Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 5, Issue 7, October 2021

Design and Implementation of Advanced Encryption Standard

Divya B N
Assistant Professor, ECE, East West Institute of Technology, Bengaluru, India, divyasridhar705@gmail.com

Davana N
ECE, East West Institute of Technology, Bengaluru, India, davana.shankarappa@gmail.com

Dhanushree K
ECE, East West Institute of Technology, Bengaluru, India, dhanushree1111@gmail.com

Apoorva V
ECE, East West Institute of Technology, Bengaluru, India, appusavee@gmail.com

Anusha V
ECE, East West Institute of Technology, Bengaluru, India, anusha98yadav@gmail.com

Abstract: With the changing times, the physical touch is changing to electronic ones. Meeting new people, working together, banking transactions, shopping, anything under this sun is now been made possible to do online. With such huge amounts of data involved in such scenarios, the safety of the data is also very important. This paper shows an encryption method to safeguard the data.

Keywords: Network security; AES algorithm; visual cryptography; FPGA

I. INTRODUCTION

There is increasing need of secure data communication in computer network Technology. Since, AES Algorithm which is considered as an important process in securing data which was introduced in 2000 was replaced by DES [1], a data encryption standard endorsed by the U.S. National Institute of Standards and Technology (NIST). The main reason that AES Algorithm is immune to brute-force attack [2].

AES can be implemented in software and hardware but, hardware implementation is used in real time application. The three major design targets with respect to hardware realization are: optimization of area or cost reduce in computational speed and low power consumption [3].

Hardware implementation of AES is very reliable, conveniently fast and suitable for high speed applications. Hardware encrypted drivers can be easily reset which reduces down time in erasing data which gives better performance [4].

In this paper we discuss the combination of two popular methodologies: The AES algorithm and visual cryptography to ensure high security of data. The paper also shows the existing techniques in this field.

II. LITERATURE SURVEY

J. Balamurugan et al [6] proposed a concept of high-speed low-cost implementation of AES on FPGA which states that implementation of 128 bit-key AES cipher. The design's target is to optimize speed and cost, but the major drawback was that the area was not completely optimized as it was estimated.

M.zeghid et al [7] proposed Modified AES algorithm for encryption. According to author, an extra feature of key stream generator is added to AES. This is shown to improve performance of encryption of an image the result was characterized by increase in entropy.

Omkar S.Dhede et al [8] implemented a method of hardware implementation of AES using minimal resource on FPGA.As hardware is more secure than the software, proposed system provides high security to the hacking . The only disadvantage proposed in the system was no flexibility for selection of key length.

Shize Guo et al [9] presented a system to prevent Analytical Side-Channel attacks on AES. This system with AES was implemented on 8-bit microcontroller, but the major drawback was there was Side-Channel attack on the AES because it couldn't recover the entire secrete key at once.

III. EXISTING ALGORITHMS

In this section we will be dealing with the existing method which includes implementation of AES.

Advanced Encryption Standard was discovered by John Daenun and Vinunt Rijmen and is popularly known as Rijndael algorithm [10]. It is a symmetric block encryption algorithm. The size of the encrypted block can be 128-bit, 192-bit or 256-bit [11]. The steps involved in Standard AES algorithm are shown in Fig. 1.

This paper implements the AES-128 encryption algorithm. The AES-128 encryption algorithm encrypts

128 bits, and this 128-bit data (16 bytes) are arranged in a 4 x 4 column order matrix [12].

A. *Add Round Key*

The 16 bytes of the matrix is now considered as 128 bits of plain text and are XORed to 128 bits of the round key and the obtained result will be imposed on the next step called Sub Bytes [13]. The key matrix used for encryption and decryption must be the same.

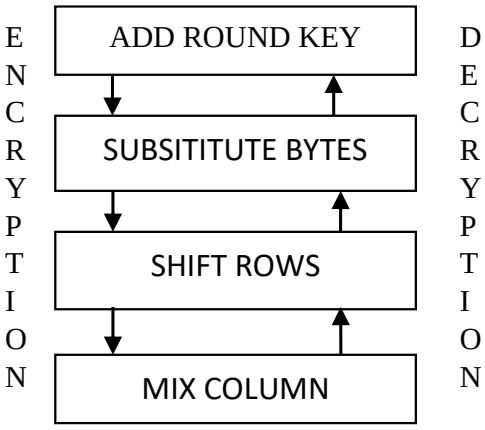

Fig 1. Standard implementation of AES algorithm

$$\begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{pmatrix} \oplus \begin{pmatrix} k11 & k12 & k13 & k14 \\ k21 & k22 & k23 & k24 \\ k31 & k32 & k33 & k34 \\ k41 & k42 & k43 & k44 \end{pmatrix} = \begin{pmatrix} r11 & r12 & r13 & r14 \\ r21 & r22 & r23 & r24 \\ r31 & r32 & r33 & r34 \\ r41 & r42 & r43 & r44 \end{pmatrix}$$

Fig 2. Add round key – a is the input matrix, k is the key matrix and r is the add round key output

B. *Sub Bytes and Inverse Sub Bytes*

During encryption, the 16-byte input matrix is substituted by look up table - a fixed table (S-box) given in design. The result is a matrix four rows and four columns [14]. For decryption inverse S-box is obtained by applying inverse transformation. The s-box and inverse s-box tables are shown in Fig. 3 and Fig. 4.

	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

Fig 3. S-box table

C. *Shift Rows and Inverse Shift Row*

For encryption, values in each row are left shifted N-1 times in a cyclic manner, where N is row number of the matrix. The shift is carried out as follows [15].

1. First row remains unchanged.
2. Second row is shifted one position to the left.
3. Third row is shifted two positions to the left.
4. Fourth row is shifted three positions to the left.

The result is a new matrix consisting of the same bytes but shifted with respect to each other, as shown in Fig. 5. For decryption, the shifting is done in cyclic right manner.

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

Fig 4. Inverse S-box table

$$\begin{pmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{pmatrix} \rightarrow \begin{pmatrix} a11 & a12 & a13 & a14 \\ a22 & a23 & a24 & a21 \\ a33 & a34 & a31 & a32 \\ a44 & a41 & a42 & a43 \end{pmatrix}$$

Fig 5. Shift-Rows of encryption of AES

D. *Mix Column and Inverse Mix Column*

Each column of four bytes is transformed using a special mathematical function. Since Mix Column approach, which is the most algorithmically intense, due to the involvement of multiplication in the Galois field [GF(2^8)].

In this transform each column of the state matrix is multiplied by a constant fixed matrix. The procedure remains the same for both encryption and decryption; only the keys change, as shown in Fig. 6 and Fig. 7.

To reduce the computation time, the Look Up table approach is implemented which is the methodology of pre-computing all possible multiplication with Galois flied and by storing the process in the form of lookup table. Let us consider the flowchart working with the look up table approach [16].

$$\begin{bmatrix} a11 & a12 & a13 & a14 \\ a22 & a23 & a24 & a21 \\ a33 & a34 & a31 & a32 \\ a44 & a41 & a42 & a43 \end{bmatrix} \otimes \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{bmatrix} s11 & s12 & s13 & s14 \\ s21 & s22 & s23 & s24 \\ s31 & s32 & s33 & s34 \\ s41 & s42 & s43 & s44 \end{bmatrix}$$

Fig 6. Shifted matrix is multiplied with the mix column key to get encrypted matrix.

$$\begin{bmatrix} s11 & s12 & s13 & s14 \\ s22 & s23 & s24 & s21 \\ s33 & s34 & s31 & s32 \\ s44 & s41 & s42 & s43 \end{bmatrix} \otimes \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} = \begin{bmatrix} a11 & a12 & a13 & a14 \\ a21 & a22 & a23 & a24 \\ a31 & a32 & a33 & a34 \\ a41 & a42 & a43 & a44 \end{bmatrix}$$

Fig 7. Encrypted matrix is multiplied with the mix column key to get decrypted matrix (shifted rows)

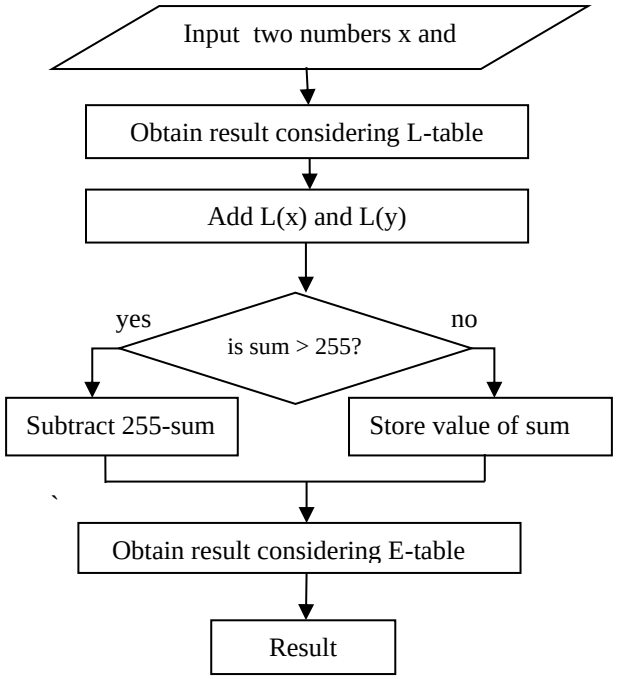

Fig 8. Flow chart of the LUT approach

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		00	19	01	32	02	1A	C6	4B	C7	1B	68	33	EE	DF	03
1	64	04	E0	0E	34	8D	81	EF	4C	71	08	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	09	78
3	65	2F	8A	05	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	06	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

Fig 9. L table used for LUT approach

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	2E	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	E4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

Fig 10. E table for LUT approach

IV. PROPOSED METHODOLOGY

The steps involved is shown in Fig. 11.

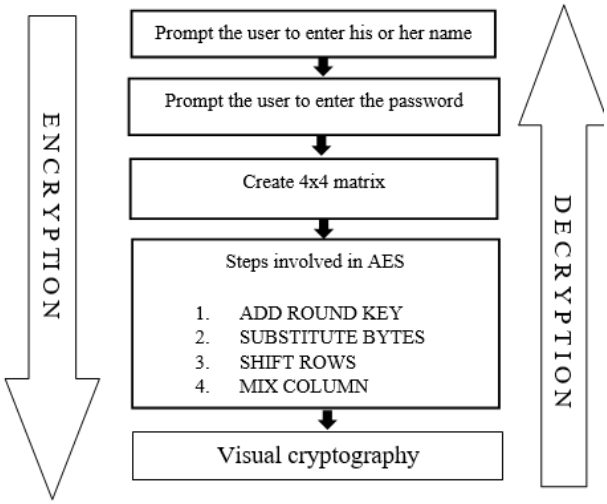

Fig 11. Flowchart of the methodology

A. *Inputs*

The user is prompted to enter the username and password. All the characters of the username are converted into a 4x4 matrix and the password is embedded within them.

B. *AES Algorithm*

AES breaks data into matrix of bytes of predetermined size and encrypt every state independently of each other. Putting together bytes into a state has no cryptographic significance, yet it is an important process without whom other operations cannot be conducted. A matrix with 4 rows and 4 columns is formed where each entry is a byte or 8 bits so that there are essentially 16 bytes in total. We restrict to description of a typical round of AES encryption. Each round comprises of four sub-processes, as described in the previous section.

C. *Visual Cryptography*

To further secure the data, an extra step of Visual cryptography is added to the algorithm.

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image. One of the best-known techniques has been credited to Moni Naor and Adi Shamir [21], who developed it in 1994.

In this method the data which is printed has to be encrypted and divided into two shares. Each pixel of original data is further divided into two sub-pixels in each share. The Distribution is done as follows:

- If the original data image has white colour pixel, then the sub-pixel of share 1 has either of black or white combination or vice versa, the same has to be chosen for share 2.
- If the original data image has black colour pixel, then the sub-pixel of share 1 has either of black and white combination or vice versa, the complement has to be chosen for share 2.

V. RESULTS

The encryption is implemented on MATLAB, while the decryption on FPGA, using Verilog. Mix column is implemented using two techniques – LUT and using adds and shifts. In adds and shifts method, the LUT is broken down into combination of multipliers (as per the conventional method) and then utilized adders and shifters in order to obtain the results. The efficiency of both have been mentioned in Table. 1.

	Area	Power	Speed
LUT	33%	24.28mW	12.726ns
Adds and Shifts	13%	11.06mW	4.356ns

Table 1. Efficiency comparison of Mix Column approaches

The screenshots of analysis are shown in Fig. 12 and 13.

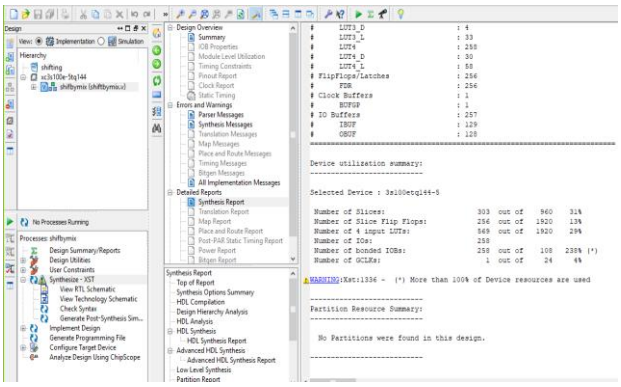

Fig 12. Synthesis report of Adds and Shifts

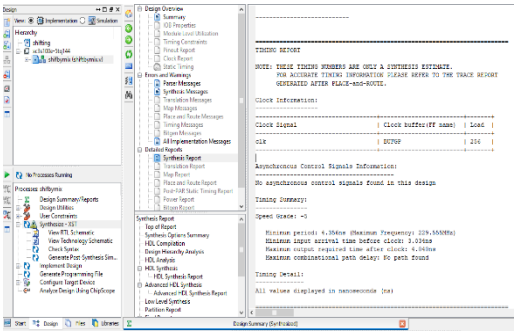

Fig 13. Synthesis report of LUT

VI. CONCLUSION

In this paper, encryption and decryption algorithm implemented by using AES 128-bit. Here, the objective is to reduce the power consumption, decrease the area and increase the speed of FPGA and reduce the computing time in Mix Column. As hardware is more secure than the software , proposed system provides high security to the hacking. As a future scope, research could be carried out to further enhance the security of the methodology elaborated in this paper. Combination of the visual cryptography method along with other forms of encryption such as DES and triple DES could be explored. Area and speed efficient VLSI designs for other stages of the AES algorithm could be studied and implemented. Overall, a lot of promising research could be continued in this field of network security.

REFERENCES

[1] Mansoor Ebrahim, Shujaat Khan, Umer Bin Khalid, “Symmetric Algorithm Survey: A Comparative Analysis”, International Journal of Computer Applications, Vol. 61 No. 20, pp. 12-19.

[2] Shize Guo, Xinjie Zhao, Fan Zhang, Tao Wang, Zhijie Jerry Shi, Francois-Xavier Standaert, and Chujiao Ma, “Exploiting the Incomplete Diffusion Feature: A Specialized Analytical Side-Channel Attack against the AES and Its Application to Microcontroller Imple- mentations”, IEEE Transactions On

Information Forensics And Security, Vol. 9, No. 6, pp. 999-1014, June-2014.

[3] Ai-Wen Luo, Qing-Ming Yi, Min Shi, "Design and Implementation of Area-optimized AES-Based on FPGA", 978-1-61284-109-0/11/2011 IEEE.

[4] Jignaesh R.Patel, Rajesh S.Bansode, Vikas Kaul,"Hybrid security algorithm for data transmission using AES-DES",.IJAIS-2012. [3]Y.Ou , C.Sur , K. H Rhee"Region based selective Encryption for Medical Imaging",1st Annual International Workshop-2007

[5] S. H Kamali, R.Shakerian, M.Hedayati,"A new modified version of Advanced Encryption Standard based algorithm for image encryption", International Conference on Electronics and information Engineering, ICEIE-2010.

[6] M.Zeghid , M.Machhout,L.Khriji,A.Baganne and R.Tourki,"A modified AES based algorithm for image encryption",International journal of computer electrical, Automation, Control and information engineering2007.

[7] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar, Mrs. M. V. Vyawahare" FPGA Implementation of AES Algorithm", IEEE-2011, Volume : 3,pp 401-405.

[8] Daemen, J.Rijmen, V.: AES Proposal: Rijndael, The Rijndael Block Cipher, AES Proposal, pp.1–45, 1999 http://csrc.nist.gov/CryptoToolkit/aes/).

[9] J. Daemen and V. Rijmen, "The Rijndael Block Cipher", AES Proposal document version 2, 1999.

[10] LAN MAN Standard committee of the IEEE Computer Society", ANSI IEEE Std 802.11, 1999 Edition," 1999.

[11] National Institute of Standards and Technology (NIST)", Advanced Encryption Standard (AES)," Nov. 2001.

[12] A.J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA Based Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists", Proc. Third Advanced Encryption Standard (AES) Candidate Conf., Apr. 2000.

[13] FIPS, Announcing the advanced encryption standard (AES).Federal Information Processing Standards Publication 197(FIPS-197), 2001.

[14] Chitu, C. Chien, D. Chien, C. Verbauwhede, I. Chang, "Hardware Implementation in FPGA of the Rijndael algorithm". Dept. of Electr. Eng., Univ. of California, Los Angeles, CA, USA

[15] N. Sklavos and 0. Koufopavlou, "Architectures and VLSI Implementations of the AES-proposal Rijndael", IEEE Trans.on Computers, vol. 51, Issue 12, pp. 1454-1459, 2002.

[16] S. R. Rupanagudi et al., "A novel and highly secure encryption methodology using a combination of AES and visual cryptography," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 1682-1688, doi: 10.1109/ICACCI.2016.7732289.