International Journal of Reviews in Computing
31st December 2012. Vol. 12

IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

PERFORMANCE EXAMINATION OF AES ENCRYPTION ALGORITHM WITH CONSTANT AND DYNAMIC ROTATION

[1]I.A. ISMIL, [2]GALAL H. GALAL -EDEEN, [2]SHERIF KHATTAB AND [2]MOHAMED ABD ELHAMID I. MOUSTAFA El BAHTITY

[1] Emeritus dean of faculty of computers and information, Zagazig University
[2] Faculty of computers and information, Cairo University

E-mail mohbahtity@gmail.com

ABTRACT

Lately, the Rijndael algorithm has been standardized by the NIST as the Advanced Encryption Standard (AES). This makes AES an essential 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 twin cipher is called Dynamic Rotation for Advanced Encryption Standard (DRAES). DRAES with variable rotation increases 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, Diffusion*

1. 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 1977 [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.

2. EVALUATION OF ADVANCED ENCRYPTION STANDARD

On the inside of the 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 number of rounds differ from key size to key size

International Journal of Reviews in Computing
31st December 2012. Vol. 12
IJRIC

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

as shown in Table 1. A block size of 128 bits is assumed. The components of the AES encryption algorithm is described next.

Table 1. Common AES Variants And Their Features

	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

2.1 SubBytes Transform

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

2.2 Shiftrow Transform:

In the **Shiftrow** 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 **ShiftRows ()** transformation proceeds as follows:

$$s''_{r,c} = s_{r,(c+\ shift\ (r,\ Nb))\ \mathrm{mod}\ \mathrm{Nb}}\ ,$$

for $0 < r < 4$ and $0 \le c < Nb$, (1)

Where the shift value shift(r, Nb) depends on row number r, as follow (128 bits is Nb = 4):

shift (1,4) =1; *shift* (2,4) = 2 ;

shift (3,4) = 3 (2)

Figure **2** illustrates the ShiftRow transformation.

2.3 MixColumns Transform

The MixColumns transformation operates on the State column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over GF (2^8) and are multiplied modulo $x^4 + 1$ with a fixed polynomial $c(x)$, given by

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

This can be written as matrix multiplication: b(x)= c(x) ⊗ a(x),

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

2.4 AddRoundKey Transform

AddRoundKey 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

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

Where W_i is a word from the key schedule, and round is a value in the range $0 \le round \le Nr$.

In the AES encryption, the initial Round Key addition occurs when round = 0, the application of the AddRoundKey transformation to the *Nr* rounds of the Cipher occurs when $1 \le round \le Nr$. The process of Addroundkey transformation is demonstrated in Fig. 4, and Fig. 5 illustrates the AES encryption and decryption processes.

Figure 4 demonstrate AddRoundKey

International Journal of Reviews in Computing
31st December 2012. Vol. 12

IJRIC
ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

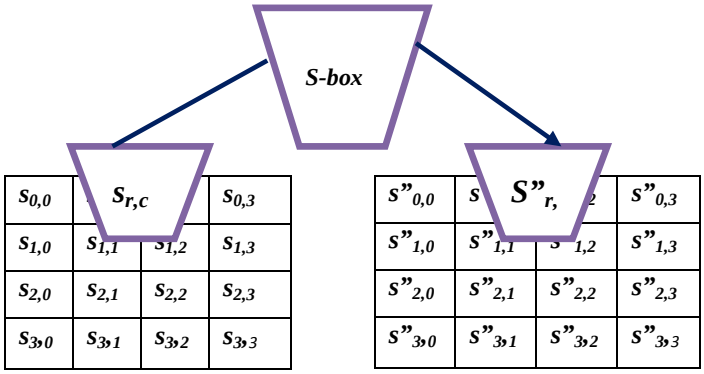

Figure 1 Subbytes () Applies The S-Box To Each Byte Of The State

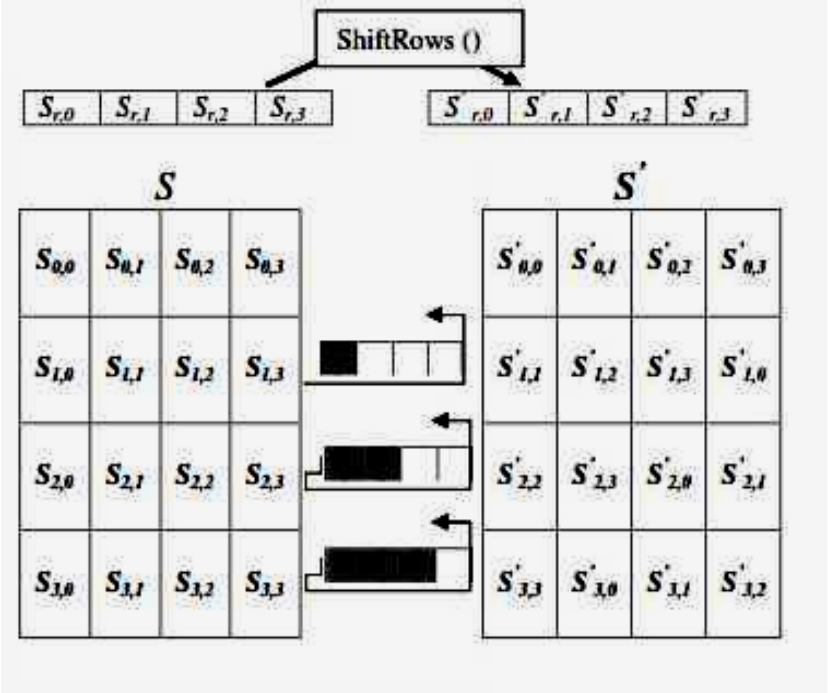

Figure 2 Shift Rows( ) Cyclically Shifts The Last Three Rows In The State

International Journal of Reviews in Computing
31st December 2012. Vol. 12

IJRIC
ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

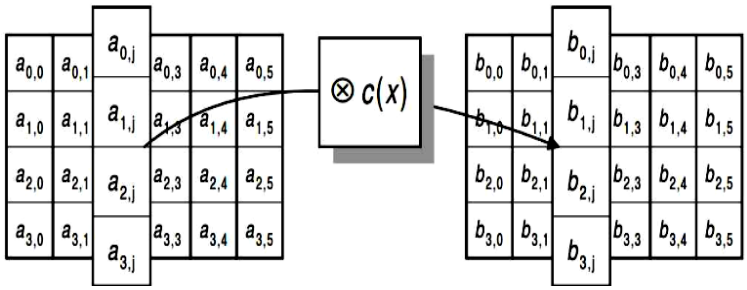

Figure3. Mixcolumn Operates On The Columns Of The State

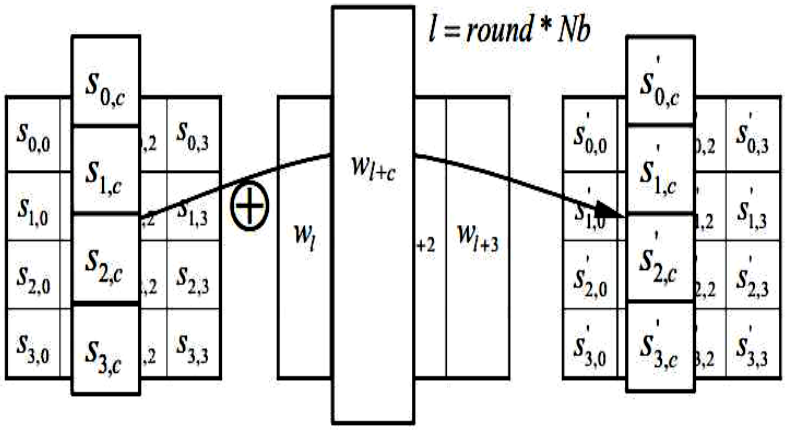

Figure 4 Addroundkey Xors Each Column Of The State With A Word From The Key Schedule

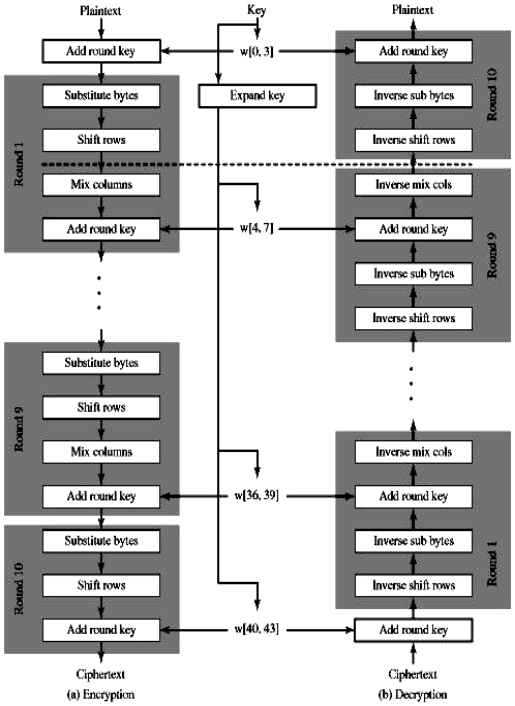

Figure 5 Structure Of AES Encryption And Encryption

International Journal of Reviews in Computing
31st December 2012. Vol. 12

IJRIC
ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

2.5 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 AddRoundKey stage and each of the 10 rounds of the cipher. The following pseudocode (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 KeyExpansion(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 = SubWord (RotWord(temp)) ^
     Rcon[i/Nk];
    else if (Nk > 6 && (i%Nk) == 4)
      temp = SubWord(temp);
    W[i] = W[i-Nk] ^ temp;       i++;       } }
```

Figure 6. Pseudocode implementation of key expansion.

3. DYNAMIC ROTATION WITH 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 play 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 ciphertext). 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 plaintext-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 in AES is used in encryption and decryption as summarized in Figure 7. The encryption includes rotation in **key expansion** (cyclic shift-left of the round key bytes) and **AddRoundKey** (shift-left of state rows). The decryption includes **inverse of the AddRoundKey** (shift-right of state rows).In this work, we modify and enhancement rotation on the advanced encryption standard (AES), in the AES the ByteSub is only nonlinear operation. Therefore, the nonlinearity SubBytes (s-box) is important for any encryption algorithms in the theoretical security.

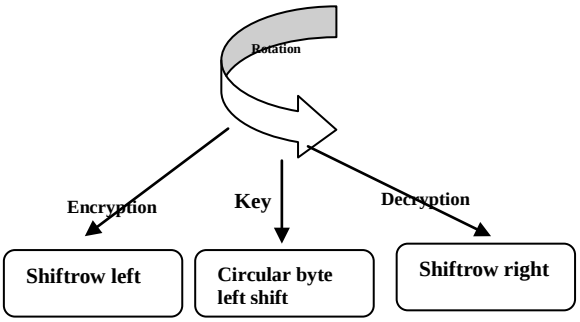

Figure 7 Rotation In Key Expansion, Encryption And Decryption

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

International Journal of Reviews in Computing
31st December 2012. Vol. 12

IJRIC
ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

enhancement makes the rotation amount dependent on data (plaintext and ciphertext) in AddRoundKey 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:

1. int Nr = Nk + 6;
2. W = new byte[4*Nb*(Nr+1)];
3. int temp; int i = 0;
4. while (i < Nk) {
5. W[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3]);
6. i++; }
7. i=Nk
8. While (i<Nb *(Nr+1))
9. if (i mod Nk == 0)
10. W[i-1] = ($b_{0,i-1}b_{1,i-1}b_{2,i-1}b_{3,i-1}$)
11. temp= ($b_{0,i-1} \oplus b_{1,i-1} \oplus b_{2,i-1} \oplus b_{3,i-1}$)
12. if (temp mod Nk == 0)
13. W[i-1] =($b_{1,i-1}b_{2,i-1}b_{3,i-1}b_{0,i-1}$) % shift left by one byte to enforce rotation
14. else if (temp mod Nk == 1)
15. W[i-1] = ($b_{1,i-1}b_{2,i-1}b_{3,i-1}b_{0,i-1}$) % shift left by one byte
16. else if (temp mod Nk ==2)
17. W[i-1] = ($b_{2,i-1}b_{3,i-1}b_{0,i-1}b_{1,i-1}$) % shift left by two bytes
18. else (temp mod 4 =3)
19. W[i-1] =($b_{3,i-1}b_{0,i-1}b_{1,2}b_{1,i-1}$) % shift left by three bytes
20. End if
21. temp = SubWord (W[i-1])) ^ Rcon[i/Nk];
22. else if (Nk > 6 && (i%Nk) == 4)
23. temp = SubWord(temp);
24. W[i] = W[i-Nk] ^ temp; i++; } }
25. End while

The RotWord rotation in key expansion occurs 10 times in DRAES similar to AES for key length of 128 bits (Nk= 4). Table 2 and Figure 8 show a comparison between AES and DRAES.

The rotation word in key expansion occurs 10 times in the Dynamic Rotation for Advanced encryption standard with (DRAES) similar to the advanced encryption standard (AES) for key length 128 bits. Table 2 and figure 6 shows the comparison between AES and DRAES

Table 2 Comparison Bet. Key Expansion In DRAES And AES.

	DRAES	AES
Number of rotations	10	10
Maximum rotation amount per round	3	1
Variable rotation amount	Yes	No
Confusion and diffusion	high	Low
predictable	weak	rise

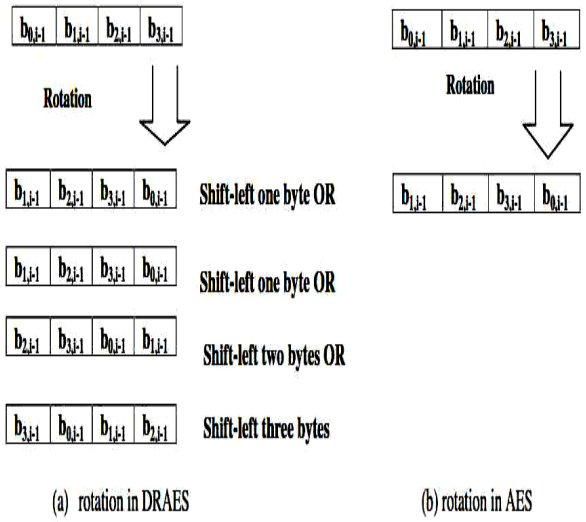

Figure 8 (a), (b). Rotation in key expansion for DRAES and AES

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

B) AddRoundKey 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, rotation amount is variable and done with the following procedure.

1. Nb=4 // the number of columns is denoted by Nb(4,6,8) and is equal to the block length divided By 32
2. State [4, Nb // the State can be pictured as a rectangular 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. AddRoundKey(state, Roundkey)
5. For round = 1 step 1 to Nr–1
6. SubBytes (state, s_box)
7. ShiftRows (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. SubBytes (state) // final round state
12. ShiftRows (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)
13. AddRoundKey (State, Roundkey);
14. End cipher

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 3 explain the variation between DRAES and AES for cipher

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

Table 3 Comparison Between DRAES & AES

	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 process	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

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

4. 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

5. 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 4 shows the SAC for both DRAES and AES when changing a single bit of plaintext while keeping the key constant. Table 5 shows the SAC for both DRAES and AES when changing a single bit of key while keeping the plaintext constant. Table 6 shows the SAC for both DRAES and AES when changing 3 bits of plaintext while keeping the key constant. Table 7 shows the SAC for both DRAES and AES when changing 3 bits of key while keeping the plaintext constant.

As shown in Table 4 (change in Plaintext by 1 bit & Key constant), at first round, 11 bits (AES) of cipher value have changed out of 128-bit cipher text. This resulted 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

The end result in table 5 (change in Plaintext by many bits & Key constant) demonstrate 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

The end result in table 6 (alter in Key by 1 bit & Plaintext constant) 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.

The outcome in table 7 (change in Key by Many bits & Plaintext constant) 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).

6. 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 secure and physically powerful with dynamic rotation when compared to Rijndael with constant rotation as publicized from results from tables that related with diffusion analysis

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

***Table** 4(A) (B) The Avalanche Effect For AES And DRAES*
(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

***Table 5** (A) (B) The Avalanche Effect For AES And DRAES*
(CHANGE IN PLAINTEXT BY MANY BITS & KEY CONSTANT)

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

***Table 6** (A) (B) The Avalanche Effect For AES And DRAES*
(Alter In Key By 1 Bit & Plaintext Constant)

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

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

Round	Number of bit altered		SAC
1	28	22%	N
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

Table 7 *(A) (B) The Avalanche Effect For AES And DRAES*
(CHANGE IN KEY BY MANY 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

REFERENCES

[1] Daniel J. Bernstein, "Understanding brute force, Department of Mathematics", Statistics, and Computer Science (M/C 249) The University of Illinois at Chicago, IL 60607-7045, 2006

[2] Neeraj Kumar, "Investigations in Brute Force Attack on Cellular Security Based on Des and Aes", IJCEM International Journal of Computational Engineering & Management, Vol. 14, October 2011, ISSN 2230-7893

[3] Alex Biryukov, Dmitry Khovratovich, Ivica Nikoli_c, "Distinguisher and Related-Key Attack on the Full AES-256",University of Luxembourg falex.biryukov, Dmitry.

ISSN: 2076-3328 www.ijric.org E-ISSN: 2076-3336

khovratovich, ivica.nikolic@uni.lug 10 August 2009

[4] Elad Barkan and Eli Biham, "In How Many Ways Can You Write Rijndael?", Computer Science Department Technion { Israel Institute of Technology Haifa 32000, Israel ,2006

[5] Krishnamurthy G N, V Ramaswamy, "Making AES Stronger: AES with Key Dependent S-Box",IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.9, September 2008

[6] Jiqiang Lu1, Orr Dunkelman2, Nathan Keller3, and Jongsung Kim4, "New Impossible Differential Attacks on AES", Volume 5365 of Lecture Notes in Computer Science, pp. 279–293, Springer-Verlag, 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] Konstantinos Drakakis, *Senior Member, IEEE*, Verónica Requena, and Gary McGuire, "On the Nonlinearity of Exponential Welch Costas Functions", IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 56, NO. 3, MARCH 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