Proposal for Complex AES Security using Key Generator and Text Permutation

Shatha Habeeb

Computer Science Department, University of Technology/ Baghdad Email: shathahabeeb@yahoo.com

Received on: 31/5 /2011 & Accepted on: 5/4/2012

ABSTRACT

Advanced Encryption Standard (AES) is a symmetric-key encryption each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key[1,2].

This research proposes a technique intended to make the Advanced Encryption Standard (AES) more safe and secure. Through the generation of random key, and permutation key sites in each round, instead of the process of expanding key locations in addition to the proposed permutation the plaintext before entering the encryption and inverse permutation for resulting ciphertext.

Keywords: AES, symmetric ,key, encryption , ciphertext.

تعضيد أمنية (AES) باستخدام التوليد العشوائي للمفتاح ومتعددة النص

الخلاصة

معيار التشفير المتقدم (AES) هو تشفير المفتاح المتماثل كل من هذه الشفرات لديها حجم كتلة 128 بت، مع أحجام المفتاح 128 و 192 و 256 بت يتم تحديد الشفرات معيار التشفير المتقدم (AES) وتكرار عدد من الجولات مساهمة في الناتج النهائي للنص المشفر. كل جولة يتكون من عدة خطوات المعالجة، بما في ذلك تلك التي تعتمد على مفتاح التشفير.

يقدم البحث تقنية الهدف منها هو جعل معيار التشفير المتقدم (AES) أكثر آمنا ومأمونا. ذلك من خلال توليد مفتاح بطريقة عشوائية، وتبديل المواقع المفتاح في كل جولة وبدلا من عملية توسيع المفتاح بالاضافة الى تبديل مواقع النص الاصلي قبل الدخول فـي التشفير ومعكـوس مصفوفة التبديل على النص المشفر.

INTRODUCTION

dvanced Encryption Standard (AES) has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum. AES operates on a 4×4 matrix of bytes, termed the *state* (versions of

2067

https://doi.org/10.30684/etj.2012.57252

Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key [3-7].

Description of the algorithm, see figure (1)

- 1. Key Expansion—round keys are derived from the cipher key using Rijndael key schedule
- 2. Initial Round
- AddRoundKey—each byte of the state is combined with the round key using bitwise xor
- 3. Rounds
- SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- AddRoundKey
- 4. Final Round (no Mix Columns)
- SubBytes
- Shift Rows
- AddRoundKey

For more explanation will present full example for AES work, see figure (2) and figure (3) [7, 8].

THE PROPOSAL SYSTEM DESIGN

In the proposed research, the aim is to strengthen (AES) algorithm. Where the proposal consists of two parts the first generation of random key. Where the user of the algorithm determines the length of the key 128-bit and all four bits number is hexadecimal or 32 number hexadecimal, and so if the length of the key 192-bit or 256-bit is equivalent to 48 or 62, No. hexadecimal, and then do permutation positions the key using a matrix consisting of 32 No. hexadecimal dimensions (4 X 8), represent the key length if 128-bit. Repeated a number of permutation rounds if the number of bits of a 128-bit repeated 10 times in each round so that the key to this allowance for the expansion of the key.

The other part includes the process of permutation or switching positions on the original text so that the matrix of either 128-bit dimensions are (8x16) or converted to hexadecimal system and its dimensions are (4×8) and then enter the encryption operations in the future be the same operations with the inverse matrix permutation. see figure (5)

DESCRIPTION OF THE PROPOSAL ALGORITHM

- 1. Key generation and permutation —round keys are derived from the cipher key using Rijndael's key schedule
- 2. plain text permutation
- 3. Initial Round
- AddRoundKey—each byte of the state is combined with the round key using bitwise xor
- 4. Rounds
- SubBytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.
- ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
- MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
- AddRoundKey
- 5. Final Round (no Mix Columns)
- SubBytes
- Shift Rows
- AddRoundKey

THE IMPLEMENTATION OF THE PROPOSAL SYSTEM

The implementation of the proposal done by using javascript run the implementation and vb6 to Description of Work Map the operations as mentioned in previous sections, the proposal aim to strength the AES algorithm in figure (6) consists of five functions, first function to choose a key length 128, 192 or 256-bit and then move to the second function to generate a key the third function is to convert the key generator to hexadecimal Fourth function is the work of the key permutation and permutation to the text of the original entrance Then complete the operations as the encryption algorithm AES, But on the other side the same for the key generation process with the use of inverse permutation key and cipher text.

In figure (6) implementation function to choose a key length 128, 192 or 256-bit to generator.

In figure (7) implementation to key generation by used randomize and convert to hexadecimal.

In figure (8) implementation initial array permutation key and plaintext.

In figure (9) implementation array of inverse initial permutation key and cipher text.

In figure (10) the full implementation of the proposal through the application of an example of generating a random key number and convert the hexadecimal system and permutation the key and the original text and then enter the other operations in the algorithm (AES)

CONCLUSIONS

By studying the AES algorithms and analysis it is work, this research present some modification on it. By implementing the proposed modified AES there is some point concluded these are:

- 1. The encryption of AES has some thing danger, that it is an algorithm depend on symmetric key, so if the key is discovered that will destroy the AES security.
- 2. from previous point, the research propose key generation method aim to reduce the danger of symmetric keys by taking short key and from it the overall key will be generated, so this short key only will be known previously by sender and receiver.
- 3. AES depend on key expansion, and this expansion is static method. in this research the dependency on a proposed key generation preserve the randomness by permutation in each round.
- 4. AES without secure dealing with plaintext, so this research aim to enter the plaintext in the security process. That by applying initial permutations on the plaintext and finally applying the inverse permutations on resulted ciphertext.

REFERENCES

[1]- Gladman, "B. The AES Algorithm (Rijndael) in C and C++, performance of the optimized implementation," from

http://fp.gladman.plus.com/cryptography_technology/rijn dael/index.htm

- [2]-<u>http://www.saylor.org/site/wp-content/uploads/2011/03/Advanced-Encryption</u>-Standard.pdf
- [3]- Satheesh Kumar, R. E.Pradeep, K.Naveen and R.Gunasekaran "A Novel Approach for Enciphering Data of Smaller Bytes" International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 2010 8201-1793
- [4]- Venkateswaran, R. Dr. V. Sundaram, Research Scholar- Ph.D Director-Computer Applications, Karpagam Academy of Higher Education Karpagam College of Engineering, Karpagam University, Affiliated to Anna University Coimbatore, Tamilnadu, India. Coimbatore, Tamilnadu, India " Information Security: Text Encryption and Decryption with Poly Substitution Method and Combining the Features of Cryptography", International Journal of Computer Applications (0975 – 8887) Volume 3 – No.7, June 2010
- [5]- Karthigaikumar, P. Asst.Professor (SG) Department of Electronics and Communication, Karunya University, Coimbatore, Soumiya Rasheed M.Tech Department of Electronics and Communication Karunya University, Coimbatore "Simulation of Image Encryption using AES Algorithm" IJCA Special Issue on "Computational Science - New Dimensions & Perspectives" NCCSE, 2011 166

[6] Joan Daemen, "The Rijndael Block Cypher: AES Proposal", Vincent Rijmen. First AES Candidate Conference (AES1),

August 1998.

- [7]-<u>http://pdfdownloads.blogspot.com/2010/10/design-of-rijndael-aes-advanced.html</u>
- [8]-http://en.wikipedia.org/wiki/ Advanced_Encryption_Standard

Proposal for Complex AES Security Using Key Generator and Text Permutation







Figure (2) This form displays the overall process of AES encryption.

Els Ett Mari Enio	bat Task iide		
The same time rails			
Q M + Q -	🖹 📓 🏠 🔎 Swith 🙀 Kwinikes 🥷) 🙆 · 🎍 🖾 · 🖵 🔣 🛍 🍇	
Coogle		🐱 🕼 Search + 🗐 + 💼 + Nove 20	🧉 Sign In 🦓 -
Of . D. Patientie	SEARCH + -	🗋 • 😹 • 🖻 • 🛃 • 😹 • 🗷	0
Davalicipt AES Exampl			8
Anthen Shtta Vaecole, s	ku edulut-sevelEncivot135-ACC.html		- 61 -
- Emil -		🛿 Logie 📓 + 🛃 + 🦦 🗃 + 🔹 🖮 🔤 🖂 📦 🚂 Logielhou	ter - 84
Mecsare: 10 20	19 of as 9c 4c 77 bit 95 d4 48 db 90 21 oc	OASCII @H-moderated	
East Date			
trub loi ib			
	Escrept Decrypt		
Output messager 40.37	27 36 32 66 23 52 6d e4 66 e6 8 36 22 36	©ASCII @Hazadetinal	
Desails			
After SubBytes 0	d Ce = 7 27	*	
	1 60 40 30 5 40 55 54		
	4 Sh 29 19		
De DO 793 AQUOS	97.26		
90 19	fe 81		
37 115	72 15		
BO ±9	3± 07		
After AddPoundKey	el 15 50 1f		
	ff d5 50 hd		
	42 97 4m 3m		
	44 76 13 12		
ALCE. INVINCE. FOR	13 37 aŭ de		
	h1 ba 47 6e		
	00.1f 04 dc		
ATTES NUCHVORB 4	I C5 01 50		
	5 1.2 53 Ta		
5	6 c0 16 15		
5	0 cb 30 23		
			Paternat
A state of the sta	Contraction of the second second second	Valued - Mont	20 6 10 K 10 10 10

Figure (3): This form displays the overall process of AES decryption.



Figure (4): The flowchart of the proposed algorithm.

Proposal for Complex AES Security Using Key Generator and Text Permutation



Figure (5): The implementation steps for proposed modifications on AES.

C. Comment and Settings Schri	halfbeikreput/STNaa. http://www.stnice.com/stylesee	
File Edit Wen Favorise Foels	lido	2
0 0 - 1 2 1	👔 🔎 Sondi 🚖 Fannta 🐵 🔝 - 🍒 💷 - 🔂 🔣 🚳	
Google	😽 🖓 Sauch 🐖 🎮 👘 💷 🛛 Hone Xi	i santi 🔍
Of - Q inse Trank	SCARDI - 🕴 🖂 - 🚚 - 🗐 - 🥥 - 🖉	a
C 100 MARS AND WITH DISPLAT.	•	6
a Minist 🔬 Cripequeerte and Settingele	sche(Devidop)/#311as.html	M 🛃 🗤
Sans -	💌 🐨 + - 🛃 + 😨 🕼 anga 📓 + 🔛 + 🔯 🖓 + 📓 🖓 Craall	name + 18 pro 🛄 🖬 🐴
key 256		
() Dane		Arrest Mater Stand
Arrest C. R. A		A A A R LOOK

Figure (6): Form of choosing the key.

Proposal for Complex AES Security Using Key Generator and Text Permutation



Figure (7): Form of key of generation and implementation.

h Est							a contraction of the second									- 19	
	Ver 1	avcepts	Taoa Help														
(Gam - O) 🔄 📓 🚱 🔎 2003					- The	🚖 Terradau 🤣 🔝 - 🍶 🔣 🛍 🖓											
Congle						🖌 🔂 and t 📑 t 💷 t mar.e								ं भूता ह			
Call + Q. Valuet Scaret				SCAFO	swort								4				
C Dec	unerts and	Settingrish	4														
	C/pcoursets and Settingdy halful (Deccord / 25: / publicat									× 🖬							
1 Mars				*			anen i	8 · E	- 10 12	1.00	BALLER &	1 15 2		initarile.	+ (3) -11	- 02	
key key key	128 192 256		_	Kej	y gene	erator	to us	e ran	domi	te hex	adeci	mal					
key key key key	128 192 256	and on Lee	0	Kej	y geni tial ai	erator Tray p	to us	e ran tatior	domi 1 key	te hex	adeci Yan T	mal ext					
key key key key	128 192 256 aracar Ho	endorn ce 12	2	Kej ini	y gene tial ai 2:	rator ray p	to us ermu	e ran tatior	domk 1 key 7	and T	adeci Ilan T	mal ext	28	34	24	16	
key key key key se, ge	128 192 256 water Fe	ndori as L' 32	2	Kej ini 25	y gene tial ar 3: 6	rray p	to us ermu 21 22	e ran tatior	domk 1 key 17	and T	adeci Ian T	ext	28 .5	34 13	24 3	16 31	
key key key key key ze ze ze	128 192 256 9 3 3 7 10 10 13 1 20	ndori as 11 32 16	2 4 18	Kej ini 19 14	y gene tial a 3: 6 24	erator eray p	to us ermu	e ran tatior	domk 1 key 1.7 .9 3	and T	adeci Plan T	ext	28 5 22	34 13 20	24 3 20	16 31 24	

Figure (8): Form of generating initial permutation Arrays applyingon key and plaintext.

Proposal for Complex AES Security Using Key Generator and Text Permutation

				Ke	w cen	erato	. to use ra	ndomi	ze hev	adeci	mol				
key key	128														
Rey	a store Fa	er dara ze													
(a) (b)			-	row of	linver	en ini	tial narms	Intion	law e	nd ch	harts	wr S			
				149 0	mver	se un	ciai per int	na cron	r Key i	uo eq	pitere	(1 04-0			
x	12	18	31	13	26	2		15	30	- 3	31	24	22	μ	22
1	12	7	9	ń	21	12	82	20	2	-7	10	3	22	1	3
25	15	14	8	23	н	28	24	23	1	.2	4	28	2	29	26
2	20	15	25	17	1	27	4	16	21	8	12.	6	19	25	13
25	15 20	14 15	8 25	23 17	11 9	28 27	24 4	23 16	7 21	2 8	4	28 6	2 19	29 25	1

Figure (9): Form of generating initial permutation arrays applying on key and cipher text

🗐 Javabenipt ALS Eau	unde - Miscosels Internet Exp	planes			🔄 ie 🗵
The Edit New Fin	voibei Taoli Heb				
(and - 0 -	🔳 🛃 🏠 🔎 Sarch	Trouber 🕘 🙆 .	👌 🖻 · 🔜 🔣 🕮 🐇		
Google			■ di seera = 题 · = ·)	610 + 33	U Souly Mart
SI - Q. Vani	(MAVE)	17ARDH	8 - 🗃 - 🞑 - 🗃 - 😫		
Carrettulg 1 ASS Ence	eule 🔶				13
attres () (ar ser	rs and setting (shahali red copyare	toni			× 🖸 ×
·		• • • 🛃 • 😰 🛃 kagn 📓		• - 📓 📷 E-mailflotifier 💌 🔗	
					2
Ederrage 601	1476E1330E44985A3535C7C0	06C00E3	() ASCII () Henderi	mai	
Key 1.47	705214F40800F05B88A28B88	DA91764	ergenerater 3128 O152 O 2	16	
Output message b2	oo 06 96 97 6o #6 'C 74 89 N	at Baladico ab	○ ASCII ⊙ Hensderi	mail	
	_				
Encrypt Decrypt	Detak:		kay partnesses	plantast permutation	
Input bits 6d 3	14 76 :1	- 11			-
50 3	35 35 57	100	wey permitteet	fin leve fermionia	
ed (6e def all		P 15 11 2 25 31 17 21	17 6 32 3 28 14 24 16	
Eay bits in 7c	52 14		22 1 32 4 19 5 28 16	19 12 25 13 1 12 0 31	
24 CO	a7 5h		1 10 29 7 5 20 17 8	9 29 6 17 18 18 23 21	
Re b6	17 58		201070.000.00 0.000.0000	· · · · · · · · · · · · · · · · · · ·	
v[0] = 10 7c !	52 14				
w[1] = f4 08 c	c0 f=				
w[2] = fb 99 s	e2 E5				
w[3] = 2d m9 :	17 da				
Schlording dis	40 02 54				
A Knowline d2 a	#0 02 5d				
w[4] = c0 0c 3	50 42				
w[5] = 3c 64 s	90 65				
w[6] = 37 1d 3	32 3m				
w[7] = ca b4 s	25 54	M1			
1				😰 Evta	ferre
Manager and State	and the second se	the second se	and the second se		

Figure (10): Form presenting all the implementation of proposed modified AES.