International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 11 | Nov -2017 www.irjet.net p-ISSN: 2395-0072

Enhanced Advanced Encryption Standard (E-AES): Using ESET

Harsh Vardhan Singh[1], Abhishek Dhama[2], Gaurav Kumar[3], Amit Kumar Sharma[4]

[123]B.Tech Student, Department of CSE, Babu Banarasi Das Institute of Engineering and Technology, Ghaziabad
[4]Assistant Professor, Department of CSE, Babu Banarasi Das Institute of Engineering and Technology, Ghaziabad

---

ABSTRACT - *Cryptography is the art of secret writing. It is conversion of data into ciphered code that can be deciphered and sent across any desired network (public or private). It is the science and art of creating secret codes. There are two types of cryptography: - Symmetric and Asymmetric cryptography. Symmetric cryptography is the fastest and commonly used type of algorithm like DES, AES, Blowfish. It is the cryptography in which only one key is their which is shared by sender and receiver. Asymmetric cryptography is the cryptography in which the two keys are their i.e., public and private key. In November 26, 2001 National Institute of Standard and Technology approved Advanced Encryption Standard also known by its original name Rijndael. The AES algorithm having the capacity of using 128,192,256-bit key to encrypt/decrypt 128 bits Block size. AES is now considered to be insecure for many applications. A 16 years old standard is still in use which is not be advisable to use because the key size is too small and possible to brute-force in finite time on modern processor. This research paper purpose a new scheme of Symmetric Key algorithm for AES using Extra Secure Encryption Technique(ESET) which capable of using cryptographic symmetric key of 2048-bits to encrypt and decrypt data in blocks of 1024-bits. This technique provides more security and increases the efficiency with different key length settings. In other words, it takes around four billion times longer to factor a 2048-bit key.*

Keywords – Advance Encryption Standard (AES), AES-2048, Cryptography, Decryption, ESET, E-AES, Encryption,2048-bit Key, 1024-Bit Data Block

1. INTRODUCTION

Cryptography is the technique where the "Simple text" i.e., the data to be secured is converted into "cipher text" which cannot be easily identified by unauthorized users. It is powerful tool in providing confidentiality, authenticity, integrity, and security from unauthorized use. The reason behind that networks often involve even greater risks from attackers due to this data is often secured with encryption, plausibly in combination with other controls.

The most important type of the encryption type is the symmetric key encryption. In the symmetric key encryption (Fig.1) both for the encryption and decryption process the same key is used. Hence the secrecy of the key is maintained and it is kept private.

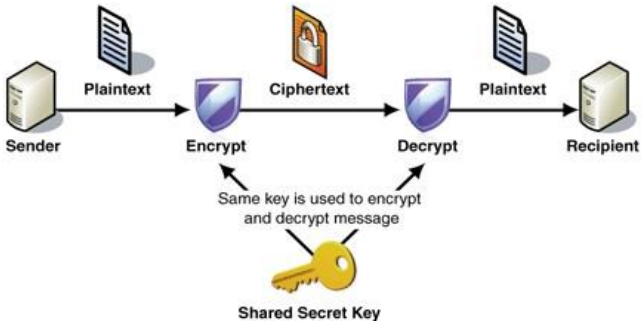

Fig-1: Symmetric key cryptography

Symmetric algorithms have many advantage over Asymmetric algorithm because it's not consuming too much of CPUs power and it works with very high speed in encryption. A block cipher is taken as the input, a key and input, and then the output block will be same in size in the symmetric key encryption. Though DES, Triple DES, AES and Blowfish are symmetric key cryptographic algorithm, and they have the ability to secure data. AES is a symmetric key cryptography which is used widely. It is approved by National Institute of Standards and Technology (NIST) in 2001 and specifies a (Federal Information Processing Standards) FIPS approved cryptographic algorithm that can be used to protect electronic data because of their fast and secure process. Various VPN network provider use AES for their secure communication.

1.1. Algorithm Specification

AES algorithm involves input block(Eb), output block and the State. Advanced Encryption algorithm involves the Cipher Key K, which is 128, 192, or 256 bits in length. This Cipher key length is represented by notation Ek. This show that the number of words in the Cipher Key is 32-bit. The length of all of these are 128 bits in length. Input block represented by notation Eb. This shows the number of words in the State is 32-bit.

For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by Er, where Er = 10 when Ek = 4, Er = 12 when Ek = 6, and Er = 14 when Ek = 8.

The only Key-Block-Round combinations that conform to this standard are given below: -

	Key Length(Ek words)	Block Size(Eb words)	Number of Rounds(Er)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Table-1: Relation between key length, block size and number of rounds

1.2. Drawback of AES

16 years old standard is still in use which is not be advisable to use. Some of the known attacks on AES are Biclique Cryptanalysis [2], Related-Key Cryptanalysis [3], and Improved Related-Key Impossible Differential Attacks [4], Cache-timing attacks on AES [5], AES power attack [6].

For AES-128, the key can be recovered with a computational complexity of 2126.1 using the biclique attack [2]. For biclique attacks on AES-192 and AES-256, the computational complexities of 2189.7 and 2254.4 respectively apply. Related-key attacks [3] can break AES-192 and AES-256 with complexities 2176 and 299.5, respectively.

On July 1, 2009, Bruce Schneier blogged[7] about a relatedkey attack on the 192-bit and 256-bit versions of AES, discovered by Alex Biryukov and Dmitry Khovratovich,[8] which exploits AES's somewhat simple key schedule and has a complexity of 2119. In December 2009 it was improved to 299.5. This is a follow-up to an attack discovered earlier in 2009 by Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić, with a complexity of 296 for one out of every 235 keys [9]. In November 2010 Endre Bangerter, David Gullasch and Stephan Krenn published a paper which described a practical approach to a "near real time" recovery of secret keys from AES-128 without the need for either cipher text or plaintext. The approach also works on AES-128 implementations that use compression tables, such as OpenSSL [10]. Like some earlier attacks this one requires the ability to run unprivileged code on the system performing the AES encryption, which may be achieved by malware infection far more easily than commandeering the root account [11].

2. PROPOSED ALGORITHM SPECIFICATION

In this research paper we proposed more advanced version of AES called ENHANCED ADVANCED ENCRYPTION STANDARD (E-AES) algorithm. In this, the length of the input block, the output block and the State is increase to 1024 bits. Now due to this the number of words are also increased. Now Eb=16 using 128-bit words.

We also increase key size is this algorithm. The length of Cipher Key Ck is now 2048-bits. Now due to this enhancement in the size of key length, the new value of Ek changes to 16 which reflects the number of 128-bit words. Now these changes in block size and key size reflects changes in number of rounds. For the purposed E-AES algorithm, the number of rounds to be performed during the execution of the algorithm is Er=64.

The Key-Block-Round combinations that conform to this standard are given below: -

	Key Length((Ek words)	Block Size (Eb words)	Number of Rounds (Er)
E-AES-2048	16	16	64

Table–2: Relation between key length, block size and number of rounds in E-AES-2048 bit Key Size

The purposed E-AES algorithm uses four different transformations which are based on byte orientation. There is a series of steps which apply in every round during the transformation plain text to Ciphertext or vice-versa: -

- Byte substitution which uses S-box called substitution table for create state matrix,
- Shifting of rows in matrix from one side to other side (Right to left)
- Mixing the data in each step within each column of the State matrix using special function,
- Finally, Addition of a Round Key to the final State matrix and proceed to next round.

2.1. Encryption of Plaintext into Ciphertext

At the start of the Encryption of Plaintext in to Cipher text in Advanced encryption, the input is copied to the State matrix. After an initial Round Key addition, the State array is transformed by implementing a round function 64 times, with the final round differing slightly from the first Nr -1 rounds. The final State is then copied to the output.

The round function is parameterized using a key schedule that consists of a one-dimensional array of four-byte words derived using the Key Expansion routine described in

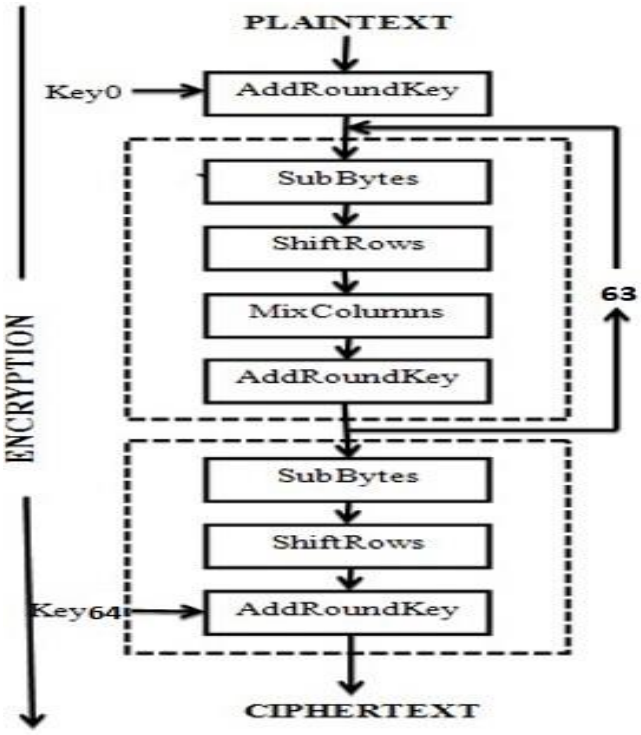

Fig-2: Encryption Process

The working of Encryption is shown in above Fig. 4. The encryption process involve these four steps - SubBytes(), ShiftRows(), MixColumns(), and AddRoundKey() – All these process of Encryption of Plaintext into Cipher text are described in the following subsections.

1) SubBytes():

The SubBytes() step involve a byte substitution that operates independently on each byte of the State matrix using a substitution table (S-box). S-box (Fig. 5), which is involve in the transformation. It is constructed by composing two transformations:

- Take the multiplicative inverse in the finite field GF(2^8)

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$$

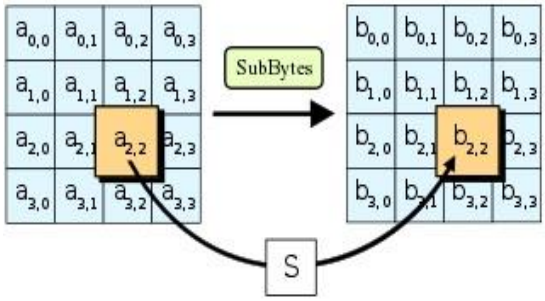

Fig-3: Process of SubBytes()

- Then apply the affine transformation over GF(2).

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Fig-4: Substitution Table(S-Box)

The SubBytes() step is very important step which is very initial step during the implementation of E-AES. S-box need to secure the plain text and perform the all the necessary operation. To understand the working, I take an example in which we transform the plain text in to their corresponding ASCII value such that is transform into understanding language of computer language.

For example, if s[1,1] ={53}, then the substitution value would be determined by the intersection of the row with index '5' and the column with index '3' in Fig. 5. This would result in s' 1,1 having a value of {ed}.

2) ShiftRow():

In the ShiftRows() function, rows of state matrix is shifted to right as seen in fig:

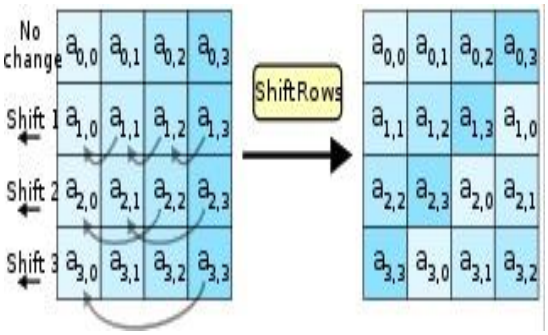

Fig-5: Process of ShiftRow()

The pseudo code for implementation of ShiftRow() :

```
ShiftRows(byte state[416,Eb])
    begin byte t[Eb]
        for r = 1 step 1 to 3for c = 0 step 1 to Nb – 1
```

```
            t[c] = state[r, (c + h[r,Eb]) mod Eb]
        end for
            for c = 0 step 1 to Eb– 1
                state[r,c] = t[c]
            end for
        end for
    End
```

3) MixColumns():

The MixColumns() function operates on the State matrix column-by-column, which use XOR function to works over the columns of one state matrix to another key matrix. The working of MixColums() is shown in fig

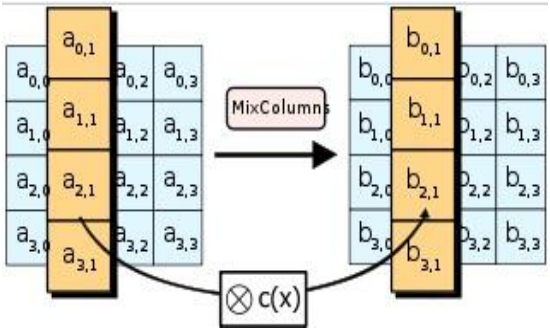

Fig-6: Process of MixColumns()

4) AddRoundKey():

In the AddRoundKey() function, a Round Key is use to add in the State matrix by a simple bitwise XOR operation. The new Round Key is use every time when the encryption perform in every state(64 times). The working of AddRoundKey() is shown in fig:

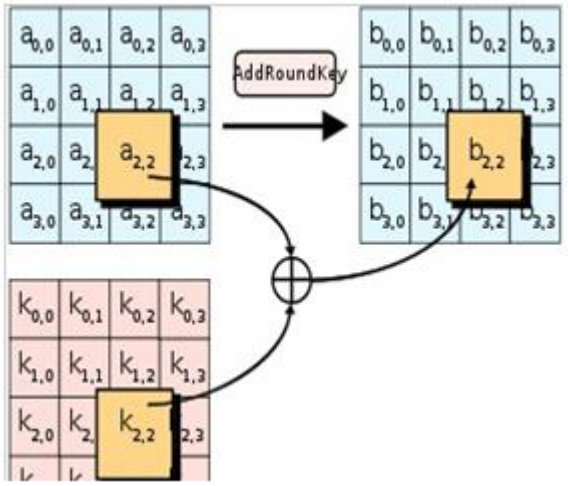

Fig-7: Process of AddRoundKey()

2.2. Key Expansion

The E-AES algorithm takes the Secret Key, K, and performs a Key Expansion routine to generate a key schedule. The expansion of the input key into the key schedule proceeds according to the pseudo code:

```
KeyExpansion(byte key[16*Ek], word w[Eb*(Er+1)], Ek)
begin
word temp
i = 0
        while (i < Ek)
            w[i]=word(key[16*i], key[16*i+1], key[16*i+2], key[16*i+3]], key[16*i+4]], key[16*i+5] ], key[16*i+6] ], key[16*i+7]), key[16*i+8]), key[16*i+9]),key[16*i+10]),key[16*i+11]), key[16*i+12]), key[16*i+13]), key[16*i+14]), key[16*i+15])
            i = i+1
        end while
    i = Ek
    while (i < Eb * (Er+1))
        temp = w[i-1]
        if (i mod Ek = 0)
            temp = SubWord(RotWord(temp))
            xor Rcon[i/Ek]
        else if (Ek > 12 and i mod Ek = 8)
            temp = SubWord(temp)
        end if
        w[i] = w[i-Ek] xor temp
        i = i + 1
    end while
end
```

The Key Expansion routine for 2048-bit (Ek = 16) is slightly different than 128,196 or 256-bit key expansion.

2.3. Decryption of Ciphertext into Plaintext

The transformation of plaintext in to cipher text in Sec can be inverted and then implementation in reverse order gives us procedure for decryption of Ciphertext into plain text. Decryption steps are implemented as shown in fig.

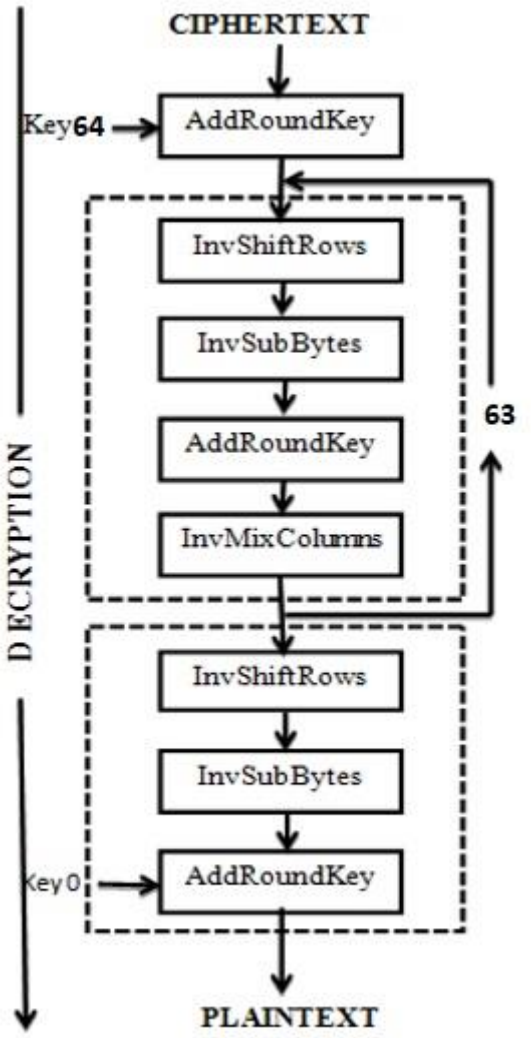

Fig-8: Process of Decryption

The transformations of Ciphertext to plain text involve inverse order of steps involved in encryption such that - InvShiftRows (), InvSubBytes(), InvMixColumns(), and AddRoundKey() – process the State and are described in the following subsections.

1) InvShiftRows():

This is the reverse of the ShiftRows() transformation. The bytes in the state matrix are cyclically shifted over different numbers of bytes (offsets). The bottom three rows are cyclically shifted by previous bytes

2) InvSubBytes():

This is the reverse on SubBytes() transformation. In this inverse of the byte substitution transformation, in which the inverse of the S-box is applied to each byte of the State matrix.

3) InvMixColumns():

This is the reverse of the MixColumns() transformation. The operation is apply on the Colum-to-Colum of state matrix.

4) InvAddRoundKey():

This is the reverse of AddRoundKey() in which state matrix performed XOR operation with CipherKey.

3. FUTURE SCOPE

E-AES uses 64 round of encryption of plain text which takes 118 times more factor in-compare of other encryption technique. So it's is very hard to decrypt the text which is encrypted using E-AES. It can also use in communication purpose between sender and receiver where confidential and important data sent. However, the symmetric key cryptography has their own disadvantages but this Enhanced Advanced Encryption Standard Using Extra Secure Encryption Technique is easy to implement and very fast in working plus 2048-bit key provide more enhanced security.

4. CONCLUSION

Due to the increasing needs for secure communications, encryption algorithm plays an important role in networking where secure data packets sent over network. There data are vulnerable to attacks. The purposed E-AES is being used in various purpose such as Archive and compression tools (7z, WINRAR, WinZip, UltraISO, Demon Tool, Nero), Encrypting File System in Windows, Disk encryption tools (DiskCryptor, BitLocker, TrueCrypt, Private Disk), security in data centre (OVH, BigRock, Hentezer). E-AES implement in communication purpose in android mobile also where short message service (SMS) plain text can be encrypted in Ciphertext and sent over to receiver. As SMSs are easily trackable and vulnerable to attack and network provider can also read our confidential messages so E-AES plays very important role to secure the communication between two parties. The larger key size makes the algorithm more secure, and the larger input block increases the throughput.

ACKNOWLEDGMENT

Proposed E-AES using ESET algorithm is enhancement of standard AES. We thank to Amit Kumar Sharma (Assistant Professor of CSE department) for their helpful guidance and precious time. We do not hold any rights on original standards version of AES, we gather the information from internet and implement their idea on existing AES algorithm.

REFERECES

[1] US National Institute of Standards and Technology Advanced Encryption Standard, Federal Information Processing Standards Publications No. 197, 2001.

[2] Andrey Bogdanov, Dmitry Khovratovich and Christian Rechberge Biclique Cryptanalysis of the Full AES 16 Aug 2011

[3] Alex Biryukov and Dmitry Khovratovich, Related-Key Cryptanalysis of the Full AES192 and AES-256, Advances in Cryptography, proceedings of ASIACRYPT2009, Lecture Notes in Computer Science 5912, pp. 1–18, Springer, 2009.

[4] Key Impossible Differential Attacks on Reduced-Round AES-192, Proceedings of
Selected Areas in Cryptography 2006, Lecture Notes in Computer Science 4356,pp. 15–27, Springer, 2007.

[5] Daniel J. Bernstein. Cache-timing attacks on AES.April2005.http://cr.yp.to/antiforgery/ca chetiming-20050414.pdf

[6] Guido Bertoni, Luca Breveglieri, Matteo Monchiero, Gianluca Palermo, and Vittorio Zaccaria, AES power attack based on induced cache miss and countermeasure. ITCC (1), 2005.

[7] Bruce Schneier (2009-07-01). "New Attack on AES". Schneier on Security, A blog covering security and security technology. Archived from the original on 8 February 2010. Retrieved 2010-03-11.

[8] Biryukov, Alex; Khovratovich, Dmitry (2009-12-04). "Related-key Cryptanalysis of the Full AES-192 and AES-256". Retrieved 2010-03-11.

[9] Nikolić, Ivica (2009). "Distinguisher and Related-Key Attack on the Full AES-256". Advances in Cryptology – CRYPTO 2009. Springer Berlin / Heidelberg. pp. 231–249. doi:10.1007/978-3-642-03356-8_14. ISBN 978-3-642-03355-1.

[10] Endre Bangerter, David Gullasch and Stephan Krenn (2010). "Cache Games – Bringing Access-Based Cache Attacks on AES to Practice".

[11] "Breaking AES-128 in realtime, no ciphertext required | Hacker News". News.ycombinator.com. Retrieved 2012-1223

[12] Added Advanced Encryption Standard (A-Aes): With 512 Bits Data Block And 512,768 And 1024 Bits Encryption Key. june 2014.www.ijictm.org/admin/html/mail/attach/2014-08-06-03-48-23.pdf