International Journal of Cyber-Security and Digital Forensics (IJCSDF) 1(3): 183-188
The Society of Digital Information and Wireless Communications (SDIWC) 2012 (ISSN: 2305-0012)

Enhancing Advanced Encryption Standard S-Box Generation Based on Round Key

Julia Juremi Ramlan Mahmod Salasiah Sulaiman Jazrin Ramli
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia
juliajuremi@gmail.com

ABSTRACT

This paper presents a new AES-like design for key-dependent AES using S-box rotation. The algorithm involves key expansion algorithm together with S-box rotation and this property can be used to make the S-box key-dependent, hence providing a better security to the block cipher. Fixed S-box allows attackers to study S-box and find weak points while by using key-dependent S-Box approach, it makes it harder for attacker to do any offline analysis of an attack of one particular set of S-boxes. The cipher structure resembles the original AES, only the S-box is made key-dependent without changing the value. This new design is tested using the NIST Statistical Test and will be further cryptanalyzed with algebraic attack in order to permit its subversion or evasion.

Keywords – AES, Key-dependent S-box, Inverse S-box, Round key, Cryptanalysis

1 INTRODUCTION

In 1997, the National Institute of Standards and Technology send out a call for candidates to replace the aging and obsolete Data Encryption Standard (DES). NIST then announced the selection of Rijndael as the proposed Advanced Encryption Standard (AES) [1]. Rijndael, submitted by Joan Daemen and Vincent Rijmen is designed for the use with keys of lengths 128, 192, and 256 bits. Although AES uses the same three key size alternatives, it limits the block length to 128 bits [2].

The input to the AES encryption and decryption algorithm is a 128-bit block. The key which is provided as the input is expanded into an array of key schedule words; with each word is four bytes. The total key schedule for 128-bit key is 44 words. Full round of encryption will go through Nr rounds (Nr=10, 12, 14) [3][4][5]. Rijndael round consists of four different stages:

1. SubByte transformation: (Sbox substitution) provides non linearity and confusion, constructed by multiplicative inverse and affine transformation.
2. ShiftRow: (rotations) provides inter-column diffusion where the bytes in the last three rows of the states are cyclically shifted.
3. MixColumn: (linear combination) provides inter-byte diffusion where each column vector is multiplied by a fixed matrix. The bytes will be treated as polynomials rather than numbers
4. AddRoundKey: (round key bytes XOR with each byte of the state and the round key) provides confusion.

The encryption process begins with an AddRoundKey stage, and followed by nine rounds of SubBytes, ShiftRows, MixColums and AddRoundKey transformation. The transformation will be performed respectively and iteratively (Nr times) depending on the key length. The final round will only include 3 stages; SubByte, ShiftRows and AddRoundKey. All of the operations are byte-oriented. The encryption and decryption structure consists of several transformation stages as shown in Fig. 1[2].

The decryption is essentially the same structure as encryption, but SubByte, ShiftRow and MixColumn are replaced by their inverses; InvSubBytes, InvShiftRows, InvMixColums, and AddRoundKey. It is the reverse order of the encryption structure.

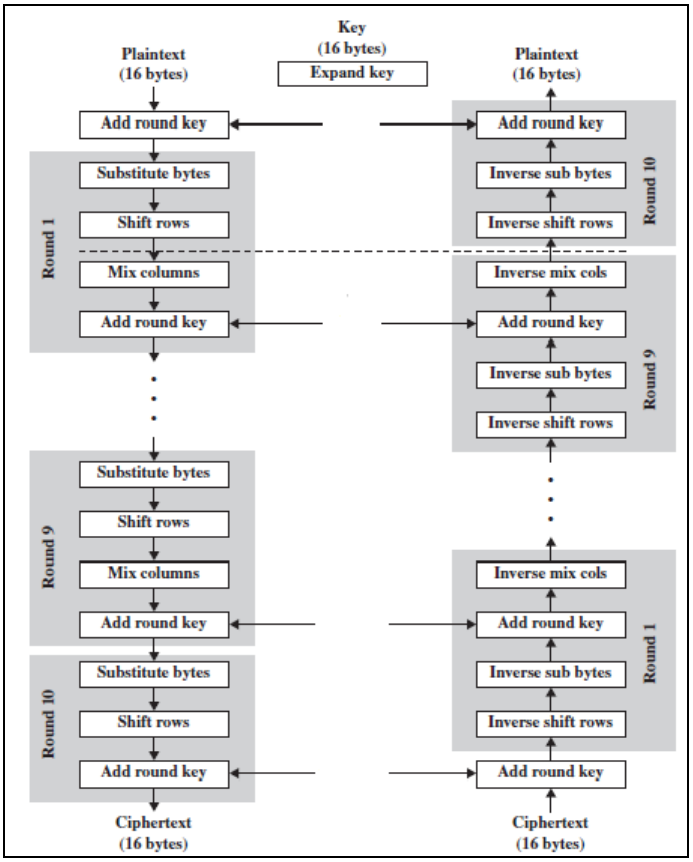

Figure 1. AES encryption and decryption

This paper introduces a new approach for designing key-dependent Advanced Encryption Standard algorithm. This paper is organized as follows: In Section 2, a proposed methodology for designing a key-dependent AES algorithm is illustrated. Section 3 will be the explanation of evaluation criteria. Section 4 suggesting some future enhancement on proposed design and Section 5 summarizes and concludes the paper.

Many efforts have been emulated to redesign and reconstruct the AES algorithm to improve its performance. [6] proposed a new key-dependent AES but the S-box is completely replaced by a new S-Box. AES original S-box has been designed and tested thoroughly for linear and differential attack and attempt on replacing the original S-box without thorough analysis will violate the AES original design and objectives [6].

2 A PROPOSED DESIGN FOR NEW KEY DEPENDENT AES

In recent years there were many cryptanalysis attempts on block ciphers. Most of the attacks prove its effectiveness either on the simplified version or the full version of AES. For this reason, we proposed this cipher to help satisfy the current and foreseeable future requirements [7]. Fig. 2 shows input for a single original AES round.

A single 128-bit block will be the input to the encryption and decryption algorithms where this block is depicted as a square matrix of bytes, and will then be copied into the state array, which will be modified at each stage of encryption or decryption.

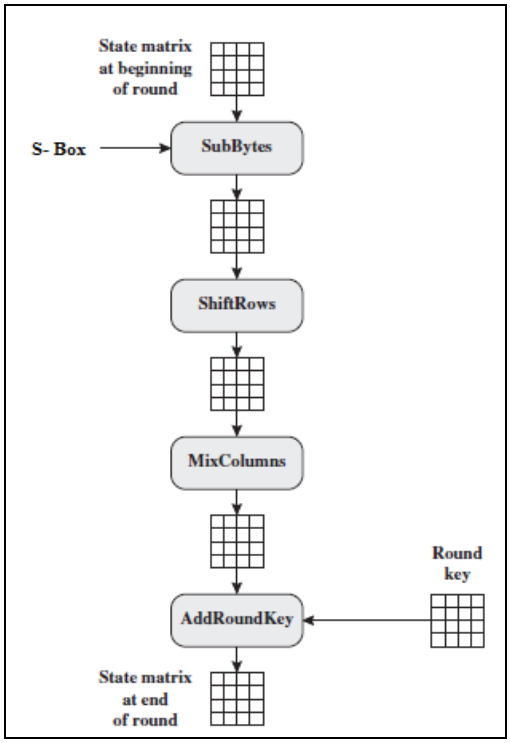

Figure 2. Input for single AES round

The encryption and decryption process of this new design resembles the original AES, it has confusion and diffusion layer and key addition layer as well

but only that the original AES consists of 4 stages while in this new design, it consists of five stages.

The extra stage is known as S-Box Rotation, and is introduced at the beginning of the round function. Fig. 2 shows input for a single round of original AES.

After the final stage of transformation, the state is copied to an output matrix. Similarly, 128-bit key is then expanded into an array of key schedule words: each word is four bytes and the total key schedule is 44 words for the 128-bit key, a round similar to a state.

Fig. 3 and Fig. 4 show the new proposed key-dependent encryption and decryption algorithm. The remaining four stages are unchanged as it is in AES.

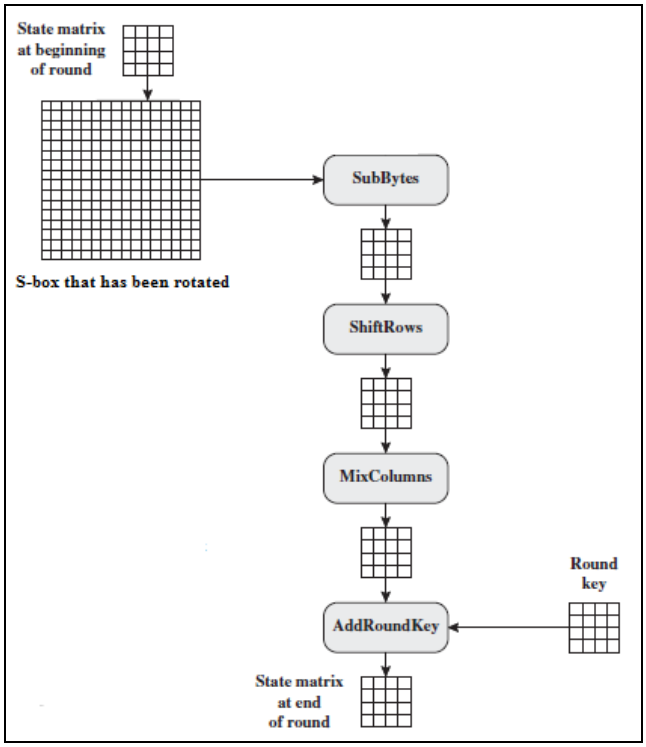

Figure 3. New proposed key-dependent encryption algorithm

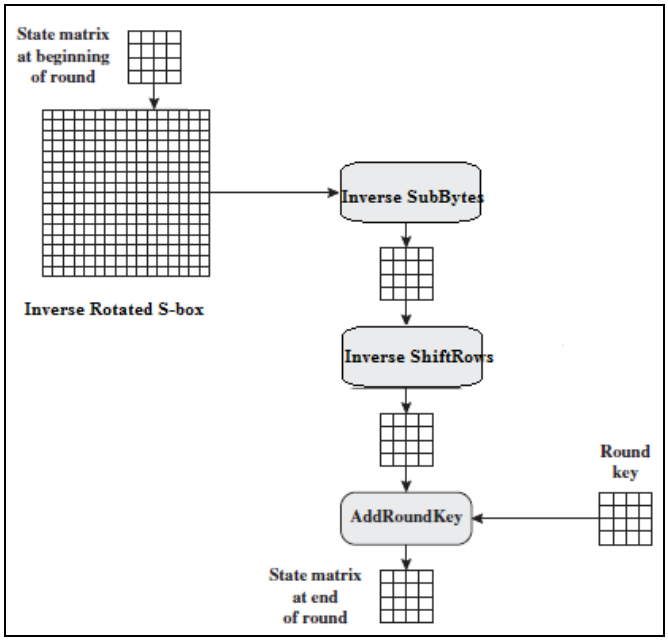

Figure 4. New proposed key-dependent decryption algorithm

2.1 S-Box Rotation and SubBytes/Inverse SubBytes Transformations

This new proposed design uses rotation in the S-Box operation. Below is the AES S-Box, shown in Fig. 5 and Fig. 6 respectively.

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

Figure 5. AES S-box

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

Figure 6. Inverse S-box

In the SubBytes process, each byte in the state is replaced with its entry in the S-box. Consider a byte, say C2 of the state. This will be replaced by 25 (in Hex) as shown in Figure 5.

25(Hex) = S-box(C2)

During the decryption process, the InverseSubBytes operation performs the inverse operation using Inverse S-box as shown in Figure 6. So, the value 25(Hex) will be replaced by the original value C2.

C2 = Inv-S-box(25)

2.2 Key-Dependent S-Box Generation

Static S-box means the same S-box will be used in each round while the key-dependent S-box means the S-box changes in each round depending on the key and number of rounds. Fixed S-box allows attackers to study S-box and find weak points while by using key-dependent S-Box approach, it makes it harder for attacker to do any offline analysis of an attack of one particular set of S-boxes. However, overall performance in terms of security and speed has not been sufficiently addressed and widely investigated [8].

By making the S-box as key-dependent, we assume the AES algorithm will be stronger. We will use the property of above S-box and apply it to an example, to make it key-dependent. The round key generated will be used for finding a value that is used to rotate the S-box.

The subkeys (roundkey) are derived from the cipher key using the key schedule algorithm [9]. Suppose for a particular round n, if the round key value is

7D558EAC0E403CD82D95275E37199242(in Hex)

Apply XOR operation on all the bytes.

9F(Hex)=7D~55~8E~AC~0E~40~3C~D8~2D~95~27~5E~37~19~92~42 (~ symbol used for XOR)

This routine get cipher key as input and generate key-dependent S-Box from cipher key. The resulting 9F(Hex) will then be used to rotate the S-box.

We will next rotate the S-box, let say, to the right by a value say 159(or 9F in Hex). The new S-Box will be as shown in Figure 7.

During the decryption, the InverseSubBytes operation performs the inverse operation using Inverse S-box to get back to the original value.

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

Figure 7. Rotated 9F(Hex)times to the right

The rotation value is now dependent on the entire round key. This property holds for all possible 256 rotations, and this property can be used to make the S-box key-dependent [6][10][11][12]

3 EVALUATION CRITERIA

Having a good block cipher does not guarantee you a better security. In particular, a block cipher must withstand cryptanalysis. Cryptanalysis is to find the weakness and to compromise any cryptosystems

(also referred as attacks). Since Rijndael has been announced as the AES and became standard in 2001, there were various attempts in cryptanalyzing the cipher. In order to make sure that our proposed design will improve the security of the AES, we will perform two main evaluation tests; NIST Statistical Test (consists of 16 test) and one cryptanalysis attack [13].

In this experiment, the data that we use should pass all the 16 NIST Statistical Test. The main test is the frequency test. All the data should pass this test before proceed to the other 15 tests. If the data fails for this frequency test, it means it will fail for the entire 16 test.

3.1 Experimental Result

20000 samples were prepared for both original AES algorithm and the new proposed algorithm. Experiments are performed with 2 different keys and the results has been observed and shown in Table. 1 and Table. 2.

Table. 1. Plaintext and ciphertext (key1) samples

Plaintext	Ciphertext
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	50 94 43 9C F4 BE 6E F4 9C 67 1E E4 54 33 4B 95
01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	10 18 0C 6E A3 4A EE 28 BA 7B 25 1C 2F A6 68 0C

Table. 2. Plaintext and ciphertext (key2) samples

Plaintext	Ciphertext
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	B2 43 B5 85 CA DD F4 4E F5 E6 6E D1 D7 08 B3 0B
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	2F 1D C4 1D DB 52 D6 9D A5 74 99 69 9B 16 31 9E

3.2 Avalanche Effect Measurements

A small change in the plaintext or key that give a large changes in the cipher text is known as avalanche effect. A block cipher is said to have a poor randomization if it does not exhibit the avalanche effect to a significant degree [14]. For a good quality block cipher, such a small change in either key or plaintext should cause a drastic change in the ciphertext.

Table.3. Avalanche effect for 1 bit change in plaintext

Number of samples	Number of times original AES gives better avalanche	Number of time proposed algorithm gives better avalanche	Number of times both original and proposed algorithm give same avalanche
20000	8754	8802	2444

3.3 Strict Avalanche Criterion (SAC)

A function is said to satisfy the strict avalanche criterion, if whenever a single input bit is complemented, each of the output bits should change with a probability of one half, as in [14]. To measure confusion and diffusion, the Strict Avalanche Criterion (SAC) test is presented, together with its results. Table. 4 show the SAC by changing one bit of plaintext in the samples:

Table. 4. Strict Avalanche Criteria for 1 bit change in the plaintext

Number of samples	Number of times original AES gives better avalanche	Number of time proposed algorithm gives better avalanche	Number of times both original and proposed algorithm give same avalanche
20000	8025	8055	3920

Results show that the enhancement on the original AES does not violate the security of the cipher. The enhanced version introduces confusion without violating the diffusion property.

4 FUTURE ENHANCEMENT

AES has been designed to have very strong resistance against the classical attacks, such as linear cryptanalysis and differential cryptanalysis. However since Rijndael is very algebraic, new algebraic attacks appeared [15]. After the new proposed algorithm successfully passes the 16 statistical tests, we will perform one cryptanalysis attack, which is the algebraic attack in attempt to break the cipher and test the security of this new design.

5 CONCLUSION

In this paper, a new design for enhancing the security of AES algorithm is proposed. This approach design will not contradict the security of the original AES algorithm by keeping all the mathematical criteria of AES remain unchanged. We try to improve the security of AES by making its S-box to be key-dependent. We also hope to perform cryptanalysis attack (algebraic attack) on this new algorithm as part of the evaluation test. The result of the cryptanalysis attack will help in permitting its subversion or evasion.

6 REFERENCES

[1] Trappe, W. and Washington, L. C.: Introduction to Cryptography with Coding Theory. United States: Prentice Hall, (2002).

[2] Daemen, J., and Rijmen, V.: The block cipher Rijndael. Proceedings of the Third International Conference on smart card Reseacrh and Applications, CARDIS'98, 1820, pp. 277-284, Berlin: Springer, (2000).

[3] Stallings, W.: Cryptography and Network Security, Prentice Hall, (2010).

[4] Daemen, J. and Rijmen, V.: The First 10 Years of Advanced Encryption. In IEEE Security and Privacy, vol. 8, pp. 72-74, November (2010).

[5] Federal Information Processing Standards Publications FIPS 197, Advanced Encryption Standard (AES), 26 Nov (2001).

[6] Fahmy, A., Shaarawy, M., El-Hadad, K., Salama, G., and Hassanain, K.: A Proposal For A Key-Dependent AES. 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications. Tunisia: SETIT (2005).

[7] Sagheer, A. M., Al-Rawi, S. S., Dawood., A. O.: Proposing of Developed Advance encryption Standard. Developments in E-systems Engineering (DeSE), pp197-202 (2011).

[8] Zhang, R., and Zhen, L.: A Block Cipher using key-dependent S-box and P-Box. ISIE 2008 IEEE International Symposium on Industrial Electronics, pp. 1463-1468 (2008).

[9] Sulaiman, S., Muda, Z., and Juremi, J.: A Proposed Approach of Key Scheduling Transformation. In Cyber Security Cyber Warfare and Digital Forensic(CyberSec), 2012 International Conference, pp. (2012).

[10] Krishnamurthy, G. N., and Ramasvamy, V.: Making AES Stronger:AES with Key Dependent S-Box. IJCSNS International Journal of Computer Science and Network Security, vol.8, pp. 388-398 (2008).

[11] Schneier, B.: Description of a New Variable Length Key, 64-Bit Block Cipher (Blowfish), Fast Software Encryption," Cambridge Security Workshop Proceedings, pp. 191-204, Springer-Verlag (1994).

[12] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., and Ferguson, N.: The Twofish Encryption Algorithm. Proc. 1st Advanced Encryption Standard (AES) Conference (1998).

[13] Rukhin, A., et al.: A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22 (2001).

[14] Forre, R.: The strict avalanche criterion: spectral properties of booleans functions and an extended definition. Advances in cryptology, in: S.Goldwasser(Ed), Crypto'88, Lecture Notes in Computer Science, vol.403, Springer-Verlag, pp.450-468 (1990).

[15] Ferguson, N., Schroeppel, R., and Whiting, D.: A simple algebraic representation of Rijndael. Selected Areas in Cryptography, pp. 103-111 (2001).