http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (2): 290-297, 2010
ISSN 1812-5638

Key Transformation Approach for Rijndael Security

Z. Muda, R. Mahmod and M.R. Sulong
Faculty of Computer Science and Information Technology,
University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia

Abstract: The aim of the study is to improve the security of Rijndael key scheduling by increasing the bit confusion and diffusion of the Rijndael subkey. Rijndael is a block cipher designed by Joan Daemen and Vincent Rijmen. It is a combination of security, performance, efficiency, implementability and flexibility that makes it the best selection for Advanced Encryption Standard (AES). However, the 128 bit Rijndael key schedule does not satisfy the frequency (bit confusion) test for majority of subkeys and does not satisfy the avalanche (bit diffusion) test for any subkeys. These contribute to some attacks in the key schedule. Thus, a new transformation method which is called Shiftrow is proposed into the 128-bit Rijndael Key Schedule based upon information principles (bit confusion and diffusion properties). The new method has shown positive results in terms of the bit confusion and diffusion of subkey and it has increased bit confusion and diffusion compared to the subkey of the original Rijndael key schedule.

Key words: Cryptography, Rijndael algorithm, key transformation approach

INTRODUCTION

Cryptography is the science of encoding and decoding secret messages (Alex-Brennen, 2004). The term cryptanalysis is the study of cryptographic algorithms or resulting ciphertext in order to determine their strengths and potential weaknesses. Often such analysis is performed to break an encryption algorithm or in order to perform key recovery. There are two types of cryptosystems: symmetric and asymmetric key. Both encrypt messages use computer algorithm and provide users with the secrecy through the use of cryptographic keys but still, there are differences in the way the keys are being used. A symmetric cryptosystem has a single key, which is used for both encrypting and decrypting messages.

Mathematical process is used in an algorithm to transform plaintext into ciphertext and vice versa, with each transformation depending on the value of the key. Data Encryption Standard (DES) is a well-known example of symmetric cryptosystem. Others are Triple DES, IDEA, RC2, RC4, RC5 and Blowfish. In contrast to symmetric cryptosystem, public-key or asymmetric cryptosystem uses complementary pair of keys to separate the process of encryption and decryption. One key is considered a pair, with the private key kept secret while the other is made public. However, this research focuses on symmetric-key cryptosystem only.

In 1997, the National Institute of Standards and Technology (NIST) US (an agency of the US Department of Commerce's Technology Administration), initiated a process to select a symmetric-key encryption algorithm to be used to protect sensitivity (unclassified) of Federal Information in furtherance of NIST's statutory responsibilities and to be adopted as Advanced Encryption Standard (AES).

In the same year, NIST announced the acceptance of fifteen candidate algorithms and requested assistance of the cryptographic research community in analyzing the candidates. NIST reviewed the results of the preliminary research and selected Rijndael, Twofish, RC6TM, MARS and Serpent as finalists. Having reviewed further public analysis of the finalists, NIST has decided to propose Rijndael as the Advanced Encryption Standard (AES). Rijndael is a block cipher designed by Joan Daemen and Vincent Rijmen.

There are many other further analysis and improvements that have been done on Rijndael (Daemen and Rijmen, 1999). McLoone and McCanny (2001) proposed Field Programmable Gate Arrays (FPGAs) Rijndael encryption design, utilizes look-up tables to implement entire Rijndael round function. In the same year, Jing *et al.* (2001) derived a new algorithm for computing inverse in GF(2m) on the standard basis. Sklavos and Koufopavlou (2002) designed alternative architectures and VLSI implementation designs. Zhang and Parhi (2002) addressed various approaches for

Corresponding Author: Zaiton Muda, Faculty of Computer Science and Information Technology, University Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan, Malaysia Tel: +60-3-8946-6514 Fax: +60-3-8946-6577

efficient hardware implementation of the AES algorithm. Phan (2004) analyzed the impossible differential cryptanalysis on the AES up to 7 round.

One of the main strengths of the block cipher is based on key scheduling (May *et al*., 2002). The goal of a strong key schedule is to overcome any perceived weakness which may be used to attack the block cipher system. Although, key schedule has been studied for many years, many mathematical properties and weaknesses of these designs are insufficiently discovered to make the block cipher fully secured. In particular, the design of the block cipher does not focus on key schedule (specifically, bit diffusion) as extensively as to the encryption process. In contrast to the serious efforts applied to algorithm design, the aspect of the key schedule has received comparatively little attention. This is despite the fact that published block ciphers are vulnerable to known attacks that exploits the weaknesses of their key schedules. Often, an attack is based on one or more important insights into aspects of cipher key behavior to deduce information about the cipher key (Daemen *et al*., 1993; Knudsen, 1994).

Ferguson *et al.* (2000) examined the security of the AES candidate Rijndael, described several new attacks and unexpected properties of the cipher and introduced the "partial sum" technique, which substantially reduces the workfactor of the dedicated square attack. According to Ferguson, compared to the cipher itself, the Rijndael key schedule appears to be a more ad hoc design and the key schedule does not achieve its stated design goals, especially for 192 and 256 bit keys and it has much slower diffusion structure than the cipher and contains relatively few non-linear elements. Similarly, Phan (2004) analyzed the attack (impossible differential cryptanalysis) on the AES up to 7 round. However, in order to make the attack look computationally infeasible, significant properties are frequently approximated. There are certain properties (e.g., bit confusion and bit diffusion) either theoretically or computationally which shall decrease the weaknesses of the key scheduling. In cryptography, confusion and diffusion are two properties for a secure cipher which was identified by Bauer (2007). These properties are bit confusion and bit diffusion that shall be applied in this research. An important theoretical underpinning for bit confusion and bit diffusion is the idea of frequency and Strict Avalanche Criterion (SAC), respectively. The SAC obviates the need for a widely used approximation, allowing more accurate evaluation of the bit diffusion to key schedule.

May *et al*. (2002) proposed and analyzed an efficient and more secure key schedule. However, there is a large unexplored area of research concerning the bit diffusion. To combat this weakness, a function, ShifRow() transformation is added to the Rijndael key schedule design to diffuse the keys in order to increase the security of a block cipher. The Shiftrow transformation operates on the rows of the state to provide diffusion (Savard, 2001).

MATERIALS AND METHODS

The focus of this stage is to identify the exact obstacles of Rijndael key schedule. For that purpose, the existing papers until the current issues of the AES as general and Rijndael in specific have been reviewed to understand and to find out the related obstacles of the block cipher and cipher key. The previous and the latest improvement of the AES are studied, thus giving a complete identification of it. After having reviewed the related papers, it was found that there is security vulnerability in the AES algorithm especially in the key schedule design. This is the core idea where the obstacle of this study is derived from. The experiment was carried out at the Faculty of Computer Science and Information Technology, University Putra Malaysia on 2008.

Input requirement: The 128 bit key is suitable as the input in this study. This is due to the fact that the different subtests in the SAC test (e.g., for different bit length; 128, 192, or 256 bits) do not seem to be obviously related (Castro *et al*., 2004). Also, in Nechvatal *et al*. (2000) stated that 128 bit input is a minimum requirement for the block size. The input can be divided into two types: random key and non-random key. The random key is generated from the Pseudo Random Number Generator (PRNG) (e.g., 2 B, 7 E, 1 5, 1 6, 2 8, A E, D 2, A 6, A B, F 7, 1 5, 8 8, 0 9, C F, 4 F, 3 C) while the non-random key is taken based on the user's own data input (e.g., 1 2, 3 4, 5 6, 7 8, 9 A, B C, D E, F 1, B 3, A A, 1 1, 0 0, 0 0, 0 0, 1 2, C D). The data inputs are provided in hexadecimal rather than in binary form to be organized easier. The inputs are inserted at the beginning of the key schedule transformation. Then, each input will produce 10 subkeys in hexadecimal form.

Performance measurement: It is important to make sure that the result satisfies the value needed in key schedule. To measure the results, it was found that two statistical tests have to be performed in this experiment to evaluate the diffusion of Rijndael key schedule; the frequency test (May *et al*., 2002) and the Strict Avalanche Criterion (SAC) test.

For the frequency test (from NIST package), the p (probability) value is used. The p-value must be greater

than or equal to 0.01. If the p-value is less than 0.01, it was caused by too many zeros exist in the data input.

For the SAC test (from SPSS software using the one-sample Kolmogorov-Smirnov test (1-sample K-S test)), D value is used that is a product of the largest absolute difference between the empirical distribution (sample observed) and theoretical distribution (hypothesis). The observed D values should be less than 1.628 or 1.949 to indicate that the bit diffusion is satisfied at the 1 or 0.1% critical level in order to increase the security of block cipher algorithm, respectively.

Design and description of Shiftrow: As mentioned by Bauer (2007), confusion and diffusion are two properties for a secure cipher. Shiftrow plays its roles by shifting the places of bytes in a state where it automatically affects the output/result of the state which contributes to the properties that will increase the security of Rijndael. So, that is why we choose Shiftrow as the method in this study.

Shiftrow is used in block cipher algorithm. The Shiftrow transformation operates on the rows of the state, providing diffusion. In this step, the rows of the state block are shifted to the left, using different offsets for each row. The first row (row 0) is offset by 0; in other words, it does not change. The shift offsets for the remaining rows depending on the length of the block. In a 128 bit block, represented as a 4×4 matrix, the second row (row 1) is shifted to the left by one byte, the third row (row 2) is shifted to the left by two bytes and the fourth row (row 3) is shifted to the left by three bytes. This transformation ensures that the different bytes in each row do not interact solely with their corresponding bytes in other rows (Savard, 2001).

In the state, both the MixColumn and Shiftrow transformation are linear operations that work together to ensure that all of the bytes in a block affect each other, thereby generating high diffusion over multiple rounds (Daemen and Rijmen, 1999). Similarly, Row shifting in the AES corresponds to permutation P in the DES. Both the Shiftrow transformation and permutation P provide diffusion (Wright, 2001). These descriptions indicate that using the Shiftrow, diffusion is attained; thus clarifying the decision to use Shiftrow in the Rijndael key scheduling in order to improve the diffusion of Rijndael subkey. In the key scheduling, the Shiftrow transformation operates in the same steps as well as in the block cipher. The only difference is the Shiftrow in the block cipher operates on the rows of the state, while Shiftrow in the key scheduling operates on the rows of the subkey. In addition, if the value of subkey distribution is less than 1.628 or 1.949, diffusion is applies. Otherwise, the diffusion is not applies.

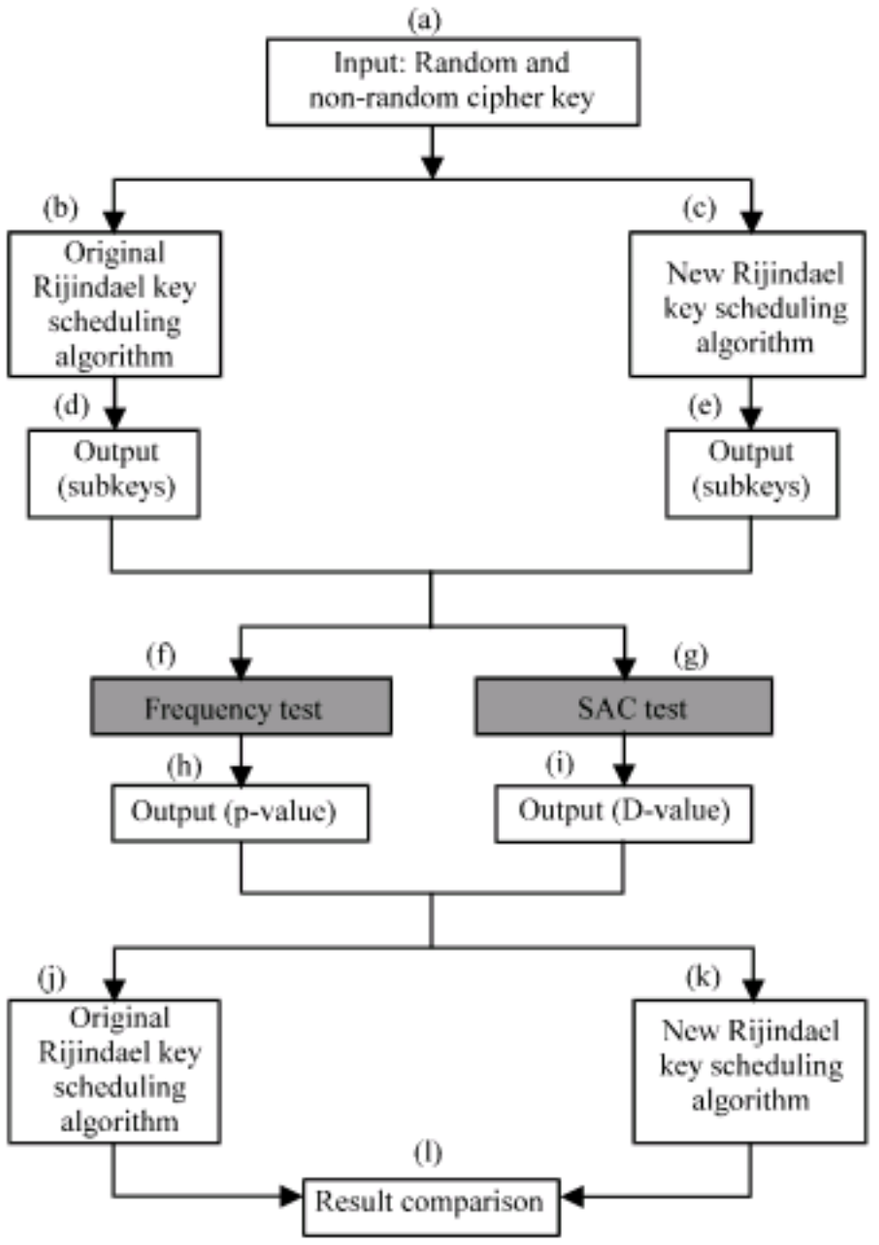

Fig. 1: The performance comparison process between the original and the new rijndael key scheduling algorithm

Implementation: Random and Non-Random data input are collected. A selected programming language (C++) is used in writing the program code (for a new key schedule approach). The complete program is run under the suitable computer platform (Windows XP) by using the selected data input. The produced output is evaluated using two tests, the frequency test and the SAC test. The values produced by these tests are measured, statistically. The whole values are then illustrated in graph and chart. The complete implementation of this experiment is shown in Fig. 1.

THE PROPOSED APPROACH

Here, we describes the new approach of Rijndael Key Expansion (Fig. 2). As mentioned earlier, the original Rijndael Key Expansion uses a round function that consists of three different byte-oriented transformations; RotWord, SubBytes and RCon. However, in the new approach, ShiftRow transformation is added into the original Rijndael Key Expansion. Thus, the new approach of Rijndael Key Expansion consists of four different

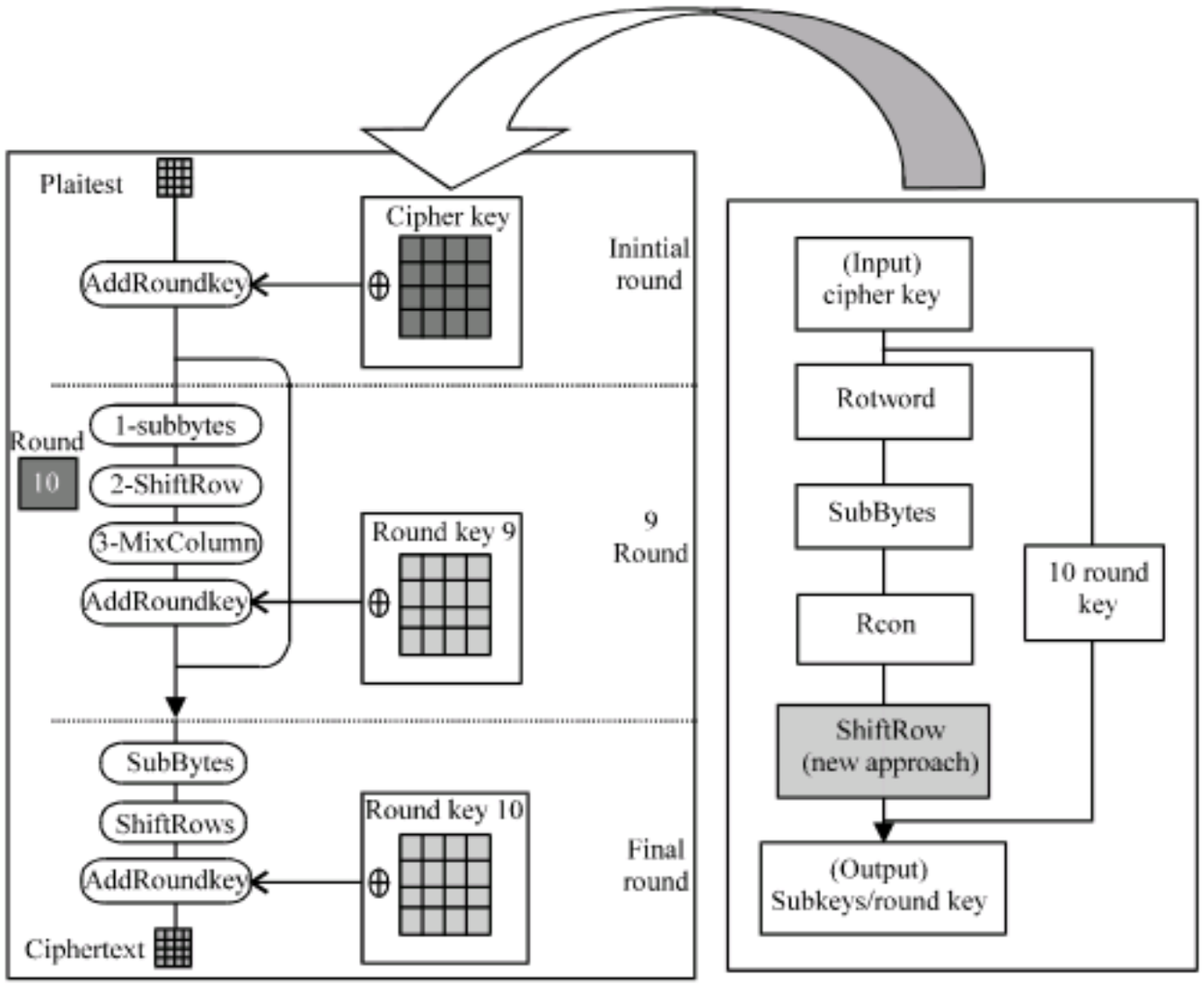

Fig. 2: The new structure of rijndael key expansion with four different byte-oriented transformations

byte-oriented transformations; RotWord, SubBytes, RCon and ShiftRow. ShiftRow is inserted and applied to the final result of 128 bit round key to diffuse the keys. This transformation is applied for each round keys (subkeys). The whole key expansion algorithm pseudo code is stated in Fig. 3.

The expanded key is a linear array of 4-byte words and is denoted by W[Nc*(Nr+1)]. For AES, (Nk = 4, 6, or 8) words contain the cipher key (input). All other words are defined recursively in terms of words with smaller indices. The key expansion function depends on the value of Nk (Fig. 3 on Line 1 and 15): there is a version for Nk equal to or below 6 and a version for Nk above 6. In this study, we shall focus only for Nk = 4. In this description, SubByte(W) (Fig. 3 on Line 13 and 16) is a function that returns a 4-byte word in which each byte is the result of applying the Rijndael S-box to the byte at the corresponding position in the input word.

The function RotWord(W) (Fig. 3 on Line 13) returns a word in which the bytes are a cyclic permutation of those in its input. It can be seen where the first Nk words are filled with the cipher key. Every following word W[i] is equal to the EXOR of the previous word W[i-1] and the word Nk positions earlier W[i-Nk]. For words in positions that are a multiple of Nk, a transformation is applied to W[i-1] prior to the EXOR and a round constant (Fig. 3 on Line 14) is EXORed. This transformation consists of a

```
Line
1   Key Expansion(byte key[4*NK], word k [Nr+1, Nc], Nc, Nk, Nr)
2   begin
3       i =0
4       while (i<Nk)
5           k[i] = word [key[4*i+3], key[4*i+2], key[4*i+1], key[4*i]]
6           i = I+1
7   end while
8   i = Nk
9   while (I< Nc*(Nr+1))
10      word temp = k[i-1]
11      if (i mod Nk = 0)
12          temp = (SubByte(RotWord(temp))xor
13          Rcon[i/Nk]))
14      end if
15          k[i] = k[i-Nk] xor temp
16      if (i mod Nk = 3)
17      // Apply the new approach "ShiftRow" transformation
18          word temp1 [4][4]
19          for (c = 3; c>= 0; c-1)
20              temp1 [c][r] = k[Nk+c]
21          ShiftRow (temp1);
22      end if
23      i = i +1
24  end while
25  end
```

Fig. 3: The key expansion algorithm pseudo code of the new structure of rijndael key

cyclic shift of the bytes in a word (RotWord), followed by the application of a table lookup to all four bytes of the

Table 1: The input and output of Rijndael key schedule

Key length (bits)	Input	Output
128	Random cipher key	10 Subkeys of random cipher key
	Non-random cipher key	10 Subkeys of non-random cipher key

word (SubByte). The new approach, ShiftRow transformation is then applied for each word (ShiftRow) which is not done in the original approach. This produced the result of subkey (output) which is the final output for the new approach of Rijndael Key Expansion (Fig. 3 on Line 13).

Input and output: Here, describes in detail the input and output of this experiment. It is very important to ensure that every input satisfies the input requirement of the experiment to produce the correct output. The input for this experiment is cipher key with the key length of 128 bits; where 128-bit cipher key is a minimum requirement for block size (Nechvatal *et al*., 2000). There are two types of cipher keys; random and non-random cipher keys which are depicted in Table 1. Random cipher key means random data is used as input to produce the subkeys as output, while the non-random cipher key means non-random data that is not produced from specific generators (e.g., user's own data) is used as input to produce the subkeys as output. Normally, the data input is in hexadecimal form (Fig. 2). However, any type of data input (e.g., the subkeys) should be transformed into binary form before it can be used in the frequency and the SAC test. The 10 subkeys are the outputs that are produced from the random and the non-random cipher key, respectively.

Test: This experiment applies two tests; the frequency and the SAC test as a benchmark in measuring the bit confusion and diffusion, respectively. The output (outcome) of this experiment is the probability value (p-value) for bit confusion and the largest absolute difference between the empirical distributions (sample observed) and theoretical distribution (hypothesis) (D_{max}) for bit diffusion, which measures the output of the original and the new approach algorithms. The results of the tests are in statistical values. The following is the details of each statistical test in measuring the 36 data inputs of 128-bit random and non-random cipher key.

The frequency test is proposed by National Institute of Standard and Technology (NIST). The purpose of this test is to determine whether the amount of ones and zeros in a sequence are approximately the same as which would be expected for a truly random sequence. The calculated values of frequency test that are based on bit ones is shown in the frequency test table (Table 2). This table is purposely created in this study as an easy way to get the p-values for each bit of ones subkey. Here, only the most frequent bit ones of subkeys are provided with their related p-values. The bit of ones (50-82 bit of ones in a subkey) that are stated in the frequency test table indicate that they are the most frequently appeared in subkeys. The experiment refers to Table 2 to get the p-values of bit of ones of each subkey.

Table 2: Frequency test table for the number of bit ones and their p-values

Bit of ones of subkeys	p-values
50	0.148
51	0.143
52	0.137
53	0.131
54	0.125
55	0119
56	0.112
57	0.105
58	0.097
59	0.088
60	0.079
61	0.068
62	0.056
63	0.040
64	0.000
65	0.125
66	0.177
67	0.217
68	0.250
69	0.280
70	0.306
71	0.331
72	0.354
73	0.375
74	0.395
75	0.415
76	0.433
77	0.451
78	0.468
79	0.484
80	0.500
81	0.515
82	0.530

RESULTS

Frequency test: In this experiment, there are 36 data sets in hexadecimal form that are converted into binary form before they can be used as data input in the frequency test. Each data set contains 10 subkeys, which means the total of subkeys are 360 (36 data sets * 10 subkeys of a data set). The amount of bit ones of each subkey then is referred to the frequency test table (Table 2) to obtain their related p-values. These are applied for all the 360 subkeys. The result of this test is shown in Table 3 (for random subkeys) and Table 4 (for non-random subkeys), where each table consists of 180 subkeys, respectively.

Both tables (Table 3, 4) show that 327 out of 360 subkeys are successfully confused (mixed) in terms of their bit confusion. This is because p-values of the subkeys are greater than 0.01 or 0.001 which indicates that bit mixing is satisfied at the 1 or 0.1% critical level.

Table 3: The results of experiment of 180 random subkeys presented in p-value

Subkey	Freq (p-values)																	
1	0.079	0.088	0.056	0.088	0.079	0.125	0.08	0.105	0.09	0.105	0.250	0.000	0.00	0.06	0.28	0.105	0.112	0.079
2	0.306	0.079	0.131	0.137	0.217	0.068	0.09	0.217	0.06	0.125	0.217	0.280	0.33	0.18	0.00	0.354	0.088	0.056
3	0.056	0.177	0.280	0.097	0.375	0.217	0.42	0.375	0.40	0.331	0.119	0.131	0.06	0.00	0.07	0.00	0.148	0.250
4	0.000	0.040	0.217	0.306	0.125	0.331	0.38	0.217	0.33	0.112	0.354	0.000	0.28	0.13	0.28	0.415	0.217	0.217
5	0.250	0.217	0.395	0.112	0.250	0.177	0.28	0.000	0.12	0.137	0.415	0.040	0.22	0.04	0.07	0.088	0.119	0.395
6	0.125	0.040	0.105	0.056	0.112	0.068	0.10	0.040	0.13	0.000	0.040	0.217	0.09	0.08	0.18	0.04	0.105	0.040
7	0.056	0.068	0.280	0.306	0.000	0.433	0.31	0.250	0.06	0.131	0.217	0.306	0.00	0.28	0.08	0.217	0.280	0.000
8	0.068	0.217	0.354	0.056	0.088	0.097	0.28	0.177	0.06	0.079	0.056	0.079	0.18	0.08	0.18	0.177	0.217	0.105
9	0.395	0.000	0.056	0.280	0.000	0.280	0.04	0.000	0.06	0.000	0.250	0.088	0.13	0.08	0.33	0.125	0.177	0.079
10	0.177	0.097	0.056	0.280	0.177	0.177	0.08	0.125	0.31	0.056	0.040	0.112	0.25	0.04	0.04	0.097	0.056	0.306

Table 4: The results of experiment of 180 non-random subkeys presented in p-value

Subkey	Freq (p-values)																	
1	0.217	0.125	0.500	0.375	0.125	0.000	0.354	0.056	0.250	0.880	0.451	0.306	0.354	0.079	0.354	0.04	0.331	0.079
2	0.056	0.375	0.177	0.105	0.040	0.306	0.177	0.056	0.177	0.137	0.040	0.079	0.097	0.306	0.354	0.354	0.415	0.530
3	0.056	0.068	0.000	0.068	0.088	0.177	0.112	0.433	0.056	0.088	0.097	0.375	0.000	0.056	0.306	0.056	0.125	0.000
4	0.068	0.177	0.125	0.306	0.088	0.097	0.000	0.041	0.217	0.217	0.125	0.306	0.125	0.040	0.079	0.00	0.105	0.375
5	0.250	0.375	0.125	0.112	0.040	0.000	0.040	0.000	0.217	0.119	0.105	0.040	0.375	0.250	0..97	0.125	0.125	0.217
6	0.217	0.000	0.790	0.177	0.056	0.097	0.250	0.306	0.125	0.040	0.125	0.217	0.000	0.177	0.112	0.131	0.112	0.105
7	0.331	0.040	0.079	0.068	0.056	0.354	0.056	0.056	0.040	0.306	0.137	0.097	0.306	0.000	0.112	0.068	0.451	0.056
8	0.177	0.331	0.375	0.156	0.079	0.250	0.056	0.079	0.097	0.395	0.143	0.125	0.068	0.217	0.125	0.088	0.28	0.000
9	0.088	0.068	0.125	0.970	0.177	0.354	0.177	0.306	0.331	0.040	0.515	0.250	0.000	0.331	0.056	0.415	0.105	0.306
10	0.068	0.177	0.125	0.500	0.000	0.375	0.079	0.433	0.000	0.280	0.395	0.105	0.415	0.00	0.000	0.079	0.068	0.250

Table 5: The result of poison distribution (D-values) of 180 random subkeys

Subkey	SAC (D-value)																	
1	1.070	1.037	1.137	1.037	1.070	1.240	1.07	0.972	1.04	0.972	1.348	1.205	1.21	1.14	1.38	0.972	0.941	1.070
2	1.421	1.070	0.849	0.819	1.311	1.103	1.04	1.311	1.04	1.240	1.311	1.384	1.46	1.28	1.21	1.497	1.384	1.137
3	1.317	1.276	1.384	1.004	1.535	1.311	1.61	1.535	1.57	1.459	0.910	0.849	1.14	1.21	1.10	1.205	0.761	1.348
4	1.205	1.171	1.311	1.421	1.240	1.459	1.54	1.311	1.46	0.941	1.497	1.205	1.38	1.24	1.38	1.612	1.311	1.311
5	1.348	1.311	1.573	0.941	1.384	1.276	1.38	1.205	0.91	0.819	1.612	1.171	1.31	1.17	1.10	1.037	0.910	1.573
6	1.240	1.171	0.972	1.137	0.941	1.103	1.00	1.171	0.85	1.205	1.171	1.311	1.04	1.07	1.28	1.171	0.972	1.171
7	1.137	1.103	1.384	1.421	1.205	1.652	0.85	1.348	1.14	0.849	1.311	1.421	1.21	1.38	1.07	1.311	1.384	1.205
8	1.103	1.311	1.497	1.137	1.037	1.004	1.38	1.276	1.14	1.070	1.137	1.070	1.28	1.07	1.28	1.276	1.311	0.972
9	1.573	1.205	1.137	1.384	1.205	1.384	1.17	1.205	1.14	1.205	1.348	1.037	1.24	1.07	1.46	1.240	1.276	1.070
10	1.276	1.004	1.137	1.384	1.276	1.276	1.07	1.240	1.42	1.137	1.171	0.941	1.35	1.17	1.17	1.004	1.137	1.421

Table 6: The result of poison distribution (D-values) of 180 non-random subkeys

Subkey	SAC (D-value)																	
1	1.311	0.879	1.813	1.535	1.240	1.205	1.497	1.137	1.348	1.037	1.692	1.421	1.497	1.070	1.497	1.171	1.459	1.070
2	1.137	1.535	1.276	0.972	1.171	1.421	1.276	1.137	1.276	0.819	1.171	1.070	1.004	1.421	1.497	0.879	1.612	1.896
3	1.137	1.103	1.205	1.103	1.037	1.276	0.941	1.652	1.137	1.037	1.070	1.535	1.205	1.137	1.421	1.137	1.240	1.205
4	1.103	1.276	1.240	1.421	1.037	1.007	1.205	1.171	1.311	1.311	1.240	1.421	1.240	1.171	1.070	1.205	0.972	1.535
5	1.384	1.535	1.240	0.971	1.171	1.205	1.171	1.205	1.311	0.910	0.972	1.171	1.535	1.348	1.004	1.240	1.240	1.311
6	1.311	1.205	1.070	1.276	1.137	1.070	1.348	1.421	1.240	1.710	0.879	1.311	1.205	1.276	0.941	0.849	0.941	0.972
7	1.459	1.171	1.070	1.103	1.276	1.497	1.137	1.370	1.171	1.421	0.819	1.004	1.421	1.205	0.941	1.103	1.692	1.137
8	1.276	1.459	1.535	1.137	1.070	1.348	1.137	1.070	1.004	1.573	0.790	1.240	1.103	1.311	1.240	1.037	1.384	1.205
9	1.037	1.103	1.240	1.004	1.276	1.497	1.276	1.421	1.459	1.171	1.854	1.348	1.205	1.459	1.137	1.612	0.972	1.421
10	1.103	1.276	1.240	1.813	1.250	1.535	1.070	1.652	1.205	1.384	1.573	.972	1.612	1.205	1.205	1.070	1.103	1.348

The p-value = 0.01 signifies that the bit of ones is more than the bit of zeros from each 128 bits of subkey. The results show that, 327 subkeys out of 360 subkeys contain more bit ones than bit of zeroes. The p-value <0.01 signifies that bit of zeroes is more than bit of ones from each 128 bits of subkey. In this experiment, 33 subkeys out of 360 subkeys contain more bit zeros than bit of ones. The p-value = 0 signifies that the total of bit ones (64 bit of ones) are equal to bit zeros (64 bit of zeros) from 128 bits of subkeys. Figure 3 shows the result of selected subkeys of the new approach (random cipher key) versus the original approach.

SAC test: The results of poison distribution of SAC Test are shown in Table 5 (for random subkeys) and Table 6 (for non-random subkeys). Both tables show the results of 360 subkeys that have been tested by SAC test and satisfied the test, where the D-value of subkeys are less than 1.628 or 1.949 which indicates that the bit diffusion is satisfied at the 1 or 0.1% critical level. Figure 4

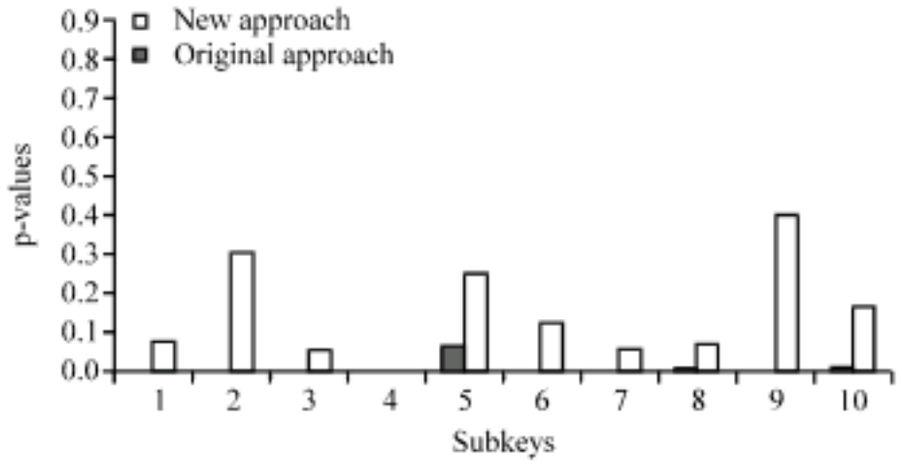

Fig. 4: The selected subkeys of the new approach (random cipher key) and original approach

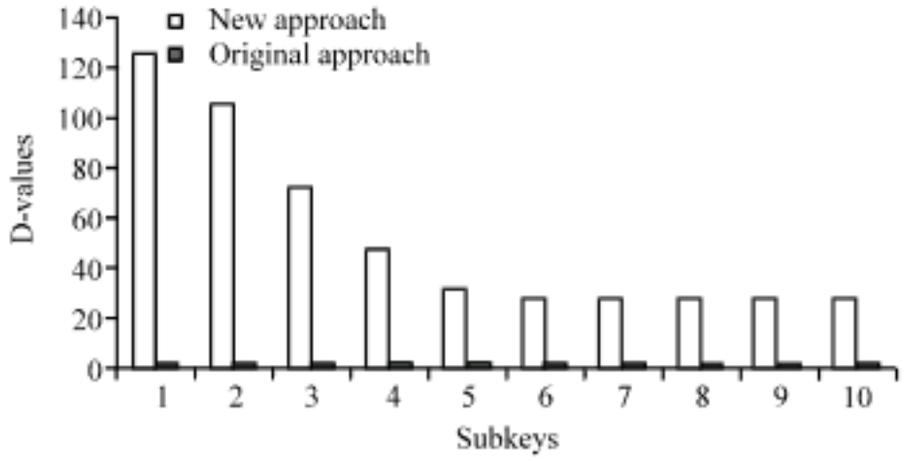

Fig. 5: The selected subkeys of the new approach (non-random cipher key) and original approach

shows the result of selected subkeys of the new approach (non-random cipher key) versus the original approach.

DISCUSSION

The bit confusion and diffusion are important properties that have to be considered in order to improve the security of the Rijndael Key Scheduling (May *et al.*, 2002). The results from the frequency and the SAC test show that the new approach achieved the aim of the study which is to improve the security of Rijndael key scheduling by increasing the bit confusion and diffusion of the Rijndael subkey.

Frequency test: After analyzing the results from both tables (Table 3 (random subkey) and Table 4 (non-random subkey)) and Fig. 5, the finding of the new approach has significant confusion outputs than the original approach where the P-values of the new approach are higher than the P-values of the original approach for nine subkeys out of ten subkeys. This is because the ShiftRow transformations form the linear mixing layer which increases the bit confusion output that contributes to the security of the Rijndael block cipher.

SAC test: Having analyzed the result from both tables (Table 5 (random subkey) and Table 6 (non-random subkey)) and Fig. 5, the finding of the new approach has significant diffusion outputs than the original approach because all the D-values of 10 subkeys of the new approach are less than 1.628 or 1.949 while none of the original approach values are less than 1.628 or 1.949. This is because the ShiftRow transformations ensure that all of the bytes in a block affect each other, which generates high diffusion over multiple rounds in order to increase the bit diffusion output that contributes to the security of the Rijndael block cipher.

CONCLUSION

This study is focused on Rijndael algorithm version of 128-bit key size with 10 rounds consideration. This research modifies the algorithm of Rijndael Key Scheduling to emphasize on the bit confusion and diffusion. After analyzed the result from both tests, the new transformation method which is called ShiftRow in the Rijndael Key Scheduling contributes to the security of the Rijndael block cipher.

REFERENCES

Alex-Brennen, V., 2004. Cryptography dictionary. http://cryptnet.net/fdp/crypto/crypto-dict/en/crypto-dict.html.

Bauer, F.L., 2007. Decrypted Secrets Methods and Maxims of Cryptology Communication. Springer, New York.

Castro, J.C.H., J.M. Sierra, A. Seznec, A. Izquierdo and A. Ribagorda, 2004. The Strict Avalanche Criterion Randomness Test. Elsevier, UK.

Daemen, J., R. Govaerts and J. Vandewalle, 1993. Weak Keys for IDEA. In: Advances in Cryptology, Stinson, D.R. (Ed.). Springer Verlag, New York, pp: 224-231.

Daemen, J. and V. Rijmen, 1999. Proposal rijndael algorithm submission. http://www.nist.gov/aes.

Ferguson, N., J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner and D. Whiting, 2000. Improved cryptanalysis of rijndael in fast software encryption. Proceedings of the 7th International Workshop on Fast Software Encryption, Apr. 10-12, Springer Verlag, pp: 213-230.

Jing, M.H., Y.H. Chen, Y.T. Chang and C.H. Hsu, 2001. The design of the fast inverse module in aes, info-tech and info-net, 2001. Proceedings of the ICII International Conference on Info-tech and Info-net, Oct. 10-Nov. 1, Beijing, China, pp: 298-303.

Knudsen, L., 1994. New Potentially Weak Keys for DES and LOKI. Advances in Cryptology-EUROCRYPT' 94, LNCS 950. Springer-Verlag, New York, pp: 419-424.

May, L., M. Henricksen, W. Millan, G. Carter and E. Dawson, 2002. Strengthening the key schedule of the AES. Proceedings of the 7th Australasian Conference on Information Security and Privacy, Jul. 3-5, Melbourne, Australia, pp: 117-134.

McLoone, W. and J.V. McCanny, 2001. Rijndael FPGA implementation utilizing look-up tables. Proceedings of the IEEE Workshop on Signal Processings Systems, Sept. 26-28, Antwerp, Belgium, pp: 349-360.

Nechvatal, J., E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti and E. Roback, 2000. Report on the Development of the Advanced Encryption Standard (AES). National Institute of Standard and Technology, USA.

Phan, R.C.W., 2004. Impossible Differential Cryptanalysis of 7-Round Advanced Encryption Standard (AES), Information Processing Letters. Vol. 91, Elsevier Science, New York, pp: 33-38.

Savard, J., 2001. The advanced Encryption standard (Rijndael). http://home.ecn.ab.ca/ not, vert, similar jsavard/crypto/co040801.htm.

Sklavos, N. and O. Koufopavlou, 2002. Architectures and VLSI implementations of the AES-proposal rijndael. IEEE Trans. Comput., 51: 1454-1459.

Wright, M.A., 2001. The advanced encryption standard. Network Security, 2001: 11-13.

Zhang, X. and K.K. Parhi, 2002. Implementations approaches for the advanced encryption standard algorithm. IEEE Circuits Syst. Magazine, 2: 24-46.