Encryption using Dual Key Transformation based on Creation of Multi S- Boxes in AES Algorithm

Nada Hussein M. Ali
Lecturer, Computer Science Department, University of Baghdad,
Baghdad, Iraq

Abdul Monem S. Rahma
Professor, Computer Science Department, University of Technology,
Baghdad, Iraq

Abdul Mohsen Jaber
Assistant Professor, Computer Science Department,
University of Technology
Baghdad, Iraq

ABSTRACT
The Advanced Encryption Standard (AES) is using in a large scale of applications that need to protect their data and information. A nonlinear substitution operation is the main factor of the AES cipher system strength. The S-Box component that used in AES is fixed, and not changeable. The purpose of the proposed approach is to use dual keys in encryption and decryption processes in SubByte transformation function. The first key generate multi random S-boxes depend on using multi keys that led to generate S-boxes provided that each one has its inverse associated with it, the second key is a random distribution of the S-boxes, the dual keys lead in increasing the complexity degree within the same delay time during the encryption and decryption processes in SubByte function. The results show that the present proposed algorithm has good cryptographic strength, with the added benefit that is resistant to linear and differential cryptanalysis, which require that the S-boxes be known beside the encryption key.

General Terms
Security, Algorithms, system strength, Plaintext, Ciphertext

Keywords
Advanced Encryption Standard (AES), S-boxes, Encryption, Decryption, Transformation, Cipher system strength, Block cipher

1. INTRODUCTION
Numerous multimedia applications, such as video surveillance, satellite communication and web cams have been drawn from rapid growth of multimedia and networking technologies. Consequently, securing multimedia transmission has become a challenging issue. The fact which states that the quality of service media (QoS) should be met to provide high media quality and low latency even when securing media transmission. A proper security algorithm should be chosen carefully for real-time multimedia transmission due to the unique characteristics of real-time multimedia data such as large data size, high bandwidth and real-time requirements. In order to preserve high media QoS, a viable security mechanism should provide three properties: high-speed processing, high compression rate and sufficient security level [1].

The best method of data protection against passive and active fraud is cryptography which has an important role in the security of data transmission. The growing number communication users have led to increasing demand for security measures to protect data transmitted over open channels. A cipher system is a set of reversible transformations, which depend on a secret key and the ciphering algorithm, from set M of a plaintext into set C of a ciphertext. In the block cipher system, the plaintext is divided into blocks and ciphering process is carried out for each block [2].

In cryptography, confusion and diffusion are two properties of the operation of a secure cipher. Diffusion is spreading of the influence of a one plaintext bit to many ciphertext bits with intention to hide the statistical structure of the plaintext. While, confusion is transformation that changes dependence of statistics on ciphertext by statistics of plaintext. In most cipher systems, the diffusion and confusion are achieved by means of round repetition. Repeating a single round contributes to cipher's simplicity. Modern block ciphers consist of four transformations: substitution, permutation, mixing, and key-adding [3].

Block cipher systems depend on the S-boxes which are fixed and have no relation with the secret key which is the only changeable parameter. Since the only nonlinear component of AES is S-boxes, they are an important source of cryptographic strength. Many researches on S-box design are focused on determination of S-box properties which yield cryptographically strong ciphers by selecting a small number of good S-boxes for use in a block cipher DES and CAST [2].

The aim of the present study is to investigate the design of a new pseudo-randomly generated key-independent S-boxes and inverse S-box generation algorithms. Preliminary results show, that our proposed algorithm has good cryptographic strength, with the added benefit that is resistant to linear and differential cryptanalysis, which require that the S-boxes be known.

2. RELATED WORKS
Several papers discussed how to solve the problem of the fixed structure S-Boxes, some of them are:

Mohammad et al. (2009) introduced AES algorithm with Variable Mapping S-box (VMS-AES) uses the key data to generate parameter. This parameter is used to shift (remapping) the substitution of S-box to another location randomly depending on the initial key and the derived sub keys data.

Kazlauskas and Kazlauskas (2009) presented a new approach to generate the AES randomly key-dependent S-boxes and their inverse. These boxes are constructed by composing two transformations: key pseudo-expansion and key-dependent S-box and inverse S-box generation. The quality of this approach has been tested by changing only one bit of the secret key to generate new S-boxes.

Hosseinkhani and Javadi (2012) introduced a new algorithm to generate dynamic S-Box from cipher key. The quality of this algorithm is tested by changing only two bits of cipher key to generate new S-Boxes.

Lambić and Živković (2013) applied an alternative S-box generation method by forming compositions of permutations

from some fixed sets. After choosing these sets, output S-boxes are obtained by making various compositions of the starting S-boxes. The sequence of the used indices of starting S-boxes is key-controlled.

3. ADVANCED ENCRYPTION STANDARD (AES) ALGORITHMS

The Advanced Encryption Standard (AES) was published by the National Institute of Standards and Technology (NIST) in 2001. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. Compared to public-key ciphers such as RSA, the structure of AES and most symmetric ciphers is quite complex and cannot be explained as easily as many other cryptographic algorithms. Figure 1 shows the AES cipher in more detail, indicating the sequence of transformations in each round and showing the corresponding decryption function [4].

The AES is a private key block cipher that processes data blocks of 128 bits with key length of 128, 192, or 256 bits. The AES algorithm's operations are performed on a 2-D array of 4 *times* 4 bytes called the State. The initial State is the plaintext and the final State is the ciphertext.
Rijndael round function operates on a state Nr times, where Nr is equal to the number of rounds that can be 10, 12 or 14 rounds, depending on Nb and Nk, where Nb is equal to the block size divided by 32. Rijndael round is composed of 4 transformations [5]:

1. SubByte: S-box substitution provides nonlinearity and confusion.
2. Shiftrow: rotations, provides inter-column diffusion.
3. MixColumn: linear combination provides inter-Byte diffusion.
4. AddRoundKey: round key bytes XOR into each byte provide confusion.

Transformations operations are byte-oriented. Decryption is applying the operations in a reverse order with respect to the order of encryption. Many people have tried to modify AES algorithm to improve its performance, to satisfy the varying criteria of different applications. Some approaches seek to maximize throughput, others minimize power consumption, and yet others minimize circuitry. S-box Construction was attempted by many designers using a fixed irreducible polynomial for higher efficiency and a smaller footprint of the AES intellectual property (IP). For long-term use, however, the fixed irreducible polynomial has been proven to make the system's golden key obvious, thus increasing the decryption rate of confidential files. The decryption methods include side channel, time channel, and power side channel attacks. Some systems can even be decrypted by an inside job [5].

3.1 Substitution Byte Transformation Function

A number of known block ciphers are of substitution-permutation (SP) type. S-boxes are used in such cipher systems as the important nonlinear component. A strong block cipher should be resistant to various attacks, such as linear and differential cryptanalysis. In SP networks, this is generally achieved if the S-boxes use satisfied number of criteria, such as strict avalanche criterion (SAC), bit independence criterion (BIC), nonlinearity, XOR Table Distribution and maximum expected linear probability (MELP) [6].

Nonlinear transformations are implemented as lookup tables (S-boxes). In the AES, the S-box generates two transformations in the Galois fields GF(2) and GF(2^8). S-box is a nonlinear transformation where each byte of the State is replaced by another byte using the substitution table [2]. The first transformation: S-box finds the multiplication inverse of the byte in the field GF (2^8). Since it is an algebraic expression, it is possible to mount algebraic attacks. Hence, it is followed by an affine transformation which is chosen in order to make the SubBytes a complex algebraic expression while preserving the nonlinearity property. Both S-box transformations can be expressed in a matrix form as:

$S^{\sim} = M \bullet S^{-1} + C$ (1)

$S^{\sim} = M \bullet S^{-1} + C$ (2)

Where the 8×1 vector $S^{\sim}$ denotes the bits of the output byte after the S-box transformations; the sign • is multiplication and the sign + is addition in the field GF(2^8). The inverse S-box transformation can be obtained by multiplying both sides of equation (3) by M^{-1} and it performs the inverse affine transformation followed by the multiplicative inverse in GF(2^8) [7]:

$S^{-1} = M^{-1} \bullet S^{\sim} + M^{-1}.C$ (3)

$S^{-1} = M^{-1} \bullet S^{\sim} + M^{-1} \bullet C$ (4)

3.2 Dual Key S-Box Encryption Algorithm

In general, S-Box is nonlinear substitution table that get number and return another number. The first step is to generate multi random S-boxes depend on using multi keys that led to generate S- boxes provided that each one has its inverse associated with it and that represent the first key. The second step is to create a random index distribution of the S-boxes that created in the first step to get more complexity and the same delay time and this represent the second key as shown in Algorithm 1.

The second key [4X4], if eight different S-Boxes are used then the key space available is 16! different keys. Figure 2 shows some possible output 4*4 keys that can be generated. Algorithm 2 demonstrates this process.
If the cryptanalyst wants to decode AES algorithm he should try to generate all possible S-Boxes and use them in SubBytes function in AES cipher system, beside the encryption key. Algorithm 1 shows the details how to create different eight S-Boxes that uses as a first key, while the algorithm 2 shows how the encryption process implemented using the second key.

4. RESULTS AND DISCUSSION

The present study outlines the work of the authors' investigation into the design of a new is to use dual keys in encryption and decryption processes in SubByte transformation function. Other systems using key-dependent S-boxes have been proposed in the past, the most well-known is Blowfish (Schneier, 1996) and Khufu (Merkle, 1991). Each of these two systems uses the cryptosystem itself to generate the S-boxes.

The proposed method to generate dual keys, The first key generate multi random S-boxes depend on using multi keys that led to generate S- boxes provided that each one has its inverse associated with it. By using different 8 keys to create different 8 S-boxes and its related constants (the values based on hexadecimal):
Rnd_Key[8]={0xf1,0x67,0x25,0x85,0xb5,0xA4,0x19,0x4c}
Cons_c[16]={0x63,0x82,0xc4,0x45,0xa5,0x7b,0xd5,0xc1,

For example the S-boxes created for the values (Key=0x85 and C=0x45) are shown in tables 2 and 3.The second key is a

random distribution of the S-boxes, the dual keys lead in increasing the complexity degree within the same delay time during the encryption and decryption processes in SubByte function.

The results show that the present proposed algorithm has good cryptographic strength, with the added benefit that is resistant to linear and differential cryptanalysis, which require that the S-boxes be known beside the encryption key. Table 4 give a summary for comparison the standard AES and the modified AES with respect to number of S-boxes, complexity, mapping using S-Box etc.

5. CONCLUSIONS

Based on the results, the following conclusions can be drawn:

1. The first key that generate multi random S-boxes depend on using multi keys make our approach resistant to linear and differential cryptanalysis. This approach will lead to generate more secure block ciphers, solve the problem of the fixed structure S-boxes, and will increase the security level of the AES block cipher system.
2. The main advantage of such approach is that an enormous number of S-boxes can be generated.
3. The second key which is a random distribution of the S-boxes, lead in increasing the complexity degree within the same delay time during the encryption and decryption processes in SubByte function.

6. ACKNOWLEDGMENTS

The authors thank anonymous reviewer for the comments and suggestions that contributed to improve the quality of the paper.

***Algorithm 1**: Modified S-box generation*

***Input**:: Different Eight values as { Rnd_Key[k], Con_c[k] , k=1,2,……..,8 }*

***Output**:: Different* Eight *S-Boxes* { $(S\text{-}box[i][j])_k$, $(S^{-1}\text{-}box[i][j])_k$, *k=1,2,.....,8 }*

Step1: Select 8 keys *Rnd_Key[k]* that each one generate a unique S-box in condition every key has its inverse to rebuild Inverse S-box

Step 2: Select 8 random values *Con_c[k]* for constant C

Step3: For every key *Rnd_Key[k]* and corresponds constant *Con_c[k]* create its own *S-box[i][j]* using affine transformation as:

$$(S\text{-}box[i][j])_k = Rnd\_Key[k] * mulp[r][c] + Con\_c[k]$$

Where *mulp[r][c]* represent the multiplicative inverse in GF(2^8) as shown in table 1.

Step4: Use the above($S\text{-}box[i][j])_k$ in encryption process .

***Algorithm 2**: Modified SubByte Transformation Function*

***Input** :plaintext Block message { State[Row][Column]Row,Colum=1,2,3,4}*

Different Eight *S-Boxes* { $(S\text{-}box[i][j])_k$, $(S^{-1}\text{-}box[i][j])_k$, *k=1,2,.....,8*

Key distribution matrix {Key_Enc[4][4]}

***Output:** ciphertext Block message*

{State[Row][Column]RowColum=1,2,3,4}

For every block to be ciphered in AES algorithm using the new created *S- box[i][j]* as follows:

Step1: For Every Row in State matrix Do

Step2: For every Column in State matrix Do

Step 3: Y=(State[Row][Column])&0x0f;
X=(State[Row][Column]>>4)&0x0f;
Where X, Y represent the index of row and column in each S-Box respectively

Step 4: The state matrix State[Row][Column] can be encrypted using the index of each S-Box *Key_Enc[4][4]* as:
State[Row][Column]=$(S\text{-}box[x][y])_{Row,\ Column}$

Table 1. The Multiplicative inverses in Rijndael's Galois field

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	--	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Table 2. AES S-box for Key=0x85 and C=0x45

X	Y																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	45	06	B0	E3	EB	27	16	43	46	47	74	34	B8	1F	12	FC
	1	C4	B5	10	4A	89	F2	FD	73	BB	CB	3C	23	BA	95	CD	3E
	2	85	6C	3D	2B	EF	C1	96	6D	77	DA	CA	3F	19	98	0A	B2
	3	39	DB	56	EA	AD	40	76	13	EE	57	2D	2F	01	67	F8	28
	4	25	F9	D1	35	79	5A	26	94	44	F0	53	B4	AC	63	05	1C
	5	08	B6	B	98	82	20	2C	69	6B	88	AB	68	E2	4B	BE	B9
	6	FA	30	5E	C5	CC	07	92	50	65	5D	93	D2	DC	15	6E	C6
	7	90	70	18	AA	71	F6	24	0E	33	C9	00	CE	CF	F4	A7	62
	8	21	58	16	E4	5B	C	29	2A	0F	41	9E	59	A0	C3	E1	81
	9	91	61	9F	A9	1A	78	E9	4F	B1	7C	02	FE	31	17	BD	4C
	A	B7	DE	BC	F1	36	A2	B3	8F	96	2E	F7	09	95	94	86	7B
	B	52	5C	93	8E	32	87	D3	8A	C2	75	42	4D	EC	AF	6F	69
	C	99	37	FF	49	9C	0D	51	97	D5	E5	64	48	AE	7F	9B	D7
	D	55	8D	1D	38	7A	DF	DA	C0	DD	3B	39	4E	84	72	D0	22
	E	FB	11	8B	83	BF	D4	E6	D8	5F	04	C8	99	F5	A1	E0	7D
	F	7E	E8	03	14	E7	1E	80	F3	54	C7	96	8C	60	ED	D6	66

Table 3. Inverse S-Box for Key=0x85 and C=0x45

X	Y															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	7A	3C	9A	F2	E9	4E	01	65	50	AB	2E	52	85	C5	77	88
	1	12	E1	0E	37	F3	6D	06	9D	72	2C	94	82	4F	D2	F5	0D
	2	55	80	DF	1B	76	40	46	05	3F	86	87	23	56	3A	A9	3B
	3	61	9C	B4	78	0B	43	A4	C1	D3	DA	30	D9	1A	22	1F	2B
	4	35	89	BA	07	48	00	08	09	CB	C3	13	5D	9F	BB	DB	97
	5	67	C6	B0	4A	F8	D0	32	39	81	8B	45	84	B1	69	62	E8
	6	FC	91	7F	4D	CA	68	FF	3D	5B	BF	57	58	21	27	6E	BE
	7	71	74	DD	17	0A	B9	36	28	95	44	D4	AF	99	EF	F0	CD
	8	F6	8F	54	E3	DC	20	AE	B5	59	14	B7	E2	FB	D1	B3	A7
	9	70	90	66	6A	AD	1D	26	C7	2D	EB	C0	CE	C4	FA	8A	92
	A	8C	ED	A5	B2	47	AC	A8	7E	53	93	73	5A	4C	34	CC	BD
	B	02	98	2F	A6	4B	11	51	A0	0C	5F	1C	18	A2	9E	5E	E4
	C	D7	25	B8	8D	10	63	6F	F9	EA	79	2A	19	64	1E	7B	7C
	D	DE	42	6B	B6	E5	C8	FE	CF	E7	29	D6	31	6C	D8	A1	D5
	E	EE	8E	5C	03	83	C9	E6	F4	F1	96	33	04	BC	FD	38	24
	F	49	A3	15	F7	7D	EC	75	AA	3E	41	60	E0	0F	16	9B	C2

Table 4. Comparison between Standard AES and modified AES

	Standard AES	Modified AES
1- Block length	16 bytes	Same
2- Numbers of rounds	10//12//14	Same
3-Key length	16//24//32	Same
4-S-box	Single	Multi (2-16)
5-Mapping using S-Box	Depend on State matrix	Depend on the index of every byte in
7-Single round details	AddRoundKey, SubByte,MixColum,	Same + Create the New S-Boxes
8-Complexity	256!	8*256!

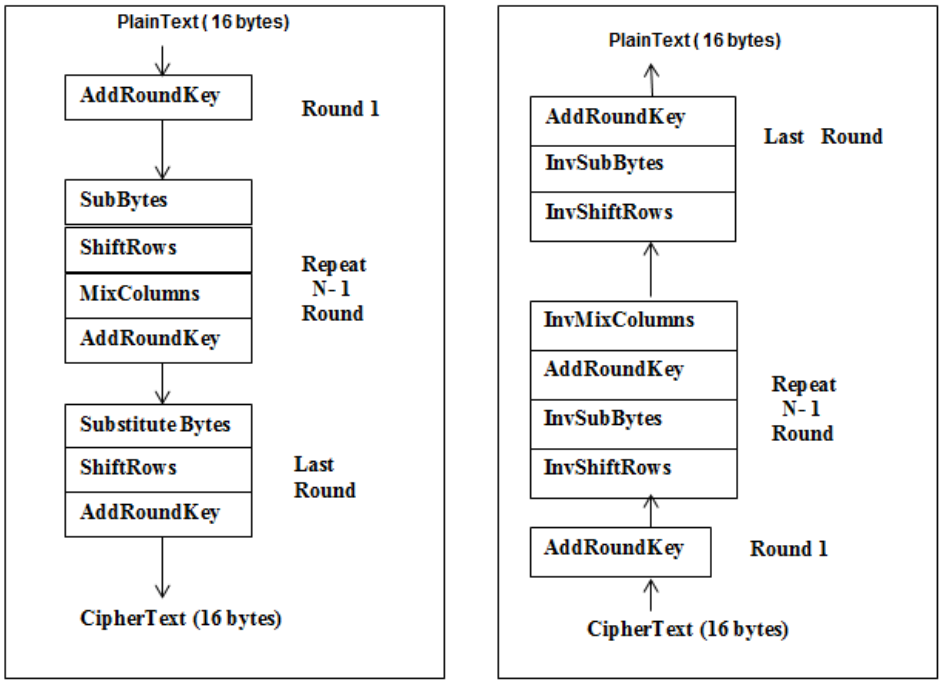

Figure 1: AES Encryption and Decryption [4]

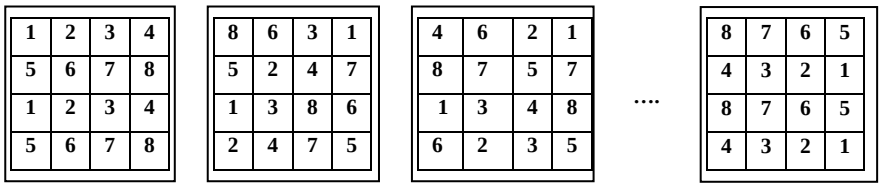

Figure 2: Possible 4 * 4 Key distribution

7. REFERENCES

[1] Choo, E. ,Lee, J. ,Lee H. and Nam, G. , 2007. SRMT: A Lightweight Encryption Scheme for Secure Real-time Multimedia Transmission. International Conference on Multimedia and Ubiquitous Engineering, Page(s): 60 – 65.

[2] Kazlauskas, K. and Kazlauskas, J. , 2009. Key-Dependent S-Box Generation in AES Block Cipher System. INFORMATICA. Vol. 20, No. 1, 23–34.

[3] Hosseinkhani, R. and Javadi, H. Haj Seyyed 2012. Using Cipher Key to Generate Dynamic S-Box in AES Cipher System. International Journal of Computer Science and Security (IJCSS). Vol. 6: Issue 1.

[4] William Stallings. 2012. Cryptography and Network Security Principles and Practice. Fifth Edition, Prentice Hall.

[5] Mohammad, F. Y. , Rohiem, A. E. and Elbayoumy, A. D. 2009. A Novel S-box of AES Algorithm Using Variable Mapping Technique. 2009. 13th International Conference on Aerospace Sciences & Aviation Technology. ASAT- 13, May 26 – 28.

[6] Lambić, D. and Živković, M. , 2013. Comparison of Random S-Box Generation Methods. Publications DE L'Institute Mathematique Nouvelle série, tome 93 (107), 109–115.

[7] Hsiao, S., F., Chen M. y. ,Tsai, and Lin C.C. 2005. System on chip implementation of the whole advanced encryption standard processor using reduced XOR-based sum-of-product operations. IEEE Proceedings, Information Security, Vol. 152, No. 1, 21–30.