A Random Key Generation Approach for Rijndael Algorithm

Dalal Naeem Hmood
Department of Computer Science, College of Science, University of Al-Nahrain, Baghdad-Iraq.
E-mail: Dal_scin@yahoo.com.

Abstract

One of security interlocution is data encryption/decryption whenever data being sent over communication lines may be protected by encrypting the message, that can be decrypted only by the authorized person receiving the message. In this paper a simple and fast approach is proposed to modify the key generation of Rijndael algorithm. The proposed approach uses randomize to generate a fixed key enough for the block size and number of rounded.

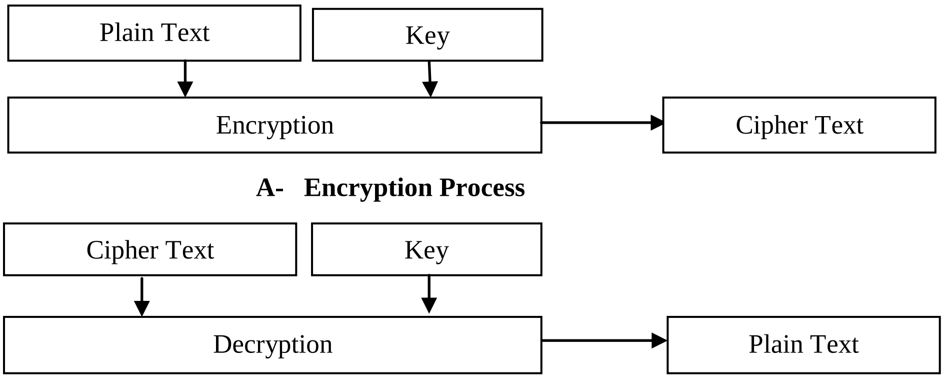
The proposed approach has been implemented successfully in interlocution system on different amount of ciphertext length.

Keywords: AES, Encryption, Interlocution, Key generation, Rijndael, Swap bits.

1. Introduction

The data being sent over communication lines may be protected by encrypting the message, that can be decrypted only by the authorized person receiving the massage. Although data encryption (also known as cryptography) has always been important, never has it been as necessary and widespread as it is today [1]. Fig.(1) depicts the encryption and decryption processes.

Block cipher algorithm can be used in networks environments, it has been paid attention to performance speed as well as security. Block cipher algorithm can operate with both software and hardware [2].



A- Encryption Process

B- Decryption Process

Fig.(1) Encryption/Decryption Processes.

There are many achievements occurred in the field of secure transmission; each suggests new method for secure chatting. The most useful ones are mentioned in the following:

In [3], cipher feedback and the skew tent map are used, the input text can be real-time encoded and the cipher text is sent via TCP/IP. With the properties of randomness of the map, and its sensitivity on system parameters and the initial conditions, the encrypted transmitting messages are difficult to be eavesdropped. In [4], built a secure chat server utilizing Public Key encryption to send secure chat messages across the internet, like AOL provide the convenience of conversing with people in real time. In [5], using OAEP/AES message encryption to provide security in real-time.

In this paper, a new approach for key generation of the Rijndael algorithm that uses randomize key is presented. The paper organized as follows: section 2, presents the proposed modification, section 3, illustrates the results, and conclusions are explained in section 4.

2. The Proposed Approach

Fig.(2), illustrates the block diagram of the proposed approach. The plaintext is forward to swap bits process then it is fed to Rijndael algorithm with modified key generation, then the ciphertext is obtained. The decryption process operates in reverse manner. The following subsections illustrate each process in details.

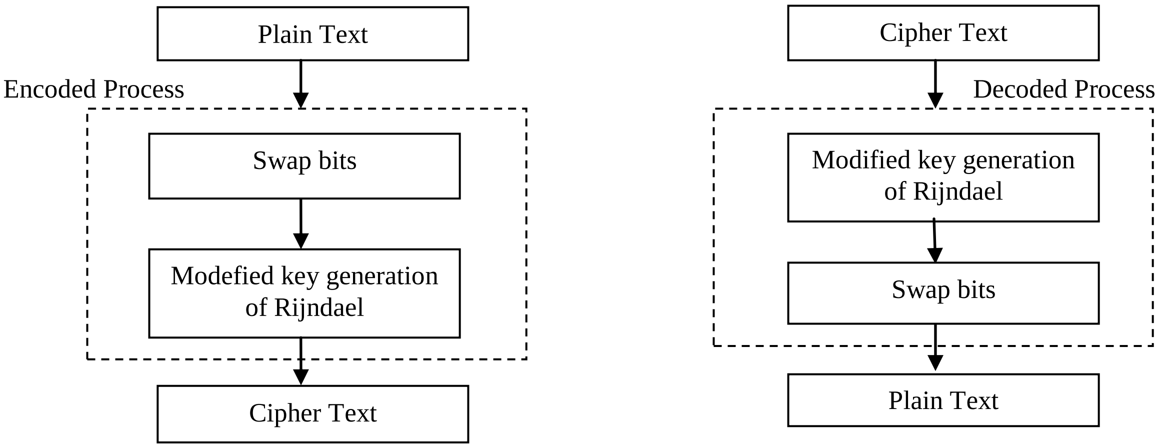


Fig.(2) Block Diagram of the Proposed Modification.

2.1 Swap bits

The best means of obtaining unpredictable random values is by measuring physical phenomena. For each character in the message represent in the binary system and then exchange the first bit with the last bit, the second bit with the previous bit of the last bit, and so on [6]. Fig.(3) illustrates the swap bits for the character has the value 99.

0	1	2	3	4	5	6	7
1	1	0	0	0	1	1	0

Swap the bits

0	1	2	3	4	5	6	7
0	1	1	0	0	0	1	1

Fig.(3) The Swap bits.

2.2 Modified Rijndael Algorithm

2.2.1 Rijndael Algorithm

The block cipher Rijndael was designed by Joan Daemen and Vincent Rijmen. The Rijndael algorithm is a new generation symmetric block cipher that supports key sizes of 128, 192 and 256 bits. For input, output or cipher key denoted by ***a***, the bytes in the resulting array will be referenced using one of the two forms, ***an*** or ***a[n]***, where n will be in one of the following ranges [1], see Table (1):

Table (1)
The Range of Cipher Key.

Key length (in bits)	*Block length Nb (in bytes)*
128	$0 \leq n< 16$
192	$0 \leq n < 24$
256	$0 \leq n < 32$

All byte values in the advanced encryption standard (AES) algorithm will be presented as the concatenation of its individual bit values (0 or 1) between braces in the order $\{b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0\}$. These bytes are interpreted as finite field elements using a polynomial representation [2], see equation 1:

$$b_7x^7+b_6x^6+b_5x^5+b_4x^4+b_3x^3+b_2x^2+b_1x+b_0= \sum_{i=0}^{7} b_i x^i \quad \ldots\ldots\ldots\ldots (1)$$

The state is a two-dimensional array of bytes consists of four rows of bytes, the column number being the block length divided by bit length (for example, divided by 32). The input is copied into the state array. The cipher or decipher are then conducted on this state array, after which its final value is copied to the output. The cipher key similarly is an array with 4 rows, but the key length divided by 32 to give the number of columns [7].

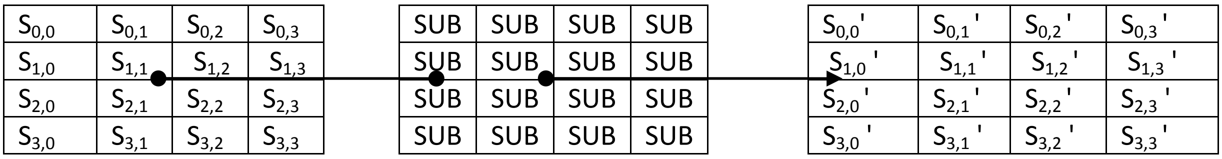
The block sizes can mirror those of the keys, see the Table (2), that representation the variable number of rounds (number of rounds depending on key/ block sizes) [7].

Table (2)
Key-Block-Round Combinations when block size (N_b)=4.

	Key length (Nk)	*Number of rounds (Nr)*
AES-128	4	10
AES-192	6	12
AES-256	8	14

AES algorithm uses four fundamental transformations occurs on each block are as follows:

1) **The Sub Bytes Transformation (Sub Bytes):-** This part is nonlinear and operates on each of the State bytes independently the invertible S-box (substitution table) is made up of two transformations, see Fig. (4).



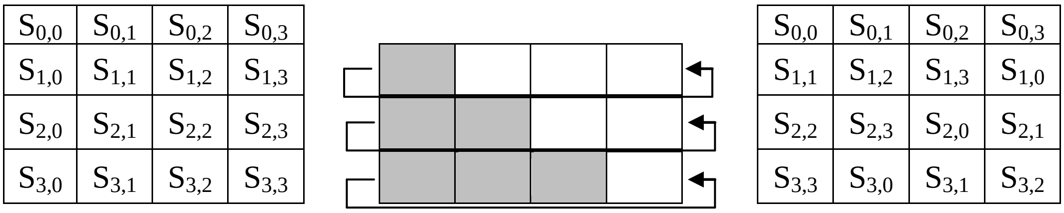
$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

SUB	SUB	SUB	SUB
SUB	SUB	SUB	SUB
SUB	SUB	SUB	SUB
SUB	SUB	SUB	SUB

$S_{0,0}'$	$S_{0,1}'$	$S_{0,2}'$	$S_{0,3}'$
$S_{1,0}'$	$S_{1,1}'$	$S_{1,2}'$	$S_{1,3}'$
$S_{2,0}'$	$S_{2,1}'$	$S_{2,2}'$	$S_{2,3}'$
$S_{3,0}'$	$S_{3,1}'$	$S_{3,2}'$	$S_{3,3}'$

Fig (4) The Sub Bytes Transformation.

2) **The Shift Row Transformation:-** this part sees the state shifted over variable offsets. The shift offset values are dependent on the block length of the State, see Fig.(5).



$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

Fig.(5) The Shift Rows Transformation.

3) **The Mix Column Transformation** the State columns take on polynomial characteristics over Galois Field values (2^8), and multiplied modulo x^4 + 1 with a fixed polynomial, see Fig.(6).

$$\begin{bmatrix} S_{3,c'} \\ S_{2,c'} \\ S_{1,c'} \\ S_{0,c'} \end{bmatrix} = \begin{bmatrix} 02 & 01 & 01 & 03 \\ 03 & 02 & 01 & 01 \\ 01 & 03 & 02 & 01 \\ 01 & 01 & 03 & 02 \end{bmatrix} \begin{bmatrix} S_{3,c'} \\ S_{2,c'} \\ S_{1,c'} \\ S_{0,c'} \end{bmatrix} \quad \text{for } 0 <= c <= Nc$$

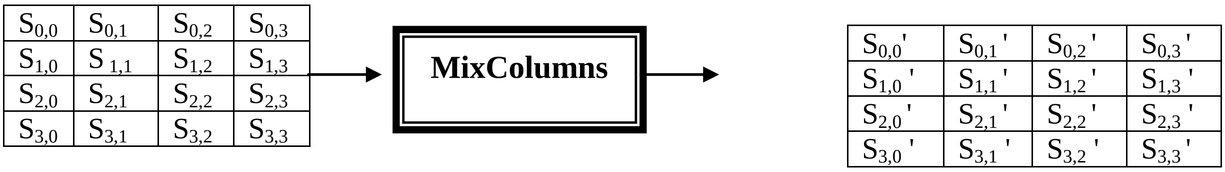

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
MixColumns
$S_{0,0}'$ $S_{0,1}'$ $S_{0,2}'$ $S_{0,3}'$
$S_{1,0}'$ $S_{1,1}'$ $S_{1,2}'$ $S_{1,3}'$
$S_{2,0}'$ $S_{2,1}'$ $S_{2,2}'$ $S_{2,3}'$
$S_{3,0}'$ $S_{3,1}'$ $S_{3,2}'$ $S_{3,3}'$

Fig.(6) The Mix Columns Transformation.

4) Finally, in the **XorRoundkey Transformation**, NC words from key schedule are each added (XORed) into the columns of the state [8], see Fig.(7).

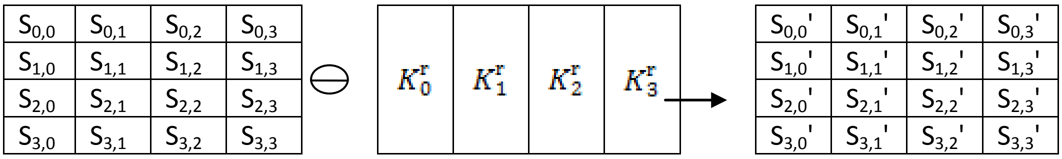

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
⊖
K_0^r K_1^r K_2^r K_3^r
$S_{0,0}'$ $S_{0,1}'$ $S_{0,2}'$ $S_{0,3}'$
$S_{1,0}'$ $S_{1,1}'$ $S_{1,2}'$ $S_{1,3}'$
$S_{2,0}'$ $S_{2,1}'$ $S_{2,2}'$ $S_{2,3}'$
$S_{3,0}'$ $S_{3,1}'$ $S_{3,2}'$ $S_{3,3}'$

Fig.(7) The XorRoundKey Transformation.

2.2 Modified Key Generation

This section presents the approach that is used to generate the key of Rijndael algorithm. The approach uses the equation 3 to generate a random number of byte. This number represent the key that are used in Rijndael algorithm.

Rand No.=**Nk*****Nb***(**Nr**+1) ....(3)

Where, **Nk**: represents key length
Nb: represents block length
and **Nr**: represents number of round

Then, the random number divided according number of rounds. For example, when applying equation 3 for the AES-128 bit, the generated random numbers are 176 numbers.

176 numbers divided to 11 round, for each round 16 number represent the cipher key to encrypted the plain text into ciphertext.

3. The Results

The proposed approach (combination of swap bits and modified Rijndael algorithm) has been implemented successfully in interlocution system (i.e. online chatting system) on different amount of ciphertext length.

Table (3), shows the comparison between Rijndael with expansion key and modified Rijndael with a random key generation on different lengths of plaintext (or ciphertext).

Table (3)
Comparison between Rijndael with expansion key and modified Rijndael with a random key generation.

Amount of Cipher Length (letters)	***Time in (sec.)***	
	Rijndael with Expansion Key	***Modified Rijndael with a Random Key Generation***
100	0.25	0.1
130	0.45	0.15
250	0.63	0.25
400	0.94	0.5
450	1.1	0.65

These results represent in the following charts, shown in figure 8 and 9 respects.

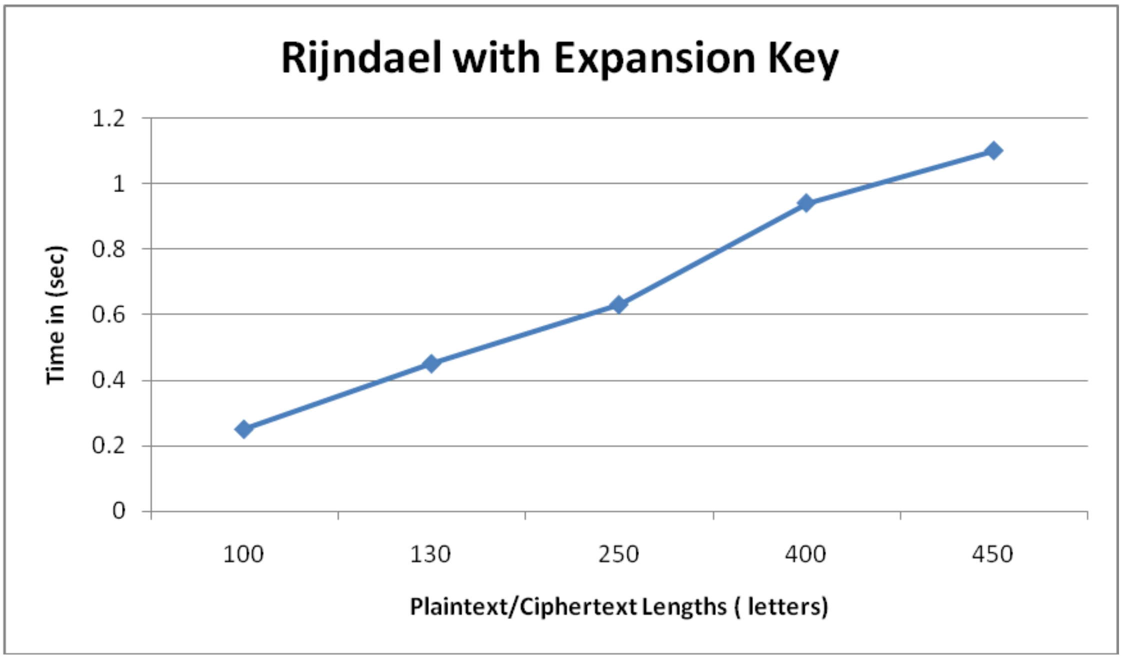


Fig.(8) Rijndael with Expansion Key Cipher Time (char per second).

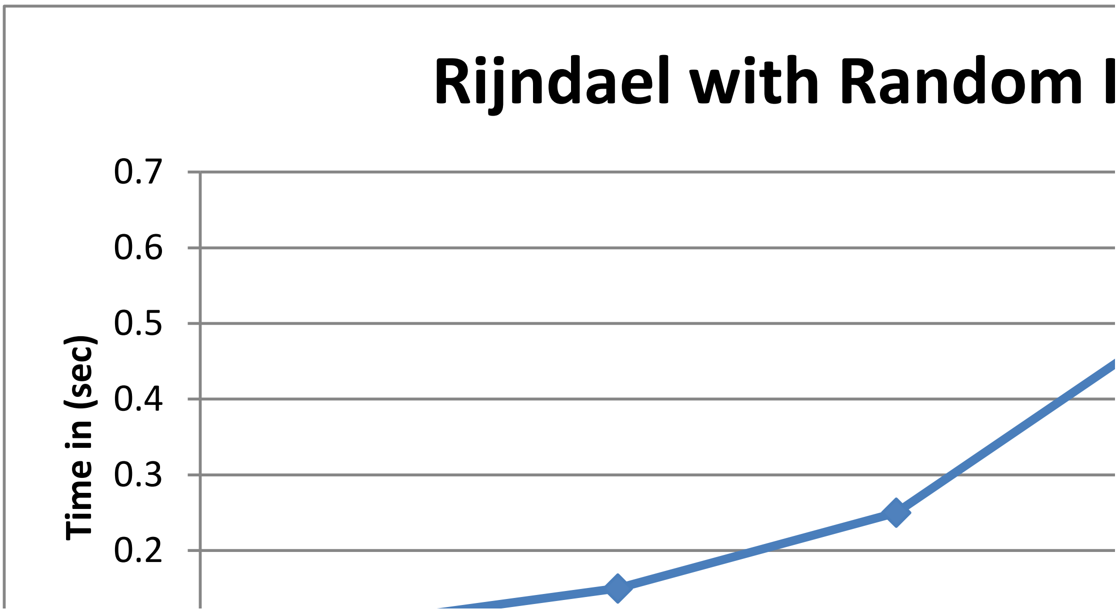


Fig.(9) Modified Rijndael with Random Key Generation Cipher Time (char per second).

Table (4), shows the complexity (represented the number of the performed operation) of Rijndael algorithm and modified key generation of Rijndael algorithm when the message length is 100 characters.

Table (4)
Comparison between Rijndael with expansion key and Rijndael with Random Key.

The Approch		*Principle Studing*		
	Amount of Cipher Length (letters)	*Size of sent message*	*Time encryption*	*Complexity*
Modified Rijndael with a Random Key Generation	100	0.1KB	0.1	352
	130	0.135KB	0.15	
	250	0.247KB	0.25	
	400	0.41KB	0.5	
	450	0.445KB	0.65	
Rijndael with Expansion Key	100	0.28KB	0.25	450
	130	0.364KB	0.45	
	250	0.76KB	0.63	
	400	0.89KB	0.94	
	450	1.09KB	1.1	

4. Conclusions

This paper presents a modified key generation for Rijndael algorithm. Modified key generation for Rijndal algorithm generates a random numbers representing cipher key. Expermintal results show that the modified key generation for Rijndael algorithm is fast and efficient method than key expansion key for Rijndael algorithm.

References

[1] Li Cheng Qing, "On the Security of Some Multimedia Encryption Schemes", PhD Thesis of City University of Hong Kong, 2008.

[2] M. Matsui, Linear cryptanalysis method for DES cipher, "Advances in Cryptology, Proceedings Eurocrypt93", LNCS 765, T. Helleseth, Ed., Springer-Verlag, pp. 386-397, 1994.

[3] H.S.Kwok, Wallace K.S. Tang, and K.F.Man, "Online Secure Chatting System Using Discrete Chaotic Map", Information Sciences: an International Journal ,14, pp285-292, 2004.

[4] Warren Fong," Final Project RSA Secure Chat Server", available at: wf007j@mail.rochester.edu, 2009.

[5] Kenny C.K. Fong1, Jonathan M. Hunt2, Soyini D. Liburd3, and Eduardo I. McLean4, "Really Secure Chat, Really!", USA, 2002.

[6] William S., "Cryptography and Network Security Principles and Practices", 4th edition, Printice Hall, 2005.

[7] Joachim Rosenthal, "A Polynomial Description of the Rijndael Advanced Encryption Standard", http://www.nd.edu/˜rosen/, February 25, 2003.

[8] Federal information processing standards publication 197, "Advanced Encryption Standard", Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, November 2001.

الخلاصة

واحدة من الأساليب لتحقيق الأمنية هي الشفرة / فتح الشفرة، عند إرسال البيانات عبر الاتصالات يجب أن تكون محمية بواسطة تشفير الرسائل القصيرة. و يمكن أن تفتح هذه الشفرة فقط من خلال الشخص المستلم المخول. يقدم هذا البحث طريقة بسيطة، وسريعة لتوليد مفتاح التشفير في خوارزمية(Rijndael Algorithm) إذ استخدمنا الطريقة العشوائية لتوليد مفتاح ثابت طوله مكافىء الى حجم البلوك والى عدد مرات التشفير. تم تطبيق هذه الطريقة في نظام محادثة و اثبتت نجاحها على مختلف الاطوال للنص. هذه الطريقة مسبوقة بطريقة تدوير البتات لكل بايت موجود بالنص المرسل.