Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Design of a AES 2D Rotations Algorithm used in Protection of Smart Cards

Umadatta. A
Student, M.Tech CSE Dept.,
Institute of Aeronautical Engineering
HYD-500043, AP, India.

Dr. N. Chandra Sekhar Reddy
Professor, CSE Dept.,
Institute of Aeronautical Engineering,
HYD-500043, AP,India.

Arvind Kumar
Professor, CSE Dept.,
Institute of Aeronautical Engineering,
HYD-500043, AP, India.

Abstract- Advanced Encryption Standard was created by two Belgian cryptographers, Vincent Rijmen and Joan Daemen, replaced the old Data Encryption Standard (DES). Advanced Encryption Standard was accepted for commercial use due to its parallel and symmetric structure and well adapted to modern processors and its suitability to smart cards. AES is a non-Feistel cipher which encrypts and decrypts a data block of 128 bits. It will use 10, 12, or 14 rounds. The key size, that can be of 128, 192, or 256 bits, depends on the number of rounds performed. We propose a modification where we rotate the bytes using 2 Dimensional rotation of the block after Step 4, thereby increasing confusion-diffusion. The scheme proposed has improved complexity that is going to increase the security for a 256 bits block case without increase in the key size.

Keywords- S-Box, Byte substitution, Shift Rows, Mix Columns, Round key addition and 2D rotation operation.

I. INTRODUCTION

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that is used to protect electronic data such as smartcards. The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption will convert the data to an unintelligible form called ciphertext; in Decryption the ciphertext converts the data back into its actual form, called as plaintext. It is based on some simple operations like Exclusive OR (XOR), bits shifts and permutations of columns [1 & 2]. The algorithm in its original form contains of four transformations in each rounds, namely, Byte Substitution, Shift Row, Mix Columns, Add Round Key (XOR). The last round does not use the Mix Columns transformation. In this paper we present a modification in AES Rijndael algorithm, by adding one more round that will increase the overall confusion-diffusion of bytes thereby increasing the complexity which cryptanalysis of the algorithm. The proposed transformations involves mathematical operations that are easy to implement in software level especially by using JAVA programming.

II. RELATED WORK

AES algorithm has highly mathematical structure. There are four steps involved in the algorithm, which perform specific transformations in the input plaintext. The algorithm uses three key lengths that are independent of block length: 128, 192, or 256 bits. It consists of 10, 12 or 14 rounds where each round consists of transformations.

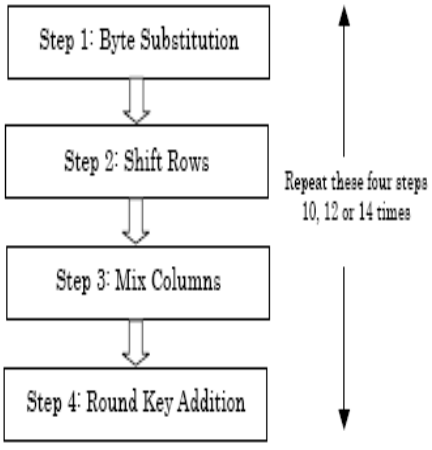

Figure 1. Steps in original AES algorithm

A. Byte Substitution

In the Byte substitution step, each byte in the matrix is reorganized using an 8-bit substitution box. This substitution box is called the Rijndael S-box. The S-box is chosen to avoid the fixed points (and so is a derangement), and also the opposite fixed points.[3]

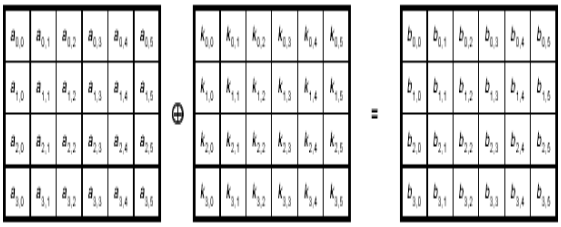

Figure 2. Byte subtitutuion

B. Shift Rows

The Shift Rows step is performed on rows of the state matrix. It cyclically shifts the bytes of each row by a certain offset. First row remains unchanged. In the second row each byte is shifted one position to the left. Similarly, in the third and fourth rows the bytes are shifted by two positions and three positions respectively.[1]

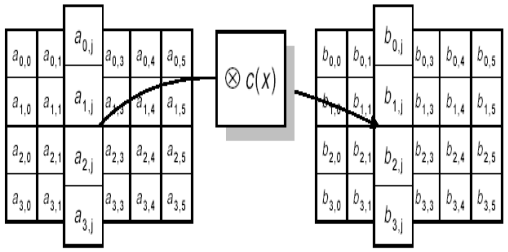

Figure 3. Shift rows

C. Mix columns

In Mix Columns step, the four bytes of each column of state matrix are combined using an invertible linear transformation[3][9].

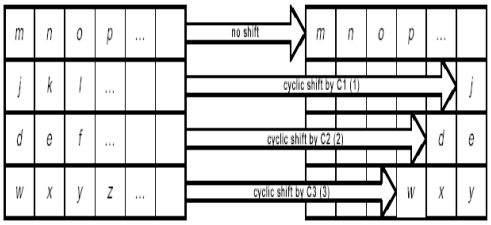

Figure 4. Mix Columns

D. Round Key Addition

A round key is generated using some operations on the cipher key which is XORed with the entire block state obtained till the Mix Column transformation[5].

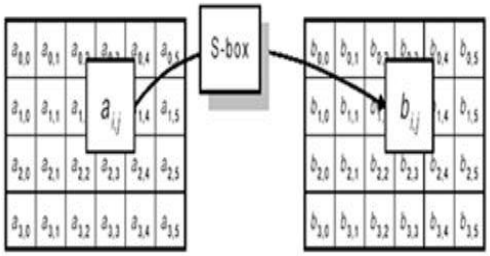

Figure 5. Round Key Addition

III. LITERATURE SURVEY

The four steps involved in the algorithm, perform the specific transformations in the input plaintext. The last round does use the Mix Columns transformation[3].

IV. PROPOSED AES ALGORITHM

In the modified approach of AES algorithm, the plaintext blocks and keys can be arranged in any square variable size of 4X4, 4X8 etc.

The modified AES consists following transformations; Byte Substitution, Shift Rows,, Mix Column, Round Key Addition and 2D Rotate Block. This new kind of rotation in the AES algorithm has been taken from the previous work on 3D Array Block ciphers[2]. We use the same approach in AES using certain mathematical operations on the matrix

structure of the block. In this kind of 2D rotation, the entire Array block is rotated by certain angle depending upon the certain value of key bits. Its use will further increase the confusion aspect when the information bytes are transformed to the ciphertext.

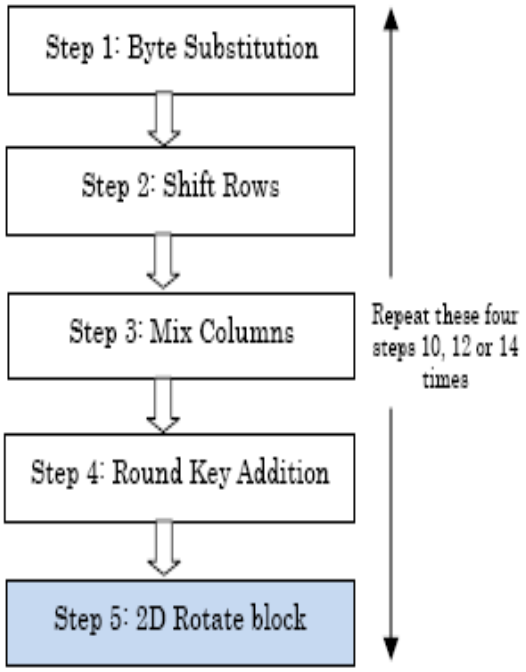

Figure 6. Architecture of modified AES

V. ANALYSIS IN 2D ROTATION OPERATION

A. Notations

We perform two operations on matrix structure,

1. rCOOM(M) that denotes :
   reverseColumnsOrderOfMatrix M i.e. function that arranges the columns of the matrix in reverse order and
2. rROOM(M) that denotes :
   reverseRowsOrderOfMatrix M i.e. function that arranges the rows of the matrix in reverse order.

We use the standard notation M' to denote the transpose of a matrix M.

B. Rotations using Encryption

In Encryption, the entire block is rotated in clockwise direction by an amount of 0°, 90°, 180°, 270° depending upon the bits of the key value 00, 01, 10, 11 respectively. This rotation will change the relative positioning of the information in the 2D array block. The modified step can be implemented by use of the mathematical operations as discussed below:

We consider an initial array of the bytes as a 2D array M of size say 4X4, which indicates number of rows and number of columns of the array then a rotation of 90° can be thought of as a two step mechanism (as shown in Fig.7). The rotation considered is as follows:

rCOOM (M)=

Row \ Column	4	3	2	1
1	D	C	B	A
2	H	G	F	E
3	L	K	J	I
4	P	O	N	M

rROOM (rCOOM (M))=

Row \ Column	4	3	2	1
1	P	O	N	L
2	L	K	J	I
3	H	G	F	E
4	D	C	B	A

Figure 7. 90° Rotation of 2D Block Array

Similarly, we reverse the order of the columns of the Matrix M (as shown in Fig.8) which is equivalent to a 180° rotation.

M'=

Row \ Column	1	2	3	4
1	A	E	I	M
2	B	F	J	N
3	C	G	K	O
4	D	H	L	P

rCOOM (M')=

Row \ Column	4	3	2	1
1	M	I	E	A
2	N	J	F	B
3	O	K	G	C
4	P	L	H	D

Figure 8. 180° Rotation of 2D block Array

rCOOM (M) operation can be used in above case,

rROOM (rCOOM (M)) = Rotation of 180°

and lastly a rotation of 270° is done by reversing the row order of the transposed matrix M (as shown in Fig. 9). Mathematically, it can be denoted by the operations as discussed below:

rROOM (M') = Rotation of 270°

M'=

Row \ Column	1	2	3	4
1	A	E	I	M
2	B	F	J	N
3	C	G	K	O
4	D	H	L	P

rROOM (M')=

Row \ Column	1	2	3	4
4	D	H	L	P
3	C	G	K	O
2	B	F	J	N
1	A	E	I	M

Figure 9. 270° Rotation of 2D block Array

VI. ALGORITHM

The primary encryption module calling every other function:

Step 1: ByteSub Transformation(SubBytes(state);)

Step 2: ShiftRow Transformation(ShiftRows(state);)

Step 3: MixColumn Transformation(MixColumns(state);)

Step 4: Round Key Addition(AddRoundKey(state, w[Nr*Nc,(Nr+1)*Nc-1]);)

Step 5:2D Rotate state(2DRotateState(state, parityCode(state));)

modifiedAES(byte in[4*Nc], byte out[4*Nc], word w[Nc * (Nr+1)])
{


```
byte state[4, Nc];
state = in;
AddRoundKey(state, w[0, Nc-1])
for(roundNo=1; roundNo<Nr; roundNo++)
{
SubBytes(state);
ShiftRows(state);
MixColumns(state);
2DRotateState(state, parityCode(state));
AddRoundKey(state, w[Nr*Nc,(Nr+1)*Nc-1]);
}
SubBytes(state);
ShiftRows(state);
AddRoundKey(state, w[Nr*Nc, (Nr+1)*Nc-1]);
2DRotateState(state, parityCode(state));
state=out;
}
```

VII. CONCLUSION

We have provided modification of AES that can be applied without increase in the size of the key block. Inclusion of one more round has further increased the complexity involved to decrypt ciphertext of AES using Brute-force attack. Backward compatibility provided in the modified approach will be beneficial till systems in communication do not upgrade to the software implementation of the same. In all, one can conclude that AES turned to a Feistel structure might prove more complex to be attacked by intruders with malified intensions.

VIII. FUTURE ENHANCEMENT

The original AES have the complexity of the order of bits used as key. The expected strength is of the order of 2^{127} for 16 bytes of key and 2^{255} for 32 bytes of key. With the introduction of the new round the complexity of the intruder will increase by the order of $3^{NumberOfRounds}$. This is because at the end of every round in modified form of the AES, the attacker needs to check for 3 possible block values that could have resulted due to 2D Array block rotation. However, this step has made the AES turned to a Feistel structure. The very most important security advantage is that no differential or linear attacks on AES have been able to break the algorithm. We have a plan to develop a new substitute box (S-Box) which will satisfy all the cryptographic properties. Also we are planning to apply hybrid cellular automata rules for block cipher.

REFERENCES

[1] Dr (Mrs) Pushpa R. Suri and Sukhvinder Singh Deora, "Design of a modified Rijndael algorithm using 2D Rotations", IJCSNS International Journal of Computer .

[2] Dr. (Mrs) Pushpa R. Suri, "A Cipher based on 3D Array Block Rotation", International Journal of Computer Science and Network Security, VOL.10 No.2, February 2010, pp. 186-191.

[3] V. Sumathy, C.Navaneethan, "Enhanced AES Algorithm for Strong Encryption", IJCSNS International Journal of Computer Science and Network Security.

[4] J.Daemen and V.Rijmen, AES Proposal: Rijndael, NIST's AES home page, http ://www:nist:gov/aes.

[5] Announcing the Advanced Encryption Standard (AES), Federal Information Processing Standards Publicatrion 197, November 2001

[6] Dr. (Mrs) Pushpa R. Suri, "3D Parity Bit Structure: a novel technique to correct maximal number of bits in a simpler way", International Journal of Computer Science and Internet Security, VOL.9 No.8, August 2011, pp. 182-186

[7] Chun Yan,Yanxia Guo, "A Research and Improvement Based on Rijndael Algorithm", 2009 First International Conference on Information Science and Engineering, Nanjing, Jiangsu China, December 26-December 28, ISBN: 978-0-7695-3887-7

[8] Naim Ajlouni et.al., "A new approach in Key Generation and Expansion in Rijndael Algorithm", The International Arab Journal of Information Technology, Vol. 3, No. 1, January 2006, pp. 35-41.

[9] Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[10] Xinmiao Zhang and Keshab K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm", 1531-636X/12, IEEE 2002.

