Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 15 Issue 6 Version 1.0 Year 2015
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Robust Performance and Resistance to Attack for the Advanced Encryption Standard using Dynamic Rotation

By Dr. Mohamed Abd Elhamid Ibr & Moustafa Mousa El Bahtity

Cairo University, Egypt

Abstract- Recently, the Rijndael algorithm has been uniform by the National Institute of Standards and Technology (NIST) as the Advanced Encryption Standard (AES). This makes AES a vital and necessary data-protection mechanism for federal agencies in the US and other countries. In AES, rotation occurs in key expansion, ciphering, and deciphering. Rotation is vital for confusion and diffusion, which play an important role in any cryptography technique. Confusion and diffusion make breaking the key complex and difficult. This paper studies the effect of reconfiguring the structure of AES, especially replacing constant rotation with variable rotation. The resulting producing another cipher is called Dynamic Rotation for Advanced Encryption Standard (DRAES). DRAES with variable rotation raises the complexity of the algorithm, and thus, increases the time consumed for brute-force attacks. We measured the diffusion of AES and DRAES algorithms. DRAES reached acceptable level of diffusion faster than AES.

Keywords: *AES, DRAES, confusion and diffusion.*

GJCST-E Classification : *E.3.*

ROBUSTPERFORMANCEANDRESISTANCETOATTACKFORTHEADVANCEDENCRYPTIONSTANDARDUSINGDYNAMICROTATION

Strictly as per the compliance and regulations of:

Open Association of Research Society, USA
Research | Diversity | Ethics

© 2015. Dr. Mohamed Abd Elhamid Ibr & Moustafa Mousa El Bahtity. This is a research/review paper, distributed under the terms of the Creative Commons Attribution-Noncommercial 3.0 Unported License http://creativecommons.org/licenses/by-nc/3.0/), permitting all non-commercial use, distribution, and reproduction inany medium, provided the original work is properly cited.

Robust Performance and Resistance to Attack for the Advanced Encryption Standard using Dynamic Rotation

Dr. Mohamed Abd Elhamid Ibr[α] & Moustafa Mousa El Bahtity[σ]

Abstract- Recently, the Rijndael algorithm has been uniform by the National Institute of Standards and Technology (NIST) as the Advanced Encryption Standard (AES). This makes AES a vital and necessary data-protection mechanism for federal agencies in the US and other countries. In AES, rotation occurs in key expansion, ciphering, and deciphering. Rotation is vital for confusion and diffusion, which play an important role in any cryptography technique. Confusion and diffusion make breaking the key complex and difficult. This paper studies the effect of reconfiguring the structure of AES, especially replacing constant rotation with variable rotation. The resulting producing another cipher is called Dynamic Rotation for Advanced Encryption Standard (DRAES). DRAES with variable rotation raises the complexity of the algorithm, and thus, increases the time consumed for brute-force attacks. We measured the diffusion of AES and DRAES algorithms. DRAES reached acceptable level of diffusion faster than AES.

Keywords: AES, DRAES, confusion and diffusion.

I. Introduction

The National Institute of Standards and Technology (NIST), a non-regulatory federal agency, standardized the Advanced Encryption Standard (AES) as Federal Information Processing Standard (FIPS) 197. Prior to AES, the Data Encryption Standard (DES) was the federal standard for block symmetric encryption FIPS 46 in 197 [7].In June 2003 the US government has approved the use of 128, 192, 256 bit key AES for secret and 192, 256-bit key AES for top-secret information.

Now, after the publication of FIPS 197, AES encryption remains the de facto standard for symmetric encryption, and non-brute-force attacks remain impossible [1, 2], at least for the foreseeable future. To date, most attack methods have focused on weaknesses or characteristics in specific implementations, called side-channel attacks, not on the algorithm itself. However, AES has been remarkably resilient to these attacks [3-6].

In the last ten years, AES has been subject to very intensive cryptanalysis, with best currently known attacks breaking 7, 10, 10 rounds for respective key sizes 128, 192, 256,with very high complexities. In this work, we propose Dynamic Rotation AES (DRAES), a modification and enhancement of the rotation in AES.

The following section contains the evaluation of AES with constant rotation. Dynamic rotation with DRAES is presented in Section III. Diffusion analysis is assessed for both AES and DRAES algorithms in Section IV. Finally, Section V contains conclusions.

II. Evaluation of Advanced Encryption Standard

On the inside ofthe AES algorithm, processes are executed on a two-dimensional array of bytes called the state. The state consists of four rows of bytes, each containing *Nb* bytes, where *Nb* is the block length divided by word size (32 bits). *Nb=4* for 128-bit block, *Nb=6* for 192-bit block, *Nb=5* for 160-bit block, and *Nb=8* for 256-bit block.

The number of words in the key is called *Nk.* Ciphering is done by a series of mathematical operations iteratively. The number of rounds (iterations) is represented by *Nr*, where *Nr* =10 when *Nk* = 4, *Nr* = 12 when *Nk* = 6, and *Nr* = 14 when *Nk* = 8. In other words, the key length and the number of rounds differ from key size to key size as shown in Table 1. A block size of 128 bits is assumed. The components of the AES encryption algorithm are described next.

Table 1 : Common AES variants and their features

Algorithm	Key length (Nk words)	Block Size (Nb words)	Number of rounds (Nr)
Aes-128-bit	4	4	10
Aes-160-bit	5	4	11
Aes-192-bit	6	4	12
Aes-256-bit	8	4	14

Author: e-mail: mohbahtity@gmail.com

a) Sub Bytes Transform

In the Sub Bytes phase, the data in the plaintext are substituted by some pre-defined values from a substitution box. The substitution box, which is used commonly, is an AES substitution box (S-box table). Figure 1 demonstrates that the substitution box (S-box) is invertible and non-linear. Sub Bytes are the only nonlinear operation in AES. Nonlinearity is important for any encryption algorithm.

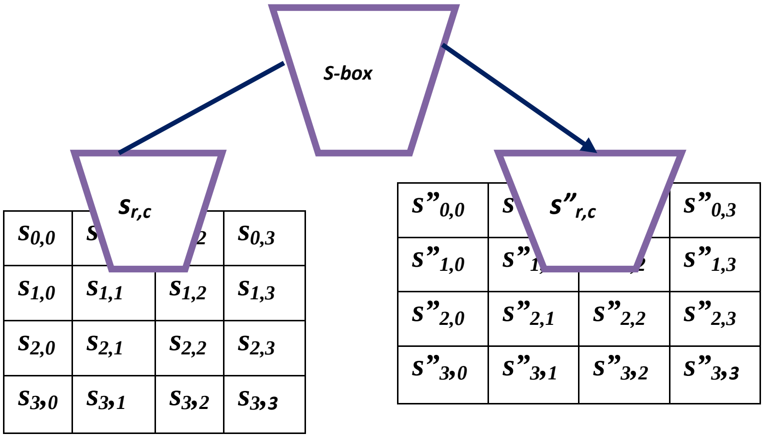

S-box
s_{r,c}
s_{0,0}
s_{0,3}
s_{1,0} s_{1,1} s_{1,2} s_{1,3}
s_{2,0} s_{2,1} s_{2,2} s_{2,3}
s_{3,0} s_{3,1} s_{3,2} s_{3,3}
s"_{r,c}
s"_{0,0}
s"_{0,3}
s"_{1,0}
s"_{1,3}
s"_{2,0} s"_{2,1} s"_{2,2} s"_{2,3}
s"_{3,0} s"_{3,1} s"_{3,2} s"_{3,3}

Figure 1 : Sub Bytes () applies the S-box to each byte of the State

b) Shift row Transform

In the Shift row transformation, the bytes in the last three rows of the State are cyclically shifted with different numbers of offsets (measured in bytes). The first row, Row 0, is not shifted.

Specifically, the Shift Rows transformation proceeds as follows:

$$S'_{r,\,c} = S_{r,\,(c+shift\,(r,\,Nb))\bmod Nb}\ ,\ \text{for } 0 < r < 4 \text{ and } 0 \le c < Nb \quad (1)$$

Where the shift value shift(r, Nb) depends on row number r, as follows:

$$shift\,(1,4) = 1;\ shift\,(2,4) = 2\ ;\ shift\,(3,4) = 3 \quad (2)$$

Figure 2 illustrates the Shift Rows transformation.

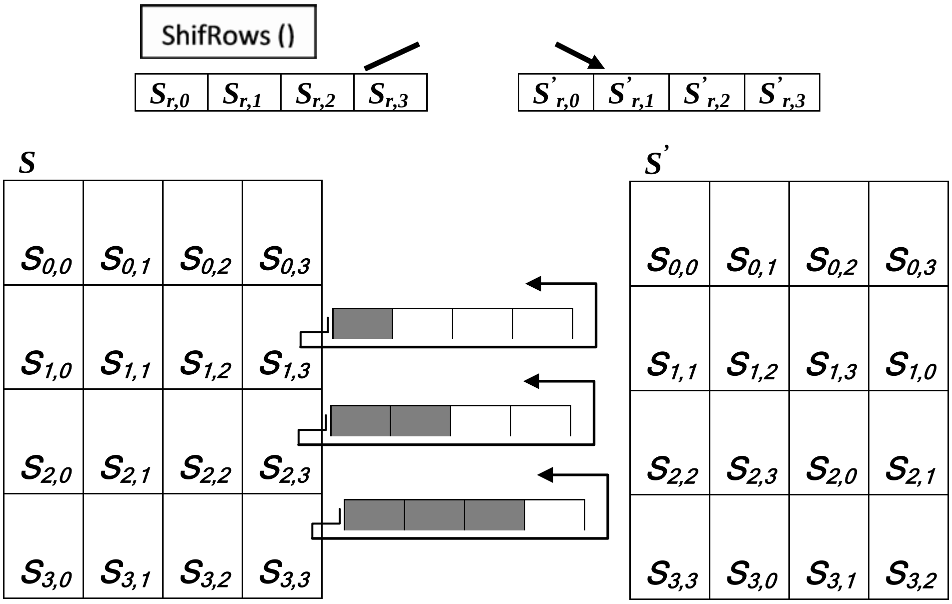

ShifRows ()
S_{r,0} S_{r,1} S_{r,2} S_{r,3}
S'_{r,0} S'_{r,1} S'_{r,2} S'_{r,3}
S
S_{0,0} S_{0,1} S_{0,2} S_{0,3}
S_{1,0} S_{1,1} S_{1,2} S_{1,3}
S_{2,0} S_{2,1} S_{2,2} S_{2,3}
S_{3,0} S_{3,1} S_{3,2} S_{3,3}
S'
S_{0,0} S_{0,1} S_{0,2} S_{0,3}
S_{1,1} S_{1,2} S_{1,3} S_{1,0}
S_{2,2} S_{2,3} S_{2,0} S_{2,1}
S_{3,3} S_{3,0} S_{3,1} S_{3,2}

Figure 2 : Shift Rows cyclically shifts the last three rows in the State

© 2015 Global Journals Inc. (US)

c) *Mix Columns Transform*

The Mix Columns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (28) and are multiplied modulo x4 + 1 with a fixed polynomial c(x), given by

$$c(x) = c_0 + c_1x + c_2x^2 + c_3x^3 \qquad (3)$$

Where$c_0=0x02_, c_1=0x01_, c_2=0x02_, c_3=0x03.$ This can be written as matrix multiplication: b(x) = c(x) ⊗ a(x),

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

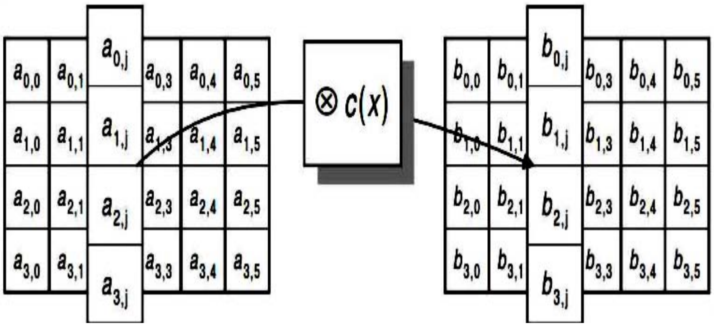


Figure 3 : Mix column operates on the columns of the State

d) *Add Round Key Transform*

Add Round Key transformation is as simple as possible and affects every bit of State.

The complexity of the round key expansion, plus the complexity of the other stages of AES, ensures security. Each Round Key consists of *Nb* words from the key schedule. Those *Nb* words are each added into the columns of the State, such that

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, S'_{3,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c}] \square \oplus [W_{round*Nb+c}] \text{ for } 0 \le c < Nb, (4)$$

Where $W_i$isa word from the key schedule, and round is a value in the range 0 ≤ *round* ≤ *Nr.* In the AES encryption, the initial Round Key addition occurs when round = 0, the application of the Add Round Key transformation to the *Nr* rounds of the Cipher occurs when 1≤ *round* ≤ *Nr.* The process of Add Round Key transformation is demonstrated in Figure4, andFigure5 illustrates the AES encryption and decryption processes.

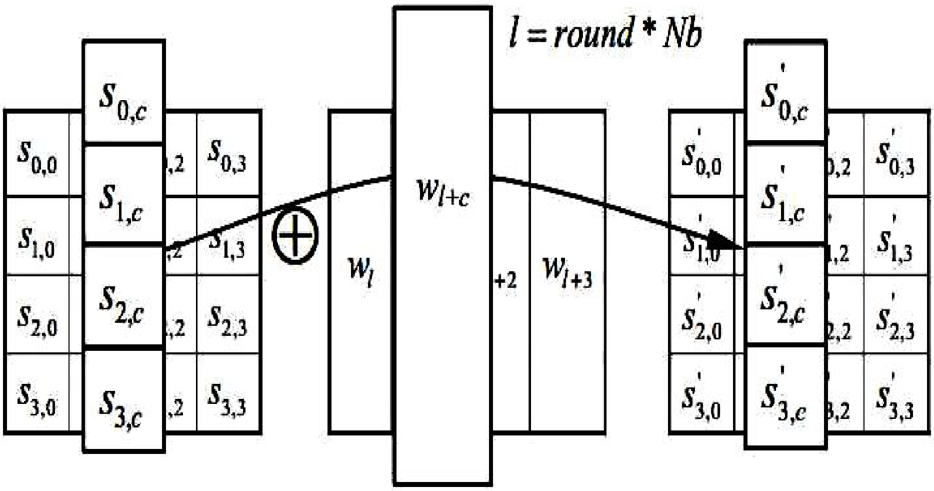


Figure 4 : Add Round Key XORs each column of state with a word from a key schedule

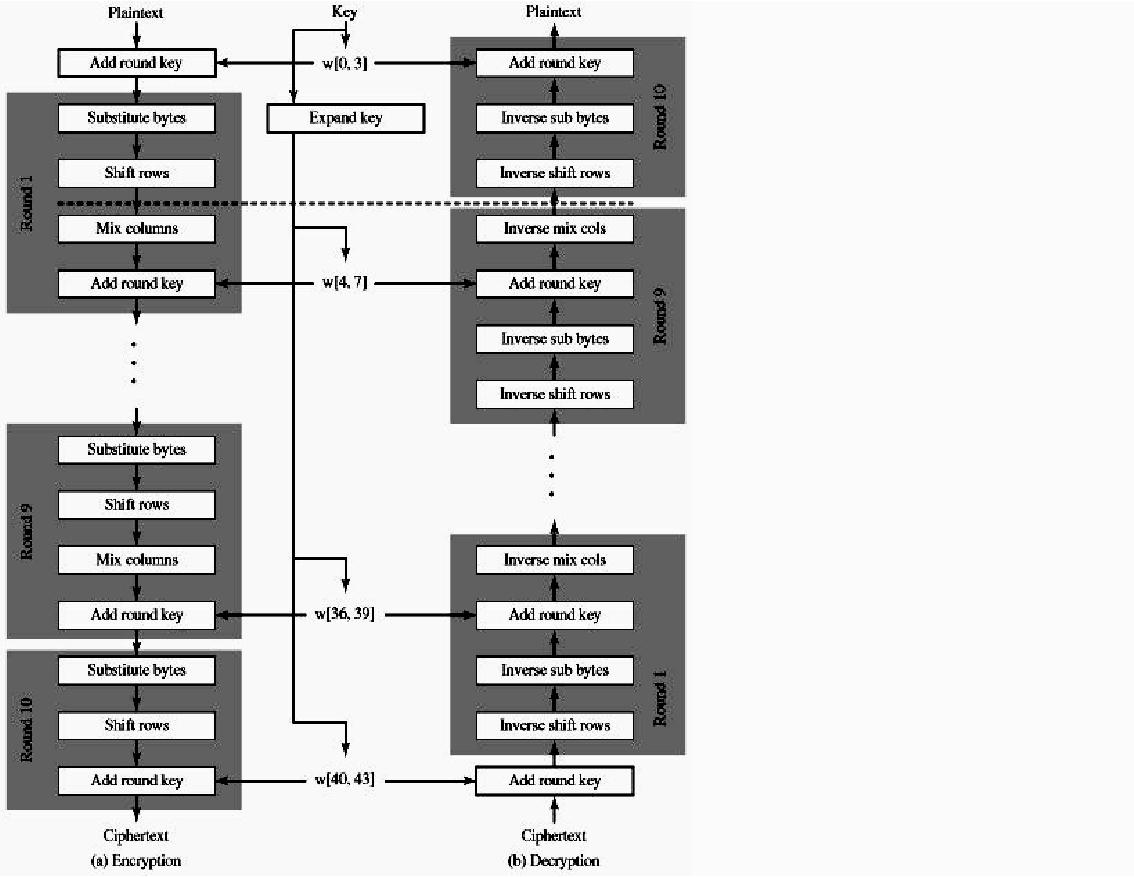


Figure 5 : Structure of AES encryption and decryption

e) *Key expansion*

Key expansion is part of the AES algorithm. It takes as input a 4-word (16-byte) key and produces a linear array of 44 words (176 bytes). This is sufficient to provide a 4-word round key for the initial Add Round Key stage and each of the 10 rounds of the cipher. The following pseudo code Figure 6 describes key expansion.

```
Constants: int Nb = 4;
Inputs:  int Nk = 4, 6, or 8; // the number of words in the key
array key of 4*Nk bytes or Nk words // input key
Output: array W of Nb*(Nr+1) words or 4*Nb*(Nr+1) bytes // expanded key
Algorithm:
void Key Expansion(byte[] key, word[] W, int Nk) {
int Nr = Nk + 6;
W = new byte[4*Nb*(Nr+1)];   int temp;   int i = 0;
while ( i < Nk) {
W[i] = word (key [4*i], key [4*i+1], key [4*i+2], key [4*i+3]);i++; }
i = Nk;
while(i < Nb*(Nr+1)) {     temp = W[i-1];
 if (i % Nk == 0)    temp = Sub Word (Rot Word(temp))
 ^ Rcon [i/Nk];
else if (Nk > 6 && (i%Nk) == 4)
temp = Sub Word(temp);
W[i] = W[i-Nk] ^ temp;      i++;}}
```

Figure 6 : Implementation of key expansion

The key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added word W [i] depends on the immediately preceding word, W [i-1], and the word four positions back [i -4]. In three out of four cases, a simple XOR is used. For a word whose position in the w array is a multiple of 4, a more complex function is used. Figure 7 illustrates the generation of the first eight words of the expanded key, using the symbol g to represent that complex function. The function g consists of the following sub functions:

- Rotation executes a one-byte circular left shift on a word. This means that an input word [b_0, b_1, b_2, b_3] is transformed into [b_1, b_2, b_3, b_0].
- SubWord achieves a byte substitution on each byte of its input word, using Sbox.
- The result of steps 1 and 2 is XORed with a Round constant (Rcon[j])

© 2015 Global Journals Inc. (US)

The round constant is a word in which the three rightmost bytes are always 0. The round constant is different for each round and is defined as Rcon[j] = (RC[j], 0, 0, 0), with RC [1] = 1, RC[j] = 2 • RC [j- 1] and with multiplication defined over the field GF (28). The values of RC[j] in hexadecimal are listed in Table 2

Table 2 : Round Constant Values

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

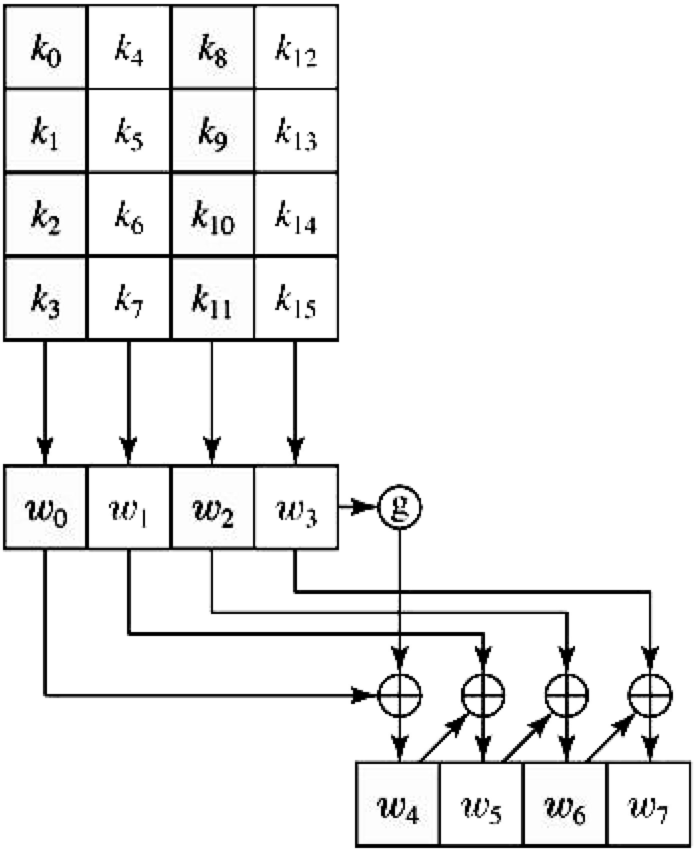


Figure 7 : AES Key Expansion

III. Dynamic Rotation AES (DRAES)

The main purpose of rotation is to mix all data elements in different columns of state. As such, rotation is important for confusion and diffusion [8], which both plays an essential role in cryptography. Confusion refers to making the output dependent on the key. Ideally, every key bit influences every output bit.

Diffusion is making the output dependent on previous input (plain and cipher ext). Ideally, every previous input bit influences each output bit. One aim of confusion is to make it very hard to find the key even if one has a large number of plain text-ciphertext pairs produced with the same key. Therefore, each bit of the ciphertext should depend on the entire key and in different ways on different bits of the key.

Rotation inAES is used in encryption and decryption as summarized in Figure 8.The encryption includes rotationin key expansion (cyclic shift-left of the round key bytes) and Add Round Key(shift-left of state rows). The decryption includes inverse of the Add Round Key (shift-right of state rows).

© 2015 Global Journals Inc. (US)

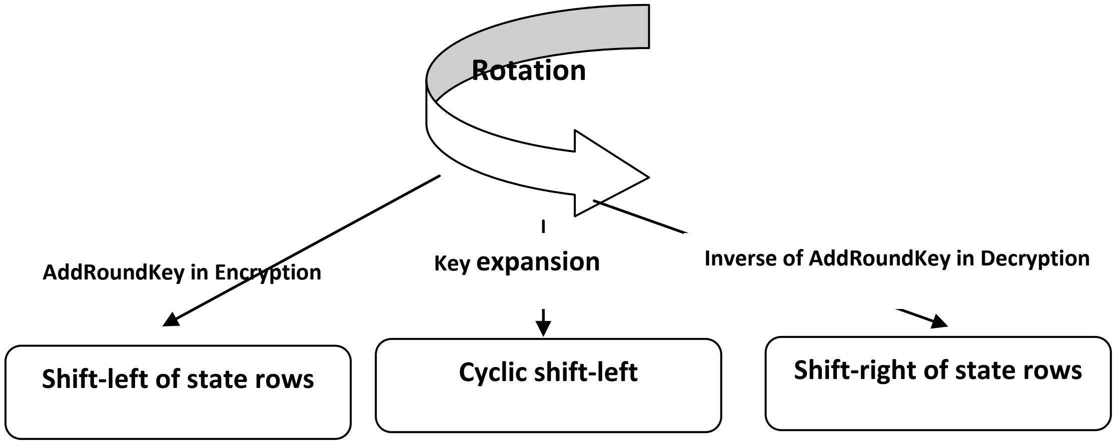

Rotation
AddRoundKey in Encryption
Key expansion
Inverse of AddRoundKey in Decryption
Shift-left of state rows
Cyclic shift-left
Shift-right of state rows

Figure 8 : Rotation in key expansion, encryption, and decryption in AES

The standard rotation rules in AES are simple, and attackers get high probability to break AES system by cryptanalysis. For example, the rotation amount (number of shifts) in AES is constant and key-independent. This rotation can be predicted and makes AES potentially vulnerable to cryptanalysis. In this work, we propose Dynamic Rotation for Advanced encryption Standard (DRAES), a modification and enhancement of the rotation in AES as follows. The proposed rotation enhancement makes the rotation amount dependent on data (plaintext and ciphertext) in Add Round Key and on the key in key expansion. When rotation occurs the round key and intermediate data are modified, and either the key value or the intermediate data affect the rotation amount.

a) Key expansion rotation in DRAES

The following algorithm describes the proposed modification in the rotation inside the key expansion.

Constants: int Nb = 4;

Inputs: int Nk = 4, 6, or 8; // the number of words in the key

array key of 4*Nk bytes or Nk words // input key

Output: array W of Nb*(Nr+1) words or 4*Nb*(Nr+1) bytes // expanded key

Algorithm:

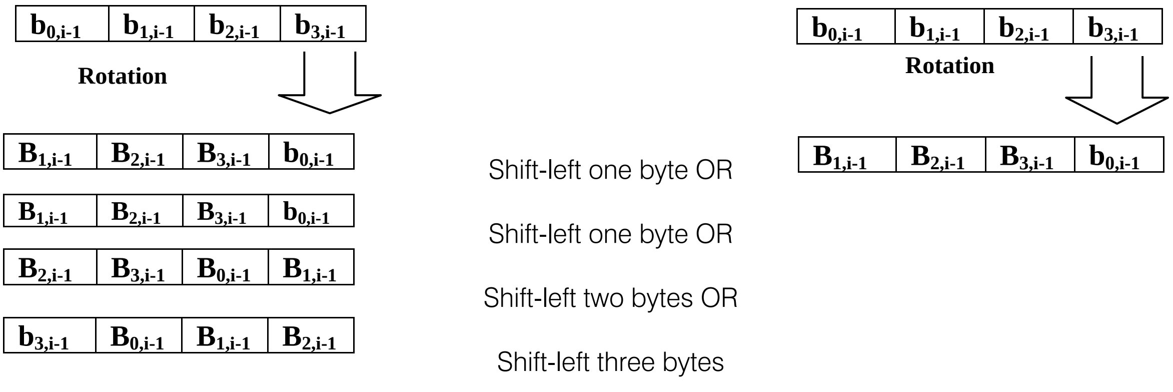
```
int Nr = Nk + 6;
W = new byte[4*Nb*(Nr+1)];
int temp;   int i = 0;
while ( i < Nk) {
W[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3]);
i++; }
i=Nk
While (i<Nb *(Nr+1))
if (i mod Nk == 0)
W[i-1] = (b0,i-1b1,i-1b2,i-1b3,i-1)
temp= ( b0,i-1 □ b1,-i1 □ b2,-i1 □ b3,-i1 )
if (temp mod Nk == 0)
W[i-1]=(b1,i-1b2,i-1b3,i-1b0,i-1) % shift left by onebyte to enforce rotation
else if (temp mod Nk == 1)
W[i-1] =(b1,i-1b2,i-1b3,i-1b0,i-1)  % shift left by one byte
else if (temp mod Nk==2)
W[i-1] =(b2,i-1b3,i-1b0,i-1b1,i-1)  % shift left by two bytes
else (temp mod 4 =3)
W[i-1] =(b3,i-1b0,i-1b1,2b1,i-1)  % shift left by three bytes
End if
temp = SubWord (W[i-1])) ^ Rcon[i/Nk];
else if (Nk > 6 && (i%Nk) == 4)
temp = SubWord(temp);
W[i] = W[i-Nk] ^ temp;     i++;     } }
End while
```

The Rot Word rotation in key expansion occurs 10 times in DRAES similar to AES for key length of 128 bits (Nk= 4). Table 3 and Figure 8 show a comparison between AES and DRAES.

© 2015 Global Journals Inc. (US)

Table 3 : Comparison between key expansion in DRAES and AES

	DRAES	AES
Number of rotations	10	10
Maximumrotation amountper round	3	1
Variable rotation amount	Yes	No
Confusion and diffusion	High	Low
Predictable	Weak	Rise


$b_{0,i-1}$ $b_{1,i-1}$ $b_{2,i-1}$ $b_{3,i-1}$
Rotation
$B_{1,i-1}$ $B_{2,i-1}$ $B_{3,i-1}$ $b_{0,i-1}$
$B_{1,i-1}$ $B_{2,i-1}$ $B_{3,i-1}$ $b_{0,i-1}$
$B_{2,i-1}$ $B_{3,i-1}$ $B_{0,i-1}$ $B_{1,i-1}$
$b_{3,i-1}$ $B_{0,i-1}$ $B_{1,i-1}$ $B_{2,i-1}$
Shift-left one byte OR
Shift-left one byte OR
Shift-left two bytes OR
Shift-left three bytes
$b_{0,i-1}$ $b_{1,i-1}$ $b_{2,i-1}$ $b_{3,i-1}$
Rotation
$B_{1,i-1}$ $B_{2,i-1}$ $B_{3,i-1}$ $b_{0,i-1}$

(a) Rotation in DRAES (b) Rotation in AES

Figure 8 : (a),(b).Rotation in key expansion for DRAES and AES

b) Add Round Key rotation in DRAES

The modification of rotation in the ciphering process is vital; the change from constant shift-row to variable shift-row make the rotation amount hard to guess, which increases confusion and diffusion. In AES, row 0 is not shifted, row 1 is shifted 1 byte, row 2 is shifted 2 bytes, and row 3 is shifted 3 bytes. In DRAES, rotationamount is variable and done with the following procedure.

```
1. Nb=4 // the number of columns is denoted by Nb and is equal to the block length divided By 32
2. State [4, Nb] //the State can be pictured as array of bytes. This array has four Rows, the number of columns is denoted by Nb and is equal to the block length divided by 32
3. NR=10 //number of round for key 128 bit, NR=12 for key 192 bit and NR=14 for key 256 bit
4. Add Round Key(state, Round key)
5. For round = 1 step 1 to Nr–1
6. Sub Bytes (state, s_box)
7. Shift Rows (state) // the rotation for each row individually in state
     I. read each row in state
    II. Sum the elements in each row in Temp using Xor
   III. If (Temp mod Nb=0)
    IV. Shift left by one-byte to enforce rotation
     V. else if (Temp mod Nb =1)
    VI. Shift left by one byte
   VII. Else if (Temp mod Nb =2)
  VIII. Shift left by Two byte
    IX. Else (Temp mod Nb =3)
     X. Shift left by three bytes
    XI. // end if and end of ShiftRows (state)
8. MixColumns (state)
9. AddRoundKey (state, RoundKey)
10. // end for
11. Sub Bytes (state) // final round state
12. Shift Rows (state)
   XII. read each row in state
  XIII. Sum the elements in each row in Temp using Xor
   XIV. If (Temp mod Nb =0)
    XV. Shift left by one byte // to enforce rotation
   XVI. else if (Temp mod Nb =1)
  XVII. Shift left by one byte
 XVIII. Else if (Temp mod Nb =2)
```

```
  XIX.   Shift left by Two byte
  XX.    Else (Temp mod Nb =3)
  XXI.   Shift left by three bytes
  XXII.  // end if and end of Shift Rows (state)
13. Add Round Key (State, Round Key);
14. End cipher
```

This figure 9 and 10 illustrate the Rotation in AES and DRAES

c) DRAES in inverse cipher

The rotation in inverse cipher is the same process for the DRAES In cipher that described in sec. b Except for the shift row instead of shift row left, the shift row is right. Table 4 explain the variation between DRAES and AES for cipher

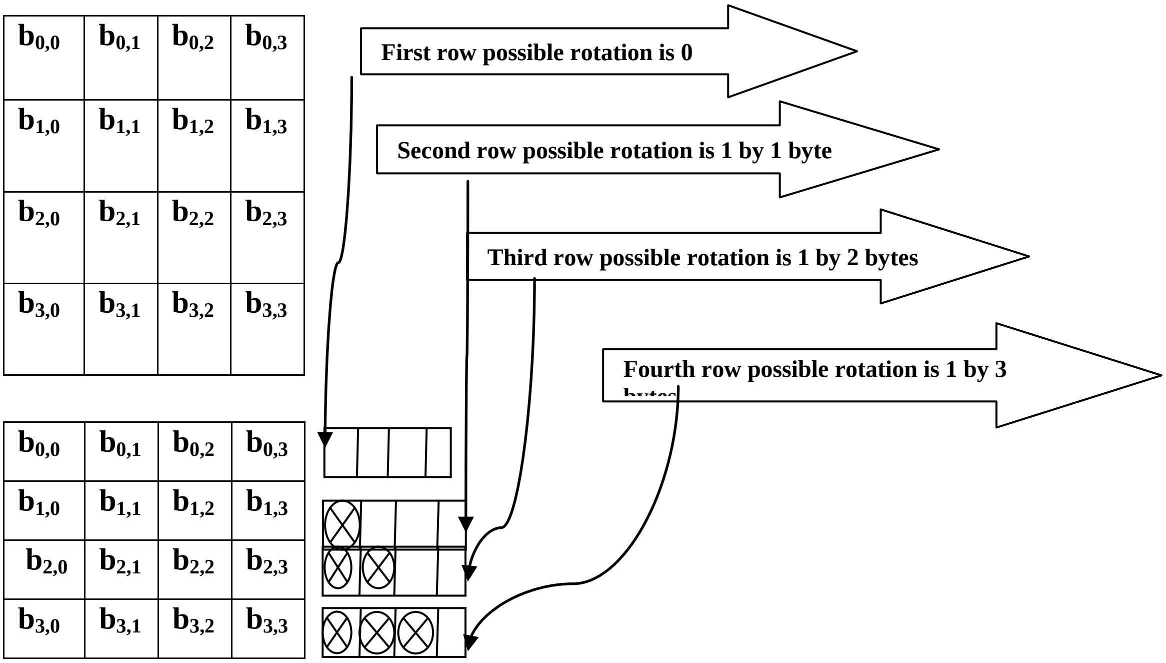

$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
First row possible rotation is 0
Second row possible rotation is 1 by 1 byte
Third row possible rotation is 1 by 2 bytes
Fourth row possible rotation is 1 by 3 bytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

Figure 9 : Possible Rotation (shift row left) states in (AES)

Table 4 : Demonstrate the conclusion between DRAES and AES for cipher

	DRAES	AES
number of row rotation for state in one round	4	3
number of row rotation for state in all round	36	27
number of row rotation for state in cipher	40	30
possible rotation for row 0 in state	$b_{0,1}$ $b_{0,2}$ $b_{0,3}$ $b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$ $b_{0,0}$ $b_{0,2}$ $b_{0,3}$ $b_{0,0}$ $b_{0,1}$ $b_{0,3}$ $b_{0,0}$ $b_{0,1}$ $b_{0,2}$	$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$

possible rotation for row 1 in state	$b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,0}$ $b_{1,2}$ $b_{1,3}$ $b_{1,0}$ $b_{1,1}$ $b_{1,3}$ $b_{1,0}$ $b_{1,1}$ $b_{1,2}$	$b_{1,1}$ $b_{1,2}$ $b_{1,3}$ $b_{1,0}$
possible rotation for row 2 in state	$b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,0}$ $b_{2,2}$ $b_{2,3}$ $b_{2,0}$ $b_{2,1}$ $b_{2,3}$ $b_{2,0}$ $b_{2,1}$ $b_{2,2}$	$b_{2,1}$ $b_{2,2}$ $b_{2,3}$ $b_{2,0}$
possible rotation for row 3 in state	$b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,0}$ $b_{3,2}$ $b_{3,3}$ $b_{3,0}$ $b_{3,1}$ $b_{3,3}$ $b_{3,0}$ $b_{3,1}$ $b_{3,2}$	$b_{3,1}$ $b_{3,2}$ $b_{3,3}$ $b_{3,0}$
confusion and diffusion	high	Low
Predictable	hard	Easy

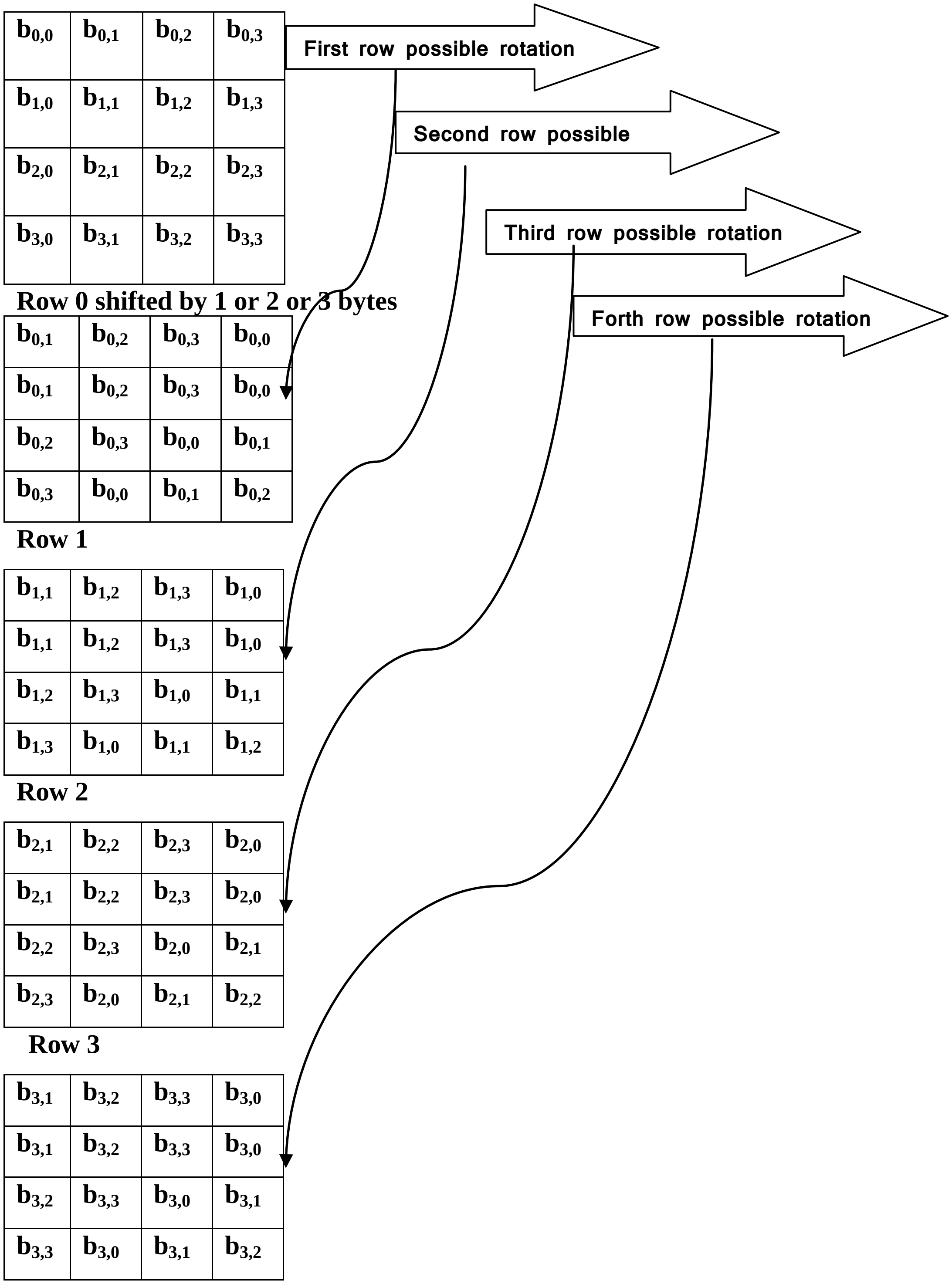
VI. DRAES with Confusion and Diffusion

A strong cipher should contain both Confusion and diffusion. Claude Shannon, develop this concepts [9]. Confusion and diffusion are two techniques that symmetric ciphers should satisfy to thwart cryptanalysis. In a block cipher with good diffusion, if one bit of the plaintext digit is changed, then affects many cipher text digits in a random mode. Cryptographic diffusion test is a kind of statistical test that evaluates a block cipher for diffusion. The performance analysis can be done with various measures such as Diffusion analysis of DRAES and AES

VII. Diffusion Analysis

Diffusion makes the ciphertext dependent on previous plaintext and ciphertext. Diffusion is important for any block cipher, more specifically AES and DRAES algorithms. The impact of diffusion can be measured by the Strict Avalanche Criterion (SAC) [10], which is satisfied when at least 50% of bits in the ciphertext are changed in response to a one-bit flip in the plaintext or key.

Table 5 shows the SAC for both DRAES and AES when changing a single bit of plaintext while keeping the key constant. Table 6 shows the SAC for both DRAES and AES when changing a single bit of key while keeping the plaintext constant. Table 7 shows the SAC for both DRAES and AES when changing 3 bits of plaintext while keeping the key constant. Table 8 shows the SAC for both DRAES and AES when changing 3 bits of key while keeping the plaintext constant.



$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

First row possible rotation
Second row possible
Third row possible rotation
Forth row possible rotation

Row 0 shifted by 1 or 2 or 3 bytes

$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,0}$
$b_{0,1}$	$b_{0,2}$	$b_{0,3}$	$b_{0,0}$
$b_{0,2}$	$b_{0,3}$	$b_{0,0}$	$b_{0,1}$
$b_{0,3}$	$b_{0,0}$	$b_{0,1}$	$b_{0,2}$

Row 1

$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,0}$
$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,0}$
$b_{1,2}$	$b_{1,3}$	$b_{1,0}$	$b_{1,1}$
$b_{1,3}$	$b_{1,0}$	$b_{1,1}$	$b_{1,2}$

Row 2

$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,0}$
$b_{2,1}$	$b_{2,2}$	$b_{2,3}$	$b_{2,0}$
$b_{2,2}$	$b_{2,3}$	$b_{2,0}$	$b_{2,1}$
$b_{2,3}$	$b_{2,0}$	$b_{2,1}$	$b_{2,2}$

Row 3

$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,0}$
$b_{3,1}$	$b_{3,2}$	$b_{3,3}$	$b_{3,0}$
$b_{3,2}$	$b_{3,3}$	$b_{3,0}$	$b_{3,1}$
$b_{3,3}$	$b_{3,0}$	$b_{3,1}$	$b_{3,2}$

Figure 10 : Possible rotation (shift row left) states in (DRAES)

Table 5 : (a) (b) Avalanche effect for DRAES and AES

(Change in Plaintext by 1 bit & Key constant)

Round	Number of bit altered		SAC
1	11	9%	N
2	50	40%	N
3	77	61%	Y
4	59	47%	N
5	60	47%	N
6	69	54%	Y
7	63	50%	Y
8	65	51%	Y
9	60	47%	N
10	66	52%	Y

(a) AES

Round	Number of bit altered		SAC
1	16	13%	N
2	72	57%	Y
3	61	48%	N
4	63	50%	Y
5	63	50%	Y
6	64	50%	Y
7	83	65%	Y
8	61	48%	N
9	63	50%	Y
10	60	47%	N

(b) DRAES

As shown in Table 5 at first round, 11 bits (AES) of cipher value have changed out of 128-bit cipher text. This results an Avalanche value of 9%. SAC is achieved at the end of third round and Avalanche values transforms around the SAC value for the remaining rounds. Similar the Avalanche effect for (DRAES), the first round 16 bit of 128-bit with Avalanche value is 13%. SAC is achieved at the end of second round and Avalanche values changes around the SAC value more rapidly in DRAES than AES

Table 6 : (a) (b) the Avalanche effect for DRAES and AES

Round	Number of bit altered		SAC
1	36	29%	N
2	63	50%	Y
3	66	52%	Y
4	55	43%	N
5	72	57%	N
6	65	51%	Y
7	54	43%	N
8	63	50%	Y
9	63	50%	Y
10	66	52%	Y

(a) AES

Round	Number of bit altered		SAC
1	64	50%	Y
2	60	47%	N
3	69	54%	Y
4	61	48%	N
5	67	53%	Y
6	58	46%	N
7	68	54%	Y
8	66	52%	Y
9	70	55%	Y
10	67	53%	Y

(b) DRAES

The conclusion result in Table 6demonstrate Avalanche effect SAC is achieved more rapidly in DRAES than AES in first round with SAC 50%, while the SAC for AES is completed in second round

Table 7 : (a) (b) details the Avalanche effect for DRAES and AES

(Varying 3 bits of plaintext & key constant)

Round	Number of bit altered		SAC
1	28	22%	N

Round	Number of bit altered		SAC
1	28	22%	N

2	65	51%	Y
3	62	49%	N
4	57	45%	N
5	61	48%	N
6	55	43%	N
7	58	46%	N
8	63	50%	Y
9	67	53%	Y
10	55	43%	N

(a) AES

2	70	55%	Y
3	57	45%	N
4	73	58%	Y
5	64	50%	Y
6	68	54%	Y
7	67	53%	Y
8	61	48%	N
9	63	50%	Y
10	68	54%	Y

(b) DRAES

The end result in Table7 display Avalanche effect SAC is achieved for DRAES and AES in same second round, but SAC 55% for DRAES is greater than SAC for AES.

Table 8 : (a) (b) illuminate the Avalanche effect for DRAES and AES

(Change in Key by 3 bits & Plaintext constant)

Round	Number of bit altered		SAC
1	58	58%	Y
2	57	57%	Y
3	66	66%	Y
4	61	61%	Y
5	56	56%	Y
6	61	61%	Y
7	55	55%	Y
8	63	63%	Y
9	65	65%	Y
10	57	57%	Y

(a) AES

Round	Number of bit altered		SAC
1	76	60%	Y
2	65	51%	Y
3	63	50%	Y
4	66	52%	Y
5	73	58%	Y
6	71	56%	Y
7	66	52%	Y
8	60	47%	N
9	63	50%	Y
10	63	50%	Y

(b) DRAES

The outcome in Table 8 present Avalanche effect SAC is achieved for DRAES and AES in same first round, but SAC 60% for DRAES is greater than SAC for AES and greater in number of bits are altered (666 bits) than AES (599 bits).

V. Conclusion

With Dynamic rotation for advanced encryption standard DRAES the confusion and diffusion is stronger than rotation that occur in AES, that mean that Rijndael is more safe and physically powerful with dynamic rotation when compared to Rijndael with constant rotation as shown from results from tables that related with diffusion analysis.

References Références Referencias

1. Bernstein, Daniel J. "Understanding brute force." In *Workshop Record of ECRYPT STVL Workshop on Symmetric Key Encryption, eSTREAM report*, vol. 36, 2005.
2. Kumar, Neeraj. "Investigations in Brute Force Attack on Cellular Security Based on Des and Aes." IJCEM International Journal of Computational Engineering & Management 14 (2011).
3. Biryukov, Alex, Dmitry Khovratovich, and Ivica Nikolic. "Distinguisher and Related-Key Attack on the Full AES-256." Advances in Cryptology-CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009, Proceedings. Vol. 5677. Springer, 2009.

© 2015 Global Journals Inc. (US)

4. Barkan, E., Biham, E.: In How Many Ways Can You Write Rijndael. In Zheng, Y., ed.: Advances in Cryptology – ASIACRYPT 2002: 8th International Conference on Theory and Application of Cryptology and Information Security. Volume 2501 of LNCS., Springer-Verlag (2002) 160–175.
5. Krishnamurthy, G. N., and V. Ramaswamy. "Making AES stronger: AES with key dependent S-box." *IJCSNS International Journal of Computer Science and Network Security* 8, no. 9 (2008): 388-398.
6. Lu, Jiqiang, Orr Dunkelman, Nathan Keller, and Jongsung Kim. "New impossible differential attacks on AES." In Progress in Cryptology-INDOCRYPT 2008, pp. 279-293. Springer Berlin Heidelberg, 2008.
7. Serge Vaudenay, "A CLASSICAL INTRODUCTION TO MODERN CRYPTOGRAPHY", Springer Science+Business Media, Inc.,2006, ISBN-13: 978-0-387-25464-7.
8. G. Lokeshwari, Dr. S. Udaya Kumar and G. Aparna "A CONFIGURABLE SECURED IMAGE ENCRYPTION TECHNIQUE USING 3D ARRAY BLOCK ROTATION", International Journal of Engineering Science and Technology (IJEST), Vol. 4 No.01 January 2012
9. Drakakis, Konstantinos, Verónica Requena, and Gary McGuire. "On the Nonlinearity of Exponential Welch Costas Functions." IEEE TRANSACTIONS ON INFORMATION THEORY 56, no. 3 (2010).
10. Mohan H. S. and A Raji Reddy , "Performance Analysis of AES and MARS Encryption Algorithms", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011.

© 2015 Global Journals Inc. (US)

Global Journals Inc. (US) Guidelines Handbook 2015

www.GlobalJournals.org