Enhancing AES Algorithm with Delay Optimization to Reduce Power Analysis Attack

Lavanya.V*, Meghana K.S*, Sahana H.R*, Radhakrishna.M* and Dr. T Vijaya Kumar**
*GAT Bengaluru
lavanyajune12@gmail.com meghanaiyengar9619@gmail.com sahanaraj9729@gmail.com radhakaug22@gmail.com
**SJBIT Bengaluru
vijayakumar_th@rediffmail.com

Abstract: The world is dependent on communication which requires the role in exchanging information and they have to be secured so that it will not be misused, this raised the concept of Encryption. One of the methodology used for encryption is AES algorithm. In this paper a cryptographical AES algorithm is implemented which is used for network security. In this algorithm a 128 bit plain text is bitwise Xored with 128bit key followed by a sequence of operations to produce a 128bit block cipher. This method is easier and faster to implement because the same key is used in both encryption and decryption , which in turn helps in high security. One of the threats in cryptography is Power analysis attack which may be because of timing attack, power consumption monitoring ,cache attack etc. The paper mainly concentrates on reducing time delay and thus reducing timing attacks.

Keywords: AES-AdvancedEncryptionStandard,HDL-Hardware discryption language, VHDL-Very high speed Hardware discryption language ,NSIT-National Institute of Standards and Technology,DES-Data Encryption Standard.

Introduction

Information can be interchanged in different ways via internet in various fields such as banking sector, medical etc. Cryptography is a method for securing transmission of information over insecure channels. Cryptography plays an important role in embedded systems application where data requires a secured connection which is usually achieved by cryptography. Cryptography involves two categories they are symmetric key cryptography (sender and receiver shares the same key) and asymmetric key cryptography (sender and receiver shares different key).symmetric key cryptography is mainly used compared to asymmetric key cryptography due its use in medical report, bank services, military ,embedded system design etc via internet. DES algorithm is replaced by AES algorithm. The AES algorithm processes data blocks of 128 bits using a cipher key of length 128 or 192 or 256 bits. Each data block consists of a 4×4 array of bytes called the state, on which the basic operations of the AES algorithm are performed. AES algorithm changes the information (plain text) to an unreadable form (cipher text).

AES Algorithm

AES algorithm is a symmetric encryption algorithm where in same key is used for both encryption and decryption. The key length for AES algorithm can be 128 bits or 192 bits or 256 bits, which are named as AES-128, AES-192 and AES-256 respectively. The older standard, DES Algorithm processed a data of 56 bits only. To overcome this disadvantage, the new standard called AES algorithm was developed. In this algorithm a single 128 bit block is used for both encryption and decryption.128 bit block is represented as 4x4 square matrix of bytes called as state array, which is modified at each stage of encryption or decryption. Finally the last stage is copied to output matrix. Similarly, the key is also represented as square matrix of bytes. This key is expanded into an array of key scheduled words.

Table 1: AES Parameters

Key Size	128 bits	192 bits	256 bits
Plaintext Block Size(bits)	128	128	128
Number of Rounds	10	12	14
Round Key Size(bits)	128	128	128
Expanded Key Size(words)	44	52	60

This paper mainly concentrates on AES 128 bits which involves 10 rounds each having sub bytes, shift row ,mixed column and add round key operations where in last round excludes mixed column operation.

Sub bytes Transformation

The Sub byte transformation operates on each elements of state bytes which uses recalculated substitution table called S-box. A simple substitution of each byte using a 16 × 16 look up table which replaces a given input byte with a byte in the substitution table.. Each byte is replaced by byte index value according to the s-box, left most 4bits indicates row and remaining 4bits (right most) indicates column. For ex. byte 95 is replaced by byte corresponding to 9^{th} row and 5^{th} column. for substitution is illustrated in the fig1.

87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

EA	04	65	85
83	45	5D	96
5C	33	98	B0
F0	2D	AD	05

Fig 1: example for sub byte transformation

S-box table contains 256 values and their corresponding resulting values. Advantage of performing the S-box computation is that it avoids complexity of hardware implementation.S-box is expressed in Table2:

Table2: S-box

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	01	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	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	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	06	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	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	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

Shift Row Transformation

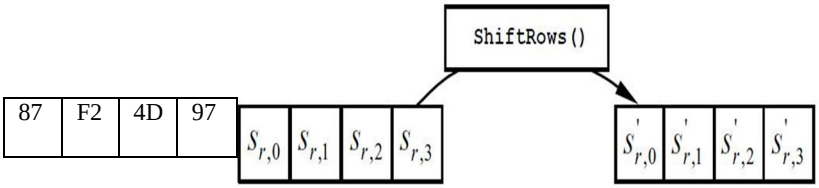
In this transformation,

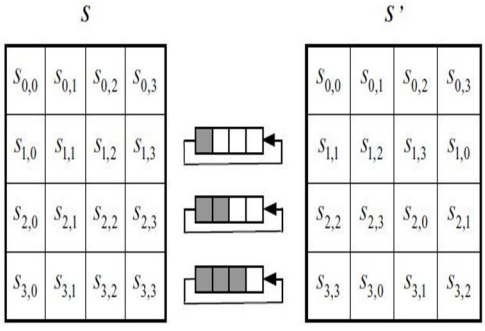
i) First row is retained as it is.

ii) Second row is circularly shifted by one byte to the left.

iii) Third row is circularly shifted by two bytes to the left.

iv) Last row is circularly shifted by three bytes to the left. Recall again that the input block is written column-wise.





87	F2	4D	97
EC	6E	4C	90
4A	C3	46	E7
8C	D8	95	A6

→

87	F2	4D	97
6E	4C	90	EC
46	E7	4A	C3
A6	8C	D8	95

Fig 2: Shift row Transformation and example

Mix column

In Mix column transformation the resultant values from shift row are compared with constant 4x4 matrix, i.e.

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

The input here is previous result of shift rows. If input is multiplied with 02, check the msb bit if one, left shift by one bit followed by conditional bitwise xor with “00011011” otherwise, left shit by one bit. If input is multiplied by ‘03’ ,then input will be split into 2x+x,i.e if we have {03}{6E} this is split into{6E}XOR{02}{6E}.For term after xor perform same operation as above mentioned for 02 and the result obtained from this is xored with {6E}.If input is multiplied by ‘01’, then input will be retained as it is.

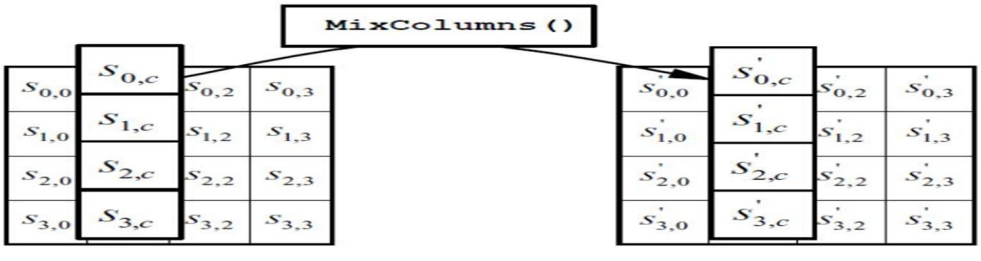


Fig 3: Mix column

6E	4C	90	EC
46	67	4A	C3
A6	8C	D8	95

→

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Fig 4: example for mixcolumn

Add round Key

In this transformation, a 128bits Round Key generated in key expansion is bitwise xored with the output of mix column which results in new state array.

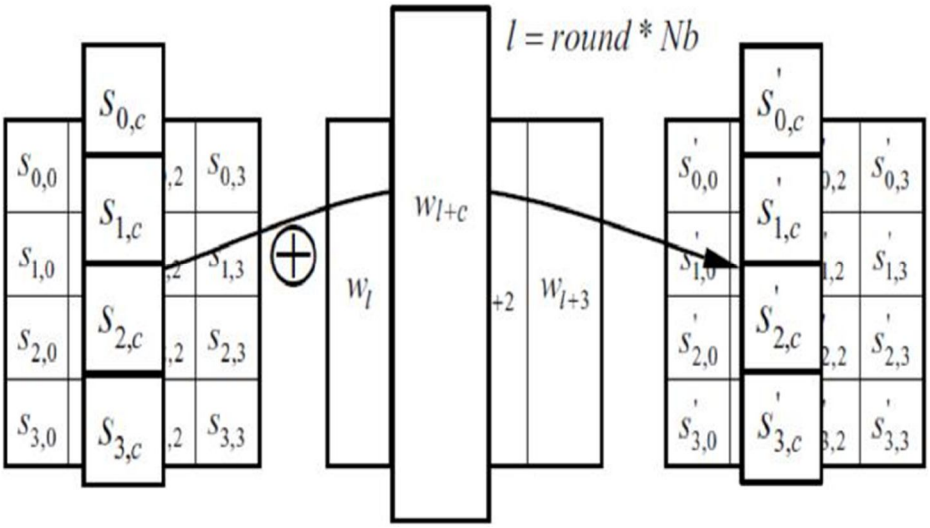


Fig 5: Add round key Transformation

47	40	A3	4c
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

Xor

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

=

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

Fig 6: Add round Transformation

In fig 6,the mix column output is xored with roundkey generated in key expansion process

Key Expansion

Key expansion process takes 16 bytes key as input and produces a linear array of 44 words. Sub Word takes a four-byte input word and does sub byte transformation to produce an output word. This is further shiftedcircularly to the left. The obtained output and the round constant word array (Rcon[i]) which contains constant values is xored with a byte of rcon.

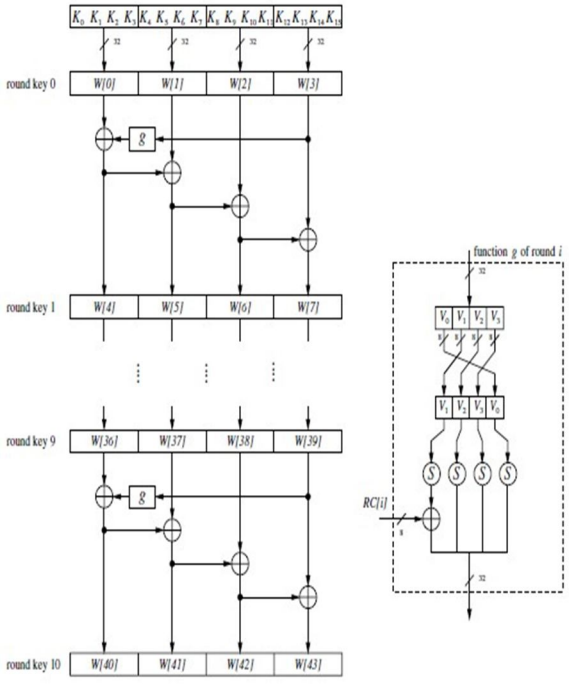


Fig 7: Key Expansion

Result

The AES-128bit algorithm with minimum delay for each round is depicted below.

Table 3: S-box

Modules	Delay
Sub byte	1.472ns
Shift row	0.34ns
Mix column	1.38ns
Key expansion	3.88ns

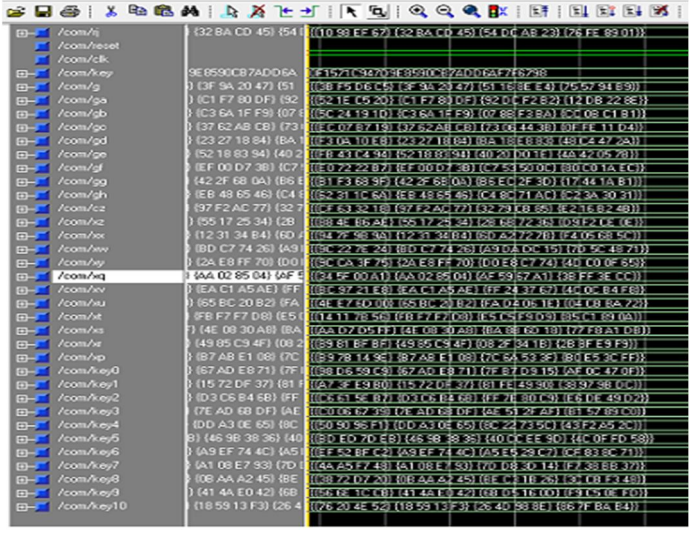


Fig 1.6 : AES Encryption process simulation

Conclusion

The AES algorithm on 128 bits message is successfully implemented. Xilinx ISE14.7i is used to synthesize and simulate VHDL implementation of AES Algorithm This implementation uses lower number of slices. The efficiency and performance was made to increase. The proposed architecture meets with satisfactory result with respect to speed and delay when compared to other designs. The throughput, low cost, flexibility of proposed architecture makes it perfectly practical for cryptographic applications. Ultimately, a synthesis and simulation of new Algorithm has been done and successfully implementation AES encryption on FPGA.

The implemented paper has reduced delay which in turn reduces power consumption and is also expected to reduce timing attacks thereby reducing power analysis attacks.

References

[1] Amravati, “Implementation of AES Algorithm Using FPGA & Its Performance Analysis”, International journal of science and research (IJSR), 2013, 2319-7064

[2] SyaliS.Kshirsagar,” Encryption and Decryption by AES algorithm using FPGA”, International journal for Emerging Trends in Engineering and Management Research (IJETEMR),June 2016, 2455-7773

[3] Abhinandan Aggarwal et al. “Implementation of AES algorithm” International journal of Engineering Research and Science, April 2016.

[4] Radhika Bajaj, Dr.U.M.Gokhale, “AES Algorithm for Encryption”, International Journal of Latest Research in Engineering and Technology, May 2016.

[5] MohiniMohurle,Prof.VishalV.Panchbhai, “Reliazation of Advanced Encryption Standard for Power and Area optimization”, International Journal on Recent and Innovation Trends in Computing and Communication, July 2016.

[6] SonaliA.Varhade, N.N.Kasat,”Implementation of AES Algorithm Using FPGA and its Performance Analysis”, International Journal of Science and Research, May 2015.

[7] Sandeep Kumar Rao, Dindayal Mahto, et al. “A Survey on Advanced Encryption Standard”, International Journal of Science and Research, January 2017.

[8] Alia Arshad, Kanwal Aslam, Dur-e-Shah war Kundi and Arshad Aziz, “FPGA Implementation of Advance Encryption Standard Using Xilinx System Generator”, Asian Journal of Applied Sciences (ISSN: 2321 – 0893),Volume 02, Issue 02, April 2014.

[9] Gangadari, Bhoopal Rao, and Shaik Rafi Ahamed. "FPGA implementation of compact S-Box for AES algorithm using composite field arithmetic." 2015 Annual IEEE India Conference (INDICON). IEEE, 2015, Pages: 1 - 5.

[10] Kalaiselvi, K and Anand Kumar. "Enhanced AES cryptosystem by using genetic algorithm and neural network in S-box." Current Trends in Advanced Computing (ICCTAC), IEEE International Conference on. IEEE, 2016, Pages: 1 - 6.

[11] Gangadari, Bhoopal Rao, et al. "Design of cryptographically secure AES S-Box using cellular automata." Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on. IEEE, 2015.

[12] Pammu, Ali Akbar, et al. "High Secured Low Power Multiplexer-LUT Based AES S-Box Implementation", 2016 International Conference on Information Systems Engineering (ICISE). IEEE, 2016, Pages: 3 - 7.

[13] Liao, Nan, et al. "A high-efficient fault attack on AES S-box.", Information Science and Technology (ICIST), 2016 Sixth International Conference on. IEEE, 2016, Pages: 210 - 215.

[14] Kester, Quist-Aphetsi, et al. "Feature Based Encryption Technique for Securing Forensic Biometric Image Data Using AES and Visual Cryptography." Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on. IEEE, 2014, Pages: 199 - 204