Volume 3, No. 4, April 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

ENHANCEMENT OF RC6 (RC6_EN) BLOCK CIPHER ALGORITHM AND COMPARISON WITH RC5 & RC6

Vikas Tyagi[1], Shrinivas Singh[2]
[1]Student,Computer Science , Subharti University Meerut (India)
itengg.vikas@gmail.com
[2]Computer Science , Subharti University Meerut (India)
shri0123@gmail.com

Abstract: In this paper, we present an enhanced version of RC6 Block Cipher Algorithm (RC6_En – RC6 enhanced version), which is a symmetric encryption algorithm [1] designed for 256-bit plain text block. RC6 uses four (w-bit) registers for storing plain text and for data-dependent rotations [2, 3], but this enhanced version (RC6_En) uses eight (w-bit) register that helps to increase the performance as well as improve security. Its salient feature includes two-variable algebraic expression modulo 2^w and 2 Box-Type operations, Box-Type I & Box-Type II. Each Box-Type operation uses two (w-bit) registers. Box-Type I works much like two registers (A & B or C & D) operation in RC6 but in Box-Type II bitwise exclusive-or is swapped by integer addition modulo 2^w used in Box-Type I and vice-versa, it improves Diffusion in each round. This enhanced version needs 2r+4 additive round-keys and uses every round-key twice for encrypting the file. This enhanced version performs better with respect to RC5[4, 5] and RC6[2, 3] when file size is larger.

Keywords: Cryptography, Data Security, Block cipher, Symmetric encryption.

INTRODUCTION

In cryptography, the use of the symmetric key encryption is common to ensure data integrity. Symmetric key encryption code can be divided into the block cipher and stream one [1][6]. RC6 is a symmetric key block cipher derived from RC5. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney and Yiqun Lisa Yin to meet the requirements of the Advanced Encryption Standard (AES) [7] competition. The algorithm was one of the five finalists and was also submitted to the NESSIE [8] and CRYPTREC [9] projects. It is a proprietary algorithm, patented by RSA Security [10].

Details of RC6:

Like RC5, RC6 is a fully parameterized family of encryption algorithms. A version of RC6 is more accurately specified as RC6-w/r/b where the word size is w bits, encryption consists of a nonnegative number of rounds r and b denotes the length of the encryption key in bytes. Since the AES submission is targeted at w = 32 and r = 20, we shall use RC6 as shorthand to refer to such versions. When any other value of w or r is intended in the text, the parameter values will be specified as RC6-w/r. Of particular relevance to the AES effort will be the versions of RC6 with 16-, 24- and 32-byte keys. For all variants, RC6-w/r/b operates on units of four w-bit words using the following basic operations [2].

The operations used in RC6 are defined as followings.

- ✓ A+B integer addition modulo 2^w
- ✓ A-B integer subtraction modulo 2^w
- ✓ A⊕B bitwise exclusive-or of w-bit words
- ✓ A*B integer multiplication modulo 2^w
- ✓ $A^{<<<}B$ rotation of the w-bit word A to the left by the amount given by the least significant lg w bits of B
- ✓ $A^{>>>}B$ rotation of the w-bit word A to the right by the amount given by the least significant lg w bits of B
- ✓ $f(x) = x(2x+1) \bmod 2^w$

RC6 is very similar to RC5 in structure, using data-dependent rotations [3], addition modulo 2^w and XOR operations; in fact, RC6 could be viewed as interweaving two parallel RC5 encryption processes. However, RC6 does use an extra multiplication operation not present in RC5 in order to make the rotation dependent on every bit in a word and not just the least significant few bits. The base-two logarithm of w will be denoted by lg w [10].

Encryption and Decryption:

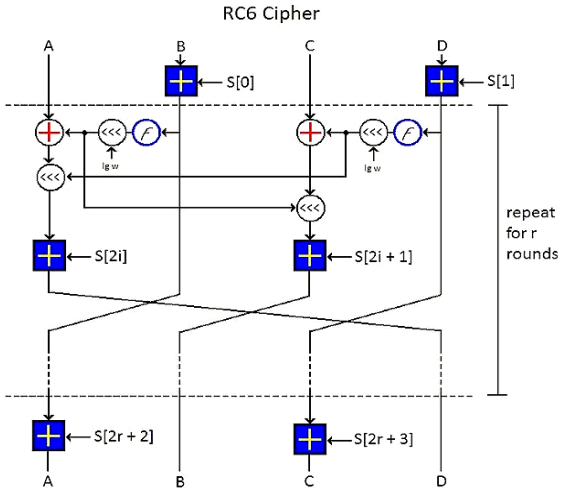

Figure 1: RC6 Block Cipher

RC6 works with four w-bit registers A, B, C, D which contain the initial input plain-text as well as the output cipher-text at the end of encryption. The first byte of plaintext or cipher-text is placed in the least-significant byte of A, the last byte of plaintext or cipher-text is placed into the most-significant byte of D [2]. Pseudo code of encryption and decryption is given below; at first we load

plain text in to registers A, B, C, D and then apply these operations to encrypt the plain text.

Encryption with RC6-w/r/b [6]
Input: Plain-text stored in four w-bit input registers A, B, C, D. r denotes the no of rounds and 2r+4 w-bit round keys S[0, 1, ..., 2r + 3]
Output: Cipher-text will be store in A, B, C, D
Procedure:
B = B + S[0]
D = D + S[1]
for i = 1 to r do
{
t = (B * (2B + 1)) <<< lg w
u = (D * (2D + 1)) <<< lg w
A = ((A ⊕ t) <<< u) + S[2i]
C = ((C ⊕ u) <<< t) + S[2i+ 1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[2r + 2]
C = C + S[2r + 3]

Operation (A, B, C, D) = (B, C, D, A) means the parallel assignment of values on the right to registers on the left. Round key generation algorithm is described below in the section of proposed system. After applying these operations on registers A, B, C, D plain-text get converted into the cipher-text and we store it in any file that is called encrypted file. Now for decryption of cipher-text load these cipher text into registers A, B, C, D and then apply these operations to convert cipher-text into plain-text.

Decryption with RC6-w/r/b [6]
Input: Cipher-text stored in four w-bit input registers A, B, C, D. r denotes the no of rounds and 2r+4 w-bit round keys S[0, 1, ..., 2r + 3]
Output: Plain-text will be store in A, B, C, D
Procedure:
C = C - S[2r + 3]
A = A - S[2r + 2]
for i = r down to 1 do
{
(A, B, C, D) = (D, A, B, C)
u = (D * (2D + 1)) <<< lg w
t = (B * (2B + 1)) <<< lg w
C = ((C - S[2i + 1]) >>> t) ⊕ u
A = ((A - S[2i]) >>> u) ⊕ t
}
D = D - S[1]
B = B - S[0]

This algorithm uses integer subtraction modulo 2^w and right rotation on registers for getting plain text; it does reverse operations on registers.

PROPOSED SYSTEM

This is a computer era and we need new technologies each and every day that produce better results and consume less time and also provide high level of security. So taking these changing demands in the consideration we propose RC6_En (an enhanced version of RC6) block cipher symmetric encryption algorithm, that helps to increase the performance of older version as well as it provide better security on confidentiality of plain text.

Features of Proposed system:

Like RC5 and RC6, this enhanced version (RC6_En) is a symmetric key block cipher that uses data-dependent rotations, modular addition and exclusive-or operations. RC6_En works with 256-bit block size, (128,192 or 256)-bit key size and 20 rounds. RC6_En works with eight-word input (plain text) block size and eight-word (cipher text) output block size. This means that it is word-oriented algorithm.

Algorithm:

This enhanced version (RC6_En) is more accurately specified as RC6_En-w/r/b in this section. Where-

w: the word size in bits,
r: non-negative number of rounds r and
b : the length of the encryption key in bytes.

The operations used in RC6_En are defined as followings.

- ✓ A+B integer addition modulo 2^w
- ✓ A-B integer subtraction modulo 2^w
- ✓ A⊕B bitwise exclusive-or of w-bit words
- ✓ A*B integer multiplication modulo 2^w
- ✓ $A^{<<<}B$ rotation of the w-bit word A to the left by the amount given by the least significant lg w bits of B
- ✓ $A^{>>>}B$ rotation of the w-bit word A to the right by the amount given by the least significant lg w bits of B
- ✓ $f(A, B) = (A^2 + B^2 – AB – 7) \bmod 2^w$, two-variable algebraic expression.
- ✓ (A, B, C, D, E, F, G, H) = (B, C, D, E, F, G, H, A) parallel assignment

Working of RC6_En is illustrated in Fig 2, in this fig there are two types of box are used Box-Type I & Box-Type II both works parallel. Box-Type II swaps the integer addition modulo 2^w & bitwise exclusive-or as in Box-Type I, it increases the diffusion of each round. RC6_En also uses two-variable algebraic expression $f(x, y) = (x^2 + y^2 – xy – 7) \bmod 2^w$ that takes 2 input (illustrated by green line in fig 1) and then the output of this function are used in bitwise exclusive-or & integer addition modulo 2^w with other registers in Box-Type I & Box-Type II preceding with left rotation. This leads to security in each round. RC6_En works with only 2r+4 additive round-keys that is same as RC6 and uses every round-key twice, once in Box-Type I & once in Box-Type II. Using same key in both Box-Types does not compromise with security but it reduces the effort for generating round-keys.

RC6_En have mainly 3 algorithms, Key-Expansion, Encryption and Decryption. Encryption & Decryption algorithms uses the basic operations described above in this section. In encryption algorithm plain-text will be stored into 8 registers (A to H) and then follow the steps of encryption algorithm (described below) to get the cipher text in these registers. In decryption algorithm we will store cipher-text into those registers and then follow the steps of decryption algorithm (described below) to get plain text.

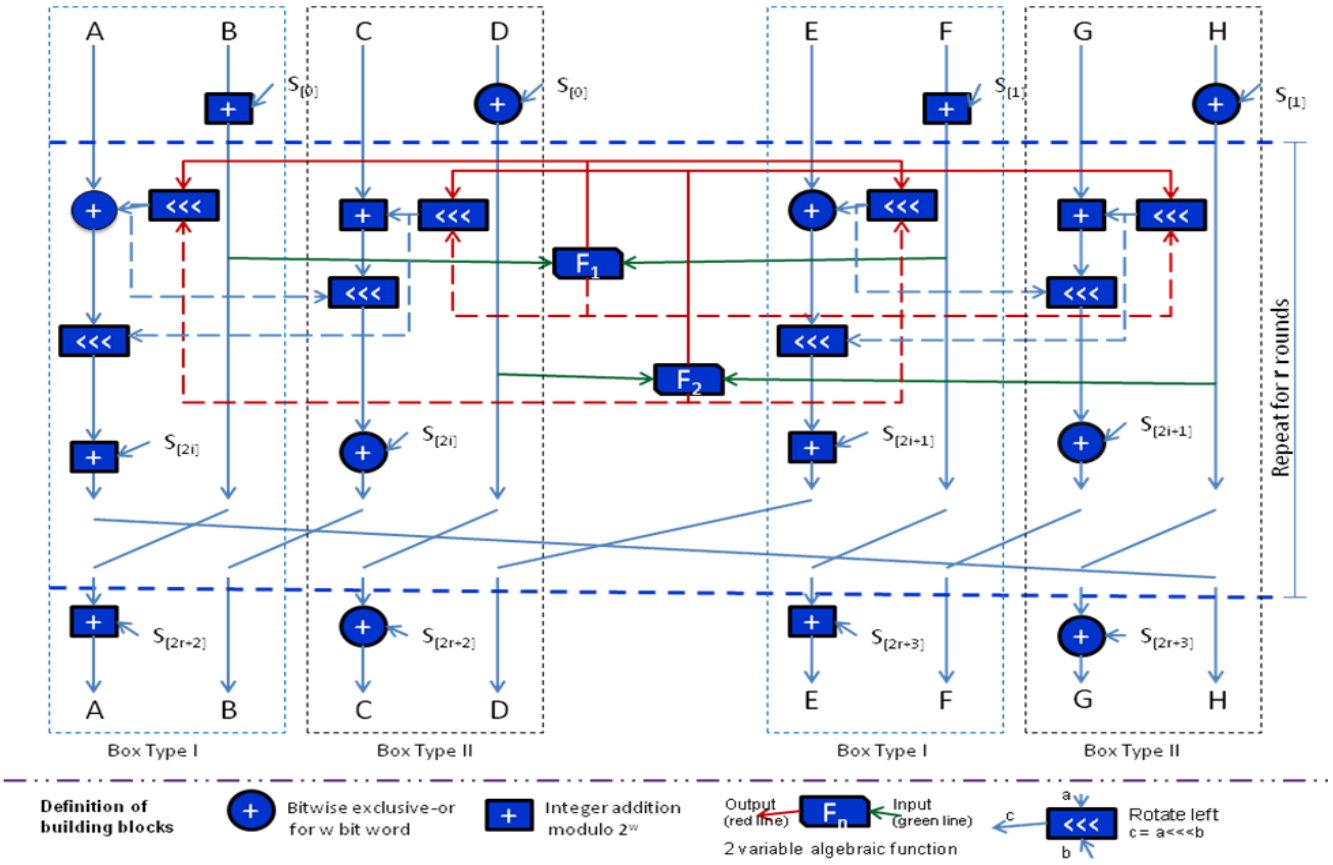

Figure 2: RC6_En Block Cipher

Key-Expansion Algorithm:

The key schedule for RC6_En-w/r/b is same as key schedule of RC6. The user supplies a key of b bytes, copy the secret key K[0...b-1] into an array L[0..c-1] of c = ceil(b/u), where u = w/8 in little-endian order. In other words, we fill up L using u consecutive key bytes of K. Any unfilled byte positions in L are zeroed. In the case that b = c = 0, set c = 1 and L[0] = 0. The number of w-bit words that will be generated for additive round keys is (2r + 4) and these are stored in the array S[0, ..., 2r + 3].

Magic Constants P_w and Q_w are defined for arbitrary w as follows:

P_w = Odd ((e -1) 2^w) (1)

Q_w = Odd ((v - 1) 2^w) (2)

Where e is the base of natural logarithms (e = 2.718281828459) and v is the golden ratio (v =1.618033988749)

Odd (x) is the odd integer nearest to x [11].

Tables 1 show these magic constants in hexadecimal using several values of w. Which are calculated by above expressions (1) & (2).

Table 1. Magic Constants values P_w and Q_w [11]

W	16	32	64
P_w	B7E1	B7E15163	B7E151628AED2A6B
Q_w	9E37	9E3779B9	9E3779B97F4A7C15

Magic Constraints play a vital role in the generation of round keys. Key-Expansion algorithm can be used to generate any number of round keys.

Input: b byte key that is preloaded into c word array L[0,1,…, c-1], r denotes the no of rounds.
Output: 2r+4 w-bit round keys S[0,1, …, 2r + 2,2r+3].
Procedure:
S[0] = P_w,
For i= 1 to 2r+3 do
{
S[i] = S[i – 1] + Q_w
}
X = Y = a = b = 0
Iteration =3 * max(c, 2r+4)
For i = 1 to Iteration do
{
X = S[a] = (S[a] + X + Y) <<< 3
Y =L[b] = (L[b] +X + Y) <<< (X + Y)
i = (a + 1) mod (2r + 4)
j = (b + 1) mod c
}

Encryption Algorithm:

Input: Plain-text stored in eight w-bit registers A, B, C, D, E, F, G & H. r denotes the no of rounds and 2r+4 w-bit round keys stored in S[0,1, …, 2r + 2,2r+3].
Output: Cipher-text will be store in these eight w-bit registers A, B, C, D, E, F, G & H.
Procedure :
B = B + S[0]

D = D ⊕ S[0]
F = F + S[1]
H = H ⊕ S[1]
For i=1 to r do
{
F1 = $B^2 + F^2$ – BF - 7
F2 = $D^2 + H^2$ – DH - 7
R1 = F1 <<< F2
R2 = F2 <<< F1
A = ((A ⊕ R1) <<< R2) + S[2i]
C = ((C + R2) <<< R1) ⊕ S[2i]
E = ((E ⊕ R1) <<< R2) + S[2i + 1]
G = ((G + R2) <<< R1) ⊕ S[2i + 1]
(A, B, C, D, E, F, G, H) = (B, C, D, E, F, G, H, A)
}
A = A + S[2r +2]
C = C ⊕ S[2r +2]
E = E + S[2r + 3]
G = G ⊕ S[2r + 3]

Decryption Algorithm:

Input: Cipher-text stored in eight w-bit registers A, B, C, D, E, F, G & H. r denotes the no of rounds and 2r+4 w-bit round keys stored in S[0,1, …, 2r + 2,2r+3].
Output: Plain-text will be store in these eight w-bit registers A, B, C, D, E, F, G & H.
Procedure:
C = C ⊕ S[2r +2]
A = A - S[2r +2]
G = G ⊕ S[2r + 3]
E = E - S[2r + 3]
For i = r down to 1 do
{
(H, G, F, E, D, C, B, A) = (G, F, E, D, C, B, A, H)
F1 = $B^2 + F^2$ – BF - 7
F2 = $D^2 + H^2$ – DH - 7
R1 = F1 <<< F2
R2 = F2 <<< F1
A = ((A - S[2i]) >>> R2) ⊕ R1
C = ((C ⊕ S[2i]) >>> R1) - R2
E = ((E - S[2i + 1])>>> R2) ⊕ R1
G = ((G ⊕ S[2i + 1]) >>> R1) – R2
}
D = D ⊕ S[0]
B = B - S[0]
H = H ⊕ S[1]
F = F - S[1]

PARAMETRIC COMPARISO

Table 2 summarizes the comparison between RC5, RC6 and the RC6_En for different design parameters such as word size, block size, number of rounds and secret key size [11][12].

Table 2. Comparison on the basis of parameters

Parameters	Algorithm Type		
	RC5	RC6	RC6_En
b (key length in bytes)	0 - 255 (standard 16)	0 - 255 (standard 16)	0 - 255 (standard 16)
r (no of rounds)	0 - 255 (standard 12) [13]	0 - 255 (standard 20) [10]	0 - 255 (standard 20)
No of round keys	2r+2	2r+4	2r+4
Block size in words	2w	4w	8w
w (word size in bits)	16, 32, 64 (standard 32)	16, 32, 64 (standard 32)	16, 32 ,64 (standard 32)
Block size in bits	32, 64, 128 (standard 64)	64, 128, 256 (standard 128)	128, 256, 512 (standard 256)
Transformation Function	Does not exist	$F(x) = x(2x+1) \bmod 2^w$	$F(x,y) = (x^2 + y^2 – xy -7) \bmod 2^w$
Used Operation	+, -, ⊕, <<<, >>>	+, -, ⊕, *, <<<, >>>	+, -, ⊕, *, <<<, >>>

IMPLEMANTATION ISSUE

Unlike many other encryption algorithms, RC6_En block cipher does not use look-up tables during encryption. This means that the code and data can readily fit within today's on-chip cache memory and typically do so with room to spare. RC6_En block cipher make use of (2r +4) word key schedule and a bare minimum of additional memory; to compute that (2r + 4) word key schedule, the key setup process requires little more than an auxiliary array of approximately the same size as the user's supplied key. In addition, since the key schedule is only (2r+4) word, it is possible to pre-compute and store the key schedules for hundreds of keys. Then switching to one of these keys only requires switching the pointer to the relevant key schedule, thereby providing key agility [11].

The two-variable algebraic function is aimed at providing a complex diffusion thereby improving the chances that simple differentials will spoil rotation amounts much sooner than is accomplished with RC5. The transformed values of B and F are used to modify the registers A and E vice-versa the transformed values of D and H are used to modify the registers C and G. increasing the nonlinearly of the. The variable rotation by transformation function value bits plays a vital role in complicating both linear and differential cryptanalysis.

CONCLUSION

This paper introduced an enhanced version of RC6 block cipher algorithm (RC6_En). It has a new architecture and implementation, such that it encrypts and decrypts 256 bit block size of data per round.

RC6_En also achieves the requirements of the Advanced Encryption Standards (AES) and computer security system developer goals. So this enhanced version of RC6 (RC6_En) is a fast and secure block ciphering algorithm. It offers good performance and high security.

REFERENCES

[1] W. Stallings, "Cryptography and Network Security: Principles and Practice", Prentice-Hall, New Jersey, 1999.

[2] Ronald L. Rivest, M.J.B. Robshaw, R. Sidney and Y.L.Yin, The RC6 ™ Block Cipher , M.I.T. Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, Version 1.1 - August 20, 1998. Available on the site: http://people.csail.mit.edu/rivest/Rc6.pdf

[3] "RC6® Block Cipher" http://www.rsa.com/rsalabs/node.asp?id=2512

[4] Ronald L. Rivest, "RC5 Encryption Algorithm", Dr Dobbs Journal, Vol. 226, PP. 146-148, Jan 1995.

[5] Ronald L. Rivest, The RC5 Encryption Algorithm, MIT Laboratory for Computer Science 545 Technology Square, Cambridge, Mass.02139 (Revised March 20, 1997). Available on the site: httu://theory.lcs.mit.cdu/-rivest/Rivest-rc5rev.pdf

[6] Gil-Ho Kim, Jong-Nam Kim, Gyeong-Yeon Cho, "An improved RC6 algorithm with the same structure of encryption and decryption" ISBN 978-89-5519-139-4, Volume **:** 02, ICACT 2009, IEEE

[7] "Report on the Development of the Advanced Encryption Standard (AES)." http://csrc.nist.gov/encryption/aes/round2/r2report.pdf

[8] "New European Schemes for Signatures, Integrity and Encrypt" https://cosic.esat.kuleuven.be/nessie

[9] "Cryptography Research and Evaluation Committees" http://www.cryptrec.go.jp/english/about.html

[10] "RC6**"** http://en.wikipedia.org/wiki/RC6

[11] Abdul Hamid M. Ragab, Nabil A. Ismail, Senior Member IEEE and Osama S. Farag Allah, "Enhancements and Implementation of RC6™ Block Cipher for Data Security", IEEE Catalogue No. 01 CH37239-0-7803-7101-1/01 © 2001 IEEE.

[12] Asma Belhaj Mohamed, Ghada Zaibi, Abdennaceur Kachouri "IMPLEMENTATION OF RC5 AND RC6 BLOCK CIPHERS ON DIGITAL IMAGES", 978-1-4577-0411-6/11 ©2011 IEEE

[13] "RC5" http://en.wikipedia.org/wiki/RC5

[14] W. Stallings, "Network and Internetwork Security: Principles and Practice", Prentice-Hall, New Jersey, 1995.

Short Bio Data for the Author

Vikas Tyagi is M.tech Computer science student from Subharti University Meerut, and working as Assistant Professor in the IMS Engineering College Ghaziabad(U.P). He has been in teaching from more than four years. He has been member of several academic and administrative bodies. During her teaching he has coordinated several Technical fests and National Conferences at Institute and University Level. He has attended several seminars, workshops and conferences at various levels. His many papers are published in various national and international conferences. His area of research includes Network Security, Networking and Information Security.

Shrinivas Singh is Head of the department of Computer Science & Engineering in Subharti University, Meerut He has been member of several academic and administrative bodies. During her teaching he has coordinated several Technical fests and National Conferences at Institute and University Level. He has attended several seminars, workshops and conferences at various levels. His many papers are published in various national and international conferences. His area of research includes Network Security, Networking and Information Security.