ISSN 2278 – 3091

International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), Vol. 4 , No.4 Pages : 28 - 33 (2015)
Special Issue of ICEEC 2015 - Held on August 24, 2015 in The Dunes, Cochin, India
http://warse.org/IJATCSE/static/pdf/Issue/iceec2015sp06.pdf

An Enhanced Advanced Encryption Standard Algorithm

Kshyamasagar Mahanta[1], Hima Bindu Maringanti[2]
[1]Department of Computer Science & Engineering, North Orissa University, India, kshyamasagar74@gmail.com
[2]Department of Computer Science & Engineering, North Orissa University, India, profhbnou2012@gmail.com

Abstract: In today's world most of the communication is done using electronic media. Data Security plays a vital role in such communication. Hence, there is a need to protect data from malicious attacks. Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication and data origin authentication. In data and telecommunications, cryptography is necessary when communicating over any unreliable medium, which includes any network particularly the internet.

Advanced Encryption Standard (AES), also known as Rijndael, is an encryption standard used for securing information which is defined by The National Institute of Standard and technology (NIST) of United States[1][5][9]. AES is a block cipher algorithm that has analyzed extensively and is now used widely[13]. AES involves the sequence of four primitive functions: Sub Bytes, Shift Rows, Mix Column and Add Round Key.

This paper presents some modifications to make AES more secure.

Key words : AES , Cryptography, Encryption, Random Number generator.

INTRODUCTION

Many Encryption algorithms are widely available and used in information security. They can be categorized into Symmetric(private) and Asymmetric (public)keys encryption. In Symmetric Key encryption or secret key encryption only one key is used to encrypt and decrypt data. In asymmetric keys two keys are used :private and public keys. Public key is used for encryption and private key is used for decryption. Public key encryption is based on mathematical functions, computationally intensive. There are many examples of strong and weak keys of cryptography algorithms like DES,AES.DES uses one 64-bits key while AES uses various 128,192,256 bits keys[5][8][13]. Asymmetric encryption techniques are almost 1000 times slower than symmetric techniques , because they require more computational computing power.

Though AES algorithm is secured, still some security issues lie with this. In 2010, Abdulkarim Amer Shtew et. al. have found such issues and modified the standard algorithm by modifying the shift row phase involved . Similarly, in 2010, El-Sayed Abdoul-Moaty ElBadawy et. al. and in 2011, Zhang Zhao et. al. have modified the standard AES algorithm by modifying the S-box generation using 1-D logistic chaos equation . In 2011, Alireza Jolfaei et. al. have identified such issues and modified the standard algorithm by modifying the S-box using the chaotic baker's map equations .

Here, we have followed a different approach by encrypting the key used in the AES algorithm. For the key encryption, we have used the RNG algorithm, which gives a unique random key for each time of encryption. And the key will be sent along with the cipher text for decryption. The position of the key in the cipher text will be chosen by another random no. generator. So that the key position will vary each time.

PROBLEM STATEMENT

To use a random key instead of constant key in AES algorithm. For each time of encryption a random key will be generated and the length of the key should be changed. Instead of sending the cipher text and key differently for decryption , we can send both at a time in a single message.

LITERATURE SURVEY

[1]. Design of High Speed 128 bit AES Algorithm for Data Encryption

In this paper software implementation of Advanced Encryption Standard (AES) algorithm is used for data encryption that can process with the data block of 128 bit and cipher key length of 128 bit[5] . The usage of 128 bit cipher key to achieve the high security, because 128 bit cipher key is difficult to broken. As result of this secure transmission of data is occurred in encryption. While computing the existing AES, the Sub Bytes transformation consumes the more memory in AES so to overcome this affine transform is used in AES flow.

[2]. Random Number Generation Using Deterministic Random Bit Generators

This Paper specifies techniques for the generation of random bits that may then be used directly or converted to random numbers when random values are required by applications using cryptography.

There are two fundamentally different strategies for generating random bits. One strategy is to produce bits non-deterministically, where every bit of output is based on a physical process that is unpredictable; this class of random bit generators (RBGs) is commonly known as non-deterministic random bit generators (NRBGs). The other strategy is to compute bits deterministically using an algorithm; this class of RBGs is known as Deterministic Random Bit Generators (DRBGs).

[3]. AES Algorithm Using 512 Bit Key Implementation for Secure Communication

The paper consist of a new version of the advanced encryption standard algorithm with efficient utilization of resources such as processor and memory. The new algorithm AES 512 consists of input block of 512 bit and key 512 bit [4][7]. Due to this provision it becomes more resistant to linear and differential encrypt analysis providing high security and throughput by consuming less memory and processor. The result show that the tremendous increase in the throughput to 230% than AES 128 bit algorithm[12].

Drawbacks of AES

Rijndael has very strong resistance against the differential cryptanalysis and linear cryptanalysis attacks since it used Wide Trail Strategy in its design. Although these linear attacks are invalid for the AES, they have been extended in several ways for recent years and new attacks have been published that are relevant to them. The newest attack combined boomerang and the rectangle attack with related-key differentials was introduced by E. Biham ,et al. in 2005[14].It uses the weaknesses of few nonlinear transformations in the key schedule algorithm of ciphers ,and can break some reduced-round versions of AES. It can break 192-bit 9-round AES by using 256 different related keys. Rijndael inherits many properties from square algorithm. So the square attack is also valid for Rijndael which can break round –reduced variants of Rijndael up to 6 or 7 rounds(i.e.AES-128 and AES-192)faster than an exhaustive key search. N. Ferguson et.al. proposed some optimizations that reduce the work factor of the attack. So, this attack breaks a 256-bit 9-round AES with 2^{77} plaintexts under 256 related keys, and 2^{224} encryptions. For the same plain text when we repeatedly encrypt , cipher text will be same all the time.

METHODOLOGY

1. AES Evaluation

i. Security – 128 minimal key sizes provides enough security.
ii. Cost – AES should have high computational efficiency.

i. Security: This refers to the effort required to crypt analyze an algorithm. The emphasis in the evaluation was on the practicality of the attack. Because the minimum key size for AES is 128 bits, brute-force attacks with current and projected technology were considered impractical. Therefore, the emphasis, with respect to this point, is cryptanalysis other than a brute-force attack.

ii. Cost: NIST intends AES to be practical in a wide range of applications. Accordingly, AES must have high computational efficiency, so as to be usable in high-speed applications.

2. The AES Cipher

The AES Algorithm is a symmetric-key cipher, in which both the sender and the receiver use a single key for encryption and decryption. The data block length is fixed to be 128 bits, while the length can be 128, 192, or 256 bits [5][8]. The detailed is shown in Table1. The AES algorithm is an iterative algorithm. Each iteration can be called a round, and the total number of rounds is 10, 12, or 14, when key length is128,192, or 256, respectively [9][10]. The 128 bit data block is divided into 16 bytes. These bytes are mapped to a 4x4 array called the State, and all the internal operations of the AES algorithm are performed on the State (K.Rahimunnisa et al, 2012)(FIPS PUB 197, 2001).

Table 1 :AES Parameters

Algorithm	Key length (Nk words)	Block size(Nb words)	Number of rounds (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

The encryption process is iterative in nature. Each iteration is known as rounds. For each round 128 bit input data and 128 bit key is required. That is, need 4 words of key in one round [10]. So the input key must be expanded to the required number of words, which depends upon the number of rounds. The output of each round serves as input of next stage. In AES System, same secret key is used for both encryption and decryption. So it provides simplicity in design. The block diagram for encryption is shown in Fig.1.

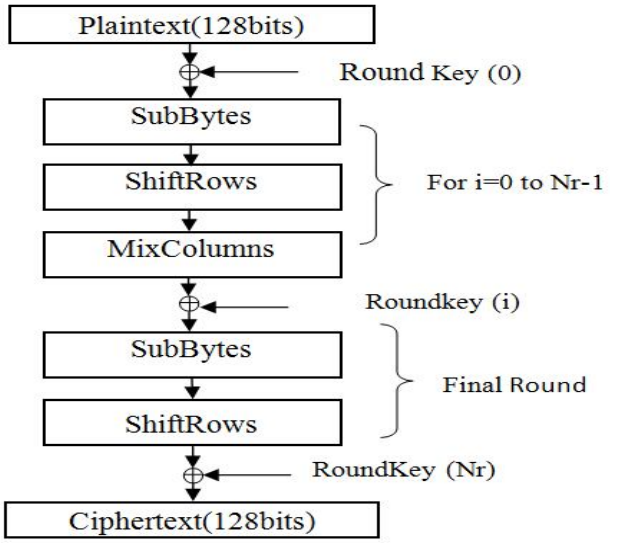
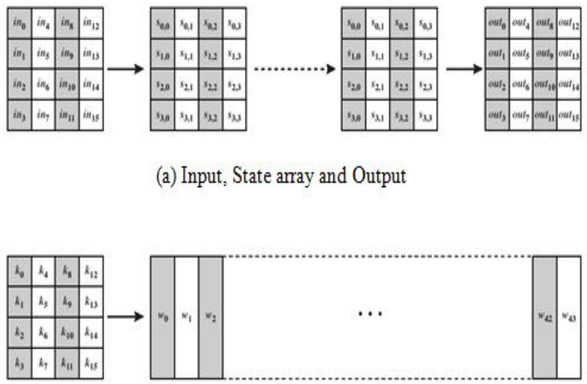


Fig.1 Detailed Block diagram of an encryption

The input to the encryption algorithm is a single 128-bit block, this block, in FIPS PUB 197, is depicted as a square matrix of bytes. This block is copied into the State array, which is modified at each stage of encryption or decryption. After the final stage, State is copied to an output matrix. These operations are depicted in Fig.2.a.

Similarly, the 128-bit is depicted as a square matrix of bytes. This key is expanded into an array of key schedule words; each word is 4 bytes and the total key schedule is 44 words for the 128-bit key (Fig.2.b). Ordering of bytes within a matrix is by column. Before delving into details, we can make several comments about overall AES structure:


in_0 in_4 in_8 in_{12}
in_1 in_5 in_9 in_{13}
in_2 in_6 in_{10} in_{14}
in_3 in_7 in_{11} in_{15}
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
out_0 out_4 out_8 out_{12}
out_1 out_5 out_9 out_{13}
out_2 out_6 out_{10} out_{14}
out_3 out_7 out_{11} out_{15}
k_0 k_4 k_8 k_{12}
k_1 k_5 k_9 k_{13}
k_2 k_6 k_{10} k_{14}
k_3 k_7 k_{11} k_{15}
w_0 w_1 w_2
w_{42} w_{43}

(a) Input, State array and Output

(b) Key and Expand Key

Fig.2 AES Data Structures

□ This cipher is not a Feistel structure.

□ The key that is provided as input is expanded into an array of 44 words (32-bits each), w[i]. 4 distinct words (128 bits) serve as a round key for each round; these are indicated in Figure. 2.

□ 4 different stages are used, 1 permutation and 3 of substitution:

□ Substitute bytes – Uses an S-box to perform a byte-to-byte substitution of the block.

□ Shift rows – A simple permutation.

□ Mix columns – A substitution that makes use of arithmetic over GF (28).

□ Add round key – A simple bitwise XOR of the current block with the portion of the expanded key.

□ The structure is quite simple. Fig.3 depicts the structure of a full encryption round.

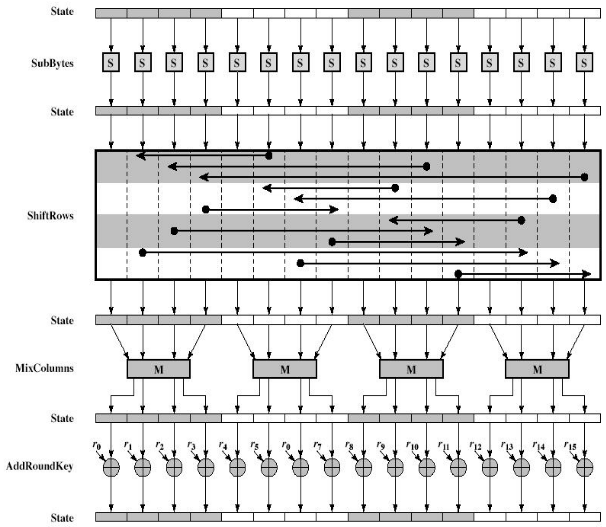

State
SubBytes
S
State
ShiftRows
State
MixColumns
M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_6 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

Fig.3 AES Encryption Round

□ Only the Add Round Key stage uses the key. Any other stage is reversible without knowledge of the key.

□ The Add Round Key is a form of Vernam cipher and by itself would not be formidable. The other 3 stages together provide confusion, diffusion, and nonlinearity, but by themselves would provide no security because they do not use the key. We can view the cipher as alternating operations of XOR encryption (Add Round Key), followed by scrambling of the block.

□ The final round of encryption consists of only 3 stages; it is the consequence of the particular structureof AES (William Stallings, 2005).

Design Steps AES Algorithm

SubBytes Transformation

AES defines a 16 x 16 matrix of byte values called an S-box that is a pre calculated substitution table contains 256 numbers (from 0 to 255) and their corresponding resulting values[9][6][10]. S-box table is as shown in Table 2. Each byte of State array is mapped into a new byte in the following way: The leftmost 4 bits of the byte are used as a row value and the right most 4 bits are used as a column value (P.Karthigaikumar et al, 2011)(P. Aatheeswaran et al, 2013). These row and column values serve as indexes into the S-box to select a unique 8-bit output value as shown in Figure.4.

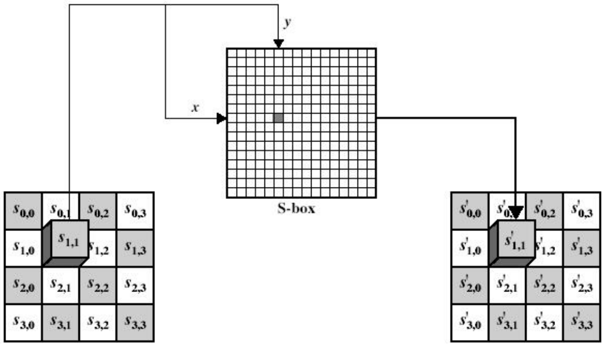

y
x
S-box
$s_{0,0}$ $s_{0,1}$ $s_{0,2}$ $s_{0,3}$
$s_{1,0}$ $s_{1,1}$ $s_{1,2}$ $s_{1,3}$
$s_{2,0}$ $s_{2,1}$ $s_{2,2}$ $s_{2,3}$
$s_{3,0}$ $s_{3,1}$ $s_{3,2}$ $s_{3,3}$
$s'_{0,0}$ $s'_{0,1}$ $s'_{0,2}$ $s'_{0,3}$
$s'_{1,0}$ $s'_{1,1}$ $s'_{1,2}$ $s'_{1,3}$
$s'_{2,0}$ $s'_{2,1}$ $s'_{2,2}$ $s'_{2,3}$
$s'_{3,0}$ $s'_{3,1}$ $s'_{3,2}$ $s'_{3,3}$

Fig.4 S-box transformation

Table 2: S-box table

		y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x	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 Rows Transformation

Every row in the state is shifted a certain amount to the left. In this operation, each row of the state is cyclically shifted to the left, depending on the row index. The first row is not shifted, the second shifted 1 byte position, the third 2 byte and the fourth 3 byte position (AES FIPS PUB 197, 2001). A graphical representation of shift rows transformation is shown Fig.5.

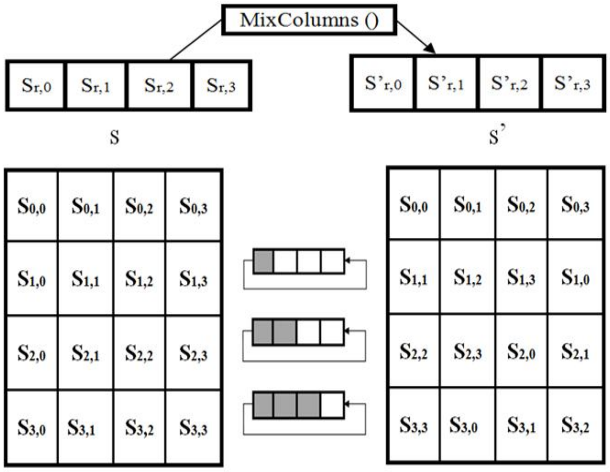


Fig.5 Shift Rows transformation

MixColumns Transformation

MixColumn operation (Fig.6) performs on the state column by column, and each column is treated as a four term Polynomial over GF ( ) As a result of this multiplication, the new four bytes in a column are generated as follows (Vasamsetti Ramoji et al, 2012):

A'= ({02}.A)^({03}.B)^({01}.C)^({01}.D)
B'= ({01}.A)^({02}.B)^({03}.C)^({01}.D)
C'= ({01}.A)^({01}.B)^({02}.C)^({03}.D)
D'= ({03}.A)^({01}.B)^({01}.C)^({02}.D)

The operation of '^' is XOR operation modulo 2 and the '.' is a multiplication of polynomials modulo an irreducible polynomial [6] m(x) = x8+ x4 + x3+ x +1 .The operation of {02}.X can be computed using Verilog HDL HDL language:
{02}.X ={X[6: 0],1'b0}^(8'h1B&{8{X[7]}})
So {03}_ X can be generated as follows (Vasamsetti Ramoji et al, 2012) :
{03}.X = ({02}.X) +{01}.X(3)

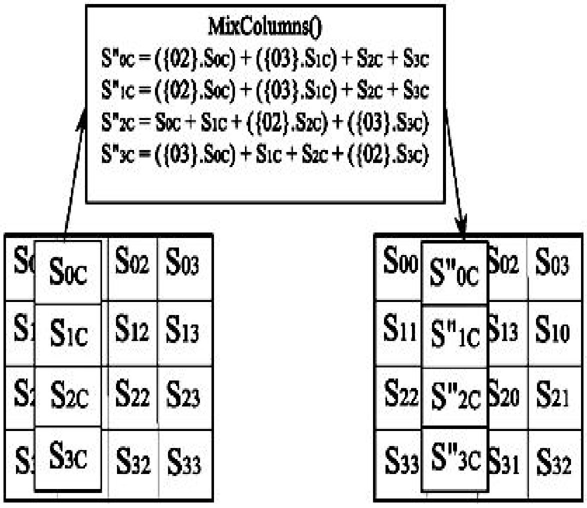

MixColumns()
S"0c = ({02}.S0c) + ({03}.S1c) + S2c + S3c
S"1c = ({02}.S0c) + ({03}.S1c) + S2c + S3c
S"2c = S0c + S1c + ({02}.S2c) + ({03}.S3c)
S"3c = ({03}.S0c) + S1c + S2c + ({02}.S3c)
S0c S02 S03
S1c S12 S13
S2c S22 S23
S3c S32 S33
S00 S"0c S02 S03
S11 S"1c S13 S10
S22 S"2c S20 S21
S33 S"3c S31 S32

Fig.6 MixColumns transformation

AddRoundKey Transformation

During the AddRoundKey transformation, the round key values are added to the State by means of a simple Exclusive or (XOR) operation (Vasamsetti Ramoj et al, 2012). Each round key consists of Nb words that are generated from the Key Expansion routine. The round key values are added to the columns of the state in the following way:

$[S'_{0,c}, s'_{1,c}, S'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}]$ XOR[Wround*Nb+c] for 0<=C<Nb [9]

In the equation above, the round value is between $0 \le round \le N_{r.}$ When round=0, the cipher key itself is used as the round key and it corresponds to the initial AddRoundKey transformation [9]. The AddRoundKey transformation is illustrated in Fig.7.

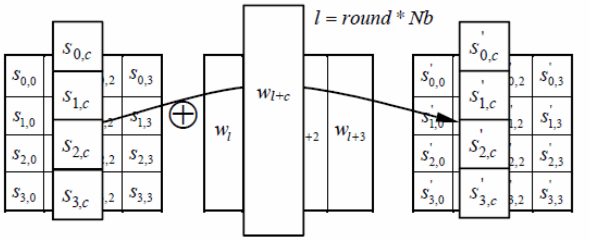


Fig.7 AddRoundKey XORs each column of the state with a word from the key schedule

Key Expansion

The AES algorithm takes the Cipher Key and performs a Key Expansion routine to generate a key schedule (P.Karthigaikumar et al, 2011). This process, as shown in Fig.8, consists of the following sub-functions:

- RotWord performs a one-byte circular left shift on a word.
- SubWord performs a byte substitution on each byte of its input word using the S-box.

The result of steps i and ii is XOR-ed with a round constant RC[j] is shown in Table 3.

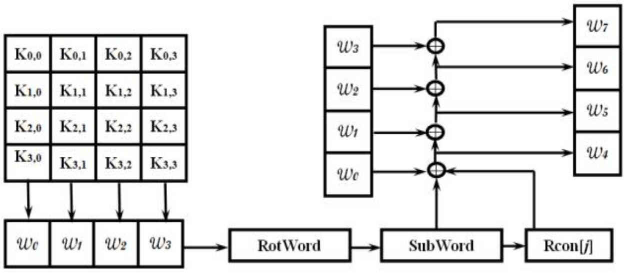


Fig.8 Key expansion process

Table 3: The value RC[j] in Hexadecimal

j	1	2	3	4	5	6	7	8	9	10
RC[j]	01	02	04	08	10	20	40	80	1B	36

Random Number Generator:

A random number generator (RNG) is a computational or physical device designed to generate a sequence of numbers or symbols that lack any pattern, i.e. appear random. Several computational methods for random number generation exist. Many fall short of the goal of true randomness — though they may meet, with varying success, some of the statistical tests for randomness intended to measure how unpredictable their results are However, carefully designed cryptographically secure computationally based methods of generating random numbers do exist, such as those based on the Yarrow algorithm and the Fortuna (PRNG) and others.

Here we have used 3 random number generators. The work of these 3 random number generators are:
1-defines the length of the key. 2-generates the random key according to the length defined by 1st random number generator.3-defines the position where the key should present in the cipher text at the time of transferring the encrypted key.

RESULTS

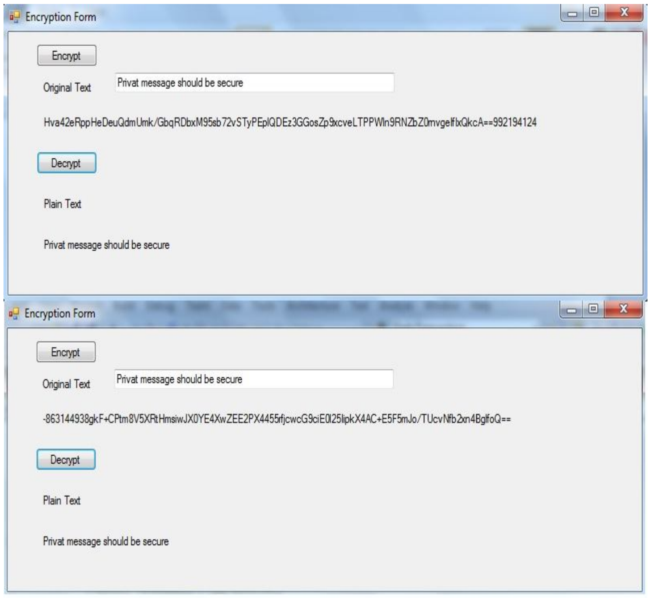
In standard AES:

Plain text: private message should be secure
Cipher text:
QffPAvP7d9ArWAyzW46dORIWfR0cS/+LT+7S4P+d39To293CilYrR/33EeDptzFhIqVDTfUIwHioSq4T0HldZKX8Fn852wNvK5sDnsnxNAc

In Enhanced AES:

CipherText:
1.
-1420322800nfBXE4uJVAYYWKx4rrCN2oU75PX5GMeMok5N9TimThIvY+yhMN0ITMaop51glRXL8c4yUAJoTf5njy7EYyT5S7PkFan0wyGaJb5vV+fQDlo=
2.
242305485OOOaawhRn4ykDgfcIxntn428vDBkwwV4Zp6Hn2b8pb+4meRfizcXZz8qJeGtf0wA7ddx7gZWlFW/nflUFLyPFZSsAkywoHK9RSo1pqASfxE=
3.
Za/Nfm+rhrA6mDlA5ldqBmtNzv/iXLcOV1GRM2+qyn64MNwRxmNikSBwQC6+vNpOhKP3XwXvM0t4kI09tZW/FIsSZesVMRkQv6RMtNPOK3c=-32829382

Fig.9 shows the result.



Original Text: Privat message should be secure

Hva42eRppHeDeuQdmUmk/GbqRDbxM95sb72vSTyPEplQDEz3GGosZp9xcveLTPPWln9RNZbZ0mvgeIfIxQkcA==992194124

Plain Text

Privat message should be secure

Original Text: Privat message should be secure

-863144938gkF+CPtm8V5XRtHmsiwJX0YE4XwZEE2PX4455rfjcwcG9ciE0I25lipkX4AC+E5F5mJo/TUcvNfb2xn4BgIfoQ==

Plain Text

Privat message should be secure

Fig.9 Result

CONCLUSION

Due to the increasing needs for secure communications, a more safe and secure cryptographic algorithms has to be proposed and implemented. The Advanced Encryption Standard (AES-128bit) is widely used nowadays in many applications. In this paper, we proposed a new variation of AES with variable length key size compared with 128-bit in the original AES-128 algorithm. We got the best result as it generates different cipher text for the same input value which is lack in standard AES. The larger key size make the algorithm more secure, and the larger input block increases the throughput. The extra increase in area can be tolerated and makes the proposed algorithm ideal applications in which high level of security and high throughput are required such as in multimedia communications.

Here we have used random key all the time and the key length is also changed .We distribute the key with the cipher text at any position and the position is changed randomly. So it will be difficult to crack by outsider. We have implemented this and we got the best result. We can also use this concept for image encryption using AES [11].

REFERENCES

[1]. Sumalatha Patil Ma* and Mala L Ma, International Journal of Current Engineering and Technology" Design of High Speed 128 bit AES Algorithm for Data Encryption" ISSN 2277 – 4106, ©2013 INPRESSCO.

[2]. Elaine Barker and John Kelsey, “Random Number Generation Using Deterministic Random Bit Generators,” NIST Special Publication 800-90A, January 2012.

[3]. Rishabh Jain1, Rahul Jejurkar2, Shrikrishna Chopade3, Someshwar Vaidya4, Mahesh Sanap5, “AES Algorithm Using 512 Bit Key Implementation for Secure Communication” International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801, Vol. 2, Issue 3, March 2014.

[4] S.Radhika, A.ChandraSekar ,“AES Algorithm Using 512 Bit Key Implemented For Secure Communication”, GJCST,Vol. 10 ,2010.

[5] Rohan Rayarikar, SanketUpadhyay, PriyankaPimpale, “SMS Encryption usingAES Algorithm on Android”, IJCA, Volume 50- No.19, 2012.

[6] Joan Daemen and Vincent Rijmen, “A Specification for Rijndael, the AES Algorithm”, Dr. Brian Gladman, v3.1, 2001.
[7] Ashwaq T. Hashim , “A Proposed 512 bits RC6 Encryption Algorithm”,IJCCCE,vol.10, no.1, 2010.

[8] M.Anand Kumar and Dr.S .Karthikeyan ,“Investigating the Efficiency of Blow-fish and Rijndael (AES) Algorithms” ,I. J. Computer Network and Information Security, 2012.

[9] Advanced Encryption Standard (AES) (2001), Federal Information Processing Standards publication 197.

[10] K.Rahimunnisa, M. Priya Zach, S. Suresh Kumar and J.jayakumar (2012) Architectural optimization of AES Transformations and key expansion.

[11] P.Karthigaikumar and Soumiya Rasheed (2011), Simulation of Image Encryption using AES Algorithm.

[12] Vasamsetti Ramoji, P.Ganesh and Ch.Appala Swamy, (2012), Highly Secured High Throughput Efficient VLSI Architecture for AES Implementations.

[13] Dr. Prerna Mahajan & Abhishek Sachdeva, A Study of Encryption Algorithms AES, DES and RSA for Security, Global Journal of Computer Science and TechnologyNetwork, Web & SecurityVolume 13 Issue 15 Version 1.0 Year 2013.

[14] A. Biryukov, “The Boomerang Attack on 5 and 6-Round Reduced AES”,LNCS 3373, pp.11-15, Springer, 2005.