International Journal of Information Technology and Knowledge Management
January-June 2012, Volume 5, No. 1, pp. 31-36

REVISED AES AND ITS MODES OF OPERATION

Mohan H.S[1] and A. Raji Reddy[2]

ABSTRACT: In cryptography, a block cipher operates on blocks of fixed length, often 64 or 128 bits. If Advanced encryption standard (AES) specification is implemented, then the resulting algorithm mode is known as Electronic Code Book (ECB). This mode is weak and less secure. In order to improve the security, alternate modes are proposed: Output Feedback (OFB), Cipher Feedback (CFB) and Counter (CTR) modes. These modes further processes the ECB output to generate a keystream, which in turn is used for encrypting the plaintext with a simple EXOR operation. Hence, these modes act as stream ciphers. In this paper we propose a method of enhancing the security of existing AES algorithm by using key multiplication rather than conventional key addition, and for the revised AES the CFB,OFB, CTR modes are applied in place of ECB mode.

Keywords: Advanced encryption standard, electronic code book, output feedback, cipher feedback, counter mode.

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.

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.

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.

This paper presents the brief introduction to the cryptography and the important terms in section 1. Section 2 describes the AES algorithm and its evaluation. Section 3 describes a proposed Key multiplication is in place of Addroundkey. Section 4 describes the application of different modes of operation for the revised AES. Experimental results are 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

[1] Research Scholar Dr. MGR University, Chennai, India, E-mail: mohan_kit@yahoo.com.

[2] Professor & Head, Department of Electronics & Communication, Madanapalle Institute of Technology & Science, Madanapalle, India, E-mail: ar_reddy@yahoo.com.

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 [3]. 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 Universities 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 merely 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 Figure 1. After an initial addroundkey, a round function is applied 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.

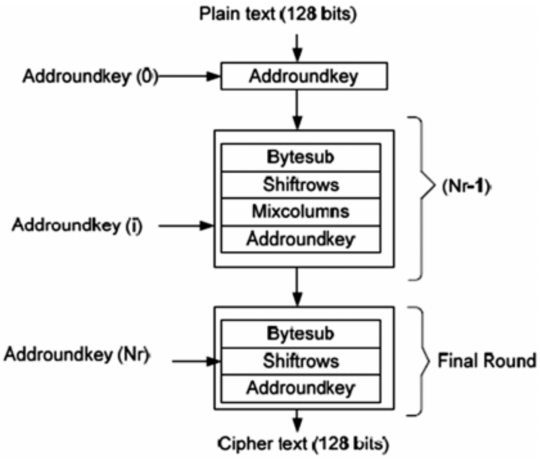


Figure 1: AES algorithm-Encryption Structure.

3. PROPOSED KEY MULTIPLICATION

3.1. Advantage of Key Multiplication

Only the AddRound Key stage makes use of the key. For this reason, the cipher begins and ends with the AddRound Key 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.

3.2. 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 0 × 1b.
  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.

4. DIFFERENT MODES OF OPERATIONS

4.1. Electronic Codebook (ECB)

The simplest of the encryption modes is the electronic codebook (ECB) mode. The message is divided into blocks and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all.

4.2. Stream Ciphers using OFB, CFB and CTR Modes

The ECB mode is the basic building block for constructing a stream cipher using OFB, CFB and CTR modes. In these modes, an Initialization vector is used as plain text to feed into ECB cipher. The ECB output is further processed to generate a keystream. This keystream is used to encrypt the plaintext.

Initialization Vector (IV)

All these modes (except ECB) require an initialization vector, or IV a sort of 'dummy block' to kick off the process for the first real block, and also to provide some randomization for the process. There is no need for the IV to be secret, in most cases, but it is important that it is never reused with the same key. For CFB, reusing an IV leaks some information about the first block of plaintext, and about any common prefix shared by the two messages. For OFB and CTR, reusing an IV completely destroys security.

4.2.1. Cipher Feedback (CFB)

The cipher feedback (CFB) mode, makes a block cipher into a self-synchronizing stream cipher. Operation is very simple and given below:

$$C_i = E_K(C_{i-1}) \oplus P_i,\ P_i = E_K(C_{i-1}) \oplus C_i,\ C_0 = IV$$

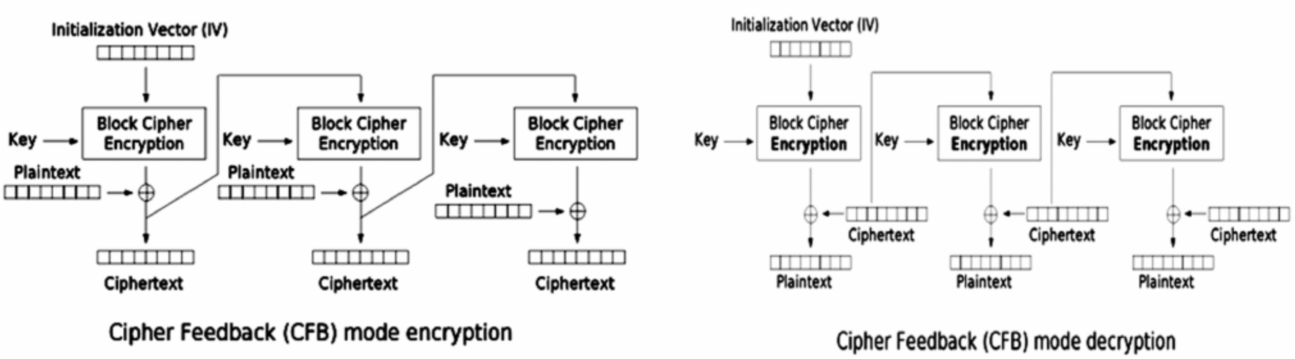

Initialization Vector (IV)
Key
Block Cipher Encryption
Plaintext
Ciphertext
Cipher Feedback (CFB) mode encryption
Initialization Vector (IV)
Key
Block Cipher Encryption
Ciphertext
Plaintext
Cipher Feedback (CFB) mode decryption

Figure 2: CFB Mode Encryption & Decryption.

In CFB mode, changes in the plaintext propagate forever in the ciphertext, and encryption cannot be parallelized. Also, decryption can be parallelized. When decrypting, a one-bit change in the ciphertext affects two plaintext blocks: a one-bit change in the corresponding plaintext block, and complete corruption of the following plaintext block. Later plaintext blocks are decrypted normally.

Because each stage of the CFB mode depends on the encrypted value of the previous ciphertext XORed with the current plaintext value, a form of pipelining is possible, since the only encryption step which requires the plaintext is the final XOR. This is useful for applications that require low latency between the arrival of plaintext and the output of the corresponding ciphertext, such as certain applications of streaming media.

4.2.2. Output Feedback (OFB)

The output feedback (OFB) mode makes a block cipher into a synchronous stream cipher: it generates keystream blocks, which are then XORed with the plaintext blocks to get the ciphertext. Just as with other stream ciphers, flipping a bit in the ciphertext produces a flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

Because of the symmetry of the XOR operation, encryption and decryption are exactly the same:

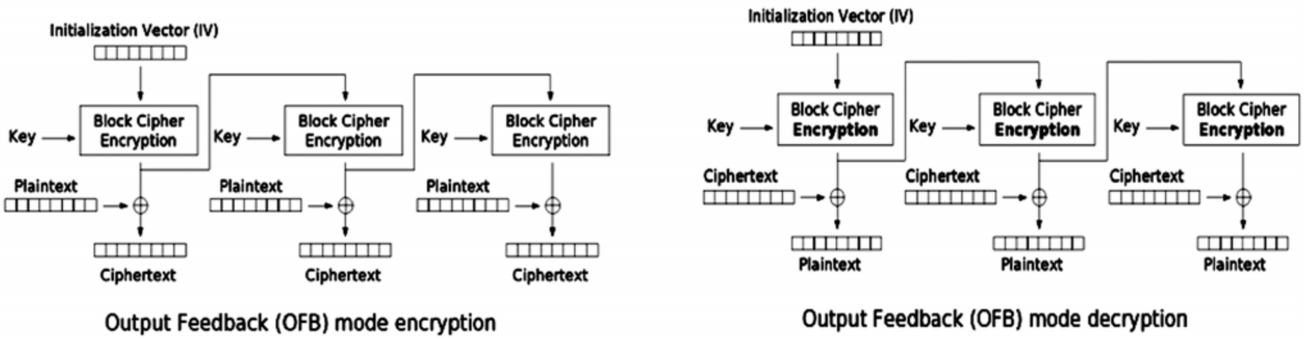

Initialization Vector (IV)
Key
Block Cipher Encryption
Plaintext
Ciphertext
Output Feedback (OFB) mode encryption
Initialization Vector (IV)
Key
Block Cipher Encryption
Ciphertext
Plaintext
Output Feedback (OFB) mode decryption

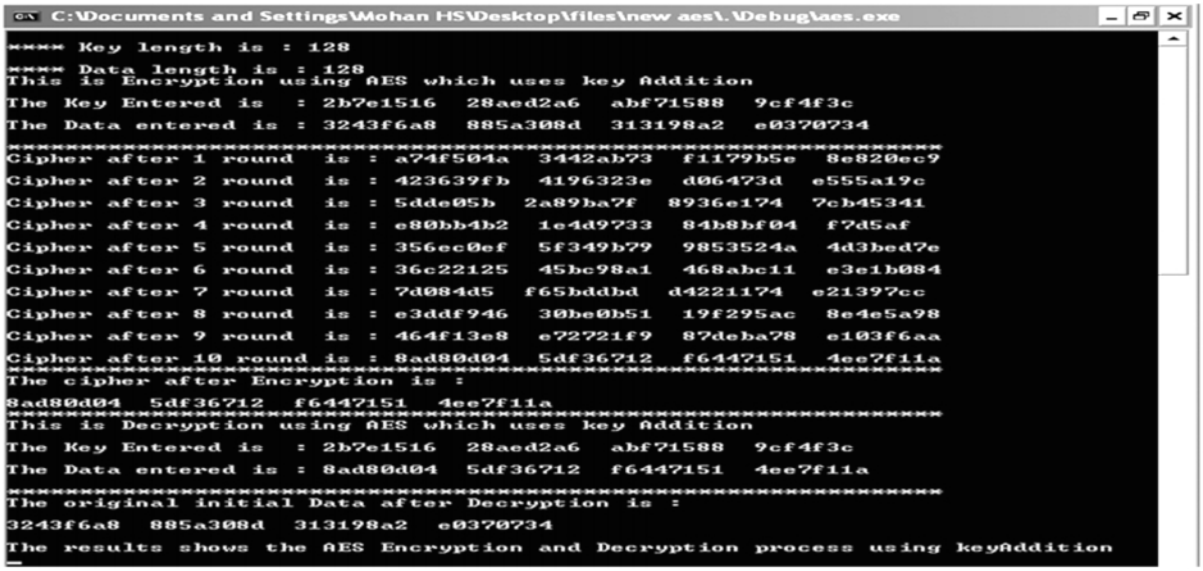
Figure 3: OFB Mode Encryption & Decryption.

Each output feedback block cipher operation depends on all previous ones, and so cannot be performed in parallel. However, because the plaintext or ciphertext is only used for the final XOR, the block cipher operations may be performed in advance, allowing the final step to be performed in parallel once the plaintext or ciphertext is available.

4.2.3. Counter (CTR)

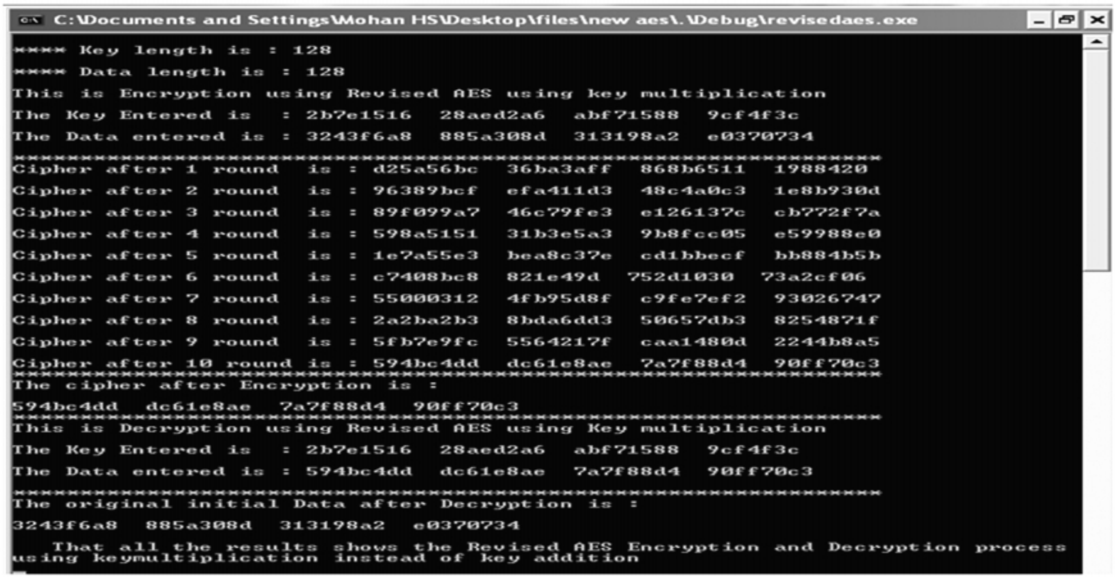
Like OFB, counter mode turns a block cipher into a stream cipher. It generates the next keystream block by encrypting successive values of a "counter". The counter can be any simple function which produces a sequence which is guaranteed not to repeat for a long time, although an actual counter is the simplest and most popular. CTR mode has similar characteristics to OFB, but also allows a random access property during decryption, and is believed to be as secure as the block cipher being used. Note that the nonce in this graph is the same thing as the initialization vector (IV) in the other graphs. The IV/nonce and the counter can be concatenated, added, or XORed together to produce the actual unique counter block for encryption.

5. EXPERIMENTAL RESULTS


C:\Documents and Settings\Mohan HS\Desktop\files\new aes\.\Debug\aes.exe

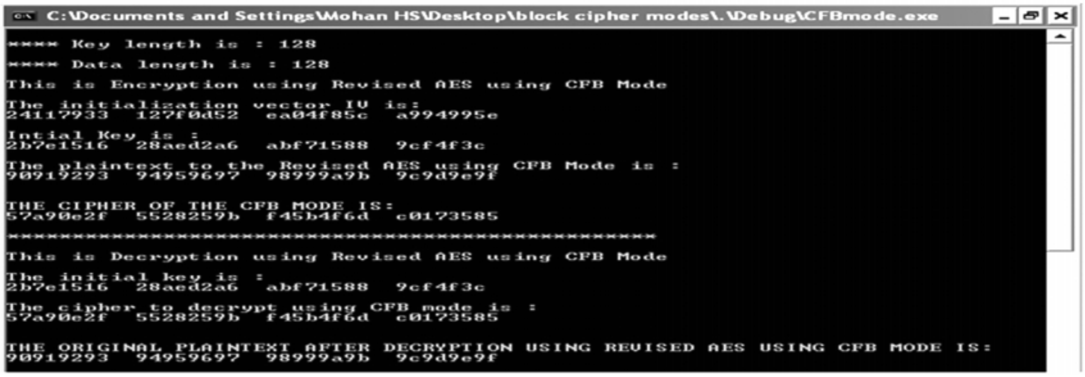
```
**** Key length is : 128
**** Data length is : 128
This is Encryption using AES which uses key Addition
The Key Entered is  : 2b7e1516  28aed2a6  abf71588  9cf4f3c
The Data entered is : 3243f6a8  885a308d  313198a2  e0370734
****************************************************************
Cipher after 1 round  is : a74f504a  3442ab73  f1179b5e  8e820ec9
Cipher after 2 round  is : 423639fb  4196323e  d06473d  e555a19c
Cipher after 3 round  is : 5dde05b  2a89ba7f  8936e174  7cb45341
Cipher after 4 round  is : e80bb4b2  1e4d9733  84b8bf04  f7d5af
Cipher after 5 round  is : 356ec0ef  5f349b79  9853524a  4d3bed7e
Cipher after 6 round  is : 36c22125  45bc98a1  468abc11  e3e1b084
Cipher after 7 round  is : 7d084d5  f65bddbd  d4221174  e21397cc
Cipher after 8 round  is : e3ddf946  30be0b51  19f295ac  8e4e5a98
Cipher after 9 round  is : 464f13e8  e72721f9  87deba78  e103f6aa
Cipher after 10 round is : 8ad80d04  5df36712  f6447151  4ee7f11a
****************************************************************
The cipher after Encryption is :
8ad80d04  5df36712  f6447151  4ee7f11a
****************************************************************
This is Decryption using AES which uses key Addition
The Key Entered is  : 2b7e1516  28aed2a6  abf71588  9cf4f3c
The Data entered is : 8ad80d04  5df36712  f6447151  4ee7f11a
****************************************************************
The original initial Data after Decryption is :
3243f6a8  885a308d  313198a2  e0370734
The results shows the AES Encryption and Decryption process using keyAddition
_
```

Figure 4: Encrypt and Decrypt using the Existing AES Which uses Key Addition & ECB Mode.


C:\Documents and Settings\Mohan HS\Desktop\files\new aes\.\Debug\revisedaes.exe

```
**** Key length is : 128
**** Data length is : 128
This is Encryption using Revised AES using key multiplication
The Key Entered is  : 2b7e1516  28aed2a6  abf71588  9cf4f3c
The Data entered is : 3243f6a8  885a308d  313198a2  e0370734
******************************************************************
Cipher after 1 round  is : d25a56bc  36ba3aff  868b6511  1988420
Cipher after 2 round  is : 96389bcf  efa411d3  48c4a0c3  1e8b930d
Cipher after 3 round  is : 89f099a7  46c79fe3  e126137c  cb772f7a
Cipher after 4 round  is : 598a5151  31b3e5a3  9b8fcc05  e59988e0
Cipher after 5 round  is : 1e7a55e3  bea8c37e  cd1bbecf  bb884b5b
Cipher after 6 round  is : c7408bc8  821e49d  752d1030  73a2cf06
Cipher after 7 round  is : 55000312  4fb95d8f  c9fe7ef2  93026747
Cipher after 8 round  is : 2a2ba2b3  8bda6dd3  50657db3  8254871f
Cipher after 9 round  is : 5fb7e9fc  5564217f  caa1480d  2244b8a5
Cipher after 10 round is : 594bc4dd  dc61e8ae  7a7f88d4  90ff70c3
******************************************************************
The cipher after Encryption is :
594bc4dd  dc61e8ae  7a7f88d4  90ff70c3
******************************************************************
This is Decryption using Revised AES using Key multiplication
The Key Entered is  : 2b7e1516  28aed2a6  abf71588  9cf4f3c
The Data entered is : 594bc4dd  dc61e8ae  7a7f88d4  90ff70c3
******************************************************************
The original initial Data after Decryption is :
3243f6a8  885a308d  313198a2  e0370734
   That all the results shows the Revised AES Encryption and Decryption process
using keymultiplication instead of key addition
```

Figure 5: Encrypt & Decrypt using Revised AES Which uses Key Multiplication & ECB Mode.

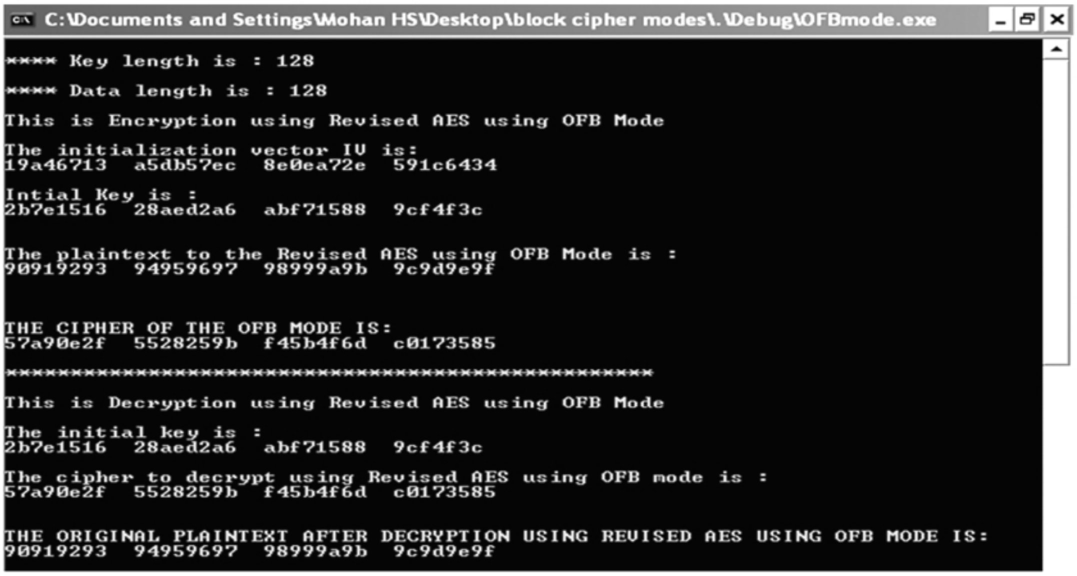

C:\Documents and Settings\Mohan HS\Desktop\block cipher modes\.\Debug\CFBmode.exe

```
**** Key length is : 128
**** Data length is : 128
This is Encryption using Revised AES using CFB Mode
The initialization vector IV is:
24117933  127f0d52  ea04f85c  a994995e
Intial Key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c
The plaintext to the Revised AES using CFB Mode is :
90919293  94959697  98999a9b  9c9d9e9f

THE CIPHER OF THE CFB MODE IS:
57a90e2f  5528259b  f45b4f6d  c0173585
**************************************************
This is Decryption using Revised AES using CFB Mode
The initial key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c
The cipher to decrypt using CFB mode is :
57a90e2f  5528259b  f45b4f6d  c0173585

THE ORIGINAL PLAINTEXT AFTER DECRYPTION USING REVISED AES USING CFB MODE IS:
90919293  94959697  98999a9b  9c9d9e9f
```

Figure 6: Encryption and Decryption using Revised Revised AES using CFB Mode.


C:\Documents and Settings\Mohan HS\Desktop\block cipher modes\.\Debug\OFBmode.exe

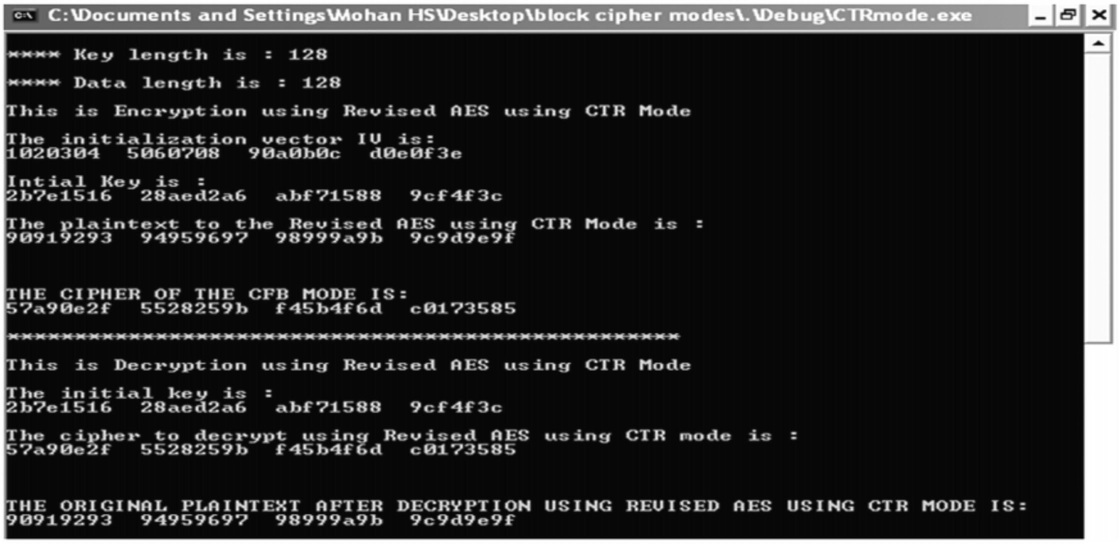
```
**** Key length is : 128
**** Data length is : 128
This is Encryption using Revised AES using OFB Mode
The initialization vector IV is:
19a46713  a5db57ec  8e0ea72e  591c6434
Intial Key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c

The plaintext to the Revised AES using OFB Mode is :
90919293  94959697  98999a9b  9c9d9e9f


THE CIPHER OF THE OFB MODE IS:
57a90e2f  5528259b  f45b4f6d  c0173585
**************************************************
This is Decryption using Revised AES using OFB Mode
The initial key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c
The cipher to decrypt using Revised AES using OFB mode is :
57a90e2f  5528259b  f45b4f6d  c0173585

THE ORIGINAL PLAINTEXT AFTER DECRYPTION USING REVISED AES USING OFB MODE IS:
90919293  94959697  98999a9b  9c9d9e9f
```

Figure 7: Encryption and Decryption using AES using OFB Mode.



C:\Documents and Settings\Mohan HS\Desktop\block cipher modes\.\Debug\CTRmode.exe

```
**** Key length is : 128

**** Data length is : 128

This is Encryption using Revised AES using CTR Mode

The initialization vector IV is:
1020304  5060708  90a0b0c  d0e0f3e

Intial Key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c

The plaintext to the Revised AES using CTR Mode is :
90919293  94959697  98999a9b  9c9d9e9f


THE CIPHER OF THE CFB MODE IS:
57a90e2f  5528259b  f45b4f6d  c0173585

**************************************************

This is Decryption using Revised AES using CTR Mode

The initial key is :
2b7e1516  28aed2a6  abf71588  9cf4f3c

The cipher to decrypt using Revised AES using CTR mode is :
57a90e2f  5528259b  f45b4f6d  c0173585


THE ORIGINAL PLAINTEXT AFTER DECRYPTION USING REVISED AES USING CTR MODE IS:
90919293  94959697  98999a9b  9c9d9e9f
```

Figure 8: Encryption and Decryption using Revised AES using CTR Mode.

6. CONCLUSION

The basic design of an encryption algorithm is based upon the strength of diffusion and confusion. This paper explored diffusion and confusion 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. This scheme is both efficient and highly secure. Test vectors are generated and tabulated for the existing AES and the revised AES algorithm. Existing AES uses ECB mode in block cipher, but as an application of the revised AES algorithm, a stream cipher is also implemented and the results using CFB, OFB and CTR mode are tabulated.

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, pp. 81-83.

[7] 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.

[8] 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, 2, No. 9, September 2010.

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

[10] 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, 8, No. 6, September 2010, pp. 128-132.

[11] R. Elumalai, A. Raji Reddy., "Improving Diffusion Power of AES Rijndael with 8 × 8 MDS Matrix", International Journal of Scientific and Engineering Research, 2, March-2011.