ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 3, March 2013

An Effective S-Box Comparison by Parity Based Fault Detection Scheme Using Composite Fields.

M. Hemalatha, PG Scholar,
S. Sridevi Sathya Priya, Assistant Professor(SG),
P. Karthigaikumar Associate Professor

***Abstract*—AES is the most effective algorithm for network security that is more confidential, reliable and robust. A key step in the Advanced Encryption Standard (AES) algorithm is the subbyte transformation and its inverse . The hardware implementation of such nonlinear parts however leads to an erroneous output due to the faults that occurs accidentally or intentionally. To overcome these faults a concurrent fault detection scheme must be adopted. In this paper, the composite field s box and inverse S box are divided into blocks and the predicted parities of these blocks are obtained. The faults are being injected in the sub byte input and the corresponding fault detection has been carried out. The comparison between the normal SBox and the composite field Sbox has been done using Xilinx 9.1 ISE and the corresponding simulation has been done using Modelsim software. The resulting area report reveals the better method for the efficient fault detection scheme of AES algorithm.**

***Index Terms*—AES, Fault Injection, Inverse S-Box, Multiple Stuck at Fault, Parity Prediction, S-Box.**

I. INTRODUCTION

The short form of Advanced Encryption Standard is AES and is defined in Federal Information Processing Standard (FIPS) 192 of United States encryption standard, which was published in Nov 2001. It got approved as a Federal Standard in May 2002. Many schemes have been introduced to detect faults in the hardware implementation of AES [3]-[16] in which the methodologies presented in [3]-[8] does not look for the hardware implementation of S Box and Inverse S Box. There are ROM based memories schemes that are mentioned in [9] and in [10]. Fault tolerant schemes have been introduced in [12]. To implement the parity based approach since it helps to protect the combinational logic blocks as in [11]. To protect the memories used for storing the expanded key and state matrix, Hamming code or Reed-Solomon error correcting code is used. But in our proposed approach ROM is not preferable for high performance implementation of AES. For high performance application, the S Box and Inverse S Box are implemented using logic gates in composite fields [17].

The approaches in [13]-[16] includes the composite field implementation of S Box and Inverse S Box. The approach in [13] is based on polynomial basis implementation using parity-based fault detection method for a specific S Box for covering all the single faults. The methodology in [14] gives the fault detection of S Box of the multiplicative inversion in the both composite fields without taking into account the transformation matrix and affine transformation matrices.

II. PRELIMINARIES

A description of S-Box and Inverse S-Box operation is done in this section. Then, the composite field realization of the two non-linear operations using polynomial basis and normal basis is explained.

A. The S-Box and Inverse S-Box

The nonlinear operation of S-Box and its inverse takes 8-bit input and produces 8-bit output. A binary field GF (2^8) is constructed using the irreducible polynomial of $P(x) = x^8 + x^4 + x^3 + x + 1$. The input and output of the S-Box be $X \in GF(2^8)$ and $Y \in GF(2^8)$. If so, then this nonlinear operation consists of a multiplicative inversion, i.e., $X^{-1} \in GF(2^8)$ followed by affine transformation which consists of a matrix A and vector b to produce the output as

$$\begin{pmatrix} s_7 \\ s_6 \\ s_5 \\ s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

B. Multiplicative Inversion Using Composite fields

The normal basis [17] and polynomial basis [18]-[21] representation of composite fields. Here, for the elements in the binary field GF (2^8) is represented as X and Y. There are

Manuscript received Feb, 2013.
***M. Hemalatha**, Electronics and Communication, Karunya University, ., Coimbatore, India,9042674036.*
***S. Sridevi Sathyapriya**, Electronics and Communication, Karunya University, Coimbatore, India.*
***P. Karthigaikumar**, Electronics and Communication, Karunya University, Coimbatore, India,*

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 3, March 2013

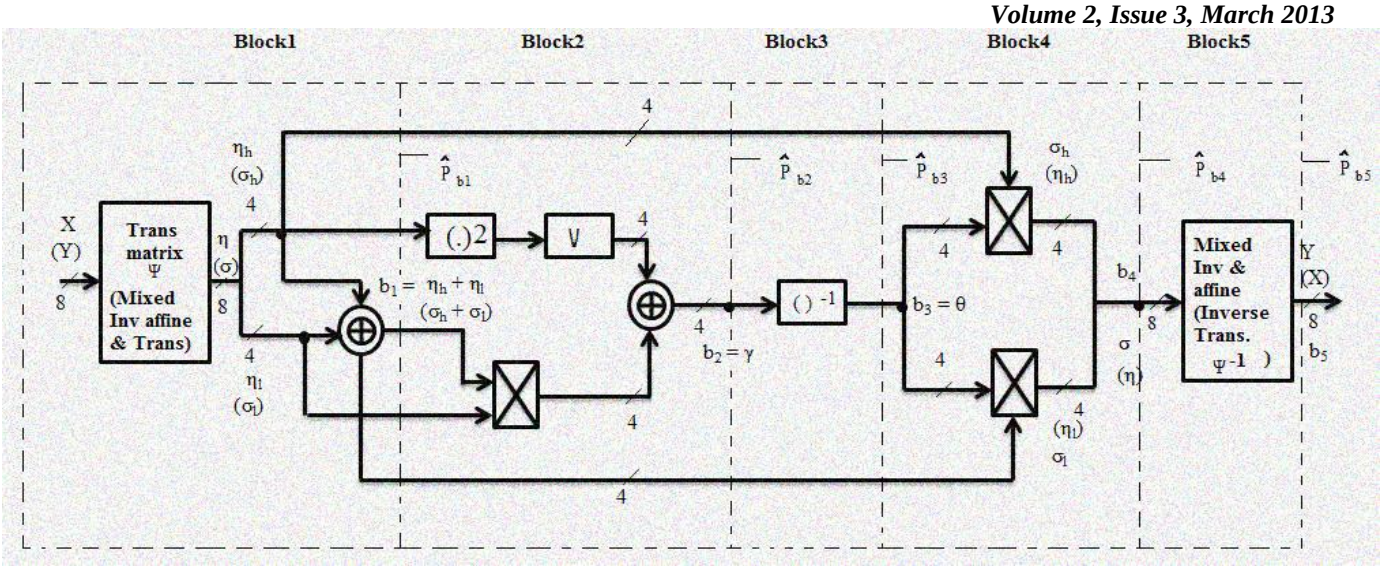

Fig.1 The composite Field Representation of S-Box (the Inverse S-Box) using polynomial basis [17] and their fault detection blocks.

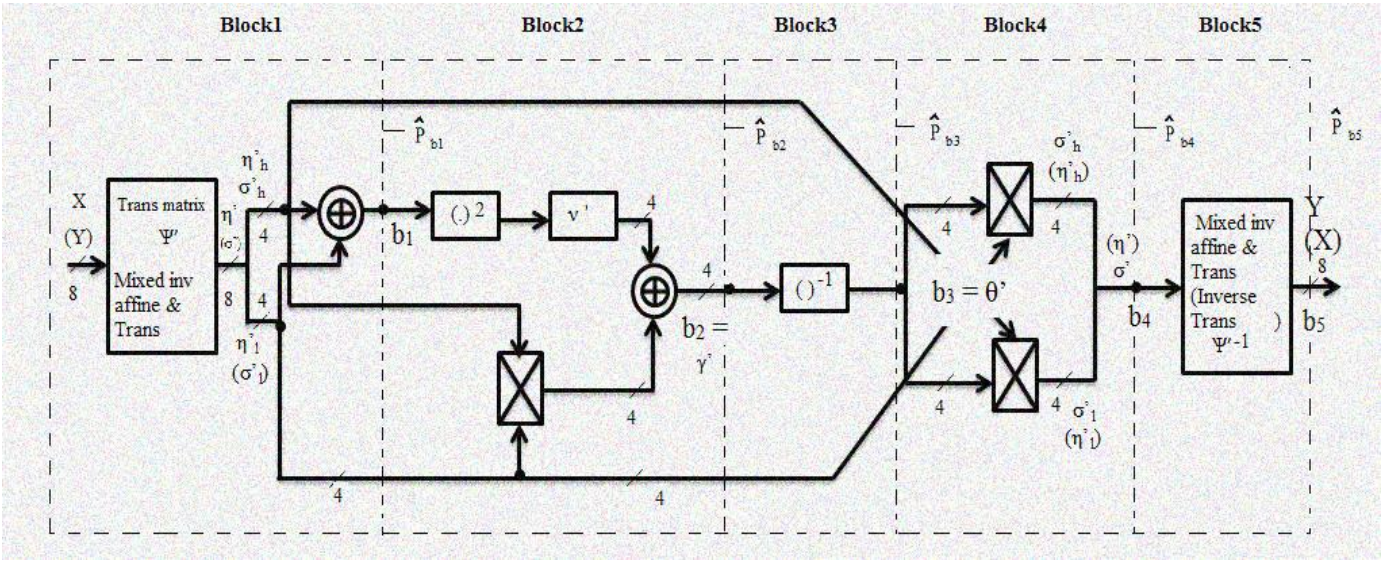

Fig.2 The Composite Field Representation of S-Box(the Inverse S-Box) using Normal basis [16] and their fault detection blocks.

other representation of GF (2^4) as η_h, η_l, ν and of GF (2^2) as θ and γ. The S-Box and Inverse S-Box for normal basis and polynomial basis are given as Figs.1 and 2. The transformation of field element X in the binary field GF (2^8) to the equivalent representation in composite fields GF (2^4). The calculation of multiplicative inversion is given as in [17] & [18].

III. FAULT DETECTION SCHEME

A multiple stuck-at fault model at logic level is used in this paper. To obtain low-overhead parity prediction, we have divided the S-box and the inverse S-box into five blocks as shown in Figs. 1and 2. In these figures, the 4 XOR gates in modulo-2 additions, are shown by two concentric circles with a plus inside. Furthermore, the multiplications in GF (2^4) are shown by rectangles with crosses inside. Let be the output of the block denoted by dotted lines in Figs. 1 and 2 for the S-box.

As seen in Fig. 1, b1 = ηh+ηl, b2 = γ, b3 = θ, b4 = σ and b5 = Y and Similarly, from Fig. 2, b1 = η'h +η'l, b2 = γ', b3 = θ', b4 = σ' and b5 = Y. One can replace η(η') with σ(σ') and X with Y for the inverse S-box. In the following, an exhaustive search for the least overhead parity predictions have been done of these blocks denoted by in both diagrammatic representations.

A. *The S-Box and Inverse S-Box using Polynomial Basis*

The implementation complexities of different blocks of the S-box and the inverse S-box and those for their predicted parities are dependent on the choice of the coefficients ν Є GF(2^4) and Φ Є GF(2^2) in the irreducible polynomials u^2+u+ ν and v^2+ v+ Φ used for the composite fields. Our goal in the following is to find ν Є GF(2^4) and Φ Є GF(2^2) for the composite fields GF(((2^2)^2)^2) for the composite fields GF(2^4) so that the area complexity of the entire fault detection implementations becomes optimum. These parameters determine the complexities of some blocks are explained as.

Blocks 1 and 5: Based on the possible values of ν and Φ in $GF(((2^2)^2)^2)$ the transformation matrices in Fig. 1 in blocks 1 and 5 of the S-box and the inverse S-box can be constructed using the algorithm presented in [21]. Using an exhaustive search, eight base elements in $GF(((2^2)^2)^2)$ (ν in $GF((2^4)^2)$) to which eight base elements in $GF(((2^2)^2)^2)$ (or $GF((2^4)^2)$) of are mapped, are found to construct the transformation matrix.

Blocks 2 and 4: In Fig. 1, block 2 of the S-box and the inverse S-box consists of a multiplication, an addition, a squaring and a multiplication by constant ν in $GF((2^2)^2)$. We have exhaustively searched for and obtained the optimum implementation for different values of νs. Moreover, block 4 in Fig. 1 is independent of the value of ν. Therefore, the complexity of the predicted parity for this block is the same for all possible νs .

Block 3: As in [2] ,the parity prediction of the polynomial basis have been derived to be the following equation $□_{b1} = x_0$, $□_{b2} = \eta_3(Б_7 + \eta_4) + (Б_7 + P_{\eta h})+\eta_1(\eta_6+\eta_4)+\eta_0 P_{\eta h}+\eta_6+\eta_7$, $□_{b3} = (\gamma_2 \gamma_1 □ \gamma_0)+\gamma_1 \gamma_3$, $□_{b4} = \eta_3(\theta_3+\theta_0)+\eta_2(P_\Theta+\theta_3)+\eta_1(\theta_2+\theta_0)+\eta_0 P_\Theta$, $□_{b5} = \sigma_7+\sigma_5+\sigma_3+\sigma_2+\sigma_0$ where $P_{\eta h} = \eta_7+\eta_6+\eta_5+\eta_4$ and $P_\Theta = \theta_3+\theta_2+\theta_1+\theta_0$. This could be applied to Inverse s-Box also.

B. The S-Box and Inverse S-Box Using Normal Basis:

The fault detection for S-Box using Normal basis has been given in [15]. The difference here is that the inverse S-Box has been taken and their parity predictions are combined. The predicted parities for normal basis are taken to be the proposition 3[2]. The area overhead to be predicted using parity prediction and is optimum.

C. Fault injection

Here, multiple stuck at fault is to be injected manually in any of the two operations in the blocks. The codings have been done in VHDL language using Xilinx ISE 9.2i. The general fault detection involves encoding the message sent and detecting the faults. Change in parity of the input and injected parity leads to the fault occurrence. In the polynomial basis two stuck at 1 fault have been introduced in the Block 1-4 section of the Code. In normal basis the two faults have been injected in multiplicative inverse section of the code.

IV. COMPARISON

A. S-Box

The Look up table S-Box and the two composite fields with fault injection have been taken and compared here. This provides the view of a better method in terms of Area overhead that could be applied in AES encryption and Decryption and make it more secure and robust when applied in Networks.

V. IMPLEMENTATION

The code for the S-Box and Inverse S-Box is done in VHDL language using XILINX ISE 9.2i and simulated by MODELSIM SE 6.5. The two methods using Polynomial basis and Normal basis as well as a LUT S-Box have been compared in terms of both area and power. Since there could be no much difference in power the table have been plotted for area and the best suitable method has been analysed. The corresponding Area report has been given.

Table I. Comparison of S-Box

Architecture	**Area (Gate Counts)**
Proposed approach – Polynomial Basis	465
Proposed approach – Normal Basis	411
Look up table	1200

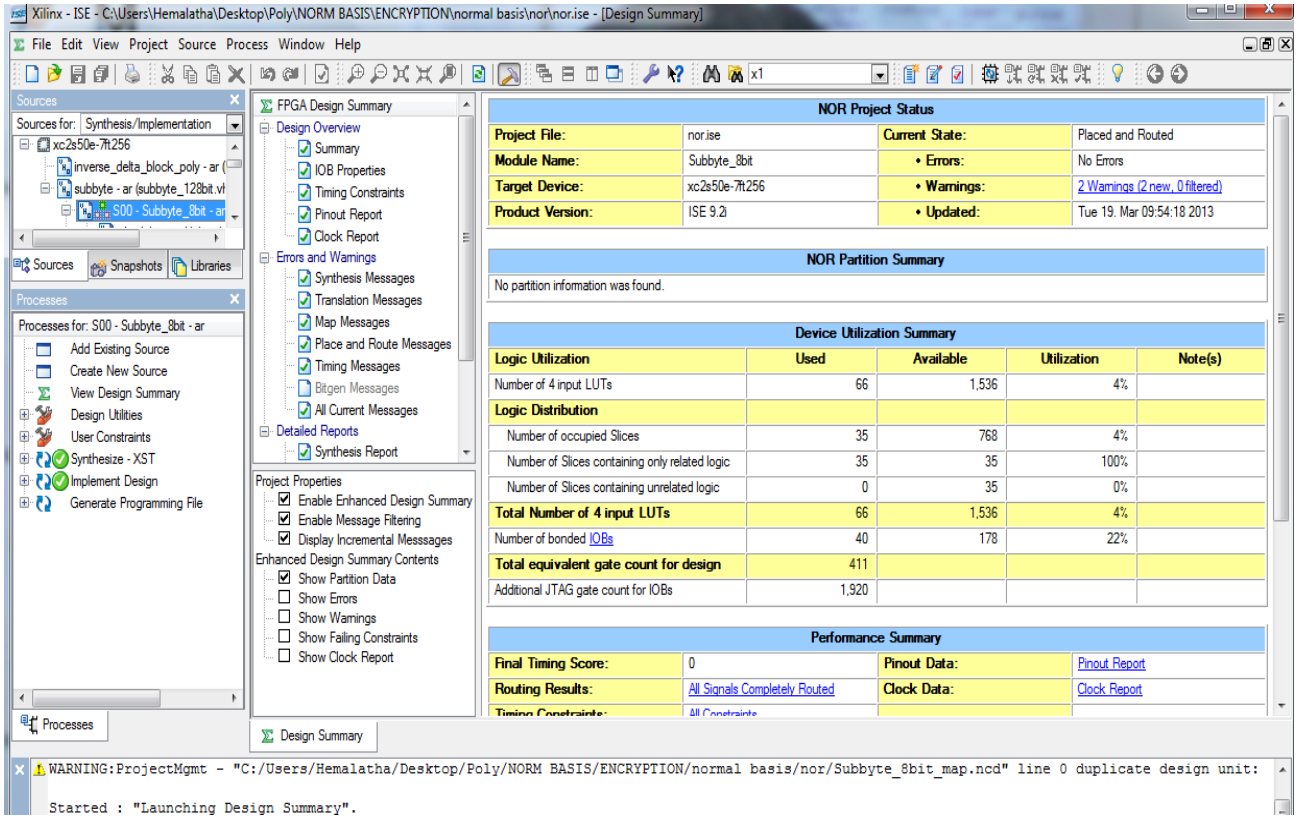

Fig.3 Area report for the Normal Basis S-Box

VI. CONCLUSION

From the above area report the normal basis S-Box is found to be the better method and hence this could be further implemented in AES encryption algorithm to find out the effective usage in Networks. This is suitable for implementing in hardware. Hence effective security could be given to networks. Further the correction of errors can also be done.

REFERENCES

[1] National Institute of Standards and Technologies, Announcing the Advanced Encryption Standard (AES) FIPS 197, Nov. 2001.
[2] Mehra Mozaffari-Kermani, Arash Reyhani-Masoleh, "A Lightweight High-Performance Fault Detection Scheme for the Advanced Encryption Standard Using Composite Fields", IEEE Trans. VLSI systems, pp. 85-91.
[3] 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.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] P. Maistri and R. Leveugle, "Double-data-rate computation as a countermeasure against fault analysis," *IEEE Trans. Computers*, vol. 57, no. 11, pp. 1528–1539, Nov. 2008.
[9] C. H. Yen and B. F.Wu, "Simple error detection methods for hardware implementation of advanced encryption standard," *IEEE Trans. Computers*, vol. 55, no. 6, pp. 720–731, Jun. 2006.
[10] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "A parity code based fault detection for an implementation of the advanced encryption standard," in *Proc. DFT*, Nov. 2002, pp. 51–59.
[11] G. Bertoni, L. Breveglieri, I. Koren, P. Maistri, and V. Piuri, "Error analysis and detection procedures for a hardware implementation of the advanced encryption standard," *IEEE Trans. Computers*, vol. 52, no. 4, pp. 492–505, Apr. 2003.
[12] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, "A fault-tolerant DFA-resistant AES core," in *Proc. ISCAS*, 2008, pp. 244–247.
[13] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "Parity-based fault detection architecture of S-box for advanced encryption standard," in *Proc. DFT*, Oct. 2006, pp. 572–580.
[14] S.-Y. Wu and H.-T. Yen, "On the S-box architectures with concurrent error detection for the advanced encryption standard," *IEICE Trans. Fundam. Electron., Commun. Comput. Sci.*, vol. E89-A, no. 10, pp. 2583–2588, Oct. 2006.
[15] A. E. Cohen, "Architectures for Cryptography Accelerators," Ph.D. dissertation, Univ. Minnesota, Twin Cities, Sep. 2007.
[16] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A lightweight concurrent fault detection scheme for the AES S-boxes using normal basis," in *Proc. CHES*, Aug. 2008, pp. 113–129.
[17] D. Canright, "A very compact S-box for AES," in *Proc. CHES*, Aug. 2005, pp. 441–455.
[18] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in *Proc. ASIACRYPT*, Dec. 2001, pp. 239–254.
[19] J.Wolkerstorfer, E. Oswald, and M. Lamberger, "An ASIC implementation of the AES SBoxes," in *Proc. CT-RSA*, 2002, pp. 67–78.
[20] V. Rijmen, Dept. ESAT, Katholieke Universiteit Leuven, Leuven, Belgium, Efficient Implementation of the Rijndael S-Box, 2000.
[21] X. Zhang and K. K. Parhi, "High-speed VLSI architectures for the AES algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. VLSI-12, no. 9, pp. 957–967, Sep. 2004.