Microelectronics Journal 45 (2014) 1354–1360

Contents lists available at ScienceDirect

Microelectronics Journal

journal homepage: www.elsevier.com/locate/mejo

A method to assess the robustness of cryptographic circuits at the design stage

M. Arévalo-Garbayo, M. Portela-García, M. García-Valderas, C. López-Ongil, L. Entrena

Electronic Technology Department, University Carlos III of Madrid (UC3M), Leganés, Madrid 28911, Spain

ARTICLE INFO

Article history:
Received 15 February 2013
Received in revised form
23 September 2013
Accepted 19 December 2013
Available online 20 January 2014

Keywords:
Cryptographic circuits
Differential fault attacks
Fault-based attack
Fault injection
Single event transient

ABSTRACT

This paper proposes the use of an FPGA-based fault injection technique, AMUSE, to study the effect of malicious attacks on cryptographic circuits. Originally, AMUSE was devised to analyze the soft error effects (SEU and SET) in digital circuits. However, many of the fault-based attacks used in cryptanalysis produce faults that can be modeled as bit-flip in memory elements or transient pulses in combinational logic, as in faults due to radiation effects. Experimental results provide information that allows the cryptographic circuit designer to detect the weakest areas in order to implement countermeasures at design stage.

1. Introduction

Cryptographic circuits are used in more and more applications in order to protect exchanged information (commercial transactions, web servers, sending personal data, etc.) or to avoid the access to secret information by unauthorized people (reverse engineering, defense sabotages, economic exploitation, etc.). The security of these circuits is a main concern, and there are many research works aimed at looking for the vulnerabilities of the cryptographic algorithms and their implementations, as well as, for solutions to overcome them.

Fault attacks have been extended successfully since 1997 [1,2], and are common methods used to obtain secret information from a crypto-circuit. This kind of method consists on provoking faults that alter the execution in such a way that the faulty output leak information related to the secret key. There is a relation between the hardware implementation of a cryptographic algorithm and fault effects in the circuit. Certain implementations might make easier the task of extracting out secret information. Assessing the robustness of a given crypto-circuit at design stage is very useful for designers and necessary to propose efficient countermeasures to reduce or remove the possible vulnerabilities.

There are very few available tools to help designers in determining the security of their implemented cryptographic algorithms during design stage. The majority of existing solutions are based on attacks to the prototyped device [3,4]. These methods, although very realistic, imply a high redesign cost if cipher is proven as insecure. Nowadays, transient fault injection is considered one of the most effective methods for successfully obtaining secret information in cryptographic devices [4–6].

Fault injection is currently executed with an external source of energy for affecting specific elements in the circuit (laser beam, light, etc.). Of course, these methods must be applied on manufactured devices in order to provide a 'security qualification', but other techniques will be welcome by designers if they are applicable during the design cycle and help designers in the selection of the best implementation.

In early stages of the design cycle, circuits are described in a high abstraction level and a fault model is required for representing the fault effect in that abstraction level. Fault injection is performed on the design description and a comparison between the fault-free behavior and the faulty behavior is run in order to check the fault attack effects in the circuit. During the design cycle of a cryptographic algorithm, there are two possibilities for checking the effect of intentional attacks on the circuit:

- Simulation CAD tools allow fault injection by means of simulator commands or specific-purpose fault-injecting blocks simulated together with the design under test. This option is highly flexible while implies a high amount of resources in terms of time and area (CPU and memory).
- Design emulation in programmable devices (like FPGAs) [7] allows fault injection from an external manager (PC host) through a specific link (USB, PCI, RS232). This option is much

E-mail addresses: jmareval@ing.uc3m.es (M. Arévalo-Garbayo), mportela@ing.uc3m.es (M. Portela-García), mgvalder@ing.uc3m.es (M. García-Valderas), celia@ing.uc3m.es (C. López-Ongil), entrena@ing.uc3m.es (L. Entrena).

0026-2692/$ - see front matter © 2014 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.mejo.2013.12.012

faster than the first one allowing intensive injection campaigns in shorter times.

In this paper we propose a method for checking the security of cryptographic algorithms during early stages in the design cycle. This method is applicable to hardware and software implemented algorithms when a netlist of the final circuit is available. With the results provided by the tool presented, the designer will sweep the design space for a given algorithm/protocol in a short time with accurate measurements. These effects are the same provoked in a digital circuit due to environmental conditions like cosmic radiation and they suppose a main concern for circuit dependability. In fact, laser beam is also a fault injection technique applied to evaluate the dependability of a given circuit.

This paper is organized as follows: Section 2 presents an overview of the existing malicious attacks justifying and describing the type of fault attack considered in this work. Section 3 describes the proposed solution to analyze the behavior of a circuit against fault attacks. Section 4 describes the performed experiments and presents the obtained results. Finally, Section 5 states conclusions.

2. Malicious attacks

Attacks on cryptographic circuits are usually performed by using physical means to obtain the secret key. Used methods can be passive or active. On the one hand, passive techniques profit from information leakage during the circuit running by measuring some physical magnitude like power consumption, electromagnetic radiation, execution time, etc. Depending on the processed data, those physical magnitudes vary and attackers can deduce secret information. These techniques are named side-channel attacks and are non-intrusive [1]. On the other hand, active attacks are more invasive and consist in modifying part of the circuit or its behavior. In case of just introducing behavior modifications, the attacks are named fault attacks and different subtypes can be distinguished. Three categories are considered in the following [4]:

- Algorithm modifications, for example bypassing part of the hardware implementation to reduce the circuit complexity.
- Differential fault attack (DFA) that consists in provoking faults in the cryptographic circuit and observing the outputs in order to find any difference with respect to the golden behavior.
- Safe-error attacks are based on analyzing those errors that do not produce any effect in the circuit behavior.

There are different fault injection techniques in order to perform fault attacks in a cryptographic circuit. Faults can be introduced by laser exposure, voltage or clock glitches, electromagnetic interferences, ion beam radiation, etc. In [5,6], summaries of the most important fault injection techniques are presented. The effect of a fault attack in the circuit can be permanent or transient, depending on its duration, and single or multiple, depending on the number of affected memory elements. Different fault injection techniques allow the insertion of different type of faults. Using laser beam provides a very high precision on the location and fault insertion timing [8].

When a laser beam impacts in a CMOS circuit, a transient fault is generated, producing a bit-flip if the fault location is a memory element, or an erroneous transition in logic values if the fault location is a combinational gate [4,6,8,10]. These fault models are the same used to study soft errors due to radiation effects in digital circuits: SEU (single event upset) and SET (single event transient), respectively. In fact, laser injection methods are used to simulate the effects of heavy ion beams [11].

An SET can be propagated through the circuit and it may eventually reach one or more memory elements where the fault effect remains stored until the following writing. Therefore, an SET can produce multiple errors in the circuit starting from a single laser injection. Furthermore, the probability of an SET reaches memory elements increase for higher circuit frequencies. As an example of the relevance of this effect in cryptographic circuits, the French company INVIA develops hardware implementations of cryptographic algorithms with optional protection against SET fault attacks [9].

The main objective of this work is to present a fault injection technique, traditionally used to analyze SEU and SET sensitivity of digital circuits, in order to check security of a cryptographic circuit under fault attacks. This method allows the designers to detect the vulnerabilities in the algorithm implementation during the design stage and therefore, to propose countermeasures and evaluate their efficiency. We will prove with the experimental results that a fault injection technique intended to evaluate the dependability of a digital circuit against SET provides a very efficient tool during the design of a cryptographic circuit.

2.1. Transient fault based attacks

An SET could not produce any effect in the circuit if its effect is masked by one of these reasons: logic masking, latch window masking or electrical masking. Logic masking occurs in a location and at instant when the performed logic function does not propagate the fault to the outputs. For example, if an SET occurs in the input of an AND gate when other input is '0', the fault effect will not propagate to the output. Latch window and electrical masking depends on the circuit delays. Latch window masking occurs when the SET effect reaches a memory element far from the active clock edge and then the erroneous value is not stored. Electrical masking occurs when the erroneous voltage pulse is filtered due to the capacitors across gates.

In early design stages, fault attacks can be analyzed by modeling the behavior of transient faults in the circuit description. Fault models could be defined at any abstraction level where the circuit is described. The accuracy of this estimation depends on the precision of the fault model. Register-transfer (RT) level is usually the abstraction level used by designers to describe digital circuits by a hardware description language like Verilog or VHDL. However, SET propagation through the circuit, and therefore, its effect in the circuit behavior depends on the gate delays. Therefore, gate delays effect is usually analyzed at lower abstraction level that includes real delay information for the target technology.

Due to the huge number of possible transient faults (different transient faults are generated depending on the location and the injection instant), a fast rapid fault injection method is required and hardware-based injection solutions are proposed. In [10], transient faults are injected by voltage manipulations. The attacked circuit is a SPARC microprocessor running an RSA algorithm and implemented in an FPGA. By means of transient fault injections, the authors obtained the private key. However, this technique does not allow the designer to control the exact location and duration of the transient fault to inject. Some works [6,12] present approaches where the design is prototyped in an FPGA and a logic fault model is used to insert transient faults in memory elements. This kind of techniques is also used for the dependability analysis of digital circuits against SEU faults. In [13], laser injection experiments are performed to evaluate a crypto-processor as well as soft error experiments.

The main novelty of this work is the capability to emulate transient faults in combinational logic as well as in memory elements, for providing some information about the effects of fault attacks in the design stage. This way, designers can improve the weaknesses found