Analyzing and Eliminating the Causes of Fault Sensitivity Analysis

Nahid Farhady Ghalaty, Aydin Aysu, Patrick Schaumont
Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, USA
{farhady,aydinay,schaum}@vt.edu

***Abstract*—Fault Sensitivity Analysis (FSA) is a new type of side-channel attack that exploits the relation between the sensitive data and the faulty behavior of a circuit, the so-called fault sensitivity. This paper analyzes the behavior of different implementations of AES S-box architectures against FSA, and proposes a systematic countermeasure against this attack. This paper has two contributions. First, we study the behavior and structure of several S-box implementations, to understand the causes behind the fault sensitivity. We identify two factors: the timing of fault sensitive paths, and the number of logic levels of fault sensitive gates within the netlist. Next, we propose a systematic countermeasure against FSA. The countermeasure masks the effect of these factors by intelligent insertion of delay elements. We evaluate our methodology by means of an FPGA prototype with built-in timing-measurement. We show that FSA can be thwarted at low hardware overhead. Compared to earlier work, our method operates at the logic-level, is systematic, and can be easily generalized to bigger circuits.**

***Index Terms*—Fault Sensitivity Analysis; S-box; Data Dependency; On-chip time measurement; FPGA.**

I. Introduction

Modern embedded cryptographic implementations, such as those in smart cards, are threatened by Side Channel Attacks (SCA). SCA is any attack based on the information leakage gained from the physical implementation of a cryptographic algorithm. This leakage could be either timing information [1], power consumption [2], electromagnetic leaks [3]and other.

Fault Sensitivity Analysis (FSA) is a recently introduced side-channel attack. This attack is based on the concept of fault sensitivity. Fault sensitivity is a condition in which the faulty output begins to show detectable correlations with internal variables of a cryptographic algorithm. The attacker can record the Fault sensitivity information during the fault injection, and can later use this information as a side channel leakage to guess the correct key [4]. For the PPRM1 S-box [5], Li et al. have proposed a correlation between the Hamming Weight of the inputs and the fault sensitivity. Utilizing this correlation, they can successfully extract the 128-bit key with less than 50 trials.

One of the most important requirements of FSA is to find the fault sensitivity correlation with the input or intermediate values [4]. In this paper, we analyze this leakage function on two implementations of S-box. The first one, PPRM1, is the same as used in Li's work [4]. The second one, Boyar-Peralta, is a recently published design with a very low gate count. To the best of our knowledge, this paper is the first to present an analysis of the structure of a design and its relation to fault sensitivity. The contribution of this paper has two parts: First, based on our analysis, we identified two factors that mainly affect the fault sensitivity of a design. These two factors are the arrival time of the signals and the number of logic levels of fault sensitive gates. Moreover, based on this observation, we propose a countermeasure against FSA that is based on masking of these factors. This countermeasure is based on inserting delay elements (typically buffers) within the gate level netlist of the circuit to make the arrival time of the signals and the depth of effective gate network uniform.

The organization of this paper is as follows. Section II briefly reviews the terminology and the procedure of FSA. Section III explains our experiments on a different S-box architecture. Section IV explains the circuit properties that affect fault sensitivity. Section V proposes a masking countermeasure against FSA based on the analysis of the behavior of two different S-box implementations. This section also shows a case study of the proposed masking method on the PPRM1 S-box. Finally, Section VI describes the experimental setup of the on-chip timing measurement and countermeasure implementation on a FPGA. Section VII reviews previous work, and Section VIII concludes the paper.

II. Preliminaries

The first fault-based attack against RSA devices has been proposed by Boneh et. al [6]. Later, Biham and Shamir have proposed Differential Fault Analysis (DFA), against DES algorithm [7]. Since then, many countermeasures have been proposed against DFA. In a DFA attack, the attacker uses a cryptographic system to encrypt the same plaintext in faulty and fault-free modes. If the results do not match, the attacker knows that at least one of these results is generated in a faulty situation. DFA attack has two major requirements [4]:

1) The attacker must be aware of the characteristics of the injected fault. The key retrieval process is not possible unless the injected fault is the expected one.
2) The attacker must also capture the value of the faulty ciphertext.

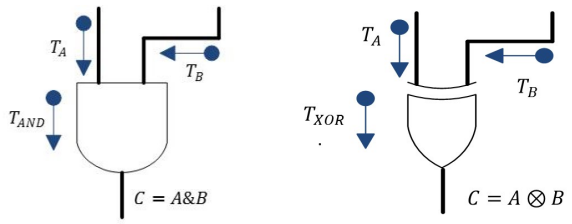
978-3-9815370-2-4/DATE14/

When either of these conditions are unfulfilled, the key retrieval process is unsuccessful. For example, a typical fault-attack countermeasure is to disable the device after the detection of a fault. This prevents observation of a faulty ciphertext.

Fault Sensitivity Analysis has been proposed by Li et. al. at CHES 2010 [4]. The attack is based on the fact that not all fault injections cause a faulty output. Injected fault may cause three possible results: a faulty output, or a fault-free output in case that the fault is injected on a path in the circuit that does not affect the result, or an irrelevant output (such as all-zeroes) if some fault prevention techniques has been employed. The following subsection explains some terminology necessary for understanding the concept of FSA:

1) **Fault Intensity**: In the FSA attack the attacker can gradually increase the intensity in which he disturbs the device e.g. if the attack exploits setup-time violations, fault injection is done by over-clocking or reducing the voltage level and fault intensity means gradually increasing the clock frequency or decreasing the supply voltage.
2) **Fault Sensitivity** : In the FSA attack procedure, while gradually increasing the fault intensity, there is a point at which the output of the device becomes faulty. This threshold is recorded as the fault sensitivity of the device for that specific input.
3) **Sensitive Data**: The input, output, or intermediate variables in a cryptographic device under attack whose value can be a function of the fault sensitivity.
4) **Critical Timing Delay**: The timing delay is defined as the time it takes for a circuit to generate the next valid state after fresh inputs have been applied to this module. The critical timing delay is defined as the highest timing delay of a logic circuit. The most important component of critical timing delay is the combinational delay of a circuit, even though there may be other factors such as clock skew.
5) **Arrival Time**: The arrival time of an input signal of a gate is the time it takes from issuing the input to a module till the time it affects the inputs of that gate. The Arrival Time of signal z is marked as T_z.

The attack procedure in FSA exploits the data dependency of fault sensitivity. In this paper, we are specifically interested in analyzing the cause of fault sensitivity. Li observed that gates become fault sensitive when their inputs have a different arrival time [4]. Figure 1(a) illustrates the case of an AND gate. If we assume that $T_A < T_B$ (which means that signal B has gone through a longer path than signal A), then T_C depends on the value of A. If $A = 0$, then $T_C = T_A + T_{AND}$. In other words, if A has the known value zero, then the arrival time for the AND gate output is defined by the arrival time of A plus a small constant delay determined by the AND gate. In other other case, when $A = 1$, the A input does not affect the eventual value of the output, and any transition on C is defined by transitions on the input B. Therefore, $T_C = T_B + T_{AND}$. Hence, we conclude that an AND-gate is fault-sensitive: its switching time depends on the value of an input bit A. The



(a) Data Dependency of Fault Sensitivity for AND Gate

(b) Data Dependency of Fault Sensitivity for XOR Gate

Figure 1. Data Dependency of Fault Sensitivity for Different Gates

Table I
DATA DEPENDENCY OF FAULT SENSITIVITY IN AND, OR AND XOR GATES

	A	B	T_C
AND	0	x	$T_C = T_A + T_{AND}$
AND	1	x	$T_C = max(T_A, T_B) + T_{AND}$
OR	1	x	$T_C = T_A + T_{OR}$
OR	0	x	$T_C = max(T_A, T_B) + T_{OR}$
XOR	x	x	$T_C = max(T_A, T_B) + T_{XOR}$

same happens for the OR gate, and it is illustrated in Table I. If $A = 1$, then $T_C = T_A + T_{OR}$. Otherwise $T_C = T_B + T_{OR}$. In contrast to the AND and OR gate, the XOR gate is not fault sensitive. For XOR gates (Figure 1(b)), the output C will propagate changes to either A or B with the same preference. Therefore, $T_C = max(T_A, T_B) + T_{XOR}$. Table I summarizes the above arguments. In the remainder of the paper, we use the term **Effective Gates** to indicate the gates that are fault sensitive such as AND and OR gates.

The FSA attack has two phases, the fault sensitivity information collection, and the key retrieval procedure. In the first phase, the attacker applies a plaintext as an input to the cryptography device, then gradually increases the fault intensity until he sees some abnormality in the output of the device. This point is recorded for the applied plaintext as the fault sensitivity. The profiling phase is performed for N different inputs. In the key retrieval phase, the attacker has a key guess, the ciphertexts and the fault sensitivity for the profiling input. The attacker finds the fault sensitivity for the potential key guess. Then, he draws the correlation graph between the actual fault sensitivity and the guessed fault sensitivity. The correct key is the best match between these two [4].

III. DATA DEPENDENCY OF FAULT SENSITIVITY ON S-BOX ARCHITECTURES

One of the most important requirements of FSA is that the attacker must be aware of the data dependencies that enable fault sensitivity. This section analyzes two different S-box implementations. The first one is the PPRM1 S-box, as studied in Li *et al.* [4]. The second architecture is the Boyar-Peralta S-box, a design with a very small gate footprint. The data presented in this section was extracted from an actual FPGA

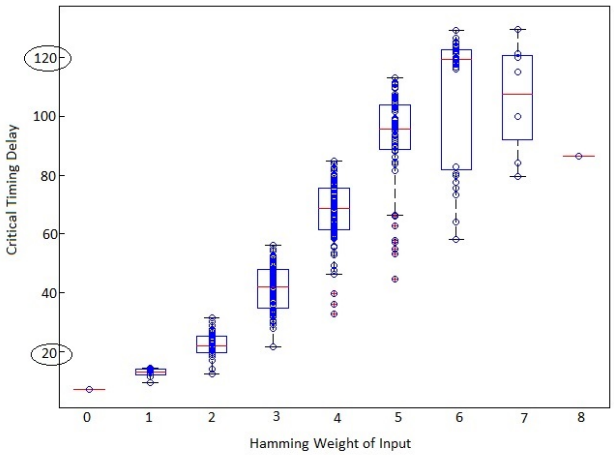


Figure 2. Data Dependency of Fault Sensitivity with Input Hamming Weight in PPRM1 S-box (Note the Y-scale Range)

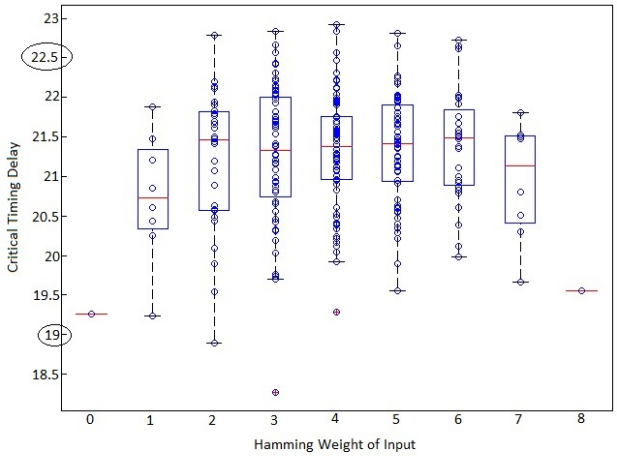


Figure 3. Data Dependency of Fault Sensitivity with Input Hamming Weight in Boyar-Peralta Sbox (Note the Y-scale Range)

prototype, which will be discussed in a later section (Section VI).

A. PPRM1 S-box

Figure 2 shows the relation between the Hamming Weights of the inputs vs. the critical timing delay of the circuit. The initial values of all S-box architectures are set to zero. The input sequence is exhaustive, and assigns every possible input value. As shown in the graph, the higher the Hamming Weight the larger the critical timing delay. This way, the attacker can extract the data dependency of fault sensitivity by different input values.

B. Boyar-Peralta S-box

Circuit minimization for the AES S-box is a widely studied hardware problem. A recent effort in this area is by Boyar and Peralta [8]. This design minimizes the total number of gates and the overall circuit depth. This design is a two step process. The first one is the non-linear gate reduction which is based on performing multiplication and inverse operations in large fields by implementing them in smaller fields [9]. We study this S-box because of its compactness, which makes it a good candidate for cost-sensitive, embedded cryptographic applications.

We measured the data dependency of fault sensitivity in Boyar-Peralta's S-box. Figure 3 shows the critical timing delay of the S-box vs. the Hamming Weight of the input data. As shown, the range of mean critical timing delay in Figure 3 is between 19.5 to 21.5 while this range is from 10 to 120 in Figure 2. Consequently, the graph in Figure 3 discloses significantly less data dependency on fault sensitivity, when compared to Figure 2. We also did not find any significant data dependencies of fault sensitivity using the factors such as:

- Use of a linear weighted combination of the input bits, rather than the sum (Hamming Weight)
- Value of the S-box outputs
- Hamming Distance among successive inputs
- The linear combination of edge triggers of input bits along with their values

The above experiments have been done on Canright S-box as well [10]. Based on the results, Canright S-box is more vulnerable to FSA attack than Boyar-Peralta S-box. The reason will be discussed in section IV.

IV. Data Analysis

As shown in Figure 3, the range of the delays for different input Hamming weights is distributed uniformly in the case of the Boyar Peralta S-Box. However, this distribution for the PPRM1 S-box is based and depends on the Hamming weight value. A careful comparison of both architectures leads to the following observations.

- The AND network is the determining factor in data dependency of fault sensitivity. Figure 4 shows that the AND network for PPRM1 has AND arrays of depth 6 down to depth 1. However, as stated in Figure 5, the AND network for Boyar-Peralta S-box has only arrays of depth 2 and 1. Based on our analysis, the depth of the AND network in a circuit is an important factor to determine its sensitivity to setup time violations. If there are several AND networks in a circuit with different depths, and the difference between their depth is large, the data dependency of fault sensitivity would increase. We know that PPRM1 S-box has AND network of depth 7 down to depth 1. If we assume that each AND gate has a delay equal to T_{AND}, then the range of timing delay for PPRM1 S-box would be from $7 \times T_{AND}$ down to T_{AND}. This number for Boyar-Peralta would be $2 \times T_{AND}$ down to T_{AND} because it has AND network of depth 2 and 1. The delay difference for Boyar-Peralta is negligible compared to PPRM1.

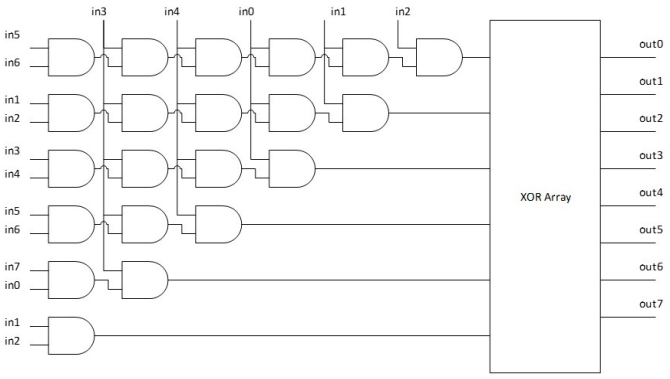


Figure 4. Partial Structure of the PPRM1 S-box Design

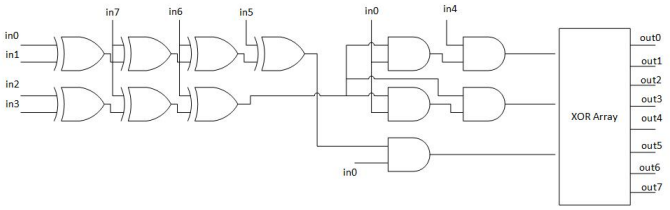


Figure 5. Partial Structure of Boyar-Peralta S-box Design

- Based on Figure 4, the architecture of the PPRM1 S-box has a layer of AND gates and after that a layer of XOR gates. The only gates that affect the fault sensitivity are the AND and OR gates. So, in case of the PPRM1 S-box only a layer of the AND network is counted for data dependency of fault sensitivity. However, for the Boyar-Peralta S-box (Figure 5), there is a layer of XOR gates before the AND network. The XOR network does not directly affect the data dependency of fault sensitivity, but the depth of the XOR network will affect the arrival time of the signals to the AND network. This increases the spread of the signals in time, *independent* of the actual data value. The net effect is that the contribution of the data-dependent delay of the AND network will decrease in relative terms. Hence, the XOR network preceding the AND network, in the case of the Boyar-Peralta S-box, will further decrease the fault sensitivity.

Based on the above discussion, the factors that affect a circuit to show data dependency of fault sensitivity are as follows:

1) The type of gates in the design: As observed in [4], the gates that cause data dependency of fault sensitivity are AND, OR and related combinations. XOR and XNOR gates do not affect the data dependency of the critical timing delay since their output always depends on both data inputs.
2) The differential depth of the effective gate network: If G_E is defined as the number of logic levels along any path from input to output that contain effective gates such as AND and OR, then the differential depth of the effective gate network is $\max G_E - \min G_E$,.
3) The arrival time of signals to the inputs of the effective gate network.

Based on the above discussion, the reason that Canright S-box is more vulnerable to FSA attack than Boyar-Peralta S-box is that Boyar-Peralta S-box is minimized for circuit depth. Meaning that the differential depth of effective gates is smaller in Boyar-Peralta S-box than Canright S-box. So, it is more difficult for the adversary to extract data dependency of fault sensitivity for Boyar-Peralta S-box.

V. Proposed Countermeasure

We propose a systematic countermeasure against the FSA attack. Finding the data dependency of fault sensitivity is the main issue for the attackers in FSA attack. Therefore, the proposed countermeasure is a masking method that aims at masking the factors that affect the data dependency of fault sensitivity. The main idea behind this countermeasure is to remove the dependency of the critical timing delay to the processed data values in the circuits. We propose a transformation that operates at two levels of abstraction, at netlist level and at gate level.

- Netlist level: The delay of the netlist must be independent of the input data.
- Gate level: The switching time of gates must be random during circuit evaluation, meaning that the switching distribution is uniform over the computation time of the circuit.

The proposed countermeasure is based on inserting delay elements in different paths of the circuit based on the statistical timing analysis of the circuit. The goal is to equalize the effective delay of each path in a circuit. The **Effective Delay of a Path** is defined as the number of effective gates in that path multiplied by their propagation delays. We start from the outputs of the circuit. For each output, we evaluate the effective delay to any input. We then find the maximum effective delay. We then insert delay elements near the input of each path such that the sum of effective path delay and inserted buffer delay becomes equal to the maximum effective delay. The number of delay elements is in inverse proportion to the length of the path.

A. FSA Resistant PPRM1 design

The PPRM1 S-box has been chosen as a case study. The delay elements are inserted in the path of each effective gate (AND gates) output until the delay of all of them becomes equal to the maximum delay of the circuit. This FSA resistant design has been analyzed to find the data dependency. As shown in Figure 6, the timing delay is now uniform for each Hamming weight of the input values. The performance cost of this method is less than 1% since the final output would be ready at the maximum time in both FSA resistant and the original design. However, the number of gates has increased by 24%.

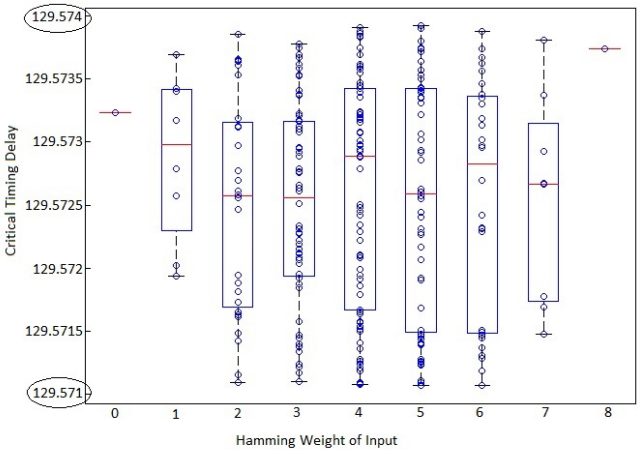


Figure 6. Timing Delays for FSA Resistant PPRM1 Design

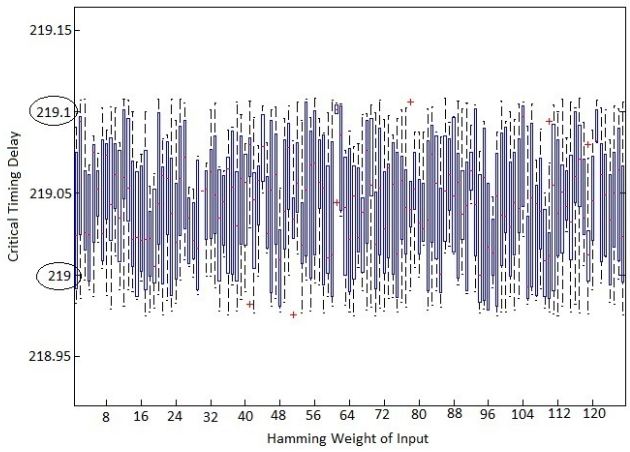


Figure 7. Timing Delays for FSA Resistant AES Design for one Round

B. FSA Resistant AES

In order to show the generality of the proposed countermeasure, we have implemented this countermeasure in a full round of Advanced Encryption Standard algorithm. AES is a symmetric key algorithm, the same key is used for both encryption and decryption [11]. The 128 bit key size AES, performs 10 rounds on the input block to generate the encrypted output, the so called ciphertext. Each round performs four operations, namely, Sub-Bytes, Shift-Rows, Mix-Columns and Add-Round-Key. We have studied the gate-level design for each of these operations to find the data dependency of fault sensitivity in their designs. The Mix-Columns algorithm shows the data dependency of fault sensitivity. The delay insertion algorithm has been applied to Mix-Columns as well. The final timing analysis of a round in AES algorithm is shown in Figure 7. The input sequence is a random number with the Hamming Weight between 0 and 128.

VI. Experimental Setup

This section clarifies the platform for calculation of Figure 3, 2, 6 and 7. Rather than using simulation tools, we created a test setup that enables us to determine the fault sensitivity of S-Box as implemented on an FPGA circuit. To make our results consistent with what we would expect from an ASIC design, we first mapped the S-Boxes under test to a gate-level netlist using the Synopsis Design Compiler [12] under the generic technology library. Next, the resulting netlist was translated in Verilog, constrained for FPGA synthesis, and integrated into our FPGA-based measurement circuit. We used the *Altera DE0-Nano* FPGA board for our implementations.

Figure 8 shows the hardware architecture of on-chip time measurement for DUT operations. The DUT can be any combinational logic. In this paper we have replaced it with different S-box implementations and a whole round of AES. The architecture consists of two blocks: a Trigger Circuit and a Measurement Circuit. The Trigger Circuit is used to generate an unstable (oscillating) feedback loop. The Measurement Circuit evaluates the period of this oscillation by comparison with a reference clock.

The feedback loop of Trigger Circuit is designed using only combinational logic without any pipeline stages. The combinational path consists of an *S-box* and decision/selection logic. The timing of this path is dominated by the *DUT* operation. Two equality checkers determine when the *DUT* output is generated. Then, the decision is captured in an *S-R latch* to prevent glitches. If enabled, the output of the *S-R latch* choose the next input. If *reg_0*, *reg_1*, *reg_2* and *reg_3* is set accordingly, the input of the *DUT* switches between *reg_0* and *reg_1*. The Measurement Circuit calculates the oscillation period. The select signal of *DUT* input multiplexer is also tied to the clock input of a counter. At every rising clock edge the value of the *cnt* register is incremented by 1. A reference counter (*ref_cnt*) using the system clock keeps the track of a known time. Then, we can calculate the oscillation time by $(\frac{ref\_cnt}{cnt}) \times clk\_period$ where *clk_period* is the clock period of the signal *system_clk*.

VII. Related Work

This section reviews previous and related work in FSA. It has been shown that most of the countermeasures against the Side Channel Attacks and fault attacks are not efficient against FSA. The authors of [13] have shown that the WDDL design is DFA resistant. But, it has been shown that this design is vulnerable against FSA [14]. In FSA, since no faulty output value is required, the attack is also resistant against the fault prevention methods employed against the DFA attack. An example of this is the AES module equipped with the concurrent error detection schemes [15] that has been broken in [16]. Certain masking techniques have been also broken by FSA. Based on [17], when the masked values are based on random numbers, the faulty ciphertext would have a non-uniform distribution and can be correlated with the input

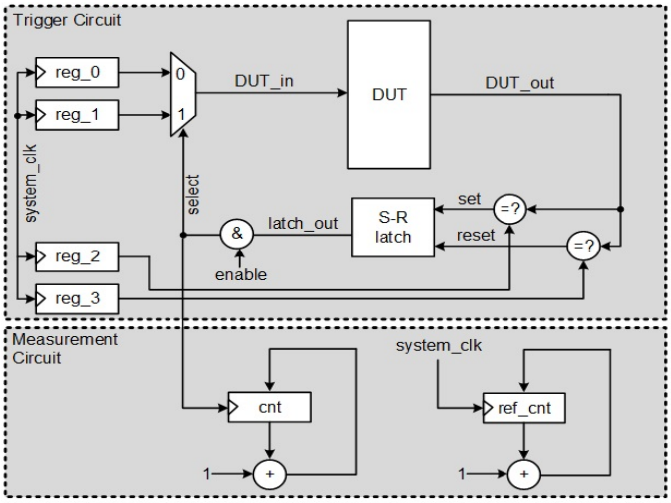


Figure 8. The architecture of the Trigger and Measurement Circuit for on-chip time measurements of S-box operations

values. The AES module with the Masked AND Operation (MAO) has been broken by [17].

The only proposed countermeasure against the FSA till now is [18]. Earlier work, namely [19], proposed the use of an enable signal to eliminate the data dependency of fault sensitivity. The results of the combinational logic is stored until the timing of the enable signal arrives. However, Li did not clarify how to generate this enable signal [19]. The authors of [18] suggested a solution for this. A one-time memory stores the timing of the combinational logic. The Delay Blocks are reconfigured based on the values of the one-time memory and set the enable signal. While the area cost of their method is 10%, their method is a post-manufacturing reconfiguration which is technology dependent. Moreover, they use an external module for generating the delay blocks based on the circuit delay. This external module is a multiplexer that decides on the number of inserted delay elements. An attacker can tap on the external module or specifically the multiplexer select signal to get the fault sensitivity information. In contrast, our proposed method is based on changing the circuit internally.

VIII. Conclusions and Future Work

This paper evaluates the cause of FSA by analyzing different S-box architectures [4]. We have demonstrated the existence of two factors, the depth of the AND/OR network in a design, as well as the arrival time of input signals to the AND/OR network. Both of these factors influence the fault sensitivity. Based on these two factors, a countermeasure has been suggested to eliminate fault sensitivity in a design based on a delay insertion algorithm which could be optimized in future works for area minimization. The delay insertion algorithm can also generate a criteria for fault sensitivity evaluation of different circuits. The proposed method has been demonstrated in a prototype setup. In our current research, we are evaluating the automation of this design transformation to larger circuits.

IX. Acknowledgments

This research was support in part by the National Science Foundation, Grant no 1115839.

References

[1] Paul Kocher, "Timing attacks on implementations of diffie-hellman, RSA, DSS, and other systems", in *Advances in CryptologyCRYPTO96*. Springer, 1996, pp. 104–113.

[2] Paul Kocher, Joshua Jaffe, and Benjamin Jun, "Differential power analysis", in *Advances in CryptologyCRYPTO99*. Springer, 1999, pp. 388–397.

[3] Karine Gandolfi, Christophe Mourtel, and Francis Olivier, "Electromagnetic analysis: Concrete results", in *Cryptographic Hardware and Embedded SystemsCHES 2001*. Springer, 2001, pp. 251–261.

[4] Yang Li, Kazuo Sakiyama, Shigeto Gomisawa, Toshinori Fukunaga, Junko Takahashi, and Kazuo Ohta, "Fault sensitivity analysis", in *Cryptographic Hardware and Embedded Systems, CHES 2010*, pp. 320–334. Springer, 2010.

[5] Sumio Morioka and Akashi Satoh, "An optimized s-box circuit architecture for low power AES design", in *Cryptographic Hardware and Embedded Systems-CHES 2002*, pp. 172–186. Springer, 2003.

[6] Dan Boneh, Richard A DeMillo, and Richard J Lipton, "On the importance of checking cryptographic protocols for faults", in *Advances in CryptologyEUROCRYPT97*. Springer, 1997, pp. 37–51.

[7] Eli Biham and Adi Shamir, *Differential cryptanalysis of the data encryption standard*, vol. 28, Springer-Verlag New York, 1993.

[8] Joan Boyar and René Peralta, "A small depth-16 circuit for the AES s-box", in *Information Security and Privacy Research*, pp. 287–298. Springer, 2012.

[9] Joan Boyar, René Peralta, and Denis Pochuev, "On the multiplicative complexity of boolean functions over the basis (and,xor,1)", *Theoretical Computer Science*, vol. 235, no. 1, pp. 43–57, 2000.

[10] David Canright, "A very compact s-box for aes", in *Cryptographic Hardware and Embedded Systems–CHES 2005*, pp. 441–455. Springer, 2005.

[11] Joan Daemen, Vincent Rijmen, and AES Proposal, "Rijndael", in *Proceedings from the First Advanced Encryption Standard Candidate Conference, National Institute of Standards and Technology (NIST)*, 1998.

[12] Design Compiler, "Synopsys inc", 2000.

[13] Nidhal Selmane, Shivam Bhasin, Sylvain Guilley, Tarik Graba, and J-L Danger, "WDDL is protected against setup time violation attacks", in *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on*. IEEE, 2009, pp. 73–83.

[14] Yang Li, Kazuo Ohta, and Kazuo Sakiyama, "Revisit fault sensitivity analysis on WDDL-AES", in *Hardware-Oriented Security and Trust (HOST), 2011 IEEE International Symposium on*. IEEE, 2011, pp. 148–153.

[15] Akashi Satoh, Takeshi Sugawara, Naofumi Homma, and Takafumi Aoki, "High-performance concurrent error detection scheme for aes hardware", in *Cryptographic Hardware and Embedded Systems–CHES 2008*, pp. 100–112. Springer, 2008.

[16] Amir Moradi, Oliver Mischke, and Christof Paar, "Collision timing attack when breaking 42 aes asic cores", Tech. Rep., Cryptology ePrint Archive, Report 2011/162, 2011. http://eprint. iacr. org, 2011.

[17] Amir Moradi, Oliver Mischke, Christof Paar, Yang Li, Kazuo Ohta, and Kazuo Sakiyama, "On the power of fault sensitivity analysis and collision side-channel attacks in a combined setting", in *Cryptographic Hardware and Embedded Systems–CHES 2011*, pp. 292–311. Springer, 2011.

[18] Sho Endo, Yang Li, Naofumi Homma, Kazuo Sakiyama, Kazuo Ohta, and Takafumi Aoki, "An efficient countermeasure against fault sensitivity analysis using configurable delay blocks", in *2012 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*. IEEE, 2012, pp. 95–102.

[19] Li Yang and Kazuo Sakiyama, "Toward effective countermeasures against an improved fault sensitivity analysis", *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 1, pp. 234–241, 2012.