Optik 127 (2016) 2341–2345

Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

Development of modified AES algorithm for data security

Puneet Kumar*, Shashi B. Rana

Department of Electronics Technology, Guru Nanak Dev University, Regional Campus Gurdaspur, Gurdaspur, Punjab 143521, India

ARTICLE INFO

Article history:
Received 23 June 2015
Accepted 24 November 2015

Keywords:
Advance encryption standard (AES)
Cryptography
Symmetric key algorithm
Symmetric cipher

ABSTRACT

Cryptography is a process by which information or messages can be sent from one user to another user which provides several security services such as confidentiality, data integrity or authentication to the wireless communication system. As there is need for secure communication, efficient cryptographic processing is required for good system performance. One of the basic fundamental tools used in information security is known as the signature. Thus, the security for internet banking, account passwords, emails accounts password etc. requires text protection in digital media. This paper presents the security and compression for the data with the advance encryption standard (AES). In our research, we increase the number of rounds (Nr) to 16 for the encryption and decryption process of AES algorithm, which results in more security to the system. Experimental results and Theoretical analysis proved that this AES technique provide high speed as well as less transfer of data over the unsecured channels.

1. Introduction

It is already known that the use of internet in the present era is increasing at higher rate and demand of security is also increasing rapidly, many users are sharing public and private information over internet. This gives rise to the need of security as the data and information is very sensitive as its transmission is needed all the time. Encryption technique is one of the most important aspects which are very useful to secure confidential information. This encryption is implemented by using some traditional encryption techniques. But traditional encryption technique has some shortcomings in terms of security [1,2]. Therefore, the network security problem can be categorised into four areas: Secrecy, integrity control, authentication and non-repudiation [3]. Cryptography in its practice and is a study of technique for the secure communication in the presence of third parties called adversaries. It is about constructing and analysing the protocols which overcome the influence of adversaries and various aspects related to the information security.

The Advanced Encryption Standard (AES) is a Federal Information Processing Standard (FIPS) which was declared after an encryption algorithm standard competition by National Institute of Standards and Technology (NIST) in 2001. AES is one of the encryption techniques which are used most frequently because of its high efficiency and simplicity. It is the highly secure algorithm. AES is a symmetric block cipher uses the same key for the encryption as well as for decryption process. It has been found that the AES is different from the data encryption standard (DES). In AES, the block and key size can be chosen independently from 128, 160, 192, 224, 256 bits whereas in case of DES it is 56 bits. AES differ from DES as it not uses the feistel network. In feistel structure, half of the data block is generally used to modify the other half of the data block and then these halves are swapped. In case of AES the entire data block is processed in parallel during each round using substitutions and permutations. It has been found that the symmetric cipher is divided into two categories: stream cipher and block cipher.

In stream cipher, one symbol is generally used such as character or bit for the encryption and decryption process. It consists of Plaintext stream, Ciphertext stream and Key stream. Whereas, for block cipher encryption is done together with the plaintext symbol of m ($m > 1$) by creating the same size ciphertext symbol grouped together. From the definition, in a block cipher single key is generally used for the encryption even if the key consist of the multiple values.

2. Literature survey

This section involves the work done by the various researchers in the field of Advance Encryption Standard (AES) cryptographic algorithm for data security. Critical analysis has been done and finally the observations have been drawn at the end of this section.

Mandal et al. [4] worked on performance evaluation of cryptographic algorithms: DES and AES. These algorithms takes significant amount of computing resources such as simulation time,

* Corresponding author. Tel.: +91 9780552945.
E-mail addresses: pkumar3397@gmail.com (P. Kumar), shashi_rana12@yahoo.co.in (S.B. Rana).

http://dx.doi.org/10.1016/j.ijleo.2015.11.188
0030-4026/

memory usage and level of encryption are of major concern. In AES, avalanche effect is high as compared with the DES which is used in the financial applications. The more research can be done in the field of image and provide more security to the system. Park et al. [5] worked on methods for practical white-box cryptography. In this attacks are even stronger then the black box model. The main limitation of this scheme was changing of look up table which is very fast and strong in case of the white box and considered for the future research. Gaspar et al. [6] worked on efficient AES S-boxes implementation for non-volatile FPGAS. They proposed an efficient method for the implementation of AES byte substitution function (S-box). The proposed a solution which requires less space and is faster than the one implementing whole S-boxes in the logic area. The main limitation of this scheme was FPGA cannot be used for the low battery purposes. Selimis et al. [7] worked on applying low power technique in AES MixColumn\InvMixColumn transformation. They investigate the use of low power resources which increases the security needs and efficiency. Thus, the data paths which are of no use for the system are deactivated and increase the flexibility of the system for the better results. Wadi et al. [8] worked on high definition image encryption algorithm based on AES modification. They discussed block cipher algorithm well known AES as it is more secure. The main limitation of this scheme was the encryption/decryption time required was more and the attacks on the encryption algorithm can reduced the rounds. Goodwin et al. [9] worked on AES implementation with increased differential power analysis (DPA) resistance and low overhead. They investigate a side channel attack that exposed to potential weaknesses for the particular power analysis. Thus, they discussed improved strength against side channel attacks with a minimal additional hardware overhead. Berna et al. [10] introduced power analysis attack on an ASIC AES implementation. They worked on side channel attack in which it is not that easy to extract the secret information. They also showed the improvement in the correlation coefficient i.e. signal to noise ratio. The limitation of this scheme was the considerable amount of noise present during the measurement of stimulated attack and the real attack which was undertaken. Lu et al. [11] worked on fast implementation of AES cryptographic algorithms in smart cards. They proposed a chip operation system (COS) known as Nexcard which has been derived from the Microsoft windows. The AES encryption may attain the 0.65ms at clock 15 MHz on INEI-NEON SU66CX322P chip without the existence of the coprocessors. Therefore, they investigate AES embedded method which proved to be more secure and efficient for the security purposes on the smart cards. The drawback for this scheme was the turnaround time in case of the CSOD was 5 s [11,12].

Up till now the research for Advance Encryption Standard has been carried out with 128 bit key used which takes 10 rounds for the encryption and decryption process. In this work the extension to the key size and number of rounds has been done. The key size of 320 bits and number of rounds has been increased to 16 instead of 10 rounds for the 128 bit key used. Here, the encryption execution time has been calculated on the basis of input file size and shows that the proposed approach has more encryption time as compared to the other cryptographic algorithms. In Section 3, Methods and analysis of AES algorithm has been explained. Section 4, includes the Modified AES algorithm and key generation process has been explained. In Section 5, comparison of our proposed approach with various other cryptographic algorithms has been analysed. Finally in Section 6, Conclusion has been drawn.

3. Method and analysis

AES is a symmetric algorithm which was introduced in 2001. In this cryptographic technique one can use the secret key of any size depending on the cipher used. As it is symmetric algorithm, uses the same key for the encryption and decryption process and keys remain secret which are hard to guess by the hackers. AES generally uses the three different key sizes which are as 128, 192 and 256 bits. It has been found that the AES parameters depends on its key size been used. The encryption of data takes place with different rounds used as in case of 128 bits 10 rounds, in 192 bits 12 rounds and in 256 bits 14 rounds have been used till now [2,8]. The term rounds refer to the numbers of iteration used during the process of encryption and decryption. AES has been considered to be the best approach towards security of the system.

3.1. Block cipher modes of operation

In AES five modes of operations are generally been used when applied to the block ciphers in varieties of applications. Various blocks have been described as:

a) **Electronic Codebook Mode (ECB)**: In this mode block same key is used for the conversion of plaintext into a single ciphertext for every block of plaintext. This mode generally operates for the messages smaller than the block length. In case the longer messages, which have to be encrypted are first break down into blocks of required length by padding the last block if required. Therefore, ECB method generally operates for small amount of data that may resist to hackers [2].
b) **Cipher Block Chaining (CBC) Mode:** In this mode users requirement is that same plaintext blocks produces the different ciphertext blocks. Therefore, cipher block chaining generally allow the XORing of each plaintext with ciphertext of the previous rounds as it uses the same key [3].
c) **Cipher Feedback (CFB) Mode:** This type of a mode generally allows the conversion of block cipher into the stream cipher. It eliminate the need of padding for the entire message to be the integral number of blocks been used for the process. In this operation the left most bits are XORed with the first segment of the plaintext in order to produce the first unit of ciphertext which is then transmitted. For the encryption process, shift register is used for converting plaintext into the ciphertext [2].
d) **Output Feedback (OFB) Mode:** This mode is similar to the CFB mode as explained above. OFB eliminates the generation of same plaintext block to same ciphertext block by adopting an internal feedback mechanism which is independent on both the plaintext and ciphertext bit strings [2].
e) **Counter (CTR) Mode:** In this type of a mode the counter value has to be different for each plaintext block that is encrypted. During the encryption process, the counter is encrypted and then XORed with the plaintext in order to produce the ciphertext block without chaning. For decryption the process is reversed as it uses the same counter values and then XORed in order to get plaintext. The main advantage of this mode is simple design; provide hardware and software efficiency and security to the system [2].

3.1.1. Description of AES transformation used in encryption/decryption

3.1.1.1. Forward and inverse substitute bytes transformation:.

3.1.1.1.1. Forward substitute bytes transformation. It is a simple look up table as it consists of 16×16 matrix bytes values known as S-box. It contains permutation of 256 eight bit values in which each individual byte of state is mapped into a new byte of array. The leftmost byte is used as a row and rightmost byte is used as a column. For example the hexadecimal value {83} where 8 is row and 3 is column results to {*EC*} value from S-box [13].

3.1.1.1.2. Inverse substitute bytes transformation. It is known as InvSubBytes. It makes use of inverse S-box look up table. The input

Table 4.1
Proposed round numbers (Nr) for different data and key lengths.

Nr	Nb = 4	Nb = 6	Nb = 8	Nb = 10
Nk = 4	10	12	14	16
Nk = 6	12	12	14	16
Nk = 8	14	14	14	16
Nk= 8	16	16	16	16

{*EC*} hexadecimal value where *E* is row and *C* is column results to {83} from the inverse S-box [13].

3.1.1.2. Forward and inverse shift row transformation.

3.1.1.2.1. Forward shift row transformation. In this shifting of rows is performed cyclically to the left. Starting from the first row which remain unchanged, then to the second row, which is shifted by one byte cyclically to the left, then to the third row, which is shifted by two byte cyclically to the left and fourth row, is shifted by three bytes cyclically to the right and son on depending on the number of rows been used. This transformation is generally performed on the matrix consist of rows and column making the system more secure [2].

3.1.1.2.2. Inverse shift row transformation. It is the opposite of the forward shift row transformation as rows are shifted cyclically to the right. In this also the first row is unchanged, second row is shifted by one byte cyclically to the right, third row is shifted by two byte cyclically to the right, fourth row by three byte and so on the shifting is performed depending on matrix being used [2].

3.1.1.3. Forward and inverse mix column transformation.

3.1.1.3.1. Forward mix column transformation. In this operation is generally performed on individual column. Here each byte of a column is plotted into a new value which is function of all the four bytes in column. This transformation can be defined by the matrix multiplication on State. Therefore, each element in the product matrix is the sum of product of element of one row and one column [2,3].

3.1.1.3.2. Inverse mix column transformation. In this the whole state is to be multiplied with pre-defined matrix called inverse polynomial matrix. The inverse transformation matrix times the forward transformation matrix equals the identity matrix [2,3].

3.1.1.4. Forward and inverse add round key transformation.

3.1.1.4.1. Forward add round key transformation. In this the 128 bits of state are bitwise XORed with a 128 bits of the round key. Here the operation is performed as column wise between the four bytes of state column and one word of the round key can also be performed as byte level operation [3].

3.1.1.4.2. Inverse add round key transformation. In this it performs XOR procedure between the cipher text and expanded key corresponding to that particular iteration. E.g., if the diagrams on the left represent the cipher and the key values, the final value after it has generated by this step is shown on the right [2].

4. Modifies AES algorithm

It is important to know that the secret key can be of any size and in our proposed AES algorithm; key size of 320 bits is used instead of three different key sizes such as 128, 192 and 256 bits. From the research it has been found that the AES parameters depend on its key size. In proposed algorithm the number of rounds has been increased to 16 as it uses the 10 rounds for 128 bit key size. The security of the system is increased by increasing the number of rounds and results in providing privacy to the unauthorized users.

The proposed table has been drawn with the increase in number of rounds which helps in providing the more security to the system

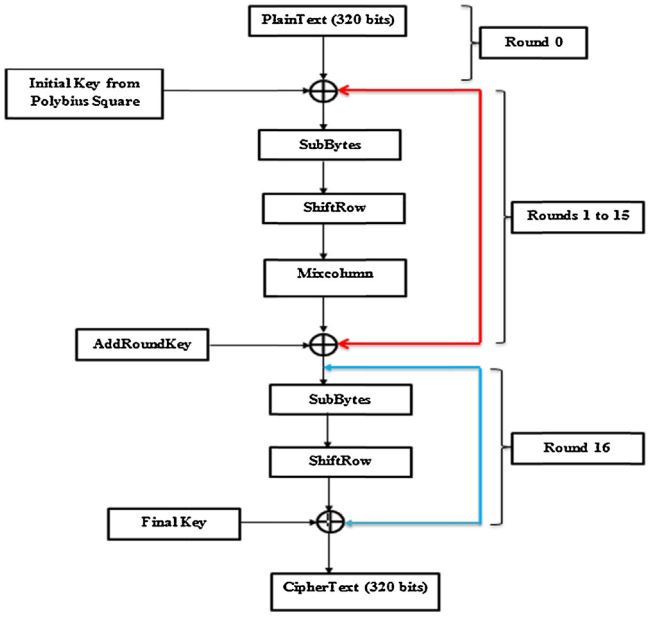


Fig. 4.1. AES proposed encryption process

and better performance. With the increase in number of rounds it will be difficult for the hackers to hack the system. It is believed that no simplification in transformation will allow breaking the AES algorithm. Therefore, key size of 320 bits has been chosen in order to provide the better results (as highlighted in Table 4.1).

4.1. Modified AES encryption process

It has been shown in Fig. 4.1.

It can be defined as the conversion of Plaintext to the Ciphertext. In AES encryption process instead of 10 rounds we have increase the number of rounds to 16. The initial key has been generated from the Polybius square. The encryption process undergoes the Subytes, ShiftRows, MixColumns and AddRound Key operations in AES which have been shown below:

4.1.1. Modified AES decryption process

Decryption is the process of converting cipherText into PlainText. Corresponding to the transformations in the encryption, Decryption process undergoes InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey are the transformations used in the decryption as shown in Fig. 4.2.

4.2. Key generation process

Polybius Square is used for generating the key with 6X6 matrix. The Polybius square consists of both the alphabets and numerals filled without repetition from the left to right and thus, help in providing the secure information [14]. The numerals are arranged in the ascending order from 0 to 9 (as shown in Table 4.2).

Table 4.2
Polybius square used for generating key.

0	1	2	3	4	5	
0	A	B	C	D	Y	4
1	E	F	G	H	Z	5
2	I	J	K	L	0	6
3	M	N	O	P	1	7
4	Q	R	S	T	2	8
5	U	V	W	X	3	9

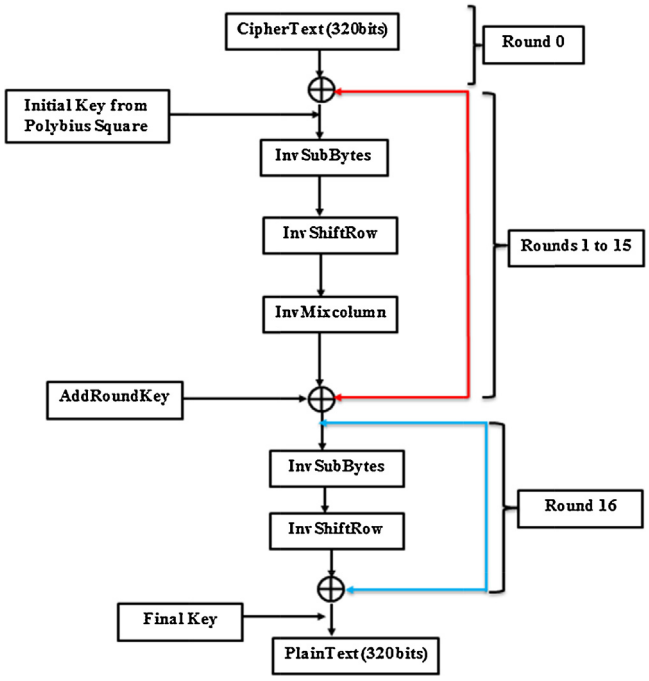


Fig. 4.2. AES decryption process

Table 4.2.1
Key generation using Polybius square.

Plaintext	S	E	C	U	R	I	T	Y	1	2	3
Position	1	2	3	4	5	6	7	8	9	10	11
Ciphertext	42	10	02	50	41	20	43	04	34	44	54

For encryption, first we have to look at the intersection of any row and column (with row number given first and column number given second) as the representation of the alphabet or numerals.

Let us take an Example: SECURITY123 is the message which is to be encoded then decoded in the original message (as shown in Table 4.2.1).

The plaintext which is in original text encrypted into the ciphertext with some codes and cannot be identify by the hacker. The Plaintext is SECURITY123 and the ciphertext is 4210025041204304344454. Similarly, the decryption process is followed. Thus, this results in generation of key used for encryption and decryption process.

5. Results and discussion

In this chapter comparison of proposed approach with various other cryptographic algorithms such as Data encryption standard (DES), Triple Data encryption standard (TDES) and Advance encryption standard (AES) has been done on the basis of Throughput for the different file size.

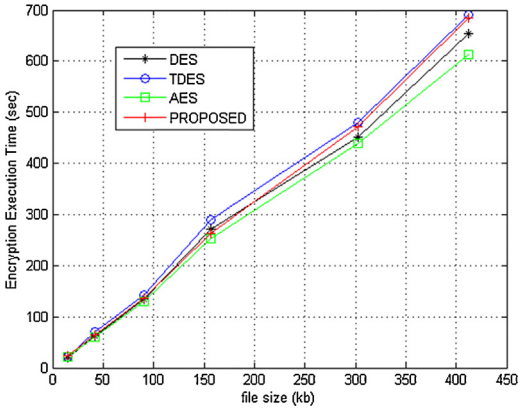
5.1. Encryption computational time

The encryption computational time can be defined as the time taken by the algorithm for conversion of plaintext to ciphertext. This encryption time can be used to calculate the encryption throughput of the algorithms. Performance parameter include the time taken by the algorithm for the encryption and decryption of input file size that is computational encryption time and computational decryption time used for the processing of file (Table 5.1).

Time for Different File Size In this the input file size of 15Kb is the encryption execution time for DES, TDES, AES and proposed

Table 5.1
Encryption execution time for different file size.

	Encryption execution time (s)			
Input file size (kb)	DES	TDES	AES	Proposed
15	20	22	21	23
42	63	71	60	65
90	134	142	129	137
157	272	289	253	263
303	451	479	439	471
412	654	691	612	684



Graph 5.1. Encryption execution time of different file sizes.

algorithm which are as 20, 22, 21, 23 s respectively. Therefore, for the file size of 412 Kb the encryption time are 654, 691, 612, 684 s, respectively. Thus, it shows that the proposed algorithm consumes more time for all the file sizes which increases the security to the system. With the help of the table: 5.1 graph has been drawn which shows that how the encryption execution time depends on the file size. Therefore, the proposed algorithm is faster than TDES. Finally comparison of proposed approach with various algorithms has been done (as shown in Graph 5.1).

6. Conclusion

In this paper study of advance encryption standard (AES) has been done. From the recent work it has been observed that by increasing the number rounds (Nr) to 16 make the system more secure and less prone to the attackers. With the increase in number of rounds it will take more computational time and will become difficult for the hacker to break the system. The generation of key has been done with the help of the Polybius square. Thus, the security of the system has been improved.

References

[1] Ajay Kakkar, M.L. Singh, P.K. Bansal, Efficient key mechanisms in multinode network for secured data transmission, Int. J. Eng. Sci. Technol. 2 (5) (2010) 787–795.
[2] Bruce Schneier, Applied Cryptography, Second ed., John Wiley & Sons, Singapore, January 1996.
[3] W. Staling, Network Security Essentials: Applications & Standards, 4th ed., Pearson Education, Upper saddle river, 2011.
[4] A.K. Mandal, C. Parakash, A. Tiwari, Performance evaluation of cryptographic algorithms: DES and AES, in: IEEE Students' Conference on Electrical, Electronics and Computer Science, 2012, pp. 1–5.
[5] J.-Y. Park, O. Yi, J.-S. Choi, Methods for practical whitebox cryptography, in: IEEE Transaction Paper, 2011, pp. 474–479.
[6] L. Gaspar, M. Drutarovsky, V. Fischer, N. Bochard, Efficient AES S-boxes implementation for non-volatile FPGAS, IEEE Transaction paper (2009) 649–653.
[7] G.N. Selimis, A.P. Fournaris, O. Koufopavlou, Applying low power techniques in AES MixColumn/InvMixColumn Transformations, in: IEEE Transaction, 2006, pp. 1088–1092.

[8] S.M. Wadi, N. Zainal, High definition image encryption algorithm based on AES modification, Springer Wireless Commun. (2014) 811–829.
[9] J. Goodwin, P.R. Wilson, Advanced encryption standard (AES) implementation with increased DPA resistance and low overhead, in: IEEE Transaction Paper, 2008, pp. 3286–3289.
[10] Mikhail J. Atallah, Marina Blanton, Nelly Fazio, Keith B. Frikken, Dynamic and efficient key management for access hierarchies, ACM Trans. Inf. Syst. Secur. 12 (3) (2009) 1–43.
[11] Chu-Hsing Lin, Dynamic key management schemes for access control in a hierarchy, Comput. Commun. 20 (1997) 1381–1385.
[12] Sheng Zhong, A practical key management scheme for access control in a user hierarchy, Comput. Secur. 21 (8) (2002) 750–759.
[13] W. Diffiee, M. Hellman, New directions in cryptography, IEEE Trans. Inf. Theory IT–22 (Nov) (1976) 644–654.
[14] Puneet Kumar, S.B. Rana, Development of modified Polybius technique for data security, Int. J. Innov. Eng. Technol. 5 (2015) 227–229.