MULTILAYERED CRYPTOGRAPHIC ALGORITHM FOR THE SECURITY ENHANCEMENT OF INFORMATION

Dr.G.Kameswari¹,Y.Amrutha²,T.Pranathi³,T.Jyothsna⁴,R.Sudheer⁵ Associate Proffessor¹,Students^{2,3,4,5} Electronics and Communication Engineering N.B.K.R Institute of Science and Technology, Andhra Pradesh, India

Abstract: The increasing need for data communication protecting in computer networks has led to development of several cryptography algorithms. The Advanced Encryption Standard (AES) is a computer security standard issued by the National Institute of Standards and Technology (NIST) intended for protecting electronic data. The AES cryptography algorithm can be used to encrypt/decrypt blocks of 128 bits and is capable of using cipher keys of 128 bits wide (AES128).

This paper represents the security enhancement of using the multilayer linear feedback shift register (LFSR) cryptographic technique in order to overcome data hacking. A hardware implementation of the AES128 encryption algorithm was proposed. A unique feature of the proposed pipelined design is that round keys, which are consumed during different iterations of encryption, are generated in parallel with the encryption process.

Keywords: Cryptography, Encryption, LFSR, AES, Pipelined

I INTRODUCTION

In today's world, hacking has become a great problem which causes the loss of information during the process of transmitting the data. This hacking creates a sense of insecurity among the users who are transmitting the data. Over the years in order to protect the data security we are inhibiting various types of algorithms, out of which the most familiar is Cryptography. Through centuries Cryptography has an interesting history which helps us to understand the encryption and decryption process in detail. Cryptography is a study of science deals with secret writing. that Cryptography provides data integrity usually means avoiding unintended users to get the information. For symmetric cryptography only one key is used in both the process of encryption and decryption whereas in asymmetric cryptography different type of keys are used. AES is short for Advanced Encryption Standard and is a United States encryption standard defined in Federal Information Processing Standard (FIPS). AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. AES is symmetric since the same key is used for encryption and the reverse transformation, decryption. The only secret necessary to keep for security is the key.

II LITERATURE SURVEY

[1] Ritu Tripathi, Sanjat Agarwal proposed that symmetric Key Cryptography, Asymmetric Key Cryptography uses two separate keys to prevent any unethical access to the data. The public key remains public and the private key is not shared. Moreover, the use of Digital Signatures in case of Asymmetric Key Cryptography provides high data confidentiality and non-repudiation.

[2] Mitali, Vijay Kumar, Arvind Sharma proposed the classical cryptography approach as well the modern ones. Cryptography along with cryptanalysis are very useful tools that make a huge difference and as such should be used with the utmost care.

[3] Babitha P.K, Thushara T, Dechakka M. P, proposed that there is a simulation problem for long bit LFSR when it is targeting to FPGA for rapid prototyping development. Definitely n bit LFSR with maximum length feedback polynomial will generate large sequence which is more secure than other but because of simulation difficulties modification in long bit LFSR needed.

[4] Dr.A.Albert Raj, Dr.Beno, R.Jaisakthi proposed that, The first layer and second layer of cryptography are completed with the help of the LFSR cryptographic technique. The ciphertext for the LFSR cryptography is generated.The work focussed on the design of effective singlelayered cryptography as well as multilayer cryptography using the LFSR cryptographic technique.

III ADVANCED ENCRYPTION

STANDARD

AES is a popular and efficient algorithm that is used in cryptography. It is a

specification for the encryption of electronic data. AES is a symmetric encryption algorithm processing data in block of 128 bits. A bit can take the values zero and one, in effect a binary digit with two possible values as opposed to decimal digits, which can take one of 10 values. Under the influence of a key, a 128-bit block is encrypted by transforming it in a unique way into a new block of the same size. . AES may configured to use different key-lengths, the standard defines 3 lengths and the resulting algorithms are named AES-128, AES-192 and AES-256 respectively to indicate the length in bits of the key. Each additional bit in the key effectively doubles the strength of the algorithm, when defined as the time necessary for an attacker to stage a brute force attack, i.e. an exhaustive search of all possible key combinations in order to find the right one.

AES is the name of the standard, and the algorithm described is a variant of '**Rijndael**'. However, in practice the algorithm is also referred to as "AES".

A. Existing AES Algorithm:

Rijndael algorithm consists of encryption, decryption and key schedule algorithm. The main operations of the encryption algorithm among the three parts of Rijndael algorithm include: bytes substitution (Sub Bytes), the row shift (Shift Rows), column mixing (Mix Columns), and the round key adding (Add Round Key).

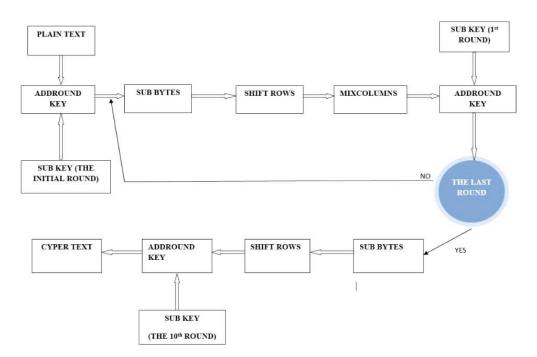


Figure 1: Basic AES Algorithm

Encryption algorithm processes Nr+1 rounds of transformation of the plaintext for the cipher text. The value of Nr in AES algorithm whose packet length is 128 bits should be 10, 12, or 14 respectively, corresponding to the key length of 128,192,256 bits. In this paper, only the (AES-128) encryption scheme with 128bit keys is considered.

- The existing method has the following disadvantages:
 - When some of the bits in both plain text and cipher text are known then it would be very easy to hack the data.
- Security is very less.

ii.

B. Proposed AES Algorithm:

The AES encryption algorithm can be divided into two parts, the key schedule and round transformation. Key schedule consists of two modules: key expansion and round key selection. Key expansion means mapping Nk bits initial key to the so-called expanded key, while the round key selection selects Nb bits of round key from the expanded key module. Round Transformation involves four modules by Byte Substitution, Byte Rotation, Mix Column and Add Round Key. Take the independent and reversible bytes substitution operation of S-box as example. Firstly, the state matrix is divided into four columns. And then byte replacement is achieved by the operation of look-up table.

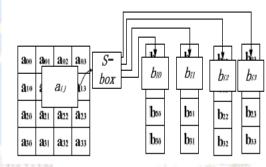


Figure2: Bytes segmentation and

replacement processing

Therefore, the original 128-bit input of plaintext and key will be replaced with four consecutive 32-bit input sequences respectively. In order to decrease the output ports, four continuous 32-bit cipher text sequences have taken place of the original 128-bit output by adding a clock controller. The improved structure is also divided into these two major processes. The initial key will be sent to the two modules: Key expansion and Key selection, while the plaintext is to be sent

to the round transformation after the round key is selected. But the operand of data transmission is turned into a 32-bit unit. The function of Mix Column can be achieved. It is a block diagram for the introduction of pipelining technology used in the round transformation.

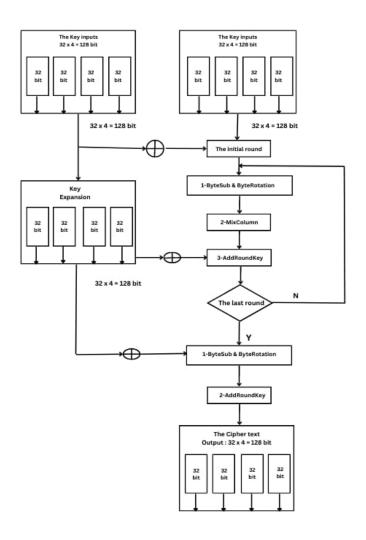


Figure3: The new improved structure of AES algorithm

The function of Mix column can be achieved. It is a block diagram for the introduction of pipelining technology used in the round transformation. In the process of pipelining, the 128-bit data is divided into four consecutive 32-bit packets that take round transformation independently. The operation of the above four groups of data can be realized in pipelining technology. The operation of the above four groups of data can be realized in pipelining technology. In brief, it can be

described as follow: store the unprocessed data in the 128-bit register, and control the clock for re-starting the 128-bit register to read the new data when the four group's operations have been overcome. Thus the 128-bit round-operating unit has been transformed into four 32-bit roundoperating elements. The internal pipelining processing should be implemented during the whole nine intermediate Round Transformations of the four packets before achieving the 128-bit cipher text.

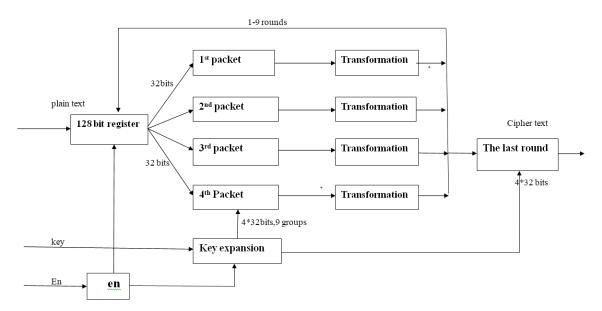


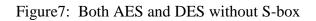
Figure4: The round processing with pipeline technology

III. STIMULATION:

	all and a second							- A1	Contraction of the second	
				2,275,014 ps						
194	Name	Value	2,274,900 ps 2	.275,000 ps	2,275,100 ps	2,275,200 ps	2,275,300 ps 2,;	275,400 ps	2,275,500 ps 2,	,275
50	🄓 clk	1								
	🕨 📑 datain[127:0]	10101011111	101010111111111111111	111111111111111	111111111111111111111111111111111111111	11111111111111111111	111111111111000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000)0000
02	🕨 📑 key[127:0]	000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	000000000000000000000000000000000000000	0000000111111111	111111111111111111111111111111111111111	111111111111111	111111111111111111111111111111111111111	1111
141.0	🕨 📑 dataout[127:0]	10010010000	1001001000011101010	00011001110010	0010001110011110	00001101101111001	0000110001110001011	01110000111000	010011000000111000	0101
. Sichard	🕨 式 r1_out[127:0]	11101001011	11101001011110100011	0100100111001	1111111101010001	10101000010101110	01111010100011111	00101100100000	001000000000000000000000000000000000000	0001
	🕨 🏹 r2_out[127:0]	10010010100	1001001010000001010	0110011100111	10100011101110001	1000001100100100	0101001010001110101	00010100000111(100011001001010010	01000
8	🕨 💐 r3_out[127:0]	00101000111	0010100011101011110	01000111100111	0011110001010001	11111111111001001	0011010011010011010	00011000000011:	011001111001101111	100
March 1	🕨 式 r4_out[127:0]	01010011011	01010011011101111	01101000111111	01110001011101011	10110111000000010	0001101101001100011	10101111101110	011100000001100100	01110
Linesper	▶ 🔩 r5_out[127:0]	10010001011	1001000101101011111	0101010001111	10100110010111101	11100101100010100	0011000010100100010	10101001111101	011101001111011000	0110
- COLORAD	▶ 式 r6_out[127:0]	00010100010	0001010001000101111	0000000100011	00001110111110011	1110101010101010110	10110110011111011	01100100000011(100110011000100001	1111
	▶ 式 r7_out[127:0]	00011011101	0001101110101010001	01011101000100	11001011010001000	10001011011111101	0001110101010101011	00000100111001	001110001011001101	0110
	▶ 式 r8_out[127:0]	11010100010	1101010001011100001	0100111000000	11000011111000111	01001011011110111	0001100100010000001	011111100001010	00001101000111011	10100
	▶ 式 r9_out[127:0]	10011001001	1001100100101101100	010110011000110	010111101011110100	00001000100001000	101001000111101010	10110000101000	001010101001011110	0111
1	▶ 式 keyout1[127:0]	00010111000	0001011100010110000	0110000101100	00101110001011000	01011001101001111	0100011101001111010	01100101100001(111000101100001011	001
	▶ 式 keyout2[127:0]	00010100000	0001010000010110000	0110000101100	0101000001011000	01011001101001111	0101111101001111010	01100101100001	100000101100001011	001
1	▶ 式 keyout3[127:0]	00010010000	0001001000010110000	0110000101100	0100100001011000	01011001101001111	0110111101001111010	01100101100001(010000101100001011	1001
	🕨 式 keyout4[127:0]	00011110000	0001111000010110000	0110000101100	0111100001011000	01011001101001111	0000111101001111010	01100101100001	110000101100001011	001
	keyout5[127:0]	00000110000	0000011000010110000	0110000101100	00001100001011000	01011001101001111	100111101001111010	01100101100000	110000101100001011	1001
	1907 - YOL Y								100 Co. 10	

Figure5: Stimulation for AES

							2,000,000 ps			
Name	Value	1,999,995 ps	1,999,996 ps	1,999,997 ps	1,999,998 ps	1,999,999 ps	2,000,000 ps	2,000,001 ps	2,000,002 ps	2,000,
▶ <table-of-contents> a[7:0]</table-of-contents>	11001100			11001100			2			
. 🕨 😽 c[7:0]	01001011			01001011						
		X1: 2,000,000 ps								
830			Fig	ure6: AE	C with S	hov				e C
			Ing	uico. Al	S with S	UUX				C.
and i										1
										2 8
<u></u>							1. 11			
									60.0	00 ns
Name	Value	0 ns	10 11 10	ns	20 ns	30 ns	2140	ns	150 ps	
Name							1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		50 ns	
🖌 🕨 🙋 aes_out[12	27:0] 52525	252525 XXXXX						25271 \(5252525 163	25	
 des_out[12] dut[127:0] dut[127:0] 	27:0] 52525 173 173		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		\ 5252525252 X		2525252a3 🛛 52	2525252525252525		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0] 52525 173 173		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 aes_out[12] out[127:0] in[127:0] key[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	
 des_out[12] des_out[12] des_out[127:0] des_out[127:0] des_out[127:0] 	27:0) 52525 173 173 145		(52525252525252 X 1D		2525252a3 🛛 52	2525252525252525 163		25271 \5252525 163	25252525252525252525252525252525252525	



767

Comparison factors	Total Availability	AES with S- Box	DES with S- Box	AES & DES without S-Box
AREA	84,352	27,789	32,281	1,872
TOTAL DELAY	-	3.793ns	74.51ns	3.874ns
TOTAL TIME	Jon Jon	177.36sec	247.75sec	11.97sec

Figure8: Comparison table of different factors

IV.CONCLUSION:

A FPGA implementation of areaoptimized AES algorithm which meets the actual application is proposed in this paper. After being coded with VHDL, the waveform simulation of the new algorithm was taken in the Model Sim SE PLUS 6.0 and Quartus7.2 platform. Ultimately, a synthesis simulation of the new algorithm has been done. The result shows that the design with the pipelining technology and special data transmission mode can optimize the chip area effectively. This design reduces power consumption to

V. REFERENCES:

[1]J.Yang, J.Ding, N.Li and Y.X.Guo, "FPGA-based design and implementation of reduced AES algorithm" IEEE Inter.Conf. Chal Envir Sci Com Engine(CESCE).,Vol.02, Issue.5-6, pp.67-70, Jun 2010.

[2] A.M.Deshpande , M.S.Deshpande and D.N.Kayatanavar, "FPGA Implementation of AES Encryption and Decryption" IEEE Inter.Conf.Cont,Auto,Com,andEner.,vol.0 1, issue04, pp.1-6,Jun.2009.

[3]Hiremath.S. and Suma.M.S., "Advanced Encryption Standard Implemented on FPGA" IEEE Inter.Conf. Comp Elec some extent, the power consumption is directly related to the chip area. Therefore the encryption device implemented in this method can meet some practical applications. As the S-box is implemented by Look Up Table in this design, the chip area and power can still be optimized. So the future work should focus on the implementation mode of S-box. Mathematics in Galois field can accomplish the bytes substitution of the AES algorithm, which could be another idea of fu<mark>rther resea</mark>rch.

Engin.(IECEE),vol.02,issue.28,pp.656-660,Dec.2009.

[4]Alessandro Cilardo "Exploring the Potential of Threshold Logic for Cryptography-Related Operations" In IEEE Transactions On Computers, Vol. 60, No. 4, (April 2011).

[5]Babitha P. K, Thushara T, Dechakka M. P. "FPGA based N-bit LFSR to generate random sequence number" in International Journal of Engineering Research and General ScienceVolume 3, Issue 3, Part- 2 , May-June, 2015, ISSN 2091-2730.

FOR

En

6]Divya Jenifer D' Souza, Minu P Abraham "A multilayered Secure for Transmission of Sensitive Information based on Steganalysis" in ELSEIVER, Procedia computer science 78 (2016).

[7]HuiXua, Xiaojun Tonga, Xianwen Menga, "An efficient chaos pseudorandom number generator applied to video encryption" in ELSEIVER, OPTIK 127 (2016).

[8]IrithPomeranz "Computing Seeds for LFSR-Based Test Generation From Nontest Cubes" in IEEE transactions on very large scale integration (vlsi) systems, vol. 24, no. 6, june 2016.

OPEN ACCESS JOURNAL