Microprocessors and Microsystems 41 (2016) 47–55

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A high-speed AES design resistant to fault injection attacks

Hassen Mestiri*, Fatma Kahri, Belgacem Bouallegue, Mohsen Machhout

Electronics and Micro-Electronics Laboratory (E.μ.E.L), Faculty of Sciences of Monastir, University of Monastir, Tunisia

ARTICLE INFO

Keywords:
Cryptographic
Secure AES algorithm
Fault detection
FPGA implementation
Embedded systems

ABSTRACT

To secure the Advanced Encryption Standard against physical attacks known as fault injection attacks, different countermeasures have been proposed. The AES is used in many embedded systems to provide security. It has become the default choice for security services in numerous applications. However, the natural and malicious injected faults reduce its robustness and may cause private information leakage. In this paper, we study the concurrent fault detection schemes for achieving a reliable AES implementation. We specifically propose a new fault detection scheme based on modification of the AES architecture. For this purpose, the round AES transformation is broken into two parts and a pipeline stage is inserted in between.

The proposed scheme is independent of the way the S-Box and the Inv_S-Box are implemented. Hence, it can be used for both the S-Box and the Inv_S-Box using Look-Up Table and those using logic gates based on Galois Fields. Our simulation results show the fault coverage reaches 98.54% for the proposed scheme. Moreover, the proposed and the previously reported fault detection schemes have been implemented on the most recent Xilinx Virtex FPGAs. Their area overhead, the frequency and throughput have been compared and it is shown that the proposed fault detection scheme outperform the previously reported ones.

1. Introduction

The Advanced Encryption Standard (AES) was finalized in October 2000 by the National Institute of Standards and Technology (NIST), when the Rijndael algorithm was adopted [1]. The AES algorithm replaced the Data Encryption Standard (DES), which had been in use since 1976. Until now, many architectures, for efficient VLSI realization of AES algorithm, have been proposed and their performances have been evaluated by using ASIC libraries and FPGA [2–4].

The objective in utilizing the AES is to secure transfer the data so that only the desired receiver with a secret key would be able to retrieve the original data. Yet, with the existence of malicious injected faults in non-secure implementation, the AES does not ensure that the confidential information is transferred reliably. In fact, several fault attacks on the AES are reported in the literature [5,6]. The Differential Fault Analysis (DFA) attacks are based on injecting faults into the structure of the AES to obtain the confidential information. The cryptanalyst injects faults during the execution of the processing algorithm. This disturbs the normal execution behavior and results in creating faulty ciphertext. Thus, the cryptanalyst can guess the secret key after a certain number of fault injections and analyzing faulty ciphertexts. To make a robust implementation against fault attacks, several fault detection schemes have been proposed to date, see, for example [7–13].

Mozaffari-Kermani et al. proposed in [7] a low-cost structure-independent fault detection schemes for the AES implementation. The authors proposed new formulations for the fault detection in the SubBytes and the Inv_SubBytes using an arithmetic relation between the input and the output of the S-Box and the Inv_S-Box. The proposed schemes are independent of the way the S-Box and the Inv_S-Box are implemented (Look-Up Table (LUT) or combinational logic). Another important contribution of the mentioned reference is that it describes an arithmetic relationship to detect faults during the execution of the AES linear transformations.

In [8], the authors proposed a lightweight concurrent fault detection scheme for the AES. In this scheme, the composite field S-Box and Inv_S-Box are divided into five blocks, thus obtaining the predicted parities of these blocks. This scheme is implemented on FPGA platform and the authors show that their method has better hardware and time complexities compared to their counterparts. However, this scheme is not suitable for the S-Box and Inv_S-Box implemented using Look-Up Table (LUT). This is because the input (respectively, the output) of every block in the S-Box (respectively, the Inv_S-Box) may not be accessible in the LUT-based

* Corresponding author. Tel.: +21699656757.
E-mail addresses: hassen.mestiri@yahoo.fr (H. Mestiri), kahrifatma@gmail.com (F. Kahri).

http://dx.doi.org/10.1016/j.micpro.2015.12.002
0141-9331/

implementations. Therefore, the fault detection scheme presented in [8] is not suitable for these implementations.

In [9], Chu et al. proposed a new error detection method using the polynomial residue number systems (PRNS) to secure the AES implementation. The PRNS error-detecting scheme yields very good error coverage and the distribution and parallelism characteristic of a PRNS architecture itself yields intrinsic resistance to some side-channel attacks. However, this scheme needed to implement three additional S-Box to calculate the residue representation which adds about 81% area overhead and makes power attacks more efficient; since the SubBytes operations contribute to much of the total power consumption in AES.

Mestiri et al. proposed in [10] a fault detection scheme, based on the information redundancy, for the AES. The authors proposed a new scheme for the fault detection in the SubBytes and the Inv_SubBytes using the relation between the input and the output of the S-Box and the Inv_S-Box. This scheme is independent of the way the S-Box and the Inv_S-Box are implemented. The authors show that their fault detection scheme reaches 99.998% fault coverage. The same authors proposed in [11] a new fault detection scheme, based on the hybrid redundancy, for the AES. They present its details implementation in each transformation of the AES. The scheme in the ShiftRows and Inv_ShiftRows is based on the bit level scrambling approach. The authors show that their scheme reaches 99.999% fault coverage.

Maistri et al. proposed in [12] a fault detection scheme based on temporal redundancy where each encryption round is computed twice and the results are compared. The encryption round is computed on both clock edges. This speeds up the computation process. Under certain assumptions, the Double-Data-Rate (DDR) scheme allows the encryption process to be computed twice without affecting the original throughput. Although this fault detection scheme can detect almost all faults, its cost is acceptable only when operating at limited frequencies. Moreover, this scheme is complex and delicate to implement at technology scales.

Rajendran et al. proposed in [13] a new CED mechanism based on the slide attack. This mechanism is independent of the implementation scheme of the S-Box. It can be applicable to all the symmetric block ciphers. It is applicable to both the encryption and decryption mechanisms. The authors say that their fault detection scheme reached 100% fault coverage, although this scheme has not been tested enough since the faults are injected only at the round operations.

In this paper, we present a new fault detection scheme for obtaining a reliable AES implementation. We summarize our contributions as:

- We have modified the AES round architecture such that the round is divided into two operations and pipelined, so that the first half round operation is checked against errors while the second half round operation is performed and vice versa.
- We have proposed a new architecture of the AES resulting in novel fault detection scheme for checking SubBytes, Inv_SubBytes, and the other transformations in the encryption and the decryption process. The proposed scheme is independent of the method the S-Box (respectively, the Inv-S-box) is implemented. Thus, it can be applied to both the LUT and composite fields implementations. It is interesting to note that the cost hardware of our scheme is lower and the clock frequency is higher than its counterparts which based on temporal redundancy. It also has higher error coverage than ones of the counterparts.
- We have simulated the proposed fault detection structure for the AES implementation. Faults are injected in all possible location such as the error detection and the encryption data paths. Through our simulations after injecting up to 4.000.000 transient bit-flips, we have shown that the proposed scheme reaches 98.54% fault coverage.
- Finally, our proposed fault detection scheme and almost all of the previously reported ones have been implemented on the recent Xilinx Virtex FPGAs, and their area and frequency overheads have been derived and compared. The FPGA implementation results show the low area overhead, the high frequency and throughput for the proposed fault detection scheme. These implementation results show also that compared to the schemes presented in [12] and [13], the complexities of the proposed detection scheme are lower.

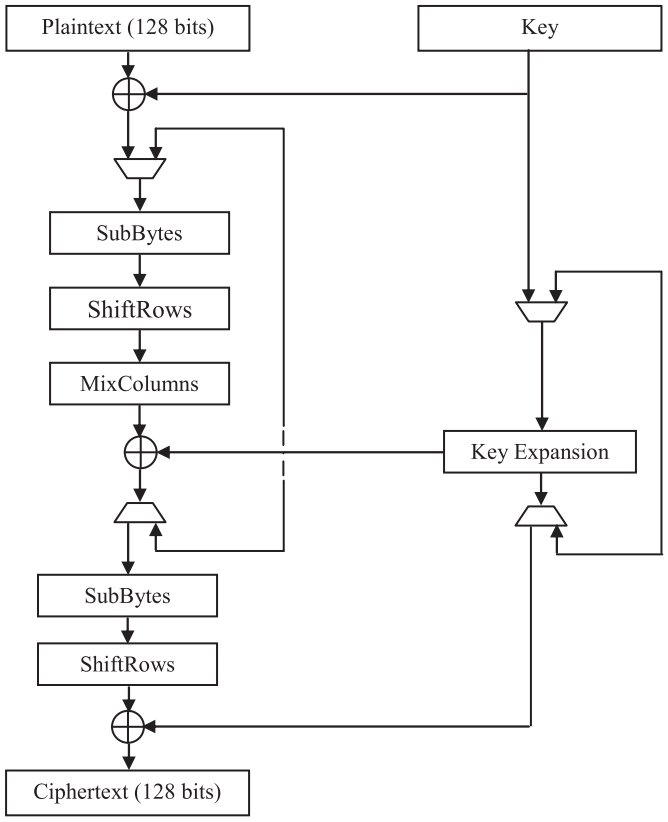

Fig. 1. Description of the AES cryptographic algorithm.

The organization of this paper is as follows. Section 2 describes the background knowledge. Section 3 presents the AES design. In Section 4, we present the proposed fault detection scheme for the AES. The detailed architecture of the first and the second half round for the encryption process are presented in Section 5. Section 6 deals with the fault coverage evaluation of the proposed design. In Section 7, the experimental synthesis results and performances are discussed and compared in terms of area, frequency and throughput. Section 8 concludes the paper.

2. Backgrounds

2.1. Advanced Encryption Standard

The AES is a symmetric block cipher that process data blocks using cipher keys with lengths of 128, 192 and 256 bits. Each data block consists of 4 × 4 array of bytes called the state. The AES is a round-based encryption algorithm. (Fig. 1)

The number of rounds is 10, 12, or 14, when the key length is 128, 192 or 256 bits, respectively. In the encryption of the AES algorithm, each round, except the final round, performs four transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey,