Improving the Diffusion power of AES Rijndael with key multiplication

Mohan H.S.
Research Scholar
Dr. MGR University, Chennai.

A. Raji Reddy
Professor, Dept of ECE MITS,
Madanapalle. India

Manjunath T.N
Research Scholar
Bharathiar University,
Coimbatore

ABSTRACT
Block ciphers are very important in communication systems as they provide confidentiality through encryption. The popular block cipher is an Advanced Encryption Standard (AES). 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. The diffusion is achieved using linear operations such as key addition, rotate byte, MDS matrix multiplication, etc. 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 measured results indicate more diffusion when compared with the existing method. Key multiplication, as a diffusion element, is a better solution in the design of encryption algorithms.

Keywords
Advanced Encryption Standard, Diffusion, Strict Avalanche Criteria.

1. INTRODUCTION
Cryptography allows people to carry over the confidence found in the physical world to the electronic world, thus allowing people to do business electronically without worries of deception. Everyday thousands of people interacts electronically, whether it is through e-mail, e-commerce, ATM machine or cellular phones. The perpetual increase of information transmitted electronically has led to an increased reliance on cryptography.

Cryptography provides four basic functions required for electronic transactions: 1) Authentication, 2) Confidentiality, 3) Integrity and 4) Non-repudiation. The cipher values of an encryption algorithm are randomized using several diffusion elements such as addition, transposition, rotation, etc. Such operations on diffusion elements are repeated several times or several rounds for achieving sufficient diffusion level. For example, DES uses 8 rounds, whereas MARS uses 32 rounds. Therefore, it is very important to understand the diffusion behavior of an encryption algorithm [10].

1.1 Symmetric Algorithms
There are two general types of key based algorithms: Symmetric and Public Key. In Symmetric algorithms encryption key can be same as the decryption key and vice versa. These are also called as secret key algorithms. Symmetric algorithms can be divided into two categories: i) some operate on the plaintext a single bit at a time which are called Stream ciphers, and ii) others operate on the plaintext in groups of bits, such groups of bits are called blocks and such algorithms are called Block ciphers [8].

1.2 Stream Ciphers and Block Ciphers
Stream ciphers are generally faster than block ciphers in hardware, and have less complex hardware circuitry. Stream ciphers are more suitable for situations where transmission errors are highly probable.

Symmetric key block ciphers are the most prominent and important elements in many cryptographic systems. Individually, they provide confidentiality. The examples of block ciphers are DES, 3-DES, FEAL, SAFER, RC5 and AES. The implementation of any basic block cipher is generally known as Electronic Code Book (ECB) mode. In order to increase the security further additional modes are also defined. They are (1) Cipher Feed Back (CFB) mode (2) Output Feed Back (OFB) mode (3) Counter mode (CTR). The counter mode has become popular in IPSec and IPv6 applications.

1.3 Confusion and Diffusion
These are the two important techniques for building any cryptographic system. Claude Shannon introduced the terms Confusion and Diffusion. According to Shannon, in an ideal cipher, "all statistics of the cipher text are independent of the particular key used". In Diffusion, each plaintext digit affects many cipher text digits, which is equivalent to saying that each cipher text digit is affected by many plain text digits.

All encryption algorithms will make use of diffusion and confusion layers. Diffusion layer is based upon simple linear operations such as multi-permutations, key additions, multiplication with known constants etc. On the other hand, confusion layer is based upon complex and linear operations such as Substitution Box (S-box) [10].

This paper proposes the key multiplication rather than key addition for improving the diffusion level in AES. A scheme is provided to implement the key multiplication. Experiments are conducted to measure the diffusion level with key multiplication. Results are compared with that of diffusion level achieved with key addition. In the remaining paper, section 2

describes the AES algorithm and its evaluation. The key addition known as "Addroundkey" of the existing AES algorithm is explained in section 3. The scheme for the proposed key multiplication is explained in section 4. Experimental results are presented and discussed in Section 5.

2. ADVANCED ENCRYPTION STANDARD ALGORITHM

2.1 Evaluation of Advanced Encryption Standard

In 1997, the National Institute of Standards and Technology (NIST) announced a program to develop and choose an Advanced Encryption Standard (AES) to replace the aging Data Encryption Standard (DES).In 1998, NIST announced the acceptance of fifteen candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates. This analysis included an initial examination of the security and efficiency characteristics for each algorithm. NIST reviewed the results of this preliminary research and selected MARS, RC6, Rijndael, Serpent and Twofish as finalists. An interesting performance comparison of these algorithms can be found in [1]. On October 2000 and having reviewed further public analysis of the finalists, NIST decided to propose Rijndael as the Advanced Encryption Standard (AES). Rijndael, designed by Joan Daemen (Proton World International Inc.) and Vincent Rijmen (Katholieke Univeriteit Leuven) of Belgium, is a block cipher with a simple and elegant structure [2].The Advanced Encryption Standard (AES), also known as the Rijndael algorithm, is a symmetric block cipher that can encrypt data blocks of 128 bits using symmetric keys of 128, 192 or 256 bits. AES was introduced to replace the Triple DES (3DES) algorithm used for a good amount of time universally. Though, if security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come. The main drawback was its slow software implementation. For reasons of both efficiency and security, a larger block size is desirable. Due to its high level security, speed, ease of implementation and flexibility, Rijndael was chosen for AES standard in the year 2001.

2.2 Rijndael Algorithm

Rijndael is a block cipher developed by Joan Daemen and Vincent Rijmen. The algorithm is flexible in supporting any combination of data and key size of 128, 192, and 256 bits. However, AES allows a 128 bit data length that can be divided into four basic operation blocks. These blocks operate on array of bytes and organized as a 4×4 matrix that is called the state. For full encryption, the data is passed through Nr rounds (Nr = 10, 12, 14) as shown in the Figure 2. These rounds are governed by the following transformations:

(i) Bytesub transformation: Is a non linear byte Substitution, using a substation table (s-box), which is constructed by multiplicative inverse and affine transformation.

(ii) Shiftrows transformation: Is a simple byte transposition, the bytes in the last three rows of the state are cyclically shifted; the offset of the left shift varies from one to three bytes.

(iii) Mixcolumns transformation: Is equivalent to a matrix multiplication of columns of the states. Each column vector is multiplied by a fixed matrix. It should be noted that the bytes are treated as polynomials rather than numbers.

(iv) Addroundkey transformation: Is a simple XOR between the working state and the roundkey. This transformation is its own inverse.

The encryption procedure consists of several steps as shown by Fig. 1. After an initial addroundkey, a round function is applied

	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

Fig 1: Key-Block-Round Combinations

to the data block (consisting of bytesub, shiftrows, mixcolumns and addroundkey transformation, respectively). It is performed iteratively (Nr times) depending on the key length. The decryption structure has exactly the same sequence of transformations as the one in the encryption structure. The transformations Inv-Bytesub, the Inv-Shiftrows, the Inv-Mixcolumns, and the Addroundkey allow the form of the key schedules to be identical for encryption and decryption [3].

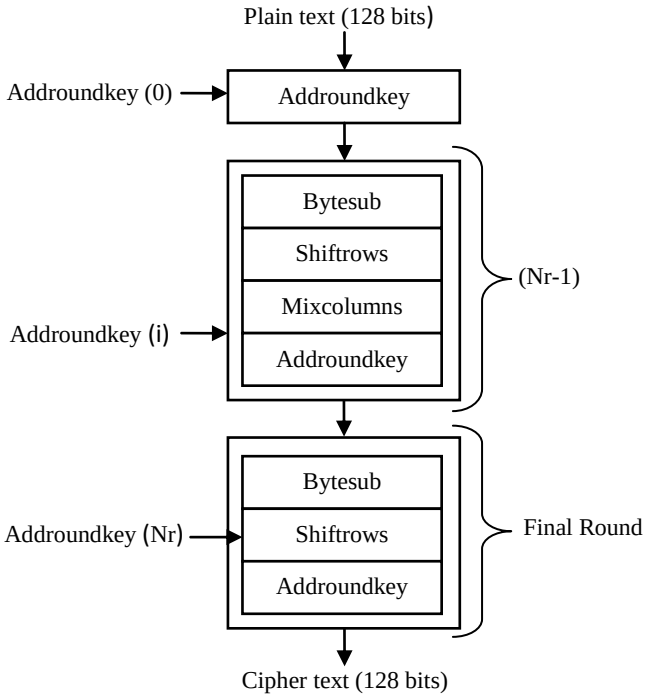


Fig 2: AES algorithm- Encryption Structure

3. EXISTING ADDROUNDKEY () IN AES

In this transformation the 128 bits of state are bitwise XORed with the 128 bits of the round key as shown in figure below. The array of round keys is derived from the cipher key by means of the key schedule. The round key length is equal to the block length. It can be viewed as a byte level operation.

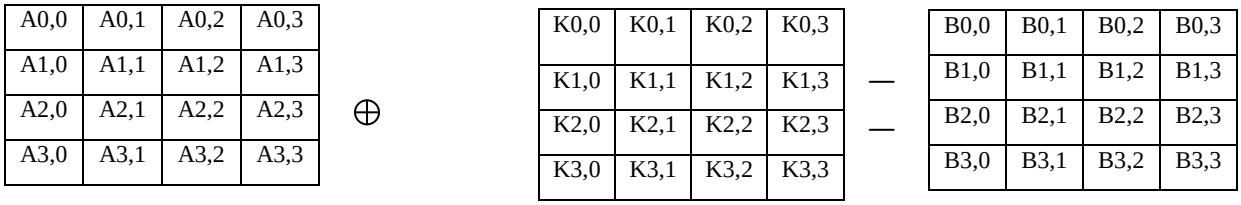
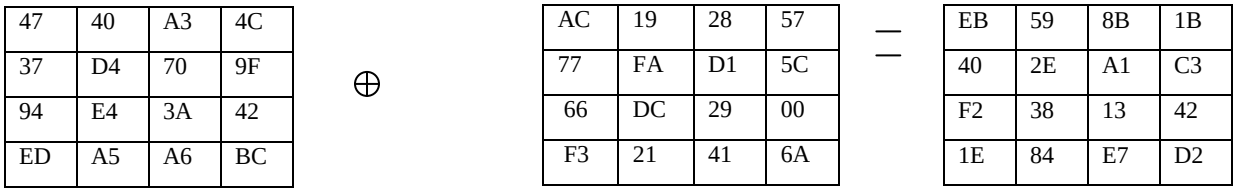


Fig 3: Add roundkey transformation

For example



The input is depicted as a square matrix of bytes. This block is copied into the state array, which is modified at each stage of encryption or decryption. After the final stage, state is copied to an output matrix. Similarly the 128-bit key is also seen as a square matrix of bytes.

4. PROPOSED KEY MULTIPLICATION

4.1 Advantage of Key multiplication

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 [13].

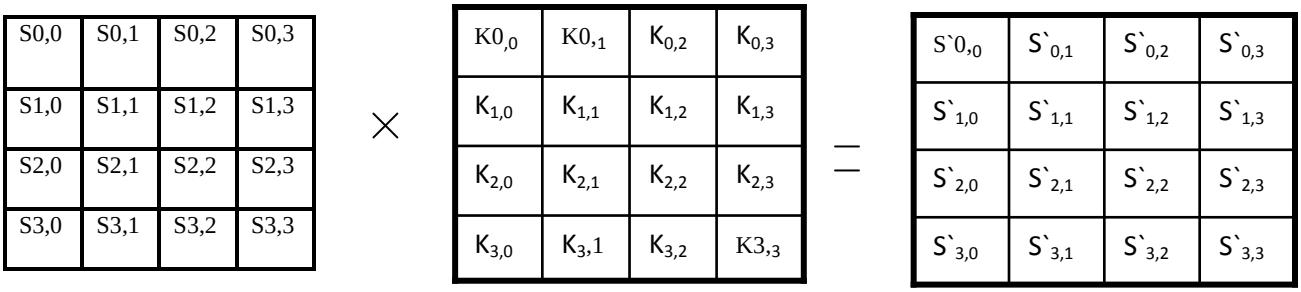
The AddRoundKey stage is, in effect, a form of Vernam cipher and by itself would not formidable. The other three stages together provide confusion, diffusion and non-linearity but by themselves provide no security, because they do not use the key. We can view the cipher as alternating operations of XOR 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.

4.2 Proposed Key Mixing

The revised AES 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 EXOR but this will produce more confusion and more Diffusion than the Keyaddition. This multiplication can be achieved using the following function

Multiplication in Rijndael's galois field is a little more 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



For example

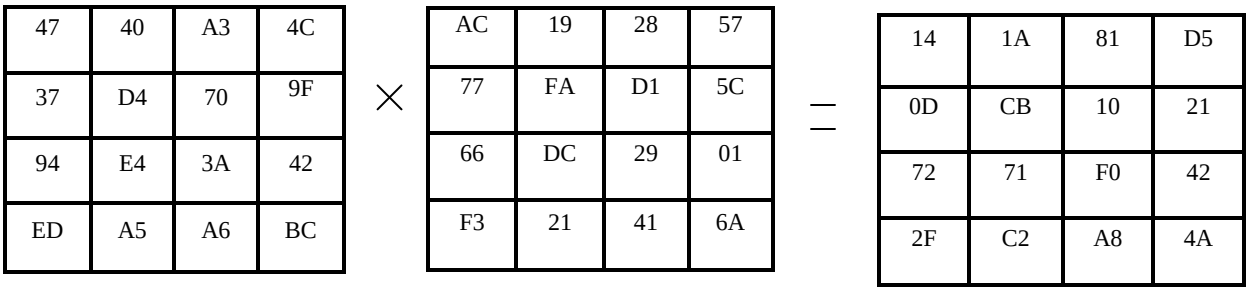


Fig 4: KeyMultiplication transformation

4.3 Problem Faced and Solution

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 inverse of keys are calculated by taking the Multiplicative inverse of each byte and it is used in the decryption.

For example consider the following initial key

0x00000000 0x00000000 0x00000000 0x00000000

The expanded key is

00000000 00000000 00000000 00000000
62626262 63636363 63636363 63636363
9bf99bf9 98fb98fb 98fb98fb c9aac9aa
9069f20b 976cf40f 34cf57ac 50fa3399
ee87757e 066a9e91 da1542ee 7b81b22b
7ff88df3 2e44da4b 2b3e7c92 8809bb90
ec14996a 6125ffb4 4b75099b 858c37a7
2135acc6 7550af1b 17626bf0 870b3c9b
0e3b9751 f9a9061d 03610afa 3338049f
b18a1d4c d47d7b66 d8b9b349 e2dade41
b43e236f ef92e98f 5be25118 cb11cf 8e

When we multiply with this key the plaintext will be lost in the first key multiplication itself. So to avoid that we replaced the {00} with the (Round+1) number.

So the new keys are:

11111111 11111111 11111111 11111111
62626262 63636363 63636363 63636363
9bf99bf9 98fb98fb 98fb98fb c9aac9aa
9069f20b 976cf40f 34cf57ac 50fa3399
ee87757e 066a9e91 da1542ee 7b81b22b
7ff88df3 2e44da4b 2b3e7c92 8809bb90
ec14996a 6125ffb4 4b75099b 858c37a7
2135acc6 7550af1b 17626bf0 870b3c9b
0e3b9751 f9a9061d 03610afa 3338049f
b18a1d4c d47d7b66 d8b9b349 e2dade41
b43e236f ef92e98f 5be25118 cb11cf 8e

5. EXPERIMENTAL RESULTS

The difference between the bits of plaintext and cipher text is the measure of diffusion level. In other words, the Hamming distance of the difference indicates the difffusion strength. The avalanche value is the number of output bits changed when one input bit in the plain text is flipped. Avalanche value is the Hamming distance of two cipher values corresponding to two cases: 1) without input bit flipping and 2) with input bit flipping. For robust encryption algorithm, higher avalanche values are desirable. An encryption algorithm implemented with either more number of diffusion elements or powerful elements exhibits higher avalanche values. Experiments are conducted to measure the avalanche values of an algorithm with key multiplication as diffusion element.

Two encryption algorithms are implemented: 1) AES with key addition as per the standard and 2) AES with key multiplication in place of key addition. Figures 5 and 6 gives the details of input plain text, key value and cipher values at the end of each round corresponding to these two implementations, respectively. Cipher values are also generated by flipping one bit of plain text and compared with earlier cipher values to obtain the avalanche values. Figures 7 and 8 gives the avalanche values obtained for key addition and multiplication, respectively. These are also shown in Table 1 for the comparing the results.

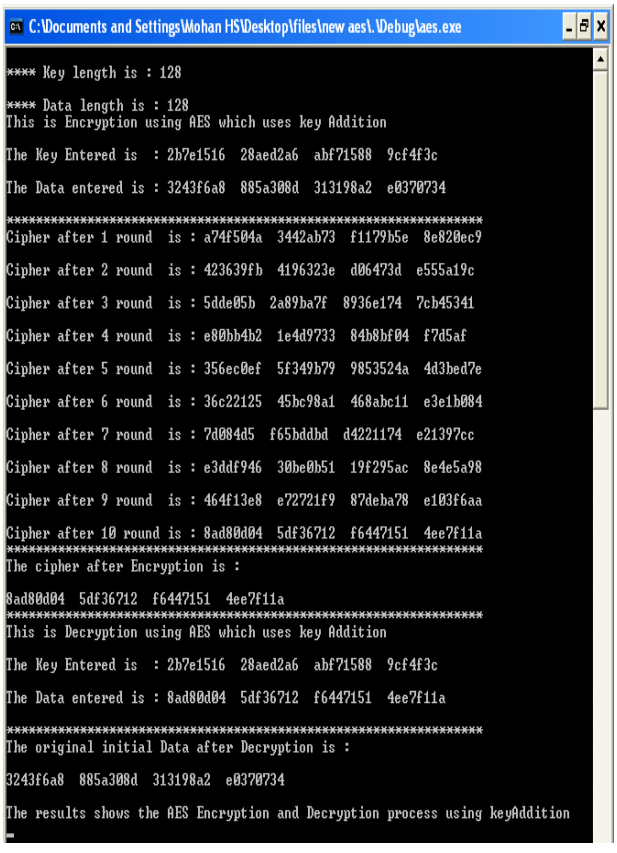


Fig 5: Encryption and Decryption using the Existing AES which uses Key Addition

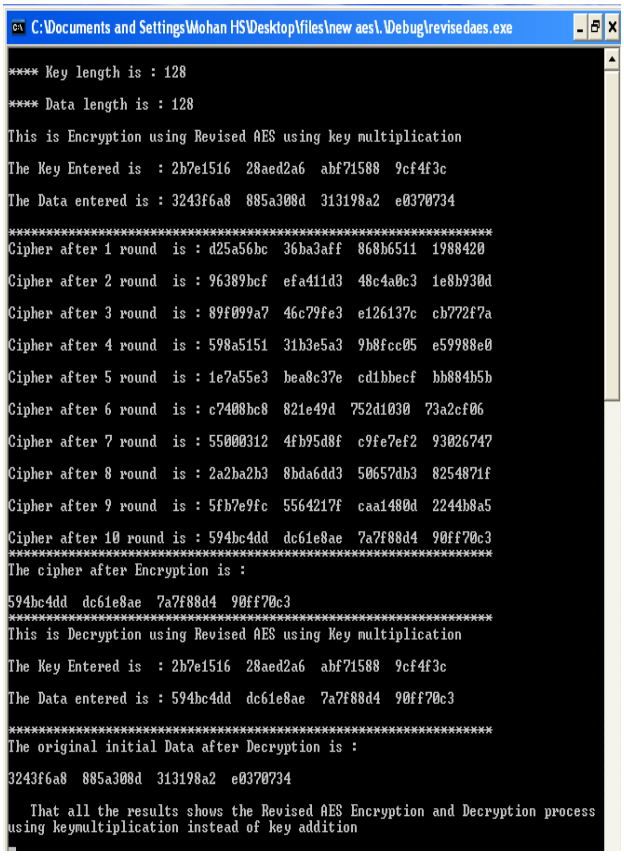


Fig 6: Encryption and Decryption using Revised AES which uses key Multiplication

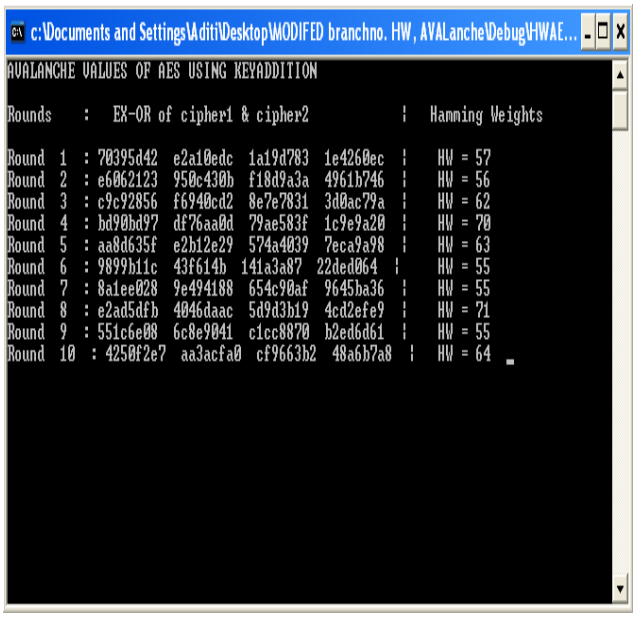


Fig 7: Results of Avalanche Effect of changing one bit using AES which uses key Addition.

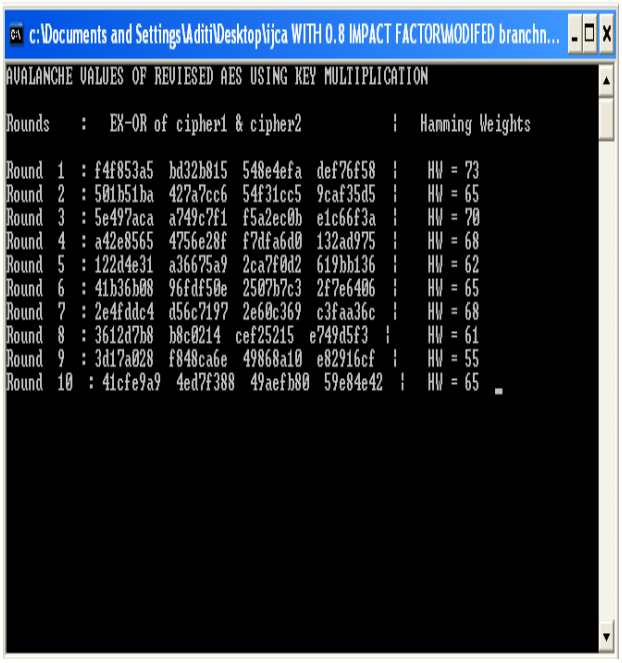


Fig 8: Results of Avalanche Effect of changing one bit in a key using Revised AES which uses key Multiplication

Table 1: Avalanche values for two encryption algorithms a) AES and b) AES with key multiplication in place of key addition.

Round Number	1	2	3	4	5	6	7	8	9	10
AES with Key addition	57	56	62	70	63	55	55	71	55	64
AES with Key Multiplication	73	65	70	68	62	65	68	61	55	65

As shown in Table 1, the avalanche values corresponding to key multiplication are higher than that of key addition. This shows that the diffusion exhibited by key multiplication is better than the key addition. It is relatively easy to meet strict avalanche criterion (SAC) using key multiplications compared to key addition. SAC states that if a single input bit is flipped, at least half of output bits should change. Therefore, key multiplication can be used as a diffusion element in the design of encryption algorithms for achieving better performance.

6. CONCLUSION

The basic design of an encryption algorithm is based upon the strength of diffusion and confusion. This paper explored diffusion elements used in the AES. Based upon these studies, we proposed a revised Advanced encryption algorithm using Key mixing using modulo multiplication instead of conventional key addition which increases the security of AES. A scheme is designed to implement key multiplication. As indicated by measured results, the diffusion level is improved with key multiplication. The results are attractive to design an encryption algorithm using key multiplication apart from the key addition in order to achieve desired diffusion level. Though key multiplication takes more CPU cycles, it finds applications on platforms with high-speed processors.

7. REFERENCES

[1] AES page available via http://www.nist.gov/CryptoToolkit.

[2] Computer Security Objects Register (CSOR): http://csrc.nist.gov/csor/.

[3] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September 3, 1999, available at [1].

[4] J. Daemen and V. Rijmen, The block cipher Rijndael, Smart Card research and Applications, LNCS 1820, Springer-Verlag, pp. 288-296.

[5] A. Lee, NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, National Institute of Standards and Technology, November 1999.

[6] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, New York, 1997, p. 81-83.

[7] J. Nechvatal, Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000.

[8] Nicolas Courtois**,** The Inverse S-box, Non-linear Polynomial Relations and Cryptanalysis of Block Ciphers, in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. pp. 170-188, Springer.

[9] M. Zeghid, M. Machhout, L. Khriji, A. Baganne, R. Tourki, "Modified AES Based Algorithm for Image Encryption", World Academy of Science, Engineering and Technology, 2007

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

[11] N. Penchalaiah , "Effective Comparison and evaluation of DES and Rijndael Algorithm (AES)", International Journal on Computer Science and Engineering, Vol. 02, No. 05, 2010, pp.1641-1645.

[12] B.D.C.N.Prasad, P E S N Krishna Prasad, "A Performance Study on AES algorithms", International Journal of Computer Science and Information Security, Vol. 8, No. 6,September 2010,pp 128-132.

[13] R. Elumalai and A. Raji Reddy. "Improving Diffusion power of AES Rijndael with 8x8 MDS matrix", International Journal of Scientific and Engineering Research, Vol. 2, Issue-3, March-2011.