International Journal of Cryptology Research 1(2): 245-255 (2009)

A Shift Column with Different Offset for Better Rijndael Security

Ramlan Mahmod, Sherif Abdulbari Ali and Abdul Azim Abd.Ghani
Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia,
43400 UPM Serdang, Selangor, Malaysia
E-mail: ramlan.mahmod@mimos.my

ABSTRACT

The strength of an encryption algorithms depends on the key's secrecy combined with the structure of the block cipher that is able to produce random output. The goal of a strong symmetric key encryption algorithm is that there is no way to decrypt the data except by knowledge of the key and there is no better way to find out that key than key exhaustion [1]. The secrecy of an encryption algorithm is measured in terms of the computational power and time required to extract the secret key. The security of the algorithm on the other hand, is based on the randomness of the output from the encryption process. This is the result of a combination of strong key and the structure of the block cipher. Rijndael, currently the Advanced Encryption Standard Algorithms (AES) is a block cipher uses a 128, 192, or 256-bit key length to encrypt 128-bit blocks of plaintext. Structurally, it has larger S-boxes, but a very simple algebraic description that make it particularly vulnerable [3]. This paper proposes a transformation function to be added to the Rijndael algorithm. It is called a ColumnShift() with different offset values that is added to the currently four transformation functions. The main objective is to increase the security of the encryption. A comparison between the Rijndael algorithm and the new approach shows that the security or the randomness by the proposed approach is better than the Rijndael.

Keywords: AES, Rijndael, Transformation Function, Randomness, Security, Cryptography.

INTRODUCTION

RIJNDAEL [2] is a substitution-linear transformation network with 10, 12 or 14 rounds, depending on the key size. A data block to be processed using Rijndael is partitioned into an array of bytes, and each of the cipher operations is byte-oriented. Rijndael's round function consists of four layers. In the first layer, an 8x8 S-box is applied to each byte. The second and third layers are linear mixing layers, in which the rows of the array are shifted, and the columns are mixed. In the fourth layer, subkey bytes are XORed into each byte of the array. In the last round, the column mixing is omitted. [3].

Randomness of the algorithm's output is required for cryptographic algorithm, where the randomness of the output is one factor of measuring security [3]. The NIST evaluation criteria of the cryptographic algorithm were divided into three major categories: Security, Cost and Algorithm and Implementation Characteristics, where the security was the most important factor in the evaluation [4]. The AES algorithm security depends only on the key's secrecy. The goal of a strong symmetric key encryption algorithm is that there is no way to decrypt the data except by knowledge of the key and there is no better way to find out that key than key exhaustion [1]. The AES algorithm have been reported for 128-bit keys, 7 rounds out of 10 have been attacked; for 192-bit keys, 7 rounds out of 12 have been attacked and for 256-bit keys, 9 rounds out of 14 have been attacked by the Related-Key and Differential cryptanalysis attacks [5], [6], [7] and [8].

This paper proposes a transformation function to be added to the Rijndael algorithm. The new transformation function is shifting the columns of the Rijndael's state after the Mixcolumn function is applied to the state. This transformation function improves the security of the Rijndael algorithm by increasing the randomness of the Rijndael's output sequence. The approach has increased the randomness of the output in comparison to the output sequence of the Rijndael algorithm.

RIJNDAEL ALGORITHM

As in the Rijndael algorithm, the length of the input block, the output block and the State is 128 bits. This is represented by ***Nb*** = 4, which reflects the number of 32-bit words (number of columns) in the State. For the AES algorithm, the length of the Cipher Key, K, is 128, 192, or 256 bits. The key length is represented by ***Nk*** = 4, 6, or 8, which reflects the number of 32-bit words (number of columns) in the Cipher Key. The number of rounds to be performed during the execution of the algorithm is dependent on the key size. The number of rounds is represented by ***Nr***, where ***Nr*** = 10 when ***Nk*** = 4, ***Nr*** = 12 when ***Nk*** = 6, and ***Nr*** = 14 when ***Nk*** = 8. In both its Cipher and Inverse Cipher, the Rijndael algorithm uses a round function that is composed of four different byte-oriented transformations:

1. SubByte(): Transformation in the Cipher that processes the state using a nonlinear byte substation table (S-box) that operates an each of the Store bytes independently.

2. ShiftRows(): Transformation in the Cipher that processes the state by cyclically shifting the last three rows of the state by different offsets.

3. MixColumns():Transformation in the Cipher that take all of the columns of the state and mixes their values (independently of one another) to produce new columns.

4. AddRoundKey():Transformation in the Cipher and Inverse Cipher in which a Round key is applied to the state using an XOR operation. The length of a round key equals the size of the state.

Encryption process	Decryption process
AddRoundKey(state) for Round = It0 Nr-1 SubBytes(state) ShiftRows(state) MixColumns(state) AddRoundKey(state) End for SubBytes(state) ShiftRaws(state) AddRoundKey(state)	AddRoundKey(state) for Round = It0 Nr-1 InvShiftRaws(state) InvSubBytes(state) AddRoundKey(state) InvMixColumns(state) End for InvShiftRows (state) InvSubBytes (state) AddRoundKey(state)

Figure 1: Cipher and Inverse Cipher

Similarly, for decryption, a round function consists InvSubBytes(s), InvShiftRows(), InvMixColumns(), which is the inverse transformation of SubBytes(), ShiftRows(), MixColumns() respectively.

THE NEW APPROACH

This section describes the new approach, which is actually a modification on the Rijndael algorithm. The Rijndael algorithm uses a round function that is composed of four different byte-oriented transformations. The modification that has been done on the Rijndael algorithm makes the new approach uses a round function that is composed of five different byte-oriented transformations (using four original transformation with added one extra transformation). The new transformation function is called ShiftColumns() which shift the columns of the State by different offset values. Thus, the new overall byte-oriented transformation functions will be as follows:

1. Byte substitution using a substitution table (S-box).
2. Shifting rows of the State array by different offsets.
3. Mixing the data within each column of the State.
4. Shifting columns of the State by different offsets.
5. Adding a Round Key to the State.

Encryption

As shown in the Figure 2, all ***Nr*** rounds are identical with the exception of the final round, which does not include the MixColumns() and ShiftColumns() transformations.

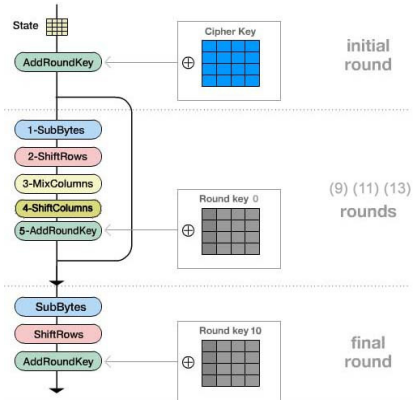

Figure 2: The cipher

In the ShiftColumns() transformation, the bytes the State are shifted over different numbers of bytes (offsets). The last two columns of the state, c_2 and c_3, are shifted to be the first two columns of the state (c_0, c_1) and the first column is shifted to be the last column of the state (c_3), then the second column is shifted to be the third column of the state (c_2). After the columns are shifted, the last two columns of the state (c_2, c_3) are inversed in backwards order. Figure 3 illustrates the ShiftColumns() transformation.

Decryption

The Cipher transformations of the new approach can be inverted and then implemented in reverse order to produce a straightforward Inverse Cipher for the new approach algorithm. The individual transformations used in the Inverse Cipher - InvShiftRows(), InvSubBytes(),InvMixColumns(), and AddRoundKey() to process the State. Figure 4 shown the inverse cipher.

To inverse the ShiftColumns() Transformation, The first two columns of the state, c_0 and c_1, are shifted to be the last two columns of the state (c_2, c_3) and the last column is shifted to be the first column of the state (c_0), then the third column is shifted to be the second column of the state (c_1). After that the columns are shifted, the first two columns of the state (c_0, c_1) are inversed in backwards order. Figure 5 illustrates the InvShiftColumns() transformation.

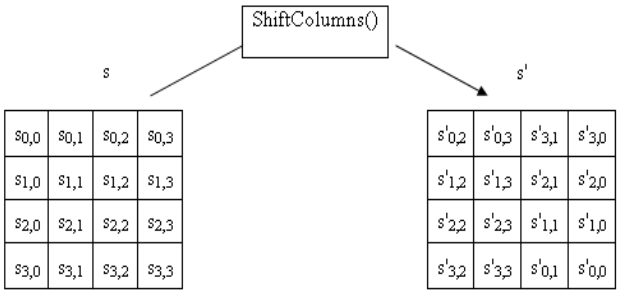

Figure 3: ShiftColumns() transformation

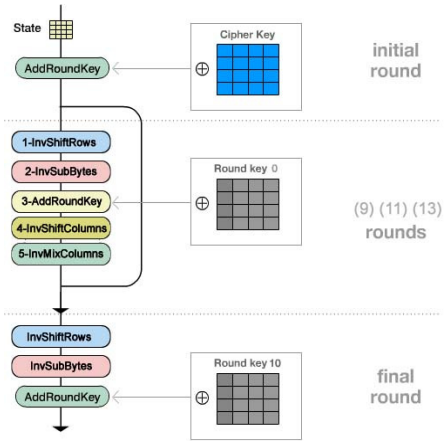

Figure 4: The inverse cipher

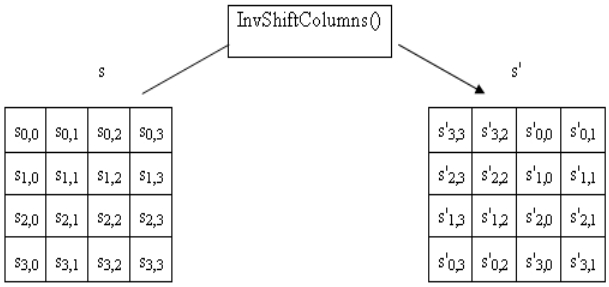

Figure 5: InvShiftColumns() transformation

SIMULATION RESULTS

The test data consists of 60 samples (files) were applied to both the new approach and the traditional Rijndael. The experimental samples were encrypted with key length 128,192 and 256. The NIST statistical tests are applied on the ciphertext files for the two cases.For each statistical test, a set of p-values (corresponding to the set of files) is produced for Rijndael algorithm and new proposed approach. For a fixed significance level, a certain percentage of p-value is expected to indicate failure. For example, if the significance level is chosen to be 0.01 then A file pass a statistical test whenever the p-value >= 0.01 and fails otherwise [3].

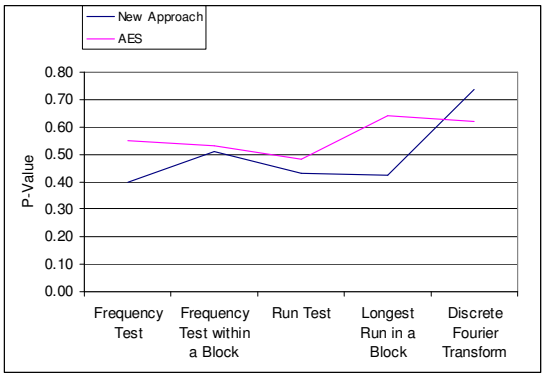

Figure 6: The results of applying statistical tests on simple plaintext files with 128-bit cipher key.

From Figure 6 found that both of the two algorithms pass the applicable statistical tests on the simple plaintext files and these results accepted but found that Rijndael algorithm in most files has p-values greater then new approach p-values in this case.

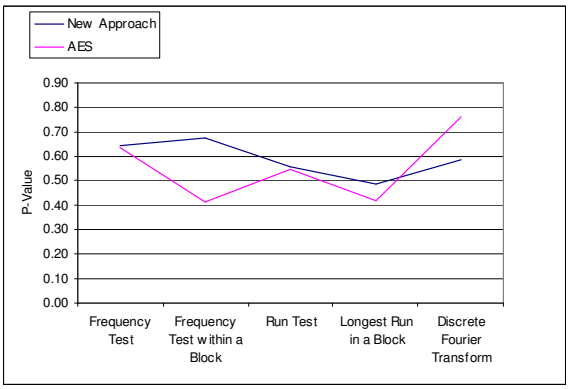

Figure 7: The results of applying statistical tests on simple plaintext files with 192-bit cipher key.

From Figure 7 found that Rijndael algorithm and the new approach passes the statistical tests, but in this case the p-value result of the new approach higher than the Rijndael algorithm in 4 tests which mean that the new approach outputs more random than Rijndael algorithm outputs.

From Figure 8 found that Rijndael algorithm and the new approach passes the statistical tests, also in this case the p-value result of the new approach higher than the Rijndael algorithm in 3 tests, therefore the new approach is consider more random outputs than the Rijndael.

From Figure 9 found that Rijndael algorithm and the new approach passes all the sixteen statistical tests, but in this case the p-value result of the new approach higher than the Rijndael algorithm in nine tests out of sixteen, therefore the new approach is consider more random outputs than the Rijndael.

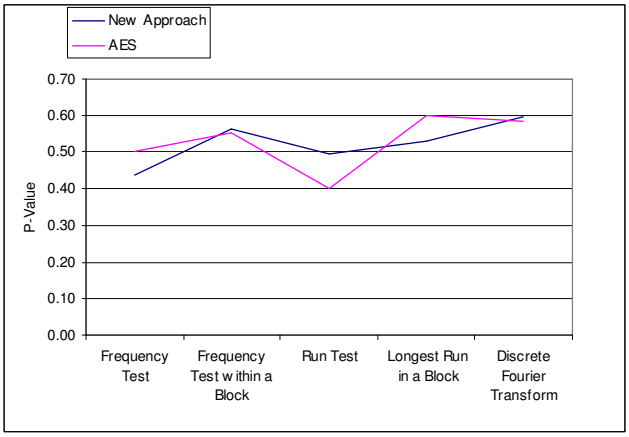

Figure 8: The results of applying statistical tests on simple plaintext files with 256-bit cipher key.

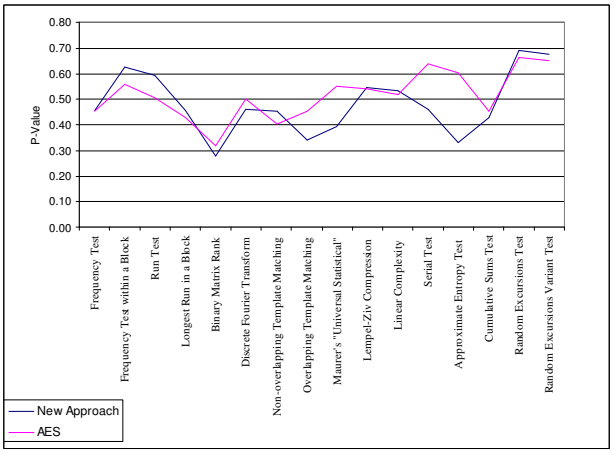

Figure 9: The results of applying statistical tests on high density plaintext files with 128-bit cipher key.

From Figure 10 found that Rijndael algorithm and the new approach passes all the sixteen statistical tests, but in this case the p-value result of the new approach higher than the Rijndael algorithm in ten tests out of sixteen, therefore the new approach is consider more random outputs than the Rijndael.

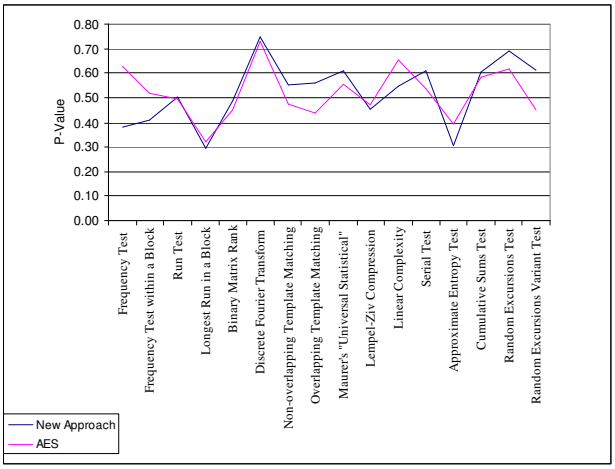

Figure 10: The results of applying statistical tests on high density plaintext files with 192-bit cipher key.

From Figure 11 found that Rijndael algorithm and the new approach passes all the sixteen statistical tests, but in this case the p-value result of the new approach higher than the Rijndael algorithm in twelve tests out of sixteen, therefore the new approach is consider more random outputs than the Rijndael.

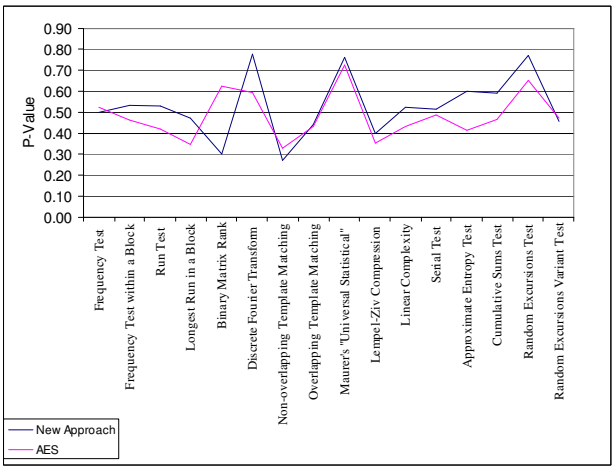

Figure 11: The results of applying statistical tests on high density plaintext files with 256-bit cipher key.

CONCLUSION

Both Rijndael and the new approach passed all 16 statistical tests for all test files (simple and long plain texts) using 128, 192, and 256 bits cipher keys. It was found that for each key length 128,192 and 256 bits, applied to a long plaintext, the new approach has shown better randomness as compared with Rijndael. This is indicated by the p-values produced by the new approach are consistently higher than p-values of the Rijndael algorithm.

The result for the 128-bits simple plaintext with 192 and 256-bits cipher key also show that the new approach has better randomness where the new approach passed all the applicable statistical tests, with its p-values higher than p-values of the Rijndael algorithm. Only for 128-bit simple plaintext with 128-bits cipher key sizes revealed that randomness sequence of the Rijndael output has more randomness than the new approach. Overall performance showed that the new approach is superior in randomness thus the security, than Rijndael for long texts with all 128, 192, and 256 bits cipher keys.

REFERENCES

Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson, Mark Vangel, David Banks, Alan Heckert, James Dray, San Vo. A Statistical Test Suite for Random and Pseudo Random Number Generators for Cryptographic Applications NIST, Special Publication 800-22 (with revisions dated May 15, 2001).

Daemen J and Rijmen V. 1999. AES Proposal: Rijndael, AES algorithm submission, September 3, 1999, available at AES home page: http://www.nise.gov/aes.

Ferguson N, Kelsey J, Lucks S , Schneier B, Stay M, Wagner D and Whiting D. 2001. Improved Cryptanalysis of Rijndael, *Fast Software Encryption: 7th International Workshop, FSE 2000*, New York, NY, USA, April 2000. Proceedings, editors: B. Schneier (Ed.) volume 1978 / 2001 of *Lecture Notes in Computer Science*, pages 213-230. Springer-Verlag.

Hee J, Kim M, Kim K, Lee J and Kang S. 2001. Improved Impossible Differential Cryptanalysis of Rijndael and Crypton, *Information Security and Cryptology - ICISC 2001: 4th International Conference* Seoul, Korea, December 6-7, 2001. Proceedings, editors: K. Kim (Ed.), volume

2288 / 2002 *of Lecture Notes in Computer Science*, pp. 39-49. Springer-Verlag.

Jamil, T. 2004. The Rijndael algorithm Potentials, *IEEE*, **23**(2): 36 – 38.

Kim J, Hong S, Sung J, Lee S, Jongin Lim J and Sung S. 2003. Impossible Differential Cryptanalysis for Block Cipher Structures, *INDOCRYPT 2003, LNCS 2904, Proceedings*, editors: T. Johansson and S. Maitra(Eds.), *Lecture Notes in Computer Science*, pp. 82-96. Springer-Verlag Berlin Heidelberg.

Nechvatal J, Barker E, Bassham L, Burr W, Dworkin M, Foti J and Roback E. 2000. Report on the Development of the Advanced Encryption Standard (AES), National Institute of Standards and Technology, October 2, 2000, available at AES discussion forum: http://aes.nist.gov/aes.

William e.burr. 2003. *Selecting the Advanced Encryption Standard*, National Institute of Standards and Technology, IEEE Security & Privacy.