A Structure-Independent Approach for Fault Detection Hardware Implementations of the Advanced Encryption Standard

Mehran Mozaffari Kermani and Arash Reyhani-Masoleh

Presented by: Mehran Mozaffari Kermani

Department of Electrical and Computer Engineering
The University of Western Ontario
London, Ontario, Canada N6A 5B9
Overview

• Introduction
• Motivation
• Multiplication: A Previous Approach
• The Proposed Structure-independent Scheme
• Simulation Results
• FPGA Implementations
• Comparison
• Conclusions
Introduction

- **AES-128**
  - 128-bit input
  - 128-bit key
  - 10 rounds
  - 4 transformations
Introduction

– Fault detection
  • Natural faults
  • Fault attacks

– Existing fault detection approaches
  • Redundant Units
    – Algorithm, Round, Operation (Transformation) Level
  • Error Detecting Codes
  • Multiplication-based Approach
Motivation

– SubBytes is a nonlinear transformation among AES transformations.

– Current fault detection schemes:
  • Dependent on the way the S-box is constructed.
  • May not be used for all the implementations of the S-box.

– The presented fault detection scheme:
  • Independent of the type of the implementation of the S-box.
  • Can be applied to both look-up table and composite field realizations of the S-box.
Multiplication: A Previous Approach

- The 8-bit input of the multiplicative inversion is multiplied by the 8-bit output and the $n$-bit result of the multiplication is compared with the $n$-bit actual result.
Multiplication: A Previous Approach

- Based on the relation of the input and output of the multiplicative inversion.

- The multiplication approach is costly
  - 64 ANDs and 84 XORs (after sub-expression sharing)
  - Approximately 93% area overhead for a typical composite field realization

- Previous schemes suggest using the two least significant bits of the result for comparison.

- We suggest using the most and least significant bits resulting in 7% area overhead reduction.
Multiplication: A Previous Approach

• Disadvantages:

  • Fault detection of the multiplicative inversion in the S-box/inverse S-box.

  • Does not include the affine/inverse affine transformation.

  • Therefore, it is not suitable whenever the output of the multiplicative inversion is not available.
The Proposed Structure-Independent (Scheme 1)

- Independent of the type of realization of the S-box/inverse S-box
- Takes the affine/inverse affine affine transformation into account
- Suitable for look-up table as well as composite field realizations of the S-box/inverse S-box
The Proposed Structure-Independent (Scheme 1)

- Theorem:

Let \( S = s_7 \alpha^7 + s_6 \alpha^6 + s_5 \alpha^5 + s_4 \alpha^4 + s_3 \alpha^3 + s_2 \alpha^2 + s_1 \alpha + s_0 \) and \( S' = s_7' \alpha^7 + s_6' \alpha^6 + s_5' \alpha^5 + s_4' \alpha^4 + s_3' \alpha^3 + s_2' \alpha^2 + s_1' \alpha + s_0' \) be the input and the output of the S-box, respectively. Then we have the following relation between the input and the output of the S-box:

\[
M s' + m = 1
\]

Where, matrix \( M \) and vector \( m \) are functions of the input of the S-box, \( s' \) is the column vector of the coordinates of the output and vector \( 1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \).
The Proposed Structure-Independent (Scheme 2)

- Although the structure-independent scheme detects all errors in the output of the S-box, its implementation is costly (64 ANDs and 111 XORs after sub-expression sharing).

- To reduce the cost, we obtain a signature (parity) of the result.
The Proposed Structure-Independent (Scheme 2)

- Single-bit parity can be used and compared with one for detecting any combination of odd number of erroneous bits at the result as follows

\[ P_{(MS'+m)} = P_b = 1 \]

- This needs 20 XORs and 8 ANDs with the delay of 4 XORs and one AND.

- Using an OR tree, the error indication flags of 16 S-boxes are ORed to obtain the flag of the SubBytes transformations.
Simulation Results

- Our simulations are based on considering the S-box and its fault detection circuit by injecting all possible errors to the output.

- For the 255 inputs, 255 possible erroneous outputs for both stuck at zero and one are evaluated.

- Our simulations show that the error coverage for one S-box is approximately 50%.

- If we consider 16 S-boxes in SubBytes, the cases in which any of the S-boxes detect an error are among the error detection cases resulting in the error coverage of 99.998%.
FPGA Implementations

- As a typical implementation, we have implemented the look-up table realization of the S-box in the AES encryption.

- XILINX ISE 8.2 and Virtex 5 family is used.

- We use pipelined distributed RAMs for implementing the SubBytes transformation.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Operation</th>
<th>SRs, SLUTs</th>
<th>Slice overhead</th>
<th>Freq. (Mhz)</th>
<th>Thro’put (Gbps)</th>
<th>Efficiency (Mbps/slice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES with SubBytes using Pipelined Distributed RAMs (xc5vlx30-3)</td>
<td>Original AES</td>
<td>2560, 8490</td>
<td>0% SR</td>
<td>482.998</td>
<td>61.8</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td>Proposed scheme for SubBytes,ShiftRows</td>
<td>2560, 9806</td>
<td>0% SR 15% SLUT</td>
<td>482.998</td>
<td>61.8</td>
<td>25.2</td>
</tr>
</tbody>
</table>
Comparison

– We have implemented the proposed scheme for SubBytes used the scheme in [Bertoni et al., 2003] for other transformations in the AES encryption.

– Our scheme and the one that uses 512*9 memory cells for the S-box are compared as follows

<table>
<thead>
<tr>
<th>Operation</th>
<th>Device</th>
<th>SRs, SLUTs</th>
<th>Slice overhead</th>
<th>Freq. (Mhz)</th>
<th>Thro’put (Gbps)</th>
<th>Efficiency (Mbps/slice)</th>
</tr>
</thead>
<tbody>
<tr>
<td>scheme in [1] using 512 × 9 ROMs for S-box</td>
<td>xc5vlx85-3</td>
<td>3600, 16138</td>
<td>40% SR 90% SLUT</td>
<td>478.286</td>
<td>61.2</td>
<td>15.2</td>
</tr>
<tr>
<td>Proposed scheme for SubBytes scheme in [1] for others</td>
<td>xc5vlx30-3</td>
<td>2560, 10559</td>
<td>0% SR 24% SLUT</td>
<td>482.998</td>
<td>61.8</td>
<td>23.4</td>
</tr>
</tbody>
</table>

Conclusions

• In this paper, a structure-independent fault detection scheme for the AES SubBytes transformation has been presented.

• This scheme can also be used for the fault detection of InvSubBytes transformation by swapping the inputs and the outputs.

• The presented scheme detects most of the random faults in the SubBytes and ShiftRows transformations, independent of the location of the faults.

• Finally, the area overhead of our structure-independent scheme is reasonable for resource constrained hardware implementations of the AES.
Thank you!