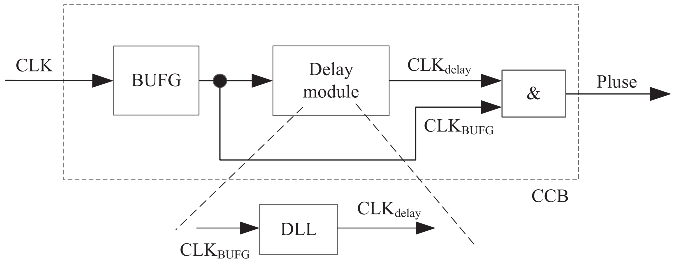

Fig. 2. Specific structure for CCB

We define the output signal of CCB Pluse $= CLK_{BUFG}$ & CLK_{delay}. Once the value of Pluse $\neq 0$, it indicates that an abnormal clock has been detected. Therefore, we can judge whether there is an attack threat by observing the value of Pluse.

Due to the cycle time shift and phase shift for the output signal of delay-locked loop (DLL) have the relationship as shown in Table I, besides, DLL has the advantages of good stability and no clock jitter accumulation, so we use a DLL to achieve the function of Delay module: that is, make the input signal delay for half a clock cycle.

Table I. The relationship between the cycle time shift and phase shift for the output clock

Phase (degree)	%Cycle time shift
0	0%
90	25%
180	50%
270	75%

If we use the "Pluse" signal controls directly the multiplexer that sends either the calculated value of the AES (after a register) or zero, since the "Pluse" signal is a glitch, the random number is sent for a very short time, thus it will not be latched by the next register. Therefore, Pluse must be latched, and the Enable signal module just achieves the latch function for Pluse. The detail implementation process is as described in Table II.

Table II. Enable signal module

Algorithm: Enable signal module	
Input: Pluse, CLK. Output: En.	
(1) flag <= ~flag.	(posedge Pluse)
(2) t_flag <= flag.	(posedge CLK)
(3) En <= 1, if (t_flag $\neq$ flag).	(posedge CLK)

So, through Enable signal module, every value of Pluse will be latched for a clock cycle.

3 Experiment and evaluation

As can be seen from Section 2, the performance of our countermeasure mainly depends on the signal detection capability of CCB, which depends on the signal latching capability of DLL used. In other words, the frequency of glitch must be included in the working frequency of our DLL, only in this way, DLL can effectively lock the abnormal clock, and our CCB can detect the abnormal clock

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018

effectively. From the existing articles, such as [4, 5, 8] and so on, we can know the frequency range of glitch for effective obtaining FS data from target AES circuit is 90 MHz~700 MHz. Based on this, and comprehensive consideration of the area overhead, we adopt the all-digital DLL proposed in [11], which operates in the frequency range of 30 MHz~1 GHz and about 4000 gates (NAND eq) area overhead. Because the DLL proposed in [11] contains a BUFG, so our CCB does not require extra BUFG (as described in Fig. 2).

We assume that a FSA occurred on cryptographic circuit, so an abnormal clock needs to be caused in the target circuit, therefore, we only need to verify our CCB can detect the abnormal clock effectively once an abnormal clock being caused, and once any abnormal clock being detected, the Enable signal module can output normal enable signal “En”, which also means that our countermeasure is effective for resisting against FSA.

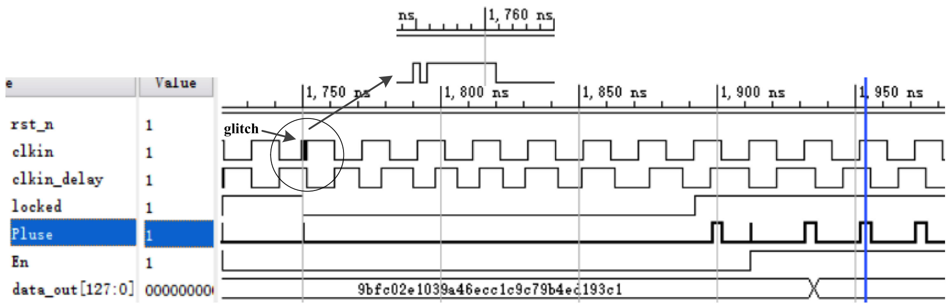

Fig. 3. Experiment for our countermeasure

We conduct several post-synthesis timing simulation experiments by Vivado software and take the detection of maximum boundary glitch (it is assumed the glitch frequency is 1 GHz) an example to verify the detection capability of our CCB, as shown in Fig. 3. The clock cycle is 20 ns. rst_n and clkin represents the reset signal and the clock signal mixed giltch, respectively. Locked represents the latch signal of DLL, and clkin_delay is the output of DLL. Pluse represents the output of CCB, and En represents the output of Enable signal module.

As can be seen in Fig. 3, when Locked is high, the DLL produces stable output clkin_delay. Once there is a glitch in normal clock, the glitch signal will have a stable impact on the output of DLL, then Pluse (the output of CCB) will periodic change, thus En will be high (after Locked signal turns to high). Indeed, as long as the frequency of glitch is between 30 MHz and 1 GHz, our CCB can detect the abnormal clock, and we can get enable signal “En”. In this case, our designed circuit (as described in Fig. 1) will output zero, this will prevent the adversary from obtaining FS data. And due to the frequency range of glitch for effective obtaining FS data from target AES circuit is 90 MHz~700 MHz, therefore, it shows that our countermeasure can thwart FSA attack.

For CDB countermeasure, the circuit designed will first store the parameters of the CDB, which are determined based on the results of static timing analysis of target circuit [5, 9]. The countermeasure proposed in [10] was based on inserting delay elements and the number of inserting delay elements also depends on the characteristics of the target circuit. So, if we use CDB countermeasure or the method proposed in [10], the circuit designed will make some change for different

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018

target AES circuit. Compared with these methods, our CCB countermeasure mainly depends on the signal latching capability of DLL used and has little relation to the target AES circuit, so, our CCB circuit designed has better versatility.

4 Hardware implementation

We carry out ASIC implementations using Synopsys Design Complier, SMIC 0.18 µm, 1.8 V. The AES circuit has a common pipelining architecture with S-boxes implemented by adopting look-up table (LUT). Table III shows the hard resource cost of each designed component.

Table III. ASIC implementation: results and analysis

<table>
<tr><th rowspan="2">Components</th><th colspan="3">Area (NAND eq.)</th></tr>
<tr><th>[5] or [9]</th><th colspan="2">This paper</th></tr>
<tr><td>AES cryptographic circuit</td><td>28702</td><td colspan="2">234821</td></tr>
<tr><td rowspan="2">CDB/CCB</td><td rowspan="2">CDB: 1068</td><td rowspan="2">CCB: 4003</td><td>Others: 3</td></tr>
<tr><td>DLL: 4000</td></tr>
<tr><td>Multiplexer</td><td>1882</td><td colspan="2">1076</td></tr>
<tr><td>Pseudo random number generator</td><td>1624</td><td colspan="2">N. A</td></tr>
<tr><td>Enable generator/signal module</td><td>35</td><td colspan="2">25</td></tr>
<tr><td>Extra total overhead</td><td>4609</td><td colspan="2">5104</td></tr>
</table>

As can be seen from Table III, the area of AES cryptographic circuit in this letter is much larger than that used in [5] or [9], this is because the AES structure adopted is different: one is pipelining architecture, and the other is iterative architecture. Due to the configurable delay block needs adopt many multiplexer [5], the area overhead of multiplexer is larger than that used in our countermeasure. Compared with the countermeasure proposed in [5] or [9], our countermeasure consumes about 495 more gates. However, if we don't consider the resource consumption of DLLs and Pseudo Random Number Generator (which can all be included in many systems), our countermeasure would consume much smaller extra area resources (about 1104 gates) than the countermeasure proposed in [5] or [9] (about 2985 gates), especially has better versatility.

5 Conclusion

In order to thwart FSA effectively, in this letter, we first analyze the principle of FSA, and then design a CCB to detect whether there is an abnormal clock in target circuit, so as to judge whether the target AES circuit is already under FSA. The experimental results show that our countermeasure can detect the abnormal clock successfully, and our countermeasure can resist FSA effectively with better versatility at the cost of about 495 more gates.

Acknowledgments

This work is supported by National Natural Science Foundation of China (No. 61774086, 61376025), Natural Science Foundation of Jiangsu Province (No. BK20160806), and Funding of Jiangsu Innovation Program for Graduate Education (No. KYLX16_0373).

DOI: 10.1587/elex.15.20180433
Received April 24, 2018
Accepted May 14, 2018
Publicized May 28, 2018
Copyedited June 10, 2018