International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.3., Issue.4., 2015
(July-Aug)

RESEARCH ARTICLE

ISSN: 2321-7758

IMPLEMENTATION OF AES ALGORITHM

VINAY P. FIRAKE[1], Dr. A. M. PATIL[2]
[1]M. E. Student,[2]HoD
Department of Electronics and Telecommunication
J. T. Mahajan College of Engg. Faizpur

ABSTRACT
An implementation of high speed AES algorithm based on FPGA is presented in this paper in order to improve the safety of data in transmission. The design uses an iterative looping approach with block 128 bits and key size of 128 bits, 192 bits and 256 bits, lookup table implementation of S-box so as to reach the purpose of improving the system computing speed the pipe-lining and parallel processing methods were used. The simulation results show that the high-speed AES encryption algorithm implemented correctly. it has less hardware resources and high cost-effective. And this system has high security and reliability. This AES system can be widely used in the terminal equipments.

Key Words—AES, FPGA, encryption, decryption, Rijndael, block cipher

I. INTRODUCTION

The importance of cryptography applied to security in electronic data transactions has gained essential relevance during the last few years. Everyday many users generate and interchange large amount of information in various fields through Internet, telephone conversations, and e-commerce transactions. These and other examples of applications deserve a security point of view, not only in the transport of such information but also in its storage. This can be achieved by various techniques such as password, cryptography and biometrics. In this sense, cryptography techniques are especially useful.

In cryptography, the AES, also known as Rijndael, is a block cipher adopted as an encryption standard by the US government, which specifies an encryption algorithm capable of protecting sensitive data [1, 2]. The AES algorithm is a sym-metric block cipher that can encrypt and decrypt information. The AES algorithm uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [3, 4]. The hardware implementation of this algorithm can provide either high performance or low cost for specific applications. At main communication channels or heavily loaded servers it is not possible to lose processing speed, which drops the efficiency of the overall system while running cryptography algorithms in software. On the other hand, a low cost and small design can be used in smart card applications, which allows a wide range of equipment to operate securely.

The AES algorithm can be efficiently implemented by hardware and software. Software implementations cost the smallest resources, but they offer a limited physical security and the slowest process. Besides, growing requirements for high speed, high volume secure communications combined with physical security, hardware implementation of cryptogra-phy takes place. An FPGA implementation is an intermediate solution between general purpose processors (GPPs) and ap-plication specific integrated

circuits (ASICs). It has advantages over both GPPs and ASICs. It provides a faster hardware solution than a GPP. Also, it has a wider applicability than ASICs since its configuring software makes use of the broad range of functionality supported by the reconfigurable device [7].

II. LITERATURE REVIEW

With the advent of new technologies in the field of FPGAs, they are increasingly preferred over ASICs. Advantages of FPGAs include the ability to re-program in the field to fix bugs. FPGA design owes support the use of third party EDA tools to perform design ow tasks such as static timing analysis, formal verification and RTL and gate level simulation. Applications of FPGAs include digital signal processing, software defined radio, aerospace and defense systems, ASIC prototyping, medical imaging, computer vision, speech recognition, cryptography, bioinformatics, computer hardware emulation, radio astronomy, metal detection and a growing range of other areas. Considering the innumerable advantages of using a FPGA platform over others as can be noted, we have chosen to review AES implementations over FPGA exclusively.

A Rijndael cipher for encryption using a basic 64-bit iterative architecture was developed and presented in this paper. The proposed architecture implemented on FPGA achieves high speed, low area and cost effectiveness. Key scheduling unit has been added as a part of this work and the number of cycles required to encrypt text has been reduced and hardware optimization achieved. The results of this paper have been expressed in table 1 [9]. In this work a 32-bit data path FPGA implementation of AES has been proposed. A significant improvement of 3.4 was achieved over the existing designs in terms of throughput. The results of this paper have been discussed in table1 [10]. A hardware implementation of AES algorithm suitable for wireless military communication has been suggested in this work. An optimized code was proposed for the Rijndael algorithm with 128-bit keys. A significant improvement in throughput and reduction of slices was achieved. The results of this paper have been discussed in table 1 [11]. In this paper a novel high-speed non-pipelined architecture for implementing both encryption and decryption operations of the Rijndael algorithm on the same FPGA implementation has been proposed. The results of this paper have been presented in table 1 [12].

This paper proposes to combine both encryption and decryption on a single FPGA implementation with focus on low area and high throughput. The Rijndaels algorithm for 128-bit key is used on the fully pipelined AES encryptor /decryptor core proposed in this work. In this paper hardware implementation of optimized area for block cipher AES has been proposed. Time sharing of resources and iteration architecture has been used to reduce the area. Designs achieving area, latency and bandwidth optimizations have been reviewed in this paper. A FPGA implementation of AES algorithm has been presented in this work incorporating these optimization techniques for better throughput and lower latency. Two new designs of FPGA implementation of AES algorithm, one achieving a very high throughput and the other with a very small area have been presented in this paper. The high through-put design supports continued throughput during key changes for both encryption and decryption processes. A new combina-tional logic to improve the efficiency of inner round pipelining has been developed in this paper. Composite field arithmetic reduced the area. A fully sub-pipelined encrypter/decrypter with three sub stage pipe-lining per round has been used to achieve higher throughput. Efficiency and high throughput issues for FPGA implementation of AES-GCM have been addressed in this paper. Both the AES engine and the modular multiplication over G.F (2m) have been discussed. The Karatsubas algorithm has been used in the multiplication. In this paper an extension of a public key cryptosystem has been proposed to support a private-key cryptosystem. A new arithmetic unit has been developed in which the polynomial modular multiplication of ECC is extended to compute the polynomial arithmetic operations over binary extended field of AES. Higher hardware efficiency was achieved. Pipelining techniques have been used in the architectural optimization for the FPGA implementation of AES presented in this paper. Significant improvement in terms of speed has been achieved by processing multiple rounds simultaneously though cost in terms of area

increased. Exclusion of Shift Row stage and on the fly key generation have been incorporated to enhance throughput. A fully pipelined architecture which uses a 128-bit cipher key has been proposed in it. This paper uses a pipelined architecture only for the outerround in the FPGA implementation of the AES algorithm. Very high throughput and efficiency are the merits of the proposed work. The fully pipelined architecture with high throughput for data security applications has been proposed in this work. The hardware is implemented on FPGA.

A high speed, non-pipelined FPGA implementation of AES-CCMP has been proposed. Where in the CCMP consists of two modes, that is Counter mode for Data Privacy and CBC mode for authenticity [13]. A high throughput design combining both encryption and decryption on a single FPGA architecture has been proposed in this paper. Low area and low cost was achieved [14]. This work proposes a new flexible AES architecture that performs both encryption and decryption. A key generation module that generates both encryption and decryption keys is provided. Flexibility is achieved so that key generation depends on the data and hence hardware need not be changed every time[15]. FPGA implementation of AES algorithm has been presented in this paper. The encryption and decryption transformations have been performed using iterative design thus cutting hardware implementation costs [16].

III. AES ALGORITHM

AES encryption algorithm includes key expansion process and encryption process. The encryption process includes an initial Add Round Key of the initial round, and then carries out several rounds of Round transformation, and the last round also carries out Round transformation as shown in Figure 2. The encryption process of AES algorithm is as follows:

1) subBytes: The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once pre-calculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. More details of the method of calculating the S-box table refers to [4]. In this design, we use a look-up table as shown in figure 1. This is a more efficient method than directly implementing the multiplicative inverse operation followed by affine transformation.

X \ Y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig. 1: S-box

2) ShiftRows: it is relatively simple. State is the interme-diate cipher result that can be pictured as a rectangular array of bytes, having four rows. In the direct ShiftRows transformation, the first line of State remains the same, the second line, third line and fourth line respectively ring shift left 1byte, 2 bytes, and 3 bytes.

3) MixColumns: it is more complex. In the direct Mix-Columns transformation, every column operates inde-pendently and every byte is mapped to a new value. This transformation is based on the matrix multiplication of State.

4) AddRoundKey: the transformation in the cipher and inverse cipher in which a round key is added to the State using an XOR operation. Round keys are values derived from the cipher key using the Key Expansion routine.

5) KeyExpansion: it is the routine used to generate a series of Round Keys from the cipher key. KeyExpansion is carried out for the word, and to this two word processing functions are introduced which are word substitution (Subword) and word rotation (RotWord). Subword takes a four-byte input word and applies an S-box to each of the four bytes to produce an output word.

RotWord takes a four-byte word and performs a cyclic permutation.

AES algorithm decryption process is shown in Figure 4. Inversed its encryption process will be able to decrypt the cipher text.

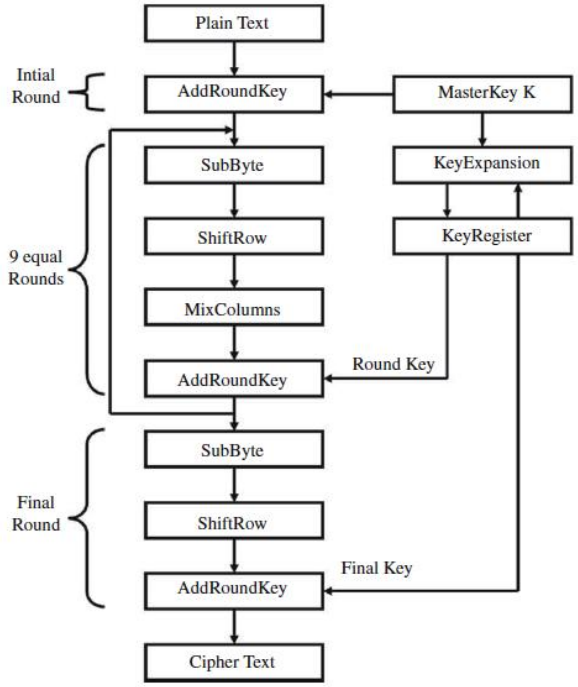


Fig. 2: AES encription process

1) AddRoundKey: AddRoundKey is its own inverse func-tion because the XOR function is its own inverse. The round keys have to be selected in reverse order. The description of the other transformations will be given as follows.
2) InvShiftRows Transformation: InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.
3) InvSubBytes transformation: The InvSubBytes transformation is done using a once-precalculated substitution table called InvS-box. That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values. InvS-box is presented in figure 3.
4) hardware resource. Different architecture should be selected according to the fields it is applied to. To make AES algorithm suitable to high-speed rate data application, we need to optimize the architecture.

Meanwhile by sharing resource and eliminating common sub expression we can reduce the hard-ware resource utilization. There are three basic architectures of AES to improve the throughput: Loop unrolled, pipelined, sub-pipelined that could be chosen.

		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	0	52	9	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	8	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	0	8c	bc	d3	0a	f7	e4	58	5	b8	b3	45	6
	7	d0	2c	1e	8f	ca	3f	0f	2	c1	af	bd	3	1	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1a
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	7	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	4	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Fig. 3: Inverse S-Box

IV. AES ALGORITHM IMPLEMENTED IN FPGA

Overall System Design: It is incompatible to implement the AES algorithm on hardware between the throughput and

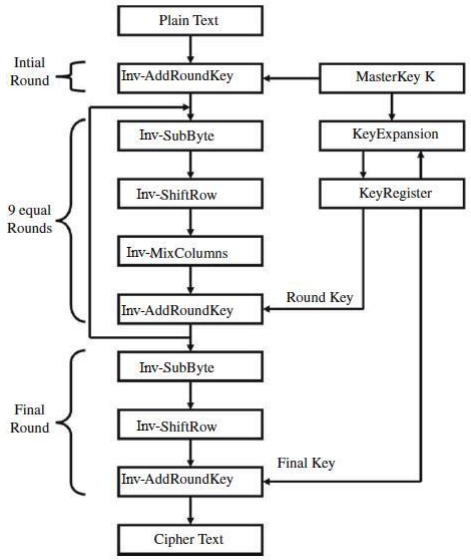


Fig. 4: AES Decryption Process

The top design of AES encryption system is shown in Figure 5. The top design includes three modules: Round module, Keyscheduel module and control module. Round module, which contains four

submodules: SubBytes, ShiftRows, Mix-Columns and AddRoundKey, performs the prime transformation process of AES. Keyscheduel module includes an S-box macro module to realize the nonlinear transformation. Control module in charge of the signals for other modules and the datas in Input/Output.

System Implementation: The design uses a synchronous clock

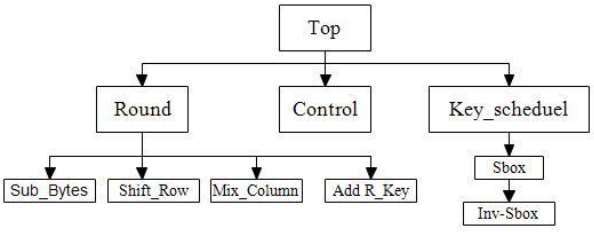


Fig. 5: Top Design of AES Encryption System

in order to make the circuit works with a unified clock and uses pipeline architecture to improve the working speed. Figure 5 shows the system implementation structure. Round module includes SubBytes, ShiftRows, MixColumns, AddRoundKey and an S-box matrix. SubBytes is a substituted operation to execute the operation and the affine transformation on finite field. ShiftRows is a cycle shift with bytes for unit. The most important process in MixColumns is the multiplication on finite field. AddRoundKey is a process that makes a 128 bits key to exclusive or the data in state one by one. S-box is a matrix that be defined to make a nonlinear replacement for SubBytes. The structure of round operation is shown in Figure 6. The pipeline scheme is utilized for implementations. In the pipeline scheme, the register arrays are assigned among the operational circuits of SubBytes, ShiftRows, MixColumns and AddRoundKey.

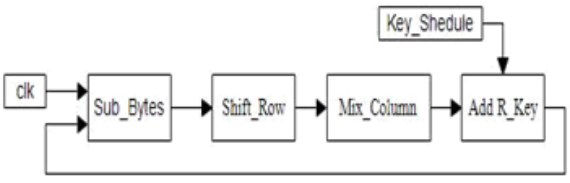


Fig. 6: Structure of round operation

TABLE I: Design Summery

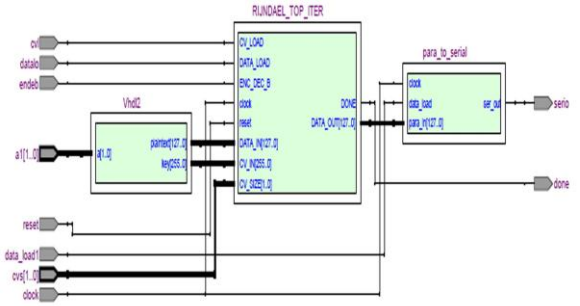
Flow status	Successful-sat Jul 04
	08:55:25 2015
	13.1.0 Build 162
Quartus II 64-bit Version	10/23/2012
	web Edition
Revision Name	Vinu
Top-Level Entity Name	Final
Family	Cyclone IV GX
Total Logic Elements	13,663/29440(46%)
Total combinational Functions	13,663/29440(46%)
Dedicated logic Resistors	554/29440(2%)
Total Resistor	554
Total Pins	12/81(15%)

TABLE II: Operating Voltage and Conditions

	Setting	Value
1	Nominal Core Voltage	1.20 V
2	Low Junction Temperature	0 C
3	High Junction Temperature	85C

V. EXPERIMENTAL RESULTS

The design has been coded by VHDL. All the results are synthesized and simulated basing on the Quatus 13.1 Web Edition and EP4CGX30BF14C6 device. The results of sim-ulating the encryption/decryption algorithm from the Quartus 13.1 web edition are shown in Figure 7. The practical results are in accordance to theoretical predictions and satisfy the encryption and decryption methodology.



. 7: RTL View of Implemented AES

To test the system a test bench is used. The test bench applies encryption/decryption input pulse to trigger the system. The output result of the encryption was found accurately after 13 clock cycles from the starting of encryption process. So the latency of encryption is only 13 clock cycles. Similarly, the latency of decryption is 13 clock cycles. overall

recourse utilization summery is shown in Table ii

Parameter	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	proposed
FPGA Device Package	XC2VP30-5-FF896	XC6SLX16	XC6SLX16	XC5VL50	XC2VP2	XV2V100	XV2V1000	VIRTEX-II PRO	EP4CGX30BF14C6
Number of Slices	1127	554	769	9028	40960	4325	58430	17314	16663
Maximum Clock Frequency (MHZ)	247.365 MHz	277.157MHz	100.8MHZ	220.7MHZ		75mhz	200mhz	28.5	1000MHZ
Data Path Bit	128	128	32	128	128				128
Power dissipation (mW)	500	252	2000	7000	3000				1200

Fig. 8: Comparison of proposed method with respect to slices, frequency, throughput etc.

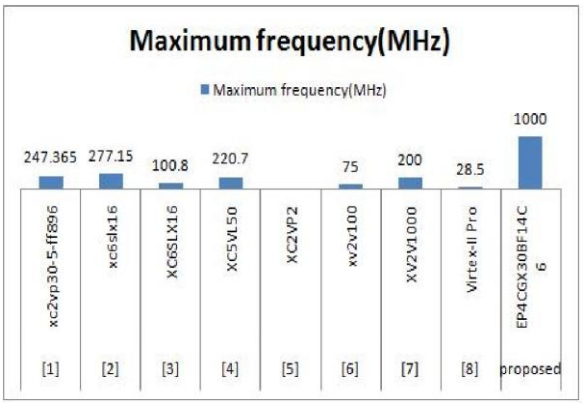


Fig. 9: Graphical Comparison of proposed method with Maximum Clock frequency.

VI. PERFORMANCE ANALYSIS

In this section, the results obtained by our design, and comparison between our results and other equivalent implementations is given and discussed. Our design for AES 128-bit encryption/decryption algorithm was synthesized, implemented by Altera tools. figure 8 summarizes the hardware resources required by main building blocks and gives detailed comparisons with the other designs [5], [6]. Considering the comparison in figure 8, our design is found to be more efficient in terms of latency, throughput and area. Therefore it allows us to process data in communication applications requiring a high security communication with low latency, high throughput and small area as per figure 9 to 11. Besides, the design is compared with another implementation using Xilinx chip [6] which uses the similar architecture with our design, but it requires a higher latency. Because Altera and Xilinx have thedifferent chip architectures, comparison between us and [6] cannot be done in the other criteria of memory, throughput.

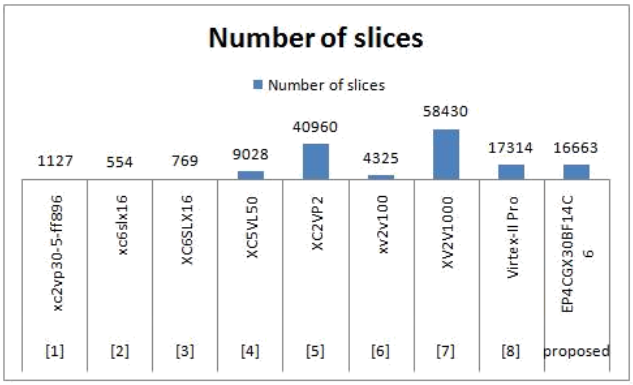


Fig. 10: Graphical Comparison of proposed method with respect to slices.

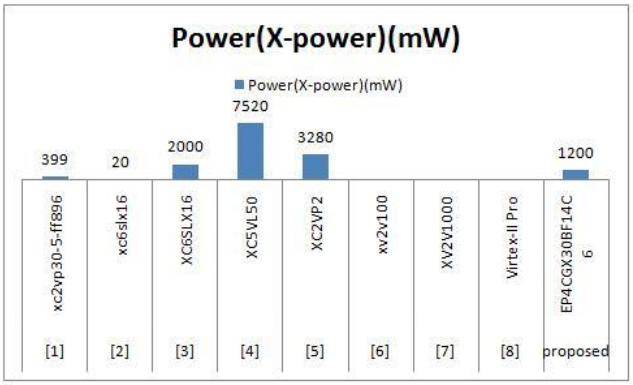


Fig. 11: Graphical Comparison of proposed method with respect to power.

CONCLUSION

The Advanced Encryption Standard algorithm is a symmetric block cipher that can process data blocks of 128 bits through the use of cipher keys with lengths of 128, 192, and 256 bits. An efficient FPGA implementation of 128 bit block and 128 bit key AES algorithm has been presented in this paper. The design is implemented on Altera using EP4CGX30BF14C6 FPGA which is based on high performance architecture. The proposed design is implemented based on the iterative approach for cryptographic algorithms. Our architecture is found to be better in terms of latency, throughput as well as area. The design is tested with the sample vectors provided by FIPS 197. The algorithm achieves a low latency and area efficient.

ACKNOWLEDGMENT

we are likes to gratefully acknowledge to all the staff member of electronics and Telecommunication department in J. T. Mahajan College of Engineering, Faizpur for allowing us to conduct the research using its all kinds of facilities.

REFERENCES

[1] Chi-Farn Chen, Hung-Yu Chang, Li-Yu Chang,"A Fuzzy-Based Method For Remote Sensing Image Contrast Enhancement", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008

[2] Shujun Fu, Qiuqi Ruan, Wenqia Wang," Remote Sensing Image Data Enhancement Based on Robust Inverse Diffusion Equation for Agriculture Applications", ICSP 2008 Proceedings.

[3] Hasanul Kabir, Abdullah Al-Wadud, and Oksam Chae," Brightness Preserving Image Contrast Enhancement Using Weighted Mixture of Global and Local Transformation Functions", The International Arab Journal of Information Technology, Vol. 7, No. 4, October 2010.

[4] Hasan Demirel, Cagri Ozcinar, and Gholamreza Anbarjafari," Satellite Image Contrast Enhancement Using Discrete Wavelet Transform and Singular Value Decomposition", IEEE Geoscience and Remote Sensing Letters, vol. 7, no. 2, pp. 333-337, April 2010.

[5] G.Praveena1, M.Venkatasrinu2," A Modified SVD-DCT Method for En-hancement of Low Contrast Satellite Images", International Journal of Computational Engineering Research, Vol. 2 Issue.5, pp. 1612-1619, September 2012.

[6] S. Srinivasan and N. Balram, "Adaptive Contrast Enhancement Using Local Region Stretching", Proc.of ASID06, , New Delhi, pp. 152-155,Oct. 2012.

[7] Ankit Aggarwal, R.S. Chauhan and Kamaljeet Kaur, "An Adaptive Im-age Enhancement Technique Preserving Brightness Level Using Gamma Correction", Advance in Electronic and Electric Engineering, Vol. 3, N0 9, pp. 1097-1108, 2013.

[8] Sayali Nimkar, Sucheta Shrivastava and Sanal Varghese, "Contrast Enhancement And Brightness Preservation Using Multi decomposition Histogram Equalization", International Journal (SIPIJ) Vol.4, No.3, Pp. 83-93, June 2013.

[9] G. Veena, V. Uma, Ch. Ganapathy Reddy, "Contrast Enhancement for Remote Sensing Images with Discrete Wavelet Transform", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-2, Issue-3, July 2013

[10] Ammu Anna Mathew, S. Kamatchi," Brightness and Resolution En-hancement of Satellite Images using SVD and DWT", International Journal of Engineering Trends and Technology (IJETT), Vol. 4, Issue. 4, pp. 712-718, April 2013.

[11] R. Kathirvel and Dr. J. Sundararajan," Brightness 3D Based Remote Sensing Image Contrast Enhancement Using SVM", Journal of Theoret-ical and Applied Information Technology, Vol. 61 No.3, pp. 545-552, March 2014.

[12] Adin Ramirez Rivera, Byungyong Ryu, and Oksam Chae, "Content Aware Dark Image Enhancement through Channel Divison",IEEE Trans-actions on Image Processing,volume 21,issue 9, 2012.

[13] Artur oza, David R. Bull, Paul R. Hill, Alin M .Achim, "Au-tomatic contrast enhancement of low-light images based on local statistics of wavelet coefficients", Digital Signal Processing (2013), www.elsevier.com/locate/dsp.

[14] Eunsung Lee, S.Kim, W.Kang, D.Seo and Jooki Paik, "Contrast En-hancement using Domonant Brightness Level and Adaptive Intensity Transformation for Remote Sensing Image",IEEE Geoscience and Re-mote sensing letters, Vol. 10, no.1, January 2013

[15] Deepak Kumar Pandey, Rajesh Nema "Efficient Contrast Enhancement using Kernel Padding and DWT with Image Fusion", International Journal of Computer Applications (0975 8887) Volume 77 No.15, September 2013.

[16] M. Mozaffari Kermani and A. Reyhani Masoleh, "A Lightweight Con-current Fault Detection Scheme for the AES S-Boxes Using Normal Basis, Proc. Intl Workshop Cryptographic Hardware and Embedded Systems (CHES 08), pp. 113-129, Aug. 2008.

[17] M. Karpovsky, K.J. Kulikowski, and A. Taubin, "Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard, Proc. Conf. Smart Card Research and Advanced Applications (CARDIS 04), vol. 153, pp. 177-192, Aug. 2004.

[18] [20] 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.

[19] C. Moratelli, F. Ghellar, E. Cota, and M. Lubaszewski, "A Fault-Tolerant DFA-Resistant AES Core, Proc. IEEE Intl Symp. Circuits and Systems (ISCAS 08), pp. 244-247, May 2008.

[20] A. Reyhani-Masoleh and M. Hasan, "Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over glosis field, IEEE Trans. Computers, vol. 53, no. 8, pp. 945-959, Aug. 2004.

[21] X. Zhang and K.K. Parhi, "High-Speed VLSI Architectures for the AES Algorithm, IEEE Trans. Very Large Scale Integration Systems, vol. 12, no. 9, pp. 957-967, Sept. 2004.

[22] R. Zimmermann and W. Fichtner, "Low-Power Logic Styles: CMOS versus Pass-Transistor Logic, IEEE J. Solid-State Circuits, vol. 32, no. 7, pp. 1079-1090, 1997.

[23] L. Breveglieri, I. Koren, and P. Maistri, "An Operation-Centered Ap-proach to Fault Detection in Symmetric Cryptography Ciphers, IEEE Trans. Computers, vol. 56, no. 5, pp. 534-540, May 2007.

[24] M. George and P. Alfke, "Linear Feedback Shift Registers in Virtex Devices, Xilinx Application Note 210.

[25] A. Satoh, S. Morioka, K. Takano, and S. Munetoh,"A Compact Rijndael Hardware Architecture with S-Box Optimization, Proc. Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 01), pp. 239-254, Dec. 2001.

[26] S. Morioka and A. Satoh, "An Optimized S-Box Circuit Architecture for Low Power AES Design, Proc. Intl Workshop Cryptographic Hardware and Embedded Systems (CHES 02), pp. 172-186, Aug. 2002.

[27] F.X. Standaert, G. Rouvroy, J.J. Quisquater, and J.D. Legat, "Efficient Implementation of Rijndael Encryption in Reconfigurable Hardware: Improvements and Design Tradeoffs, Proc. Intl Workshop Cryptographic Hardware and Embedded Systems (CHES 03), pp. 334-350, Sept. 2003.

[28] D. Canright,"A Very Compact S-Box for AES," Proc. Intl Workshop Cryptographic Hardware and Embedded Systems (CHES 05), pp. 441-455, Sept. 2005.

[29] C. Moratelli, E. Cota, and M. Lubaszewski, "A Cryptography Core Tolerant to DFA Fault Attacks, Proc. Ann. Symp. Integrated Circuits and Systems Design (SBCCI 06), pp. 190-195, Sept. 2006.

[30] M. Mozaffari-Kermani and A. Reyhani-Masoleh, "A Structure- Inde-pendent Approach for Fault Detection Hardware Implementations of the Advanced Encryption Standard, Proc. Intl Workshop Fault Diagnosis and Tolerance in Cryptography (FDTC 07), pp. 47-53, Sept. 2007.

[31] Krishnamurthy G N,"Making AES Stronger: AES with Key Dependent S-Box, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.9, September 2008

[32] Aida Janad , D. Anas Tarahi "AES immunity Enhancement against algebraic attacks by using dynamic S-Boxes.

[33] Jingmei Liu, Baodian Wei, Xiangguo Cheng, Xinmei Wang, " An AES S-box to Increase Complexity and Cryptographic Analysis

Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA05).

[34] S Shivkumar, Dr. G. Umamaheswari," Performance Comparison of Advanced Encryption Standard(AES) and AES key dependent S-box Simulation using MATLAB, 2011 IEEE.