

Enhanced AES Algorithm for Strong Encryption

V.Sumathy and C.Navaneethan

KingstonEngineeringCollege

Abstract

In this paper we present a new AES encryption and decryption process for both plain text and key Expansion. This algorithm is widely accepted due to its strong encryption, complex processing and its resistance to Brute-force attack. The proposed modifications are implemented on the rounds of the algorithm and Hash Based key expansions are made. These modifications enhance the degree of complexity of the encryption and decryption process, thereby making it difficult for the attacker to predict a pattern in the algorithm.

KeyWords

S-Box, Key Matrix, Modified SubBytes, Modified ShiftRows, Modified MixColumns, Modified Key Expansion, Modified Inverse MixColumns, Modified Inverse ShiftRows, Modified Inverse SubBytes.

1. Introduction

Rijndael Algorithm

The Advanced Encryption Standard comprises three block ciphers, AES-128, AES-192 and AES-256. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits. The block-size has a maximum of 256 bits, but the key-size has no theoretical maximum. The cipher uses number of encryption rounds which converts plain text to cipher text. The output of each round is the input to the next round. The output of the final round is the encrypted plain text known as cipher text. The input given by the user is entered in a matrix known as State Matrix. Following are the four steps.

1.1 SubBytes Step

SubBytes, also known as Byte substitution is the first iterative step of the algorithm in each round. In the SubBytes step, each byte in the matrix is reorganized using an 8-bit substitution box. This substitution box is called the Rijndael S-box. This operation provides the non-linearity in the cipher. The S-box used is derived from the multiplicative inverse over GF(28), known to have good non-linearity properties. To avoid attacks based on simple algebraic properties, the S-box is constructed by combining the inverse function with an invertible affine transformation. The S-box is also chosen to avoid any fixed points (and so is a derangement), and also any opposite fixed points. [5]

1.2 Shift Rows Step

The Shift Rows step is performed on the rows of the state matrix. It cyclically shifts the bytes in each row by a certain offset. The first row remains unchanged. Each byte of the second row is shifted one position to the left. Similarly, the third and fourth rows are shifted by two positions and three positions respectively. The shifting pattern for block of size 128 bits and 192 bits is the same.

1.3 Mix Columns Step

In the Mix Columns step, the four bytes of each column of the state matrix are combined using an invertible linear transformation.[5] A randomly generated polynomial is arranged in a 4*4 matrix. The same polynomial is used during decryption. Each column of the state matrix is XOR-ed with the corresponding column of the polynomial matrix. The result is updated in the same column. The output matrix is the input to Add Round Key.

1.4 Add RoundKey

A round key is generated by performing various operations on the cipher key. This round key is XOR-ed with each byte of the state matrix. For every round a new round key is generated using some operations on the cipher key.

2. PROPOSED MODIFIED AES ALGORITHM

The proposed changes in the current AES algorithm are as follows:

The cipher key is arranged in a 4*4 matrix. Each row of this matrix is used for specific operations in the proposed modifications. The first two rows of the cipher key matrix are used in the modified Sub Bytes round. The third row is used in the modified Shift Rows round. The last row is used in the modified Mix Columns round. The Add RoundKey step used in the original AES algorithm remains unchanged. Hash based key expansion used in proposed system. The plain text is there by arranged in a 4*4 matrix known as State matrix (M).

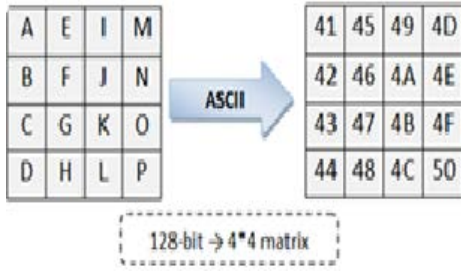


Fig. 1 ASCII Conversion

2.1 Modified Sub Bytes

Converting the first row of the key matrix into its binary equivalent, four groups of 8-bit binary values are generated as shown below. The first 4-bits from each 8-bit value are separated out as shown in figure. These bits are grouped into 4 groups: $g_0g_7, g_1g_6, g_2g_5, g_3g_4$ where g_0, g_1, g_2 and g_3 represent the row number and g_7, g_6, g_5 and g_4 represent the column number of state matrix respectively. The data at location $M(g_0, g_7)$ is substituted from S-box. The substitution is carried out according to the original Sub Bytes round of AES algorithm. This is repeated for the remaining locations viz. $M(g_1, g_6), M(g_2, g_5)$ and $M(g_3, g_4)$. Thus, using the first row of the cipher matrix 4 data elements are substituted in the state matrix. Similarly, the entire process is carried out using the second row of cipher key matrix.

- CIPHER KEY: 11223344556677889900AABBCCDDEEFF (128-bit cipher key)

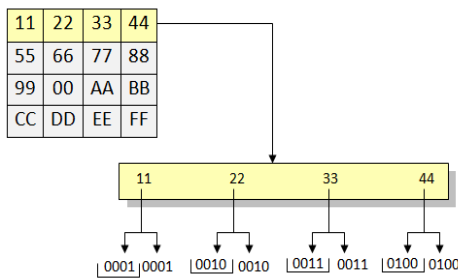


Fig. 2 Modified Sub Bytes

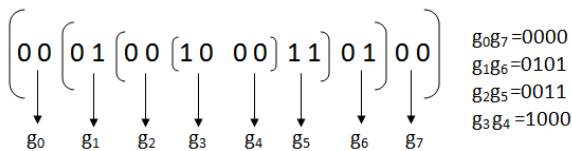
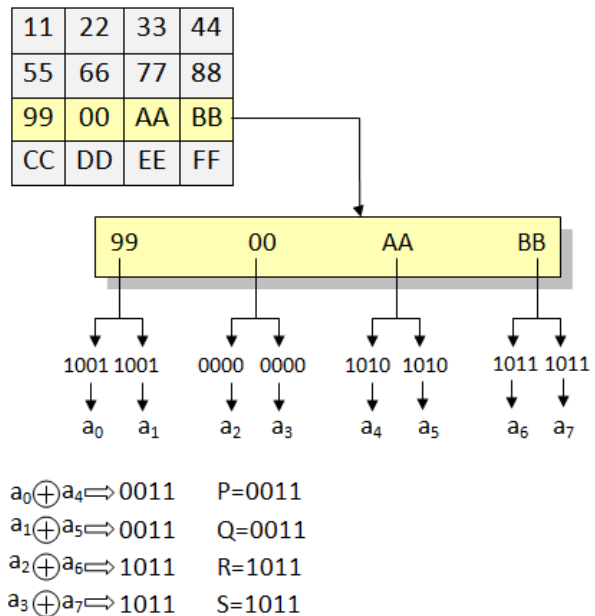


Fig. 3 Modified Sub Bytes

2.2 Modified Shift Rows

In this step the third row of the key matrix is converted to its binary equivalent form. The binary string is grouped into 8 groups, each group having 4 bits starting from first

bit as shown in the figure. The bits of a_0 are XOR-ed with those of a_4 to obtain a 4 bit binary result, P, as shown above. Similarly, Q, R and S are generated from the remaining groups i.e. $[a_1, a_5], [a_2, a_6]$ and $[a_3, a_7]$ respectively. Bits of P and R are used to identify the row number whereas the bits of Q and S give the number of cyclic left shifts. The first two bits of P represent the row number which is to be cyclically left shifted. For this row one less than the number of ones in Q is calculated. This gives the number of shifts for the row represented by P. The same process is repeated for R. Here, if first two bits (say a_0 and a_1) represent the same row which was previously shifted then a_1 and a_2 are considered which is the next immediate bit. Again, if the row number is same then a_2 and a_3 are checked followed by a_3 and a_1 . After checking all the bits, if the row number comes out to be the same, then the same row is shifted by one less than the number of ones in S.



Shift row $a_{ij}=a_{00}$ by 1 and $a_{ij}=a_{10}$ by 2
Mirror image of rows $a_{ij}=a_{01}$ and $a_{ij}=a_{11}$

Fig. 4 Modified Shift Rows

Therefore, the maximum number of rows shifted is 2. Mirror operation is performed on the remaining rows. In this operation, individual data is converted from each row into its binary form and read it from right to left, finally converting it back to hexadecimal value. An example is shown below. Consider data element 4A (Hexadecimal) = 0100 1010 (binary form) Reading from right to left we get 0101 0010 = 52 (Hexadecimal)

2.3 Modified MixColumns

In this round the elements of the fourth row of key matrix is grouped as shown in the figure. These groups are then converted into their respective decimal value and modulus-4 operation is performed on each group. The remainder (r) will always lie in range of 0 to 3 i.e. $0 \leq r \leq 3$. 0, 1, 2 and 3 represent the first, second, third and fourth column respectively. The Mix Column round of the original algorithm is carried out on the selected column. The maximum number of columns mixed in this step is 4.

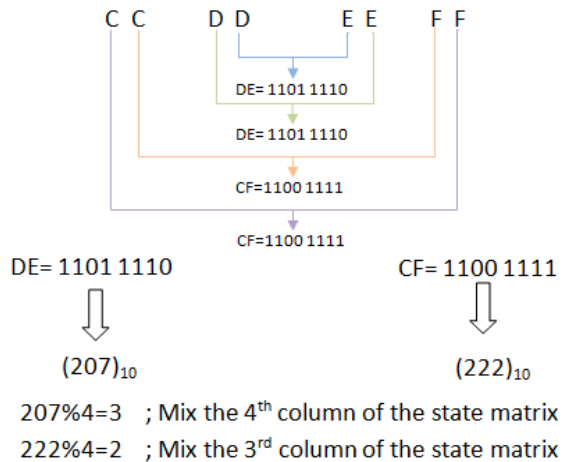


Fig. 5 Modified Mix Column

2.4 Modified Key Expansion

In Proposed algorithm we are using hash based key expansion algorithm. First we compute hash code for the key used in AES algorithm. Then hash code combined with key that will be compressed into 16 bytes in length. We are using MD5 hash method for hash code creation. It will provide strong encryption for our proposed system. Then hash based 16 byte key converted into words. Using the below function, successive words are generated for add round key used in each round.

1. RotWord performs a one byte circular left shift on a word.
2. SubByte performs a byte substitution on each byte of its input word using s box
3. Result of above steps is XORed with a round constant, $Rcon[j]$.

3. DECRYPTION PROCESS

The decryption process takes place in exactly reverse order of the encryption process. When the encrypted data reaches the receiver, the receiver first XORs the data arranged in the 4×4 matrix with the key matrix. The result obtained is subjected to the following rounds:

3.1 Inverse MixColumns

This process works same as modified MixColumns but in exactly reverse way. The last row of key matrix is considered in this step. The operations are carried out on the data matrix in exactly the reverse order as that of encryption process.

3.2 Inverse ShiftRows

In this round, the bits of the third row of the key matrix are grouped as explained in Shift Rows step mentioned above. It should be noted that the bits are grouped according to the figure shown above. In this round the row pointed by bits in R and the row pointed by bits in P is cyclically shifted to right respectively. The remaining rows are then mirrored randomly in any order.

3.3 Inverse SubByte

In this round, the second row of the key matrix is used. The operations carried out in the Sub Bytes round as explained above, remains unchanged. The order of substitution is taken from inside to outside which means the coordinates are substituted in order of $M(g3, g4)$, $M(g2, g5)$, $M(g1, g6)$ and $M(g0, g7)$. We use inverse S-Box for substitution of the data in the matrix.

4. Implementation

The proposed changes to AES algorithm can be implemented in any programming language. Here we have implemented it using java language. The pseudo codes for the step modifications are given below.

4.1 SubBytes Step

```

modifiedSubstituteByte(byte key[][] , byte state[][])
{
  for(number i=0; i<2; i++)
  {
    byte tempkey[8];
    // this loop will get first four bits of selected block //of key
    matrix
    for(number j=0;j<4;j++)
    {
      // it will store first two bit tempkey[j*2]=first
      & two bit of key[i][j];
      // it will store next two bit tempkey[(j*2)+1]=third &
      fourth bit of key[i][j];
    }
    for(number j=0;j<4;j++)
    {
      substitute (tempkey[j],tempkey[7-j]) byte of state matrix using
      S-Box ;
    }
  }
}

```

4.2 ShiftRows Step

```

modifiedShiftRows(byte key[][] , byte state[][])

```

```

{
byte tempkey[4]; number
count[2]; for(number
i=0;i<2;i++)
{
temp[i*2]=XOR(first four bits of key[2][0],first four bits of
key[2][2]);
temp[(i*2)+1]=XOR(next four bits of key[2][0],next four
bits of key[2][2]);
count[i]=countNumberOfOnes(temp[(i*2)+1]);
for(number i=0;i<2;i++)
{
number row=findRow(temp[i*2]);
cyclicLeftShift(row , count[i]);
}
number remainingRows[]=findRemainingRows();
doMirrorRow(remainingRows);
}
}

```

4.3 MixColumns Step

```

modifiedMixColumn(byte key[][] , byte state[][])
{
byte tempkey[8];
for(number j=0;j<4;j++)
{
tempkey[j*2]=first four bits of key[3][j];
tempkey[(j*2)+1]=remaining four bits of key[3][j];
}
for(number i=0;i<4;i++)
{
byte temp=tempkey[i],tempkey[7-i];
mix the (temp%4) column of state matrix with the
randomly generated polynomial
}
}

```

4.4 Key Expansion Step

```

Modified keyexpansion (byte key[16],word w[44])
{
h=H(key);
Append(hash code,key)
Follow key expansion procedure
Word t
For (i=0;i<4;i++)
W[i]=(key[4*i],key[4*i+1],key[4*i+2], key[4*i+3]);
For(i=4;i<44;i++)
{
t=w[i-1];
if(i mod 4 =0)
t= subword(rotword(t))XOR Rcon[i/4];
w[i]=w[i-4] ~t
}
}

```

5. STRENGTH

Even though the modifications are performed on the original AES algorithm, the security of the original algorithm remains intact. The cipher key in AES is of 128 bits. Therefore to break the cipher key it requires 2¹²⁸ possibilities and tests to be carried out. This is theoretically almost impossible. Therefore, the Brute-force Attack fails on the AES algorithm.

The modifications made to the existing AES have made sure that there is no fixed pattern in any of the steps of

the algorithm. The modifications have provided the algorithm with strong diffusion and confusion. Therefore, statistical analysis of the ciphertext fails. The most important security advantage is that no differential or linear attacks on AES have been able to break the algorithm.

6. CONCLUSION

The modifications proposed by us can be implemented without increasing the size of the key block. Though the original algorithm is very secure the proposed changes in the processing of the algorithm and key expansion will help to encrypt the data by making stronger diffusion and confusion. It also increases the complexity of the algorithm multiple times. These proposed changes once implemented will make it very difficult to decrypt the ciphertext without proper decryption key.

FutureWork

This algorithm is more secure than Data Encryption Standard (DES) because the key is used in the beginning of algorithm unlike it is in DES at a later stage and the number of rounds in DES is 16 whereas it is less in our algorithm. Also our algorithm contains less number of operations as compared to DES algorithm and hence it will take less time for encryption and decryption as compared to DES algorithm. Our algorithm contains less numbers of rounds where as it is 10 in Advanced Encryption Standard (AES) and in this regard it is better than that of AES algorithm and takes less time. Also our algorithm contains more number of non linear functions (confusions or substitutions) than that of AES algorithm, which builds up the better security aspect of cryptography. Because use of non linear functions (substitutions or confusions) have better security than that of linear functions (Permutations or diffusions). We have planned to develop a new substitute box (S-Box) which will satisfy all the cryptographic properties. Also we have planned to apply hybrid cellular automata rules for block cipher.

REFERENCES

- [1] J. Daemen and V. Rijmen, AES Proposal: Rijndael, NIST's AES home page, <http://www.nist.gov/aes>.
- [2] "Announcing the Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197, November 2001
- [3] Xinmiao Zhang and Keshab K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm", 1531-636X/12, IEEE 2002.
- [4] Dr (Mrs) Pushpa R. Suri and Sukhvinder Singh Deora, "Design of a modified Rijndael algorithm using 2D Rotations", IJCSNS International Journal of Computer

Science and Network Security, VOL.11 No.9, September 2011.

- [5] Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
- [6] 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.
- [7] 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
- [8] Debasis Das and Abhishek Ray, "A Parallel Encryption Algorithm for Block Ciphers Based on Reversible Programmable Cellular Automata". Journal of Computer Science and Engineering, Volume 1, Issue 1, May 2010, pages: 82-90.



Mrs V.Sumathy received B.E in Computer Science and Engineering Specialization in April 2002. She was awarded Honor in M.E CSE in the April 2010. She is a Life Member in professional Bodies like ISTE. She is working as Asst. Professor in the Department of CSE in Kingston Engineering College, Vellore-

TamilNadu. Her research interests are in the areas of Networks & Network Security.



Mr.C.Navaneethan received B.E in Computer Science and Engineering Specialization in April 2004. He was awarded Honor in M.E CSE in the July 2006. Currently, he is pursuing Doctoral in Computer Science and Engineering Discipline. He is a Life Member in professional Bodies like IAENG,

IACSIT, and CSTA. He is working as Asst. Professor in the Department of CSE in Kingston Engineering College, Vellore-TamilNadu. His research interests are in the areas of Wireless Sensor Networks & Network Security.