International Journal of Cryptology Research 1(2): 215-223 (2009)

A Faster Version of Rijndael Cryptographic Algorithm Using Cyclic Shift and Bit Wise Operations

Fakariah Hani Mohd Ali, Ramlan Mahmod,
Mohammad Rushdan and Ismail Abdullah
Faculty of Computer Science and Information Technology and
Institute for Mathematical Research,
Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
E-mail: hani_fakariah@yahoo.com, ramlan@fsktm.upm.edu.my,
mrushdan,isbah@fsas.upm.edu.my

ABSTRACT

Doing arithmetic in finite field is the key part to the implementation of communication and coding system including the newly developed Rijndael the Advanced Encryption Standard (AES). This encryption standard uses KeyExpansion, ByteSub, Mixcolumn and Shiftrow functions which consists of XOR, inverse, multiplying and swap modules. Among them, inverse and multiplier are the most complex modules with longer delay. These modules are included in the Mixcolumn function. From the proposal of AES, the Mixcolumn function was suggested to solve the problem of delay by using look-up tables. This function can be integrated into a bigger table to replace the calculations of inverse and multiply operations, if it provides enough memory. In fact, too many tables are needed for various irreducible polynomials that this system is not flexible and expandable. The area for lookup tables becomes huge when multiple round units are implemented. This research proposes the use of cyclic shift and bit wise XOR operation as new approach to replace the lookup table. The principle benefit of using this new approach over the transform from Rijndael block cipher is speed. This new approach has shown the excellent result, which faster then Rijndael. The new approach algorithm speed increment has consistently increased in between 18% to 22% microsecond for encryption and 30% to 34% for decryption compared to Rijndael algorithm depending upon the key length.

Index Terms – AES, speed, cyclic shift, exclusive OR, mixcolumn transformation, table lookup

INTRODUCTION

Rijndael is a block cipher designed by Joan Daemen and Vincent Rijmen. According to Nechvatal *et al.*, (2000), Rijndael's combination of security, performance, efficiency, implementability and flexibility makes it an appropriate selection for the AES for use in the technology of today and in the future. There are many further analyses and improvements have been done on Rijndael (Daemen *et al.,*1999) (Federal Information Processing Standards Publication (FIPS) 197, 2001). McLoone *et al.,* (2001) proposed a Field Programmable Gate Arrays (FPGAs) Rijndael encryption design which utilizes look-up tables to implement the entire Rijndael Round function. Jing *et al.* (2001) proposed a new algorithm for computing inverse in GF (2^m) on the standard basis. They proposed a set of multiplier and inverse in GF (2^m) to increase the computing speed. Sklavos *et al.* (2002) proposed an alternative architectures and VLSI implementation designs. These designs operate for both encryption and decryption process in the same device. Xinmiao *et al.* (2002) addresses various approaches for efficient hardware implementation of the AES algorithm.

In the computational operation in AES, several steps have to use inverse and multiplication functions. These modules are included in the Mixcolumn function. Inverse and multiplier are the most complex modules with longer delay. Mixcolumn transformation is suggested to use look-up tables to solve the problem of delay. That function can be integrated into a bigger table to replace the calculations of inverse and multiply operations, if it provides enough memory. In fact, too many tables are needed for various irreducible polynomials so that this system is not flexible and expandable (Jing *et al.*, 2001). According to Xinmiao *et al.* (2002), the area for lookup tables becomes huge when multiple round units are implemented. In this research we proposed the use of cyclic shift and XOR operation to replace the use of look-up tables. This approach is used to speed up Mixcolumn transformation.

The remainder of this paper is organized as follows: Section 2 reviews Rijndael Mixcolumn transformation. Section 3 introduces the new approach formula where cyclic shift and bit wise XOR operation had been used to replace the lookup table operation and the project methodology. Section 4 compares the results which using current and new approach algorithm

RIJNDAEL MIXCOLUMN TRANSFORMATION

The MixColumns() transformation operates on the State column-by-column, treating each column as a four-term polynomial . The columns are considered as polynomials over GF (2^8) and multiplied modulo x^4 + 1 with a fixed polynomial a (x), given by

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}.$$

This can be written as a matrix multiplication. Let

$$s'(x) = a(x) \otimes s(x):$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{o,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for} \quad 0 \le c < Nb.$$

InvMixcolumns () transformation

InvMixColumns() is the inverse of the MixColumns() transformation. InvMixColumns() operates on the State column-by-column, treating each column as a four term polynomial. The columns are considered as polynomials over GF (2^8) and multiplied modulo x^4 + 1 with a fixed polynomial $a^{-1}(x)$, given by

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}.$$

This can be written as a matrix multiplication. Let

$$s'(x) = a^{-1}(x) \otimes s(x):$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{o,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for} \quad 0 \le c < Nb.$$

PROJECT METHODOLOGY

The coefficients of Mixcolumn transformation are limited to '01','02' and '03' and '0E','0D','0B' and '09' for the inverse, which this multiplication implemented using lookup table. In this research, the cyclic shift and bit wise XOR operation have been used as new approach to replace the lookup table operation.

Cyclic Shift and Bit wise Exclusive OR (XOR)

Let x be a block of n bits. A cyclic shift to the left by m bits is performed by taking the first m bits from the left side of the block and attaching them to the right side. Accordingly, a cyclic shift to the right by m bits is performed by taking the first m bits from the right side and attaching them to the left side.

Table 1 shows the multiplication of '03' and it is followed by Table 2 which shows that by using cyclic shift operation, then XOR by the number itself, it will produce the same result as Table 1.

TABLE 1: Finite Field multiplication of '03' in binary and hexadecimal

D	H	Binary (A)	'03' (C)	Y=A x C	Hex
1	1	00000001	11	00000011	03
2	2	00000010	11	00000110	06
3	3	00000011	11	00000101	05
4	4	00000100	11	00001100	0C
5	5	00000101	11	00001111	0F
6	6	00000110	11	00001010	0A
7	7	00000111	11	00001001	09
8	8	00001000	11	00011000	18
9	9	00001001	11	00011011	1B
10	A	00001010	11	00011110	1E
11	B	00001011	11	00011101	1D
12	C	00001100	11	00010100	14
13	D	00001101	11	00010111	17
14	E	00001110	11	00010010	12
15	F	00001111	11	00010001	11

TABLE 2: New approach for '03'

D	H	Binary (A)	W=A <<1	D	Z=W⊕ A	H
1	1	00000001	00000010	2	00000011	03
2	2	00000010	00000100	4	00000110	06
3	3	00000011	00000110	6	00000101	05
4	4	00000100	00001000	8	00001100	0C
5	5	00000101	00001010	10	00001111	0F
6	6	00000110	00001100	12	00001010	0A
7	7	00000111	00001110	14	00001001	09
8	8	00001000	00010000	16	00011000	18
9	9	00001001	00010010	18	00011011	1B
10	A	00001010	00010100	20	00011110	1E
11	B	00001011	00010110	22	00011101	1D
12	C	00001100	00011000	24	00010100	14
13	D	00001101	00011010	26	00010111	17
14	E	00001110	00011100	28	00010010	12
15	F	00001111	00011110	30	00010001	11

So that for '0E','0D','0B' and '09' finite field multiplication:

'0E' = (((A<<1) XOR A) <<1) XOR A) << 1
'0D' = ((A<<1) XOR A) <<2) XOR A
'0B' = (A<<1) XOR A) XOR (A<<3)
'09' = ((A << 3) XOR A)

It was found that in new approach technique, GF (2^8) multiplications, which is the best method that can be implemented by table look-up has been replaced by cyclic shift and XOR operation.

RESULTS COMPARISON

The following tables will show the ciphering results. The results are divided into two types of plaintext; simple and long plaintext. However in this paper, certain results only have been shown. The following figures show the result in graphical forms based on key and block length.

Half paragraph of plaintext

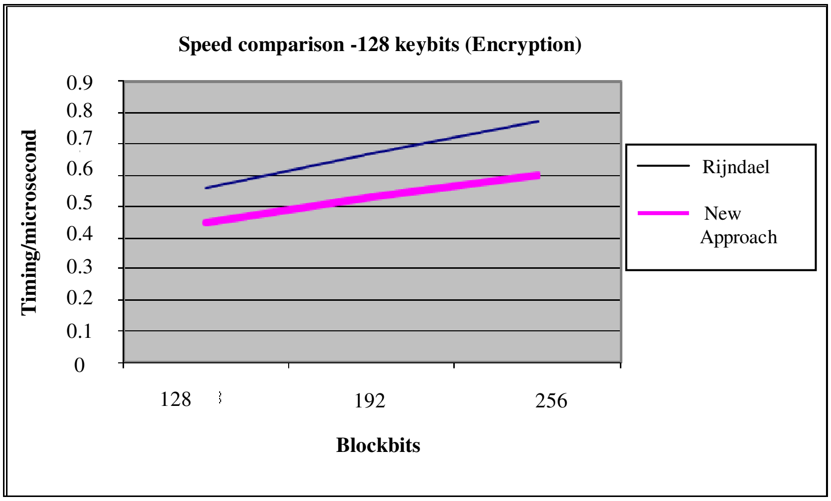

Figure 1: Half Paragraph of plaintext using 128 bit key length and 128,192 and 256 bits block length during encryption

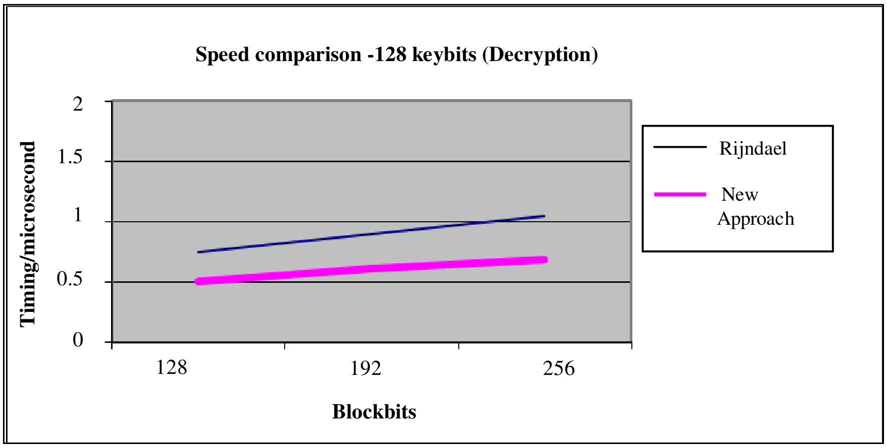

Figure 2: Half Paragraph of plaintext using 128 bit key length and 128,192 and 256 bits block length during decryption

One page of plaintexts

Following figure shows the results of one page plaintext as input during the encryption process.

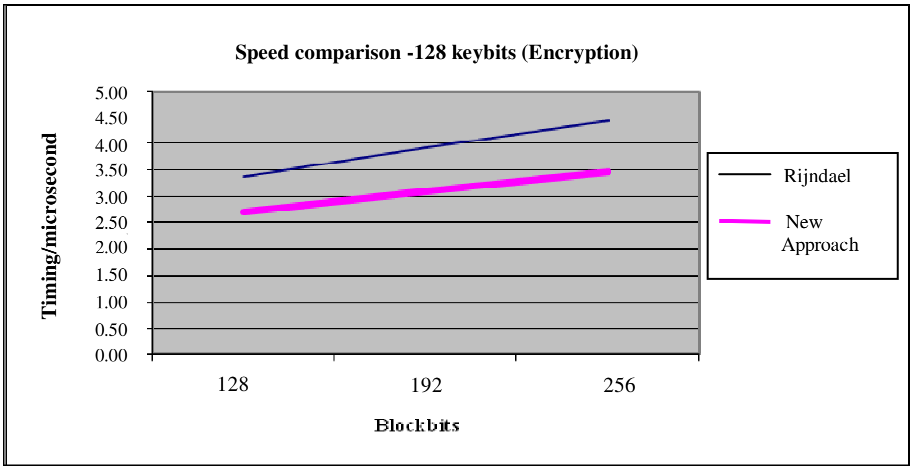

Figure 3: One Paragraph of plaintext using 128 bit key length and 128,192 and 256 bits block length during encryption

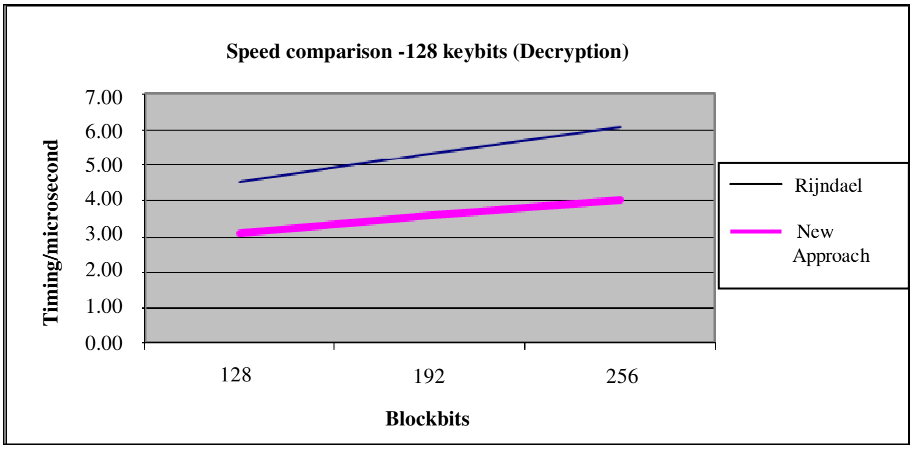

Figure 4: One Paragraph of plaintext using 128 bit key length and 128,192 and 256 bits block length during decryption

CONCLUSION

As a conclusion, this research work has achieved its objective. According to result, it was found that this new approach has shown the excellent results. It is faster than Rijndael .The new approach algorithm speed increment has consistency increased in between 18% to 22% microsecond for encryption and 30% to 34 % for decryption compared to Rijndael algorithm.

REFERENCES

AES Development Effort available at http://csrc.nist.gov/CryptoToolkit /aes/. Accessed on April 2001

Daemen J. and Rijmen V. 1999. AES Proposal:Rijndael,AES Algorithm Submission, available at AES home page: http://www.nist.gov/aes. Accessed on Sept 3, 1999.

Eskicioglu.A, Litwin L. 2001.Cryptography, Potentials,*IEEE*, 20 (1): 36-38.

Federal Information Processing Standards Publication (FIPS PUB) 46,"Data Encryption Standards," *National Bureau Of Standards,* Jan 1977 available at http://www.itl.nist.gov/fipspubs/fip46-2.htm

Federal Information Processing Standards Publication (FIPS) 1977,"Advanced Encryption

Standard (AES)", November 26, 2001 available at http://csrc.nist.gov/CryptoToolkit/aes/

Ferguson N., Kelsey J., Lucks S., Schneier B., Stay M., Wagner D., and Whiting D. 2000. Improved Cryptanalysis of Rijndael *Seventh Fast Software Encryption Workshop, Springer-Verlag,* 2000 available at**:** http://www.schneier.com/paper-el.html.

Gladman B. 2002. A Specification for the AES algorithm, v3.3, 1st May 2002, pp.1-25 At http://fp.gladman.plus.com/cryptography _technology/ rijndael/spec.v311.pdf .

Juan C.A.,Thales e-security. 2002. The Advanced Encryption Standard-Implementation and Transition to a New Cryptographic Benchmark *Network Security*, Issue 7, 1 July 2002 pp. 7-9.

Jing M.H, Chen Y.H, Chang Y.T, and Hsu C.H. 2001. The Design of A Fast Inverse Module in AES, *Info-Tech and Info-Net, 2001.Proceedings.ICII 2001 – Beijing 2001 International Conference*, **3**: 298-303.

McLoone W., McCanny J.V. 2001. Rijndael FPGA Implementation Utilizing Look-Up Tables, *Signal Processing Systems, 2001 IEEE Workshop,* 26-28 September 2001, pp. 349-360.

Marie A.Wright. 2001. The Advanced Encryption Standard, *Network Security*, Issue 10, October 31,2001, pp 11-13.

Marie A.Wright.1999. The Evolution of the Advanced Encryption Standard, *Network Security*, Issue 11, November 1999, pp 11-14.

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), available at AES home page: http://www.nist.gov/aes. Accessed on October 2, pp 120-126.

Rejeb J., Ramaswamy V.2003. Efficient Rijndael implementation for high-speed optical networks *Telecommunications, 2003. ICT 2003. 10th International Conference*, 1: 641 – 645.

Smart D. 2003. *Cryptography: an introduction*, McGraw Hill, London.

Sklavos N., Koufopavlou O.2002. Architectures and VSLI implementations of the AES-Proposal Rijndael, Computers, *IEEE Transactions*, 51(12): 1454-1459.

Sanchez A.C., Sanchez R. R. 2001.The Rijndael Block Cipher (AES Proposal): A Comparison with DES, *Security Technology, 2001 IEEE 35th International Carnahan Conference,* pp.229-234

Xinmiao Z., Parhi K.K. 2002. Implementation Approaches for the Advanced Encryption Standard Algorithm, *Circuits and System Magazine, IEEE*, **2**(4): 24-46.