INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN: 2278 – 0211 (Online)

A Concurrent Fault Detection Scheme For The Aes Using Composite Fields

N. Chandrakumar
Student, VLSID, SITAMS, Chittoor, A.P, India
G. Srihari
Assistant Professor, VLSID, SITAMS, Chittoor, A.P, India

Abstract:
The faults that accidentally or maliciously occur in the hardware implementations of the Advanced Encryption Standard (AES) may cause erroneous encrypted/decrypted output. The use of appropriate fault detection schemes for the AES makes it robust to internal defects and fault attacks. In this paper, we present a lightweight concurrent fault detection scheme for the AES. In the proposed approach, the composite field S-box and inverse S-box are divided into blocks and the predicted parities of these blocks are obtained. Through exhaustive searches among all available composite fields, we have found the optimum solutions for the least overhead parity-based fault detection structures. A low-cost parity-based fault detection scheme for the S-box and the inverse S-box using composite fields. For increasing the error coverage, the predicted parities of the five blocks of the S-box and the inverse S-box are obtained (three predicted parities for the multiplicative inversion and two for the transformation and affine matrices). It is interesting to note that the cost of our multi-bit parity prediction approach is lower than its counterparts which use single-bit parity. It also has higher error coverage than the approaches using single-bit parities. We have implemented both the proposed fault detection S-box and inverse S- box and other Counterparts. The complexities of the proposed fault detection scheme are lower. The least area and delay overhead fault detection structures for the optimum composite fields using both polynomial basis and normal basis.

Key words*: AES, composite fields, error coverage, fault detection*

1.Introduction

The Advanced Encryption Standard (AES) has been lately accepted by NIST as the symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plain text input, which is limited to 128 bits, and a key that can be specified to be 128 (AES-128), 192 or 256 bits to generate the cipher text. In the AES-128, which is hereafter referred to as the AES, the cipher text is generated after 10 rounds, where each encryption round (except for the final round) consists of four transformations. The four transformations in the AES encryption include Sub Bytes (implemented by 16 S-boxes), Shift Rows, Mix Columns, and AddRoundKey. Furthermore, to obtain the original plaintext from the cipher text, the AES decryption algorithm is utilized. The decryption transformations are the reverse of the encrypted ones. Among the transformations in the AES, only the S-boxes in the encryption and the inverse S-boxes in the decryption are nonlinear. It is interesting to note that these transformations occupy much of the total AES encryption/decryption area.

Therefore, the fault detection schemes for their hardware implementations play an important role in making the standard robust to the internal and malicious faults. There exist many schemes for detecting the faults in the hardware implementation of the AES, among them, the schemes presented in are independent of the ways the AES S-box and inverse S-box are implemented in hardware. Moreover, there exist other fault detection schemes that are suitable for a specific implementation of the S-box and the inverse S-box. The approach in and the one in which is extended in are based on using memories (ROMs) for the S-box and the inverse S-box. Moreover, a fault tolerant scheme which is resistant to fault attacks is presented in. To protect the combinational logic blocks used in the four transformations of the AES, either the parity-based scheme proposed in or the duplication approach is implemented. Furthermore, to protect the memories used for storing the expanded key and the state matrix, either the Hamming or Reed– Solomon error correcting code is utilized. It is noted that our proposed fault detection approach is only applied to the composite field S-box and inverse S-box. Whereas, the scheme presented in uses memories Using ROMs may not be preferable for high performance AES implementations. Therefore, for applications requiring high performance, the S-box and the inverse S-box are implemented using logic gates in composite fields the schemes are suitable for the composite field implementation of the S-box and the inverse S-box. The approach is based on using the parity-based fault detection method for a specific S-box using composite field and polynomial basis for covering all the single faults. In the scheme of, the fault detection of the multiplicative inversion of the S-box is considered for two specific composite fields. The transformation and affine matrices are excluded in this approach. Moreover, in predicted parities have been used for the multiplicative inversion of a specific S-box using composite field and polynomial basis. Furthermore, the transformation matrices are also considered. Finally, in the parity-based approach in, through exhaustive search among all the fault detection S-boxes utilizing five predicted parities using normal basis, the most compact one is obtained.

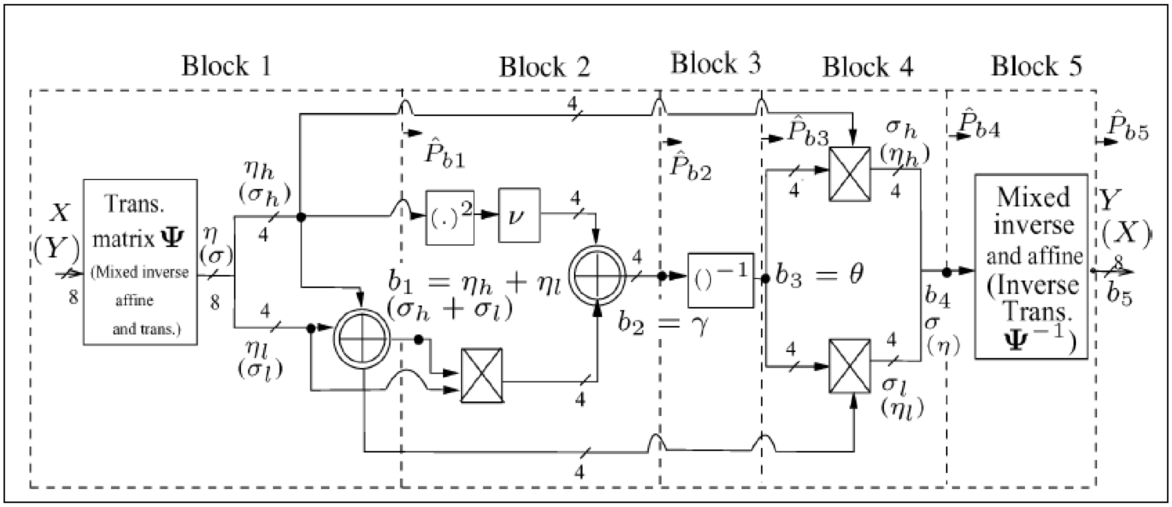

Figure1: The S-Box (The Inverse S-Box) Using Composite Fields And Polynomial Basis And There Fault Detection Blocks

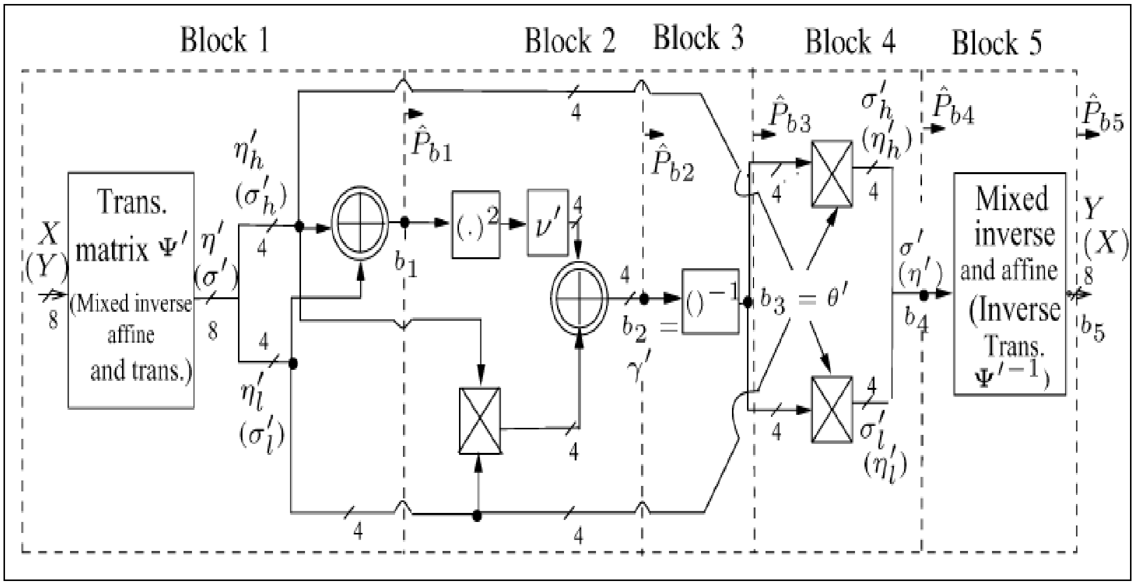

Figure 2: The S-box (the inverse S-box) using composite fields and normal basis and their fault detection blocks

2.Preliminaries

In this section, we describe the S-box and the inverse S-box operations and their composite-field realizations. The S-box and the inverse S-box are nonlinear operations which take 8-bit inputs and generate 8-bit outputs. In the S-box, the irreducible polynomial. The composite fields can be represented using normal basis or polynomial basis. The S-box and inverse S-box for the polynomial and normal bases are shown in Figs. 1 and 2, respectively

3.Fault Detection Scheme

To obtain low-overhead parity prediction, we have divided the S-box and the inverse S-box into five blocks as shown in Figs. 1 and 2. In these figures, the modulo-2 additions, consisting of 4 XOR gates, are shown by two concentric circles with a plus inside. and b5 =Y similarly, from Fig. 2 and b5 =Y One can replace $\eta(\eta\ \cdot)$win σ(σ') and with for the inverse S-box. In the following, we have exhaustively searched for the least overhead parity predictions of these blocks denoted by in Figs. 1 and 2

4.Error Simulations

If exactly one bit error appears at the output of the S-box (respectively inverse S-box), the presented fault detection scheme is able to detect it and the error coverage is about 100%. This is because in this case, the error indication flag of the corresponding block alarms the error. However, due to the technological constraints, single stuck-at error may not be applicable for an attacker to gain more information. Thus, multiple bits will actually be flipped and hence multiple stuck-at errors are also considered in this paper covering both natural faults and fault attacks. For the calculation of the error coverage for the multiple errors, we define **Pi** as the probability of error detection in the block $1 \leq i \leq 5$, in Figs. 1 and 2. Then, the probability of not detecting the errors in block i is (1-Pi) . For randomly distributed errors in the S-box (respectively inverse S-box), this probability for each block is independent of those of other blocks. Therefore, one can derive the equation for the error coverage of the randomly distributed errors as

Operations	Field	Errors covered	Error Coverage
S-box (Inverse S-box)	PB1	485,008(485,106)	97.002 %(97.021%)
	PB2	485,039(485,015)	97.008 %(97.002%)
	NB	485,015(485,174)	97.003 %(97.035%)

Table 1: Error Simulation Results Of The Optimum S-Box And Inverse S-Box After Injecting 5000000 Errors

$$EC\% = 100 \times \left(1 - \prod_{i\varepsilon s} (1 - Pi) \right)\%$$

Where **S** is the set of the block numbers where the faults are injected. For randomly distributed errors, the error coverage for each block is $p_i \approx 1/2$. Then, the representation of can be simplified as

$EC\% = 100 \times \left(1 - \left(\frac{1}{2}\right) n\right)\%$, where, is the number of blocks.

Therefore, if multiple errors are randomly distributed in all blocks, the error coverage reaches 97% using n=5 error indication flags. We have performed error simulations for the S-boxes and the inverse S-boxes using the optimum composite field obtained in the previous section to confirm our above theoretical computation. In our simulations, we use stuck-at error model at the outputs of the five blocks forcing one or multiple nodes to be stuck at logic one (for stuck-at one) or zero (for stuck-at zero) independent of the error-free values. We use Fibonacci implementation of the LFSRs for injecting random multiple errors, where, the numbers, the locations and the types of the errors are randomly chosen. In this regard, the maximum sequence length polynomial for the feedback is selected. The injected errors are transient, i.e., they last for one clock cycle. However, the results would be the same if permanent errors are considered. The results of the error simulations using Xilinx ISE version 9.1i Simulator (ISim)^{2} are presented in Table I. As seen in this table, up to 500 000 random errors are injected for both the S-box and the inverse S-box. It is noted that in these tables,
Optimum polynomial basis denoted by ,
GF ((2)$^{2)2}$)^{2}) denoted by 97% which is the same as our theoretical computation in this section. This error coverage will be increased if the outputs of more than one S-box (respectively inverse S-box) of the AES implementation are erroneous. In this case, the errors detected in any of 16 S-boxes (respectively inverse S-boxes) contribute to the total error coverage. Thus, error coverage of very close to 100% is achieved

5.ASIC And FPGA Implementations And Comparisons

In this section, we compare the areas and the delays of the presented scheme with those of the previously reported ones in both application-specific integrated circuit (ASIC) and field programmable gate array (FPGA) implementations. We have implemented the S-boxes using memories and the ones presented in (the hardware optimization of which use polynomial basis representation in composite fields. We have also implemented the fault detection schemes proposed in (both united and parity-based), and which are based on the ROM-based implementation of the S-box. The results of the implementations for both original and fault detection schemes (FDS) in terms of delay and area have been tabulated in Tables II and III. As seen in these tables, the original structures are not divided into blocks and full optimization of the original entire architecture as a single block is performed in both ASIC and FPGA. This allows us to find the actual overhead of the presented fault detection scheme as compared to the original structures which are not divided into five blocks. We have used 0.18-μm CMOS technology for the ASIC implementations. These architectures have been coded in VHDL as the design entry to the Synopsys Design Analyzer. The results are tabulated in Table II. Moreover, for the FPGA implementations in Table III, the Xilinx Virtex-II Pro FPGA (xc2vp2-7) is utilized in the Xilinx ISE version 9.1i. Furthermore, the synthesis is performed using the XST. As seen in Tables II and III, we have implemented the fault detection scheme presented based on using redundant units for the S-box (united S-box). Furthermore, the fault detection scheme proposed in [10] is implemented. This scheme uses 512 9 memory cells to generate the predicted parity bit and the 8-bit output of the S-box. One can obtain from Tables II and III that for both of these schemes, the area overhead is more than 100%. As mentioned in the introduction, the approach in utilizes the scheme in for protecting the combinational logic elements, whose implementation results are also shown in Tables II and III. Additionally, for certain AES implementations containing storage elements, one can use the error correcting code-based approach presented in addition to the proposed scheme in this paper to make a more reliable AES implementation. Moreover, the parity-based scheme in which only realizes the multiplicative inversion using memories is implemented. As seen in these tables, we have also implemented the schemes it is noted that the scheme in for the multiplicative inversion and does not present the parity predictions for the transformation matrices. Moreover, we have applied the presented fault detection scheme to the S-boxes. As seen in bold faces in Tables II and III, with the error coverage of close to 100%, the presented low-complexity fault detection S-boxes (presented in Section III) are the most compact ones among the other S-boxes. The optimum S-box and inverse S-box using normal basis have the least Hardware complexity with the fault detection scheme. Moreover, as seen in the tables, the optimum structures using composite fields and polynomial basis (PB1 and PB2) have the least post place and route timing overhead among other schemes. It is noted that using sub-pipelining for the presented fault detection scheme in this paper, one can reach much faster hardware implementations of the composite field fault detection structures.

Operation	**Architecture**		**Area (µm^2) / Delay (NS)**	
	Structure	**FDS**	**Original**	**FDS**
S-box	ROM SB	Untied S-box [2],[8]	169 X 10^3 / 5.4	344 X 10^3 / 7.7
	ROM SB	Two 256 X 9 ROMs [10]	169 X 10^3 / 5.4	378 X 10^3 / 5.8
	ROM (MULT. INV.)	Parity-based SB	185 X 10^3 / 5.4	191 X 10^3 / 5.9
	PB[20]	[13] (Mult. Inv.)	5315/12.0	6869 / 12.8
	PB[20]	[14]	5315 /12.0	7047 / 14.1
	PB[20]	[12] for the original SB	5315/12.0	6763 / 14.1
	PB [18]	Proposed scheme applied	5642/11.3	7113 / 13.0
	PB[22]	Proposed scheme applied	5547 /12.9	7034 / 13.8
	NB	[15]	5179 / 10.6	6712 / 12.5
	PB1	This work	5290 / 9.2	6723 / 11.5
	PB2	This work	5290 / 9.4	6739 / 11.5
Inverse S-box	NB	This work	5187 / 13.2	6480 / 14.5
	PB1	This work	5225 / 10.9	6537 / 13.0
	PB2	This work	5274 / 9.4	6619 / 11.3

Table 2: ASIC Implementation Of The Fault Detection Schemes Of The AES Encryption Using 0.18-µm CMOS Technology

Operation	Architecture		Slice / Delay (ns)	
	Structure	FDS	Original	FDS
S-box	ROM SB	Untied S-box [2],[8]	69 / 3.826	150 /5.398
	ROM SB	Two 256 X 9 ROMs [10]	69 / 3.826	159 / 4.282
	ROM (MULT. INV.)	Parity-based SB	88 / 5.734	100 / 6.370
	PB [20]	[13] (Mult. Inv.)	33 / 9.375	44 / 9.860
	PB [20]	[14]	33 / 9.375	47 / 9.996
	PB [20]	[12] for the original SB	33 / 9.375	42 / 10.317
	PB [18]	Proposed scheme applied	38 / 9.986	50 / 9.582
	PB [22]	Proposed scheme applied	37 / 7.284	47 / 7.465
	NB	[15]	31 / 7.284	39 / 10.026
	PB1	This work	31 / 9.339	40 / 7.465
	PB2	This work	32 / 7.356	41 / 8.150
Inverse S-box	NB	This work	31 / 7.736	38 / 7.964
	PB1	This work	32 / 6.992	42 / 7.423
	PB2	This work	32 / 7.550	44 / 8.181

Table 3: XILINX VIRTEX-II PRO FPGA Implementations Of Fault Detection Schemes For The S-Box And Inverse S-Box Of The AES Encryption

AES encryption	Optimum s-box	Area (µm2)		Freq. (MHz)
		S -	All	
Original without fault detection	PB1	692781	859471	79.4
	PB2	704490	871180	91.8
	NB	680590	845426	73.5
Presented scheme for subBytes (shift	PB1	956233	-	78.8
	PB2	972217	-	89.2
	NB	946476	-	69.2
Presented scheme for subBytes (shift Rows) scheme	PB1	-	9881	68.2
	PB2	-	9921	70.22
	NB	-	9405	60.3

Table 4: ASIC Implementation Of The Fault Detection Schemes Of The AES Encryption Using 0.18-µm MOS Technology

We have also implemented the AES encryption using the presented optimum S-boxes excluding the key expansion. Then, we have added the proposed scheme for Sub Bytes and Shift Rows considering that Shift Rows are the rewiring from the output of Sub Bytes. The results are presented in Tables IV and V. As one can notice, the S-boxes occupy more than three fourths of the AES encryption. As shown in these tables, the most compact AES encryption with and without the fault detection scheme is for normal basis. Furthermore, the frequency degradation is negligible. Moreover, the original AES encryption for PB_2 and the ones with fault detection for $\hat{P}_{b1}$ and PB_2 have the highest working frequencies. In addition, as seen in the tables, we have applied the presented scheme to Sub Bytes and Shift Rows and used the scheme in for the other transformations

AES Encryption	Optimum s-box	Slice		Freq. (MHz)
		S - boxes	All	
Original without fault detection	PB1	5248 (77)%	6760	81.1
	PB2	5417 (78)%	6913	89.8
	NB	5112 (78)%	6579	75.8
Presented scheme for subBytes (shift Rows)	PB1	6896	-	79.3
	PB2	6958	-	84.0
	NB	6342	-	73.2
Presented scheme for subBytes (shift Rows) scheme in 10	PB1	-	9881	65.8
	PB2	-	9921	64.8
scheme in 10	NB	-	9405	60.8

Table 5: XILINX VIRTEX-II PRO FPGA Implementations Of Fault Detection Schemes Of The AES Encryption

6.Conclusion

In this paper, we have presented a high performance parity based concurrent fault detection scheme for the AES using the S-box and the inverse S-box in composite fields. Using exhaustive searches, we have found the least complexity S-boxes and inverse S-boxes as well as their fault detection circuits. Our error simulation results show that very high error coverage’s for the presented scheme are obtained. Moreover, a number of fault detection Schemes from the literature have been implemented on ASIC and FPGA and compared with the ones presented here.Our implementations show that the optimum S-boxes and the inverse S-boxes using normal basis are more compact than the ones using polynomial basis. However, the ones using polynomial basis result in the fastest implementations. We have also implemented the AES encryption using the proposed fault detection scheme. The results of the ASIC and FPGA mapping show that the costs of the presented scheme are reasonable with Acceptable post place and route delays.

7.Acknowledgment

The authors would like to thank the reviewers for their comments.

8.References

1. National Institute of Standards and Technologies, Announcing the Advanced Encryption Standard (AES) FIPS 197, Nov. 2001.
2. R. Karri, K. Wu, P. Mishra, and K. Yongkook, “Fault-based side-channel cryptanalysis tolerant Rijndael symmetric block cipher architecture,” in Proc. DFT, Oct. 2001, pp. 418–426.
3. R. Karri, K. Wu, P. Mishra, and Y. Kim, “Concurrent error detection schemes for fault-based side-channel cryptanalysis of symmetric block ciphers,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 21, no. 12, pp. 1509–1517, Dec. 2002.
4. A. Satoh, T. Sugawara, N. Homma, and T. Aoki, “High-performance concurrent error detection scheme for AES hardware,” in Proc. CHES, Aug. 2008, pp. 100–112.
5. L. Breveglieri, I. Koren, and P. Maistri, “Incorporating error detection and online reconfiguration into a regular architecture for the advanced encryption standard,” in Proc. DFT, Oct. 2005, pp. 72–80.
6. M. Karpovsky, K. J. Kulikowski, and A. Taubin, “Differential fault analysis attack resistant architectures for the advanced encryption standard,” in Proc. CARDIS, Aug. 2004, vol. 153, pp. 177–192.