Improvements of Rijndael Algorithm Through Key Multiplication

Dr. Mohan.H.S[1], Sneha.M[2]

[1]Professor & Head of the Department, Department of Information Science and Engineering, SJB Institute of Technology, Bangalore-60.

[2]Fourth Semester M.Tech, Department of Information Science and Engineering, SJB Institute of Technology, Bangalore-60.

Abstract—There are basically two types of cryptographic algorithms; symmetric and asymmetric algorithms. The first algorithm which was introduced in symmetric algorithms was Data Encryption Standard (DES), which was later proved to be no longer secured by NIST. Hence the Advanced Encryption Standard (AES) was proposed and Rijndael was selected to become this standard by NIST. Each cipher uses several rounds of fixed operations to achieve desired security level. The number of rounds in a block cipher is decided based upon the resistivity levels against the known attacks. The very first level of attack on an encryption algorithm is to search for repetitive cipher values and relate them to plaintext. The diffusion enables to spread out the repetitive plain text patterns in the cipher values.

In this paper we propose a method of enhancing the diffusion power by key multiplication rather than conventional key addition used in the Advanced Encryption Standard algorithm. The paper discusses the problems associated with the key multiplication and provides the possible solutions. The strength is measured interms of diffusion analysis and time analysis. Through the results obtained it is proved that the proposed algorithm is stronger than the existing one.

Index Terms— AES, Cryptography, CPU Cycles, DES, Diffusion, NIST, SAC.

1 Introduction

IN this age of universal electronic connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud, there is indeed no time at which security does not matter. There are two main reasons to have these algorithms; first, the explosive growth in computer systems and their interconnections via networks has increased the dependence on both organizations and individuals on the information stored and communicated using these systems. This in turn has lead to the awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks. Second, the disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security.

We can classify[7] security attacks into two broad categories

1. Passive attacks
2. Active attacks

The first type of attack involves monitoring of transmission. These types are very difficult to detect, so we need to take security measures so that the data being transmitted is protected by means of encryption. It involves Release of message contents and Traffic analysis. Second type of attack involves some modification of data stream or the creation of a false stream and can be subdivided into four categories:

1. Masquerade
2. Replay
3. Modification of messages
4. Denial of service

Active attacks are easy to detect but difficult to prevent unlike passive attacks. Instead the goal is to detect them and to recover from any disruption or delays caused by them.

Cryptography, over the ages, has been practiced by many who have devised ad-hoc techniques to meet some of the information security requirements. The last twenty years have been period of transition as the discipline to a broader area. There are now several international scientific conferences devoted exclusively to cryptography and also an International Association for Crypto-logic Research (IACR), aimed at fostering research in the area.

There are two general types of cryptographic algorithms.

1. Symmetric algorithms.
2. Asymmetric algorithms.

The current Digital Encryption Standard (DES)[7] does no longer satisfy the need for data security because of its short 56-bit key. Such short keys can today be broken by brute force attacks. NIST issued a new version of DES called 3DES. With its 168-bit key length it overcomes the drawback of DES. The principle drawback of 3DES is that the algorithm is relatively sluggish in software. The original DES was designed for mid-1970s hardware implementation and does not produce efficient software code. 3DES which has three times as many rounds as DES is correspondingly slower. A secondary drawback is that both DES and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable.

Because of these drawbacks NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES), which should have security strength equal to or better than 3DES and significantly improved efficiency. In addition to these general requirements NIST specified that AES must be a symmetric block cipher with a block length of 128-bits and support for key lengths of 128, 192 and 256 bits.

In the first round of evaluation, 15 proposed algorithms were accepted. A second round narrowed the field to 5 algorithms (MARS, RC6, Serpent, Twofish, and Rijndael). NIST completed its evaluation process and published a final standard in November of 2001. NIST selected Rijndael as the proposed AES algorithm. The criteria used by NIST in the final evaluation[7] are:

- Rijndael has no known security attacks.
- Rijndael performs encryption and decryption very well across a variety of platforms including 8-bit and 64-bit platforms, and DSPs.
- Rijndael is very well suited for restricted-space environments where either encryption or decryption is implemented (but not both).
- Rijndael has the highest throughput of any of the finalists for feedback modes and second highest for non feedback modes.
- The operations used by Rijndael are among the easiest to defend against power and timing attacks.
- Encryption and decryption in Rijndael differs.
- Rijndael supports on the fly subkey computation for encryption.
- Rijndael supports fully block sizes and key sizes of 128, 192 and 256 bits in any combination.
- Rijndael has an excellent potential for parallelism for a single block encryption.

2 AES ALGORITHM

AES algorithms are symmetric cipher algorithms with variable key sizes and blocks, also with number of rounds to encrypt and decrypt the data than DES algorithms. There are numerous algorithms in AES. From them we have chosen the following algorithms for finding the performance analysis on time, key sizes, key setup time, encryption, and decryption and so on.

2.1 Rijndael Algorithm

This algorithm was developed by Joan Daemen, Vincent Rijmen. This algorithm supports different key sizes of 128, 192 and 256 bits but block length of 128-bit only is supported. The number of rounds will also change respectively to 10, 12, 14 based on the key size used for encryption. Rijndael was designed to have the following characteristics:

- Resistance against all known attacks
- Speed and code compactness on a wide range of platforms
- Design simplicity

In this project Rijndael algorithm is taken with 128, 192 and 256-bit keys and block size of 128-bits. The key, block size and number of rounds are chosen[8] as follows:

TABLE 1
KEY-BLOCK-ROUND COMBINATION

	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

If you consider 128 bit key, after an initial round, during which the first round key is XORed to the plain text (Addroundkey operation), nine equally structured rounds follow. Each round consists of the following operations:

1. Substitute Bytes: The Subbytes operation is a nonlinear substitution. This is a major reason for the security of the AES. There are different ways of interpreting the Subbytes operation. In this application report, it is sufficient to consider the Subbytes step as a lookup in a table. With the help of this lookup table, the 16 bytes of the state (the input data) are substituted by the corresponding values found in the table.
2. Shiftrows: As implied by its name, the Shiftrows operation processes different rows. A simple rotate with a different rotate width is performed. The second row of the 4x4 byte input data (the state) is shifted one byte position to the left in the matrix, the third row is shifted two byte positions to the left, and the fourth row is shifted three byte positions to the left. The first row is not changed.
3. Mixcolumns: Probably the most complex operation from a software implementation perspective is the Mixcolumns step. Opposed to the Shiftrows operation, which works on rows in the 4x4 state matrix, the Mixcolumns operation processes columns. In principle, only a matrix multiplication needs to be executed. To make this operation reversible, the usual addition and multiplication are not used. In AES, Galois field operations are used. This paper does not go into the mathematical details, it is only important to know that in a Galois field, an addition corresponds to an XOR and a multiplication to a more complex equivalent. The fact that there are many instances of 01 in the multiplication matrix of

the Mixcolumns operation makes this step easily computable.

4. Add Round Key: The Addroundkey operation is simple. The corresponding bytes of the input data and the expanded key are XORed.

3 Proposed Algorithm

In AES, the key is used by key addition operation only. No other diffusion element makes use of the key. For this reason, the cipher begins and ends with the AddRoundKey stage. Any other stage applied at the beginning or end, is reversible without knowledge of the key and so would add no security. The AddRoundKey stage is, in effect, a form of Vernam cipher and by itself would not formidable. The other three stages together provide confusion, diffusion and non-linearity but by themselves provide no security, because they do not use the key.

We can view the cipher as alternating operations of XOR encryption (Add Round Key) of a block, followed by scrambling of the block (the other three stages), followed by XOR encryption, and so, on. This scheme is both efficient and highly secure.

3.1 Proposed KeyMultiplication Round

The proposed Rijndael algorithm consists of Key Multiplication function instead of Key Addition. It is done by multiplying each byte of the state with the corresponding byte in the Key. This will consume some time than the Keyaddition, which is a simple XOR but this will produce more confusion and more Diffusion than the Keyaddition. The structure of proposed algorithm is as shown in Fig.1.

As shown in the Fig.1 the KeyAdditionRound is replaced by KeyMultiplication round.

Multiplication in Rijndael's Galois field[4] is complicated. The procedure is as follows:

- Take two eight-bit numbers, a and b, and an eight-bit product p.
- Set the product to zero.
- Make a copy of a and b, which we will simply call a and b in the rest of this algorithm.
- Run the following loop eight times:
    1. If the low bit of b is set, exclusive or the product p by the value of a .
    2. Keep track of whether the high (eighth from left) bit of a is set to one .
    3. Rotate a one bit to the left, discarding the high bit, and making the low bit have a value of zero.
    4. If a's hi bit had a value of one prior to this rotation, exclusive or a with the hexadecimal number 0x1b.
    5. Rotate b one bit to the right, discarding the low bit, and making the high (eighth from left) bit have a value of zero.
- The product p now has the product of a and b.

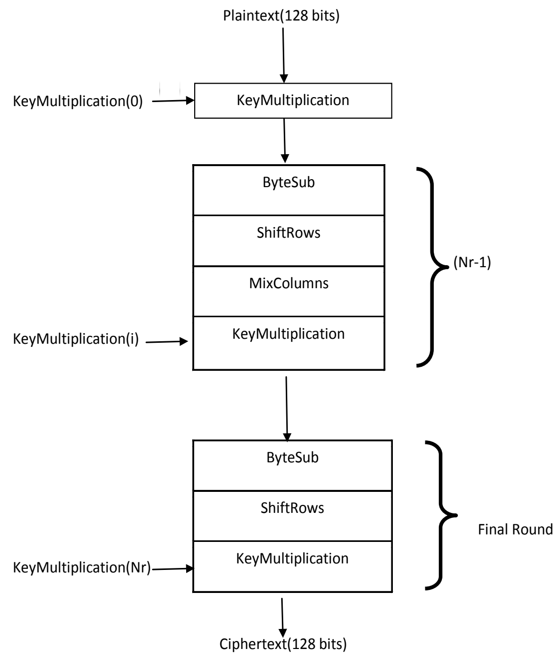

Fig.1 Structure of proposed Rijndael algorithm

3.2 Problem Faced

The Keys can have any value from {00} to {ff}. So if a key has a value {00}, then multiplication of that part of key with the state byte will give {00}. This leads to loss of that particular data. To avoid this, the key when it is expanded it is checked for any {00} value in it. If it is present then it is replaced with the number (ROUND +1).Here ROUND represents the round number. Here why it is taken as a (ROUND + 1) means, because if we have {00} in the 0th round then if we replace the key value with the round number then again it will give {00}. So (ROUND + 1) is used. The inverses of keys are calculated by taking the Multiplicative inverse of each byte and it is used in the decryption.

4 Experimental Results

The proposed algorithm is tested with the following two measures:

1. SAC value
2. Number of CPU cycles taken by encryption, decryption and key setup functions of both the algorithms.

4.1 SAC Value

Strict Avalanche Criteria (SAC) is the Hamming distance of two cipher values corresponding to cases, without input bit flipping and with input bit flipping.

Strict Avalanche criteria states that if a single input bit is flipped at least half of the output bits should be changed.

There two types of SAC:

i) First order SAC: It is a change in output bit when a single input bit is flipped.
ii) Higher order SAC: It is a change in output bit when many input bits are flipped.

In this paper first order SAC is considered for analysis.

Diffusion analysis of existing algorithm based on first order SAC for AES algorithm with key addition:

```
Enter the key length(128,192 or 256):128
Enter the Key in hexadecimal: 2b7e1516  28aed2a6  345e678a  d234a567

Enter PlainText in hexadecimal: 2eb45678  3456d789  23de34e7  23cd34e6
This is Encryption using AES which uses key Addition

The Key Entered is  : 2b7e1516  28aed2a6  345e678a  d234a567

The Data entered is : 2eb45678  3456d789  23de34e7  23cd34e6

*****************************************************************
Cipher after 1 round  is : 645059cc  5374c737  5dd56c5d  9d777eed

Cipher after 2 round  is : 87d9214c  a1ccacfb  598f47d8  912ed667

Cipher after 3 round  is : e75b72f6  bc1ec2cc  be1a8c20  6c7d1311

Cipher after 4 round  is : cbfc21fc  c23d1d17  a5d36c67  2d127955

Cipher after 5 round  is : 16c39132  2e685c5b  4313c498  1208f6da

Cipher after 6 round  is : 604e8e3e  e9778c55  f73dd813  c4a17d12

Cipher after 7 round  is : b6c9f363  de2689c5  95a100f2  242080c4

Cipher after 8 round  is : 3eb72051  cb73ea5c  d0daa79c  d4e99236

Cipher after 9 round  is : eaead192  1e2686fd  491ca020  68fac00d

Cipher after 10 round is : 2cc1c480  2bb02a4  bd30bb5e  899061ec
*****************************************************************
The cipher after Encryption is :

2cc1c480  2bb02a4  bd30bb5e  899061ec
*****************************************************************
This is Decryption using AES which uses key Addition

The Key Entered is  : 2b7e1516  28aed2a6  345e678a  d234a567

The Data entered is : 2cc1c480  2bb02a4  bd30bb5e  899061ec
```

Fig.2 Output after each round using key addition

In Fig.2, 128-bit input plaintext and key is shown and also the output after each round. The final ciphertext after 10th round is also shown in the figure.

```
Cipher after 3 round  is : e75b72f6  bc1ec2cc  be1a8c20  6c7d1311

Cipher after 4 round  is : cbfc21fc  c23d1d17  a5d36c67  2d127955

Cipher after 5 round  is : 16c39132  2e685c5b  4313c498  1208f6da

Cipher after 6 round  is : 604e8e3e  e9778c55  f73dd813  c4a17d12

Cipher after 7 round  is : b6c9f363  de2689c5  95a100f2  242080c4

Cipher after 8 round  is : 3eb72051  cb73ea5c  d0daa79c  d4e99236

Cipher after 9 round  is : eaead192  1e2686fd  491ca020  68fac00d

Cipher after 10 round is : 2cc1c480  2bb02a4  bd30bb5e  899061ec
*****************************************************************
The cipher after Encryption is :

2cc1c480  2bb02a4  bd30bb5e  899061ec
*****************************************************************
This is Decryption using AES which uses key Addition

The Key Entered is  : 2b7e1516  28aed2a6  345e678a  d234a567

The Data entered is : 2cc1c480  2bb02a4  bd30bb5e  899061ec

*****************************************************************
The original initial Data after Decryption is :

2eb45678  3456d789  23de34e7  23cd34e6

The results shows the AES Encryption and Decryption process using keyAddition

Performing the diffusion analysis with key addition
Enter the Key in hexadecimal by changing only one bit: 2b7d1516 28aed2a6 345e678a d234a567


The Key Entered is  : 2b7d1516  28aed2a6  345e678a  d234a567
```

Fig.3 Input key bit changed in key for diffusion analysis

In Fig.3, the input flipped in the input key for the diffusion analysis is shown.

```
File  Edit  View  Terminal  Tabs  Help
The Data entered is : 2eb45678  3456d789  23de34e7  23cd34e6

*****************************************************************
Cipher after 1 round  is : 645359cc  5377c737  620d1e82  7ddd457f

Cipher after 2 round  is : a40d0833  29a62eb2  64704896  65c63986

Cipher after 3 round  is : ca4d94fc  b81cb9b2  ace47163  bcf6bb5a

Cipher after 4 round  is : 38fc6c2d  5e3a720f  a9ac301c  342431f8

Cipher after 5 round  is : 7cf4172e  36fbb0d1  9878ace  2f36d681

Cipher after 6 round  is : 491d52c6  e544a62d  7f4bc3cd  986259da

Cipher after 7 round  is : aa9b045d  65b9d2be  8b2a3049  cf183da

Cipher after 8 round  is : 2a59bbc8  c73c3e59  53502b9  198dc18

Cipher after 9 round  is : 7847ba42  e4a93895  9a73a1f3  8f868007

Cipher after 10 round is : a537ddbb  aba8bec7  8c8b7120  db72256
*****************************************************************
The cipher after Encryption is :

a537ddbb  aba8bec7  8c8b7120  db72256
*****************************************************************
 NUMBER        THE NUMBER OF       SAC VALUE
OF ROUNDS      BITS THAT DIFFER

  1                40                31.250000
  2                69                53.906250
  3                61                47.656250
  4                62                48.437500
  5                59                46.093750
  6                60                46.875000
  7                69                53.906250
  8                66                51.562500
  9                67                52.343750
 10                66                51.562500[root@localhost aes_key_addition]#
```

Fig.4 SAC values for AES with key addition

ISSN 2229-5518

The Fig.4 shows the SAC values for each round using AES with key addition. It is clear from the figure that 51% is achieved in the final round that is 66 bits of 128 bits are affected by flipping one input bit.

Diffusion analysis of proposed algorithm based on first order SAC for AES algorithm with key multiplication:

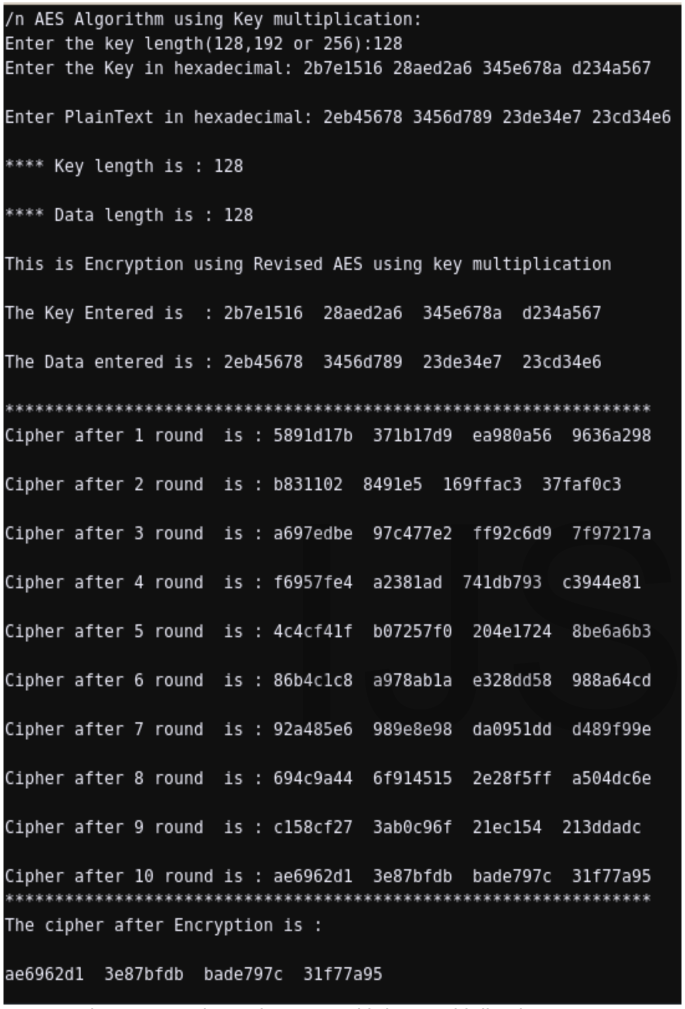

Fig.5 Encryption using AES with key multiplication

The Fig.5 shows the output after each round and finally ciphertext after 10th round using the proposed AES algorithm with key multiplication.

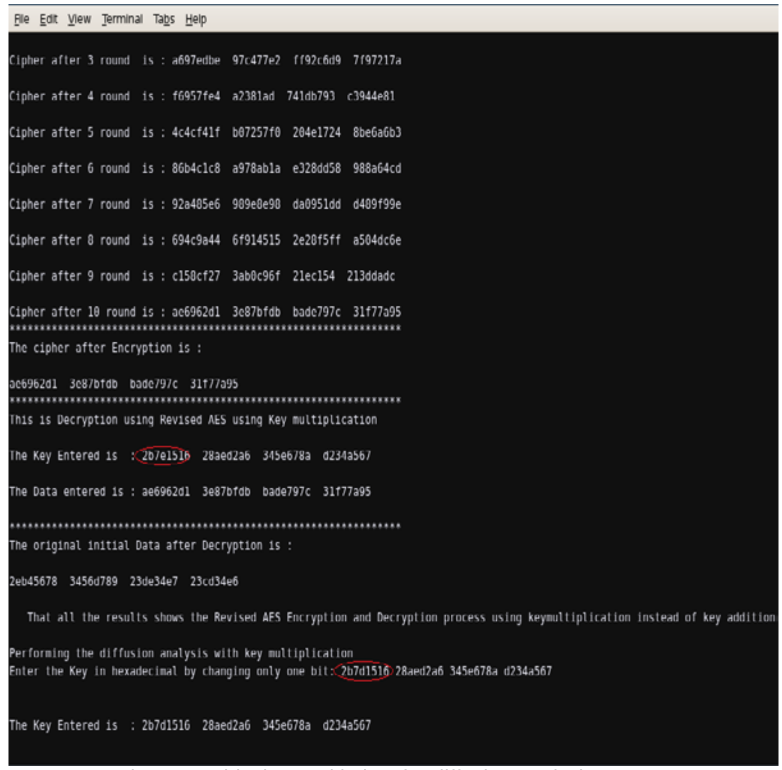

Fig.6 Input bit changed in key for diffusion analysis

The figure shows the input flipped in the input key for the diffusion analysis

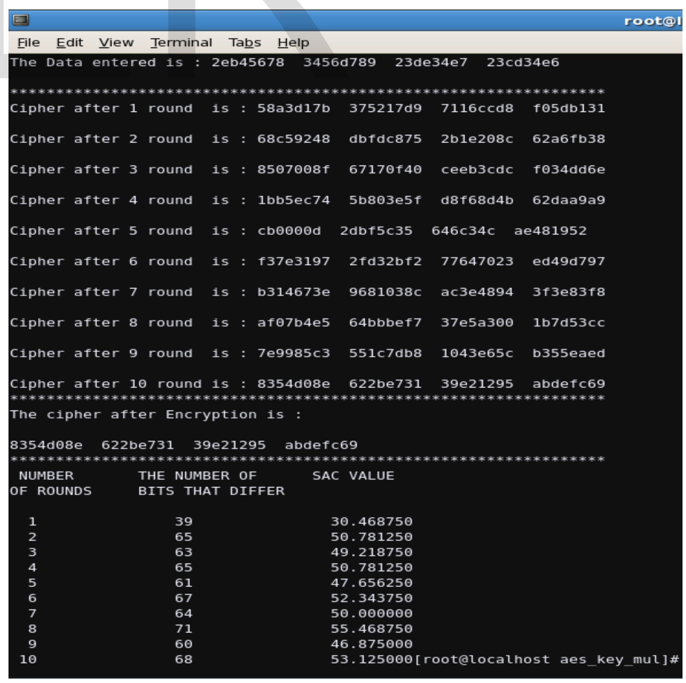

Fig.7 SAC values for AES with key multiplication

The figure shows the SAC values for each round using AES with key multiplication. It is clear from the figure that 53% is achieved in the final round that is 68 bits of 128 bits are affected by flipping one input bit, which is more compared to the AES algorithm with key addition.

4.2 Time Analysis

The below tables and graph shows the comparison of AES with key addition and Key multiplication the number of CPU cycles taken by encryption, decryption and key setup functions take:

1) Key setup

TABLE 2

KEY SETUP

AES Algorithm	Encrypt 128 (Cycles)	Encrypt 192 (Cycles)	Encrypt 256 (Cycles)
With Key addition	88.8	86.2	86.06
With Key multiplication	87.26	86.88	86.65

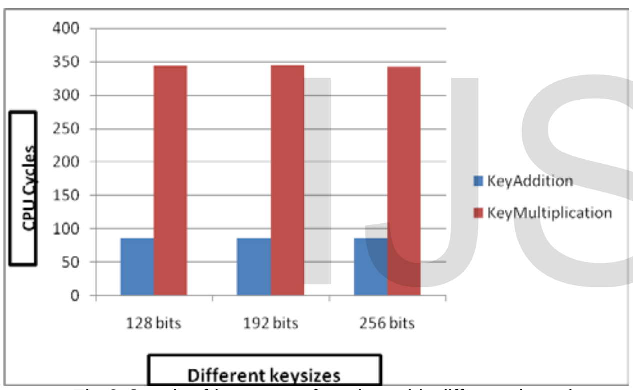

Fig.8 Graph of key setup function with different key sizes

2) Encryption

TABLE 3

ENCRYPTION

AES Algorithm	Encrypt 128 (Cycles)	Encrypt 192 (Cycles)	Encrypt 256 (Cycles)
With Key addition	88.8	86.2	86.06
With Key multiplication	87.26	86.88	86.65

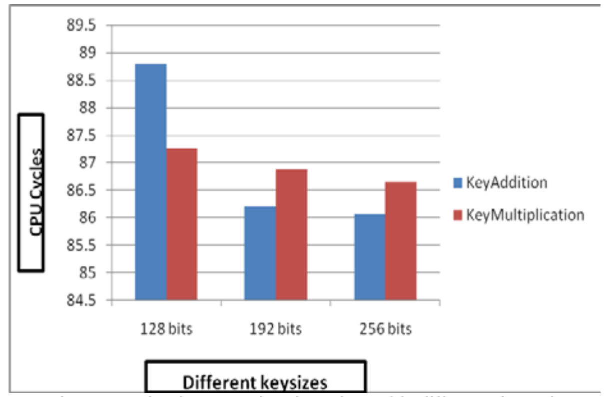

Fig.9 Graph of encryption function with different key sizes

3) Decryption

TABLE 3

DECRYPTION

AES Algorithm	Decrypt 128 (Cycles)	Decrypt 192 (Cycles)	Decrypt 256 (Cycles)
With Key addition	87.48	86.28	85.39
With Key multiplication	86.80	87.71	87.03

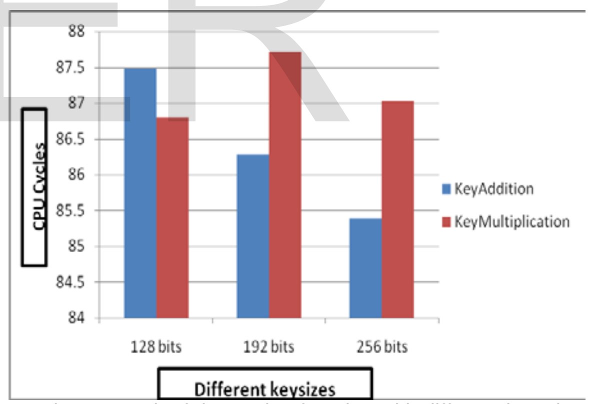

Fig.10 Graph of decryption function with different key sizes

5 CONCLUSION AND FUTURE WORK

The results of diffusion analysis indicate that the AES algorithm with key multiplication has better diffusion level than AES with key addition, which can be shown in the table TABLE 4 as shown.

TABLE 4
AVELANCHE VALUES

Rounds number	1	2	3	4	5	6	7	8	9	10
AES with key addition	40	69	61	62	59	60	69	66	67	66
AES with key multi-plication	39	65	63	65	61	67	64	71	60	68

As shown in the table there is comparable difference in the number of bits affected when one bit is flipped in the input. AES with key multiplication shows that more bits are getting affected. The diffusion level will obviously increase if the length of the key is more (192 or 256).

Through the results tabulated and graphs plotted for time analysis, it is clear that there is no much difference between the existing and the proposed algorithm for encryption and decryption time. For key setup function the more CPU cycles are consumed in AES with key multiplication compared to AES with key addition. This is because the extra searching for zeros in the key and replacing them as explained in the previous section is done in key multiplication. This is justified when compared to the improved strength of the algorithm.

As future work the input plaintext matrix size from 4x4 can be changed to 8x8, which will also enhances the security given by the existing AES algorithm.

REFERENCES

[1] B.D.C.N.Prasad, P E S N Krishna Prasad, P Sita Rama Murty and K Madhavi, "A Performance Study on AES Algorithms," (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 6, September 2010.

[2] Mohan H. S. and A Raji Reddy, "Generating the New S-box and Analyzing the Diffusion Strength to Improve the Security of AES Algorithm," (IJCSIS) International Journal of Computer Science and Information Security,Vol.2, No.9, September 2010.

[3] 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 ISSN (Online): 1694-0814.

[4] Mohan H. S., A Raji Reddy and Manjunath T.N, "Improving the Diffusion power of AES Rijndael with key multiplication," International Journal of Computer Applications (0975 – 8887) Volume 30– No.5, September 2011.

[5] Vikas Kaul, S K Narayankhedkar, S Achrekar, S Agarwal, P, "Security Enhancement Algorithms for Data Transmission for Next Generation Networks," IJCA 2012.

[6] K.Rahimunnisa, Dr. S. Sureshkumar, K.Rajeshkumar, "Implementation of AES with New S-Box and Performance Analysis with the Modified S-Box," International Conference on VLSI, Communication & Instrumentation (ICVCI), 2011.

[7] Cryptography and Network Security -"William Stallings", Third Edition.

[8] A. Lee, NIST Special Publication 800-21,"Guideline for Implementing Cryptography in the Federal Government", National Institute of Standards and Technology, November 1999. Page is available at http://csrc.nist.gov/publications/.

[9] **Rijndael:** Joan Daemen, Vincent Rijmen, "AES Proposal: Rijndael", Proton World Int.l, Belgium, Katholieke Universiteit Leuven, Belgium, September 1999.

Authors Profile

Mohan H.S. received his Bachelor's degree in Computer Science and Engineering from Malnad College of Engineering, Hassan during the year 1999, M. Tech in computer Science and Engineering from Jawaharlal Nehru National College of Engineering, Shimoga during the year 2004 and Ph.D degree in Dr.MGR University, Chennai during the year. He is Head of the Department and professor in the Dept of Information Science and Engineering at SJB Institute of Technology, Bangalore-60. He is having total 12 years of teaching experience. His area of interests are Networks Security, Image processing, Data Structures, Computer Graphics, finite automata and formal languages, Compiler Design. He has obtained a best teacher award for his teaching during the year 2008 at SJBIT Bangalore-60. He has published and presented papers in journals, international and national level conferences.

Sneha.M received her Bachelor's degree in Computer Science and Engineering from R N S Institutes of Technology, Bangalore during the year 2006 and currently pursuing M. Tech in Computer Network Engineering in SJB Institute of Technology, Bangalore-60.