A Proposed 512 bits RC6 Encryption Algorithm

Ashwaq T. Hashim* **Janan A. Mahdi *** **Salma H. Abdullah****

Received on: 20/4/2008
Accepted on: 12/11/2009

Abstract

In this paper, a new secret-key block cipher called 512 bits RC6 is proposed which is an evolutionary improvement of the 128 bits RC6 designed to meet the requirements of the Advanced Encryption Standard (AES) to increase security and improve performance. The inner loop is based around the same round found in the 128 bits RC6.

The proposed algorithm includes two things the first is doubling the pervious 128 bits RC6 to 256 bits RC6 and the second adapting a Feistal network which is iterated 256 bits RC6 20 times. A desirable property of an encryption algorithm is that a small change in either the plaintext or the key should produce a significant change in the ciphertext. This is so called valanche Effect. An Avalanche Effect of 128 bit RC6 is about 43.25%. If we change the same amount of information in key for 512 bits RC6 then the Avalanche Effect is about 257.27%. The proposed algorithm is resistant to matching and a dictionary attack which increases the security of the previous 128 bits RC6 algorithm by using a block size of 512 bits instead of 128 bits.

خوارزمية التشفير 512 bits RC6 المقترحة

الخلاصة

في هذا البحث تم اقتراح تشفير كتلي ذو المفتاح سري يعرف 512 bits RC6 وهو تحسين تطويري لخوارزمية التشفير 128-bit RC6 صمم لتحقيق متطلبات معيار التشفير المتقدم(AES) لزيادة الامنية وتحسين الاداء. الحلقة الداخلية تعتمد تقريبا لدورة مشابهة لما موجود في 128 bits RC6.الخوارزمية المقترحة تتضمن شيئين الاول هو مضاعفة 128 bits RC6 السابقة الى 256 bits RC6 والثاني هو تبني شبكة Feistal والتي تكرر 256 bits RC6 عشرين مرة. الخاصية المطلوبة لخوارزمية التشفير هي ان تغيير صغير في النص الصريح او المفتاح ينتج عنه تغيير مهم في النص المشفر يعرف‌هذا ايضاً بAvalanche Effect. ال Avalanche Effect في128 bits RC6 تقريبا 62.18 %. اذا غيرنا نفس كمية المعلومات في المفتاح 512 bits RC6 يصبح Avalanche Effect تقريبا 257.27% . الخوارزمية المقترحة مقاومة الى هجوم matchingو dictionary مما يزيد من الامنية للخوارزمية 128 bits RC6 السابقة بواسطة استخدام حجم كتلة 512 bits بدلا من 128 bits.

*Control and Systems Eng. Deptartment. University of Technology
**Information and Technology Eng. Dept. University of Technology

1. Introduction

Symmetric-key block ciphers have long been used as a fundamental cryptographic element for providing information security. Although they are primarily designed for providing data confidentiality, their versatility allows them to serve as a main component in the construction of many cryptographic systems such as pseudo random number generators, message authentication , protocols, stream ciphers, and hash functions. There are many symmetric-key block ciphers which offer different levels of security, flexibility, and efficiency. Among the many symmetric-key block ciphers currently available some (such as DES, FEAL, Blowfish, IDEA, RC6, Twofish, Mars, Rijndeal, and Serpent) have received the greatest practical interest [1].

In this paper, a symmetric-key block cipher, called 512 bits RC6, will be presented, with a block size of 512 bits. The proposed cipher uses a variety of operations to provide a combination of high security, high speed, and implementation flexibility. The main theme behind the proposed design is to get the best security/performance tradeoff by utilizing the strongest techniques available today for designing block ciphers.

The proposed algorithm re-uses the key schedule that was used in 256 bits RC6. This already had an excellent track record as a rock-solid key schedule and had been studied widely.

2. Feistel Network (FN)

Most symmetric-key block ciphers (such as DES, RC5, CAST, and Blowfish) are based on a “Feistel” network constructing and a “round function”. Figure (1) shows the general design of a Feistel cipher. The input is broken into two equal size blocks, generally called left (L) and right (R), which are then repeatedly cycled through the algorithm. At each cycle, a hash function (f) is applied to the right block and the key, and the result of the hash is XOR-ed into the left block. The blocks are then swapped. The XOR-ed result becomes the new right block and the unaltered right block becomes the left block. The process is then repeated a number of times.

Different round functions provide different levels of security, efficiency, and flexibility. The strength of a Feistel cipher depends heavily on the degree of diffusion and non-linearity properties provided by the round function. The strength of various FN systems especially the resistance of FN against Differential Cryptanalysis (DC) and Linear Cryptanalysis (LC)) is tied directly to the architecture of the S–boxes of FN [2].

This paper presents some existing modern cipher such as ABC which is a substitution-permutation network comprising 17 rounds with 3 different kinds of round functions. It is derived from MMB and SAFFER block cipher [3] and Unbalanced Feistel Networks and Block-Cipher Design (UFNs) consisting of a series of rounds in which one part of the block operates on the rest of the block [4]. It also presents: an Ultra-Lightweight block cipher which is It is an example of an SP-network and consists of 31 rounds. The block length is 64 bits and two key lengths of 80 and 128 bits are supported [5].

3. The RC6 Algorithm

RC6 was the simplest of the AES finalists. However one area of advantage for RC6 seems to be the small memory requirements. It is the superior performance of RC6 on processors such as these that make RC6 particularly suited to the high-end smart cards of today.

RC6 is a fully parameterized family of encryption algorithms [6]. 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 The AES submission is targeted at w = 32 and r = 20.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 six basic operations. The base-two logarithm of w will be denoted by lgw.

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 rotate the w-bit word a to the left by the amount
given by the least significant lgw bits of b

a>>>b rotate the w-bit word a to the right by the amount given by the least significant lgw bits of b [6].

3.1 Encryption of RC6

RC6 works with four 32-bit registers A;B;C;D which contain the initial input plaintext as well as the output ciphertext at the end of the encryption [see figure (2)]. The first byte of plaintext or ciphertext is placed in the least-significant byte of A; the last byte of plaintext or ciphertext is placed into the most-significant byte of D. Use the (A;B;C;D) = (B;C;D;A) to mean the parallel assignment of values on the right to registers on the left [7].

Encryption with RC6 w/r/b
Input: Plaintext stored in four w-bit input registers A;B;C;D
Number r of rounds
w-bit round keys S[0; : : : ; 2r + 3]
Output: Ciphertext stored in A;B;C;D
Procedure:
B = B + S[0]
D = D + S[1]
for i = 1 to r do
begin
t = (B × (2B + 1))<<<lgw
u = (D × (2D + 1))<<<lgw
A = ((A ⊕ t)<<<u) + S[2i]
C = ((C ⊕ u)<<<t) + S[2i+ 1]
(A;B;C;D) = (B;C;D;A)
end
A = A + S[2r + 2]
C = C + S[2r + 3]

3.2 Decryption of RC6

The algorithm of the RC6 decryption is as follows [7]:

Decryption with RC6 w/r/b
Input: Ciphertext stored in four 32-bit input registers A;B;C;D
Number r of rounds
w-bit round keys S[0; : : : ; 2r + 3]
Output: Plaintext stored in A;B;C;D
Procedure:
C = C - S[2r + 3]
A = A- S[2r + 2]
for i = r down to 1 do
begin
(A;B;C;D) = (D; A;B;C)
u = (D × (2D + 1))<<<lgw
t = (B × (2B + 1))<<<lgw
C = ((C - S[2i + 1])>>>t) ⊕u
A = ((A - S[2i])>>>u) ⊕ t
end
D = D - S[1]
B = B - S[0]

3.3 Key schedule for RC6

The user supplies a key of b bytes. Sufficient zero bytes are appended to give a key length equal to a non-zero integral number of words; these key bytes are then loaded in little-endian fashion into an array of c w-bit 13 s L[0]; : : :; L[c _ 1].

Thus the first byte of key is stored as the low-order byte of L[0], etc., and L[c_1] is padded with high-order zero bytes if necessary. Note that if b = 0 then c = 1 and L[0] = 0. The number of w-bit words that will be generated for the additive round keys is 2r + 4 and these are stored in the array S[0; :::; 2r + 3].

The constants P32 = B7E15163 and Q32 = 9E3779B9 (hexadecimal) are the same "magic constants" as used in the RC5 key schedule. The value of P_{32} is derived from the binary expansion of e - 2, where e is the base of the natural logarithm function. The value of Q_{32} is derived from the binary expansion of Ø-1, where Ø is the Golden Ratio [7].

Key schedule for RC6-w/r/b
Input: User-supplied b byte key preloaded into the c-word
array L[0; : : : ; c _ 1]
Number r of rounds
Output: w-bit round keys S[0; : : : ; 2r + 3]
Procedure: S[0] = Pw
for i = 1 to 2r + 3 do
S[i] = S[i _ 1] + Qw
A = B = i = j = 0
v = 3_ max{c,2r + 4}
for s = 1 to v do
{
A = S[i] = (S[i] + A + B)<<<3
B = L[j] = (L[j] + A + B)<<<(A + B)
i = (i + 1)mod(2r + 4)
j = (j + 1)modc
}

4. The Proposed 512 bits RC6 Algorithm

In general the proposed algorithm differs from the previous RC6 algorithm which encrypts and decrypts 128-bit block size. The proposed algorithm could be used to encrypt and decrypt 512 bits block size. Figure (3) shows the structure of the proposed algorithm which in fact is a Feistel network. It consists of splitting the plaintext into two 256-bits halves. Feistel ciphers are a special class of iterated block ciphers where the ciphertext is calculated from the plaintext by repeated application of the same transformation or round function. The round function is applied to one half using a subkey and the output of F function is XORed with the other half. The two halves are then swapped. Each round follows the same pattern except for the last round where there is no swapping.

A nice feature of a Feistel cipher is that encryption and decryption are structurally identical, though the subkeys used during encryption at each round are taken in reverse order during decryption.

From figure (3), the first round and the last round require four subkeys of 64 bits while the others require two subkeys. This is because the first and last round, as in the pervious RC6 algorithm, used these subkeys to perform diffusion before the first and after the last round. The proposed algorithm required to double the previous RC6 by moving from 32 bits to 64 bit registers and then using Feistel network.

4.1 The Proposed 256 bits of Improved RC6 Algorithm

In 256 bits proposed RC6 the word is doubled to 64 bits instead of 32 bits in the previous 128 RC6 encryption algorithm. The first and the last round differ from others where B and D are adding to the S0, S1 respectively on the other hand in the last round the A and C adding to S42, S43 respectively. All the operation is on 64 bits as shown in figure (4).

4.2 Key Scheduling of Proposed Algorithm

The key schedule of proposed 512 bits RC6-w/r/b is practically identical to the key schedule of RC5-w/r/b. Indeed, the only difference is that for 512 bits RC6-w/r/b, the number of bits per word is(w = 6) bits instead of the previous RC6 algorithm (w=5) bits. The length of

word is 64 bits are and then 44 words are derived from the user-supplied key for use during encryption and decryption.

The algorithm of key schedule is presented below in full detail. The user supplies a key of b bytes, where $0 \leq b \leq 255$. From this key, 2r + 4 64-bits words (w bits each) are derived and stored in the array S[0; : : : ; 2r + 3].this array is used in both encryption and decryption.

5. Security of proposed 512 bits RC6 Algorithm

The most important requirement is stated succinctly in the AES announcement: '*The security provided by an algorithm is the most important factor in the evaluation.*'

5.1 Matching ciphertext attack

The block size of 512 bits makes the proposed algorithm resistant to the matching ciphertext attack. Where after encryption of 2^{256} blocks, equal ciphertexts can be expected and information is leacked about plaintext but, the previous RC6 algorithm with 128 bits block size will require 2^{64} ciphertext [8].

5.2 Dictionary Attacks

As the block size is 128 bits, a dictionary will require 2^{128} different plaintexts to allow the attacker to encrypt or decrypt an arbitrary message under an unknown key, while the proposed algorithm will require 2^{512} different plaintexts. This attack applies to any deterministic block cipher with 128-bit blocks regardless of its design [8].

5.3 Avalanche Effect

This section is prepared for making a statistical test on the ciphertext produced from encryption the following plaintext in hexadecimal:

**0.00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000**

**1.11111111111111111111111111111111
11111111111111111111111111111111
11111111111111111111111111111111
1111111111111111111111111111**

**2.22222222222222222222222222222222
22222222222222222222222222222222
22222222222222222222222222222222
2222222222222222222222222222**

**3.33333333333333333333333333333333
33333333333333333333333333333333
33333333333333333333333333333333
333333333333333333333333333**

**4.44444444444444444444444444444444
44444444444444444444444444444444
44444444444444444444444444444444
444444444444444444444444444**

**5.55555555555555555555555555555555
55555555555555555555555555555555
55555555555555555555555555555555
555555555555555555555555555**

**6.66666666666666666666666666666666
66666666666666666666666666666666
66666666666666666666666666666666
666666666666666666666666666**

**7.77777777777777777777777777777777
77777777777777777777777777777777
77777777777777777777777777777777
777777777777777777777777777**

**8.88888888888888888888888888888888
88888888888888888888888888888888
88888888888888888888888888888888
888888888888888888888888888**

**9.99999999999999999999999999999999
99999999999999999999999999999999
99999999999999999999999999999999
999999999999999999999999999**

**10.
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaa**

And using a key of length 32 bytes where the key is "**ffffffffffffffffffffffffffffffff**".

Horst Feistel referred to the avalanche effect as: "*a small change in the key [that] gives rise to a large change in the ciphertext*" [9]. Table 1 shows the avalanche effect on the plaintext when only one bit is changed in the key by using the proposed 512 bits RC6 algorithm.

Table 2 show the avalanche effect on the plaintext when only one bit is changed in the key by using the proposed 256 bits RC6 algorithm.

Table 3 shows the avalanche effect on the plaintext when only one bit is changed in the key by using the previous 128 bits RC6 algorithm.

Table (1) shows that the average of avalanche effect of the proposed 512 bits algorithm is 257.27% and table (2) shows that the average of the avalanche effect of the proposed 256 bits algorithm is 133.72% while table(3) shows that the average of the avalanche effect of the previous 128 bits RC6 algorithm is 62.18%.

5.4 Measuring the Complexity

The strength of a cipher is determined by the computational complexity of the algorithms used to solve the cipher. The strength of an algorithm is measured by its time (T) and space (S) requirements. The time and space requirements grow, as the size of input is increases.

The evaluation of on improved algorithm can be done through analyzing the computing time, which is performed by studying the frequency of execution of its statements given various sets of data. The common notion for evaluating an algorithm is the concept of the order of magnitude of the time, complexity and its expression by sympototic notation. If T (n) is the time for an algorithm on n inputs, then it is written [10, 11]:

$$\mathbf{T(n) = \theta (f(n))}$$

In this section the complexity is of one round from cryptanalysis of the view to the two algorithms Serpent and improved algorithm are computed as the following:

- **The Complexity of one round of RC6 Algorithm**

$\theta (4 (2^{4n-1})) * \theta (4(n/4)) * \theta (2 (2^{4n+1})))$

$\theta (\log [4(2^{4n-1}) + 4(n/2)+ 2(2^{4n+1})])$

$\theta (\log [2^{16n-2} + 4(n/2) + 2^{8n+2})])$

$\theta (\log [2^{24n}])$

$\equiv \theta (e^{24n})$ where n=32

- **The Complexity of one round of proposed 256 bits RC6 encryption algorithm:**

$\equiv \theta (e^{24n})$ where n=64

The proposed 512 bits RC6 algorithm also increased the complexity of previous RC6 algorithm by using an additional operation (XOR) for each round.

6 Time requirement

In this section the time requirements are computed for the proposed 512 bits RC6 algorithm, 256-bits RC6 and the previous 256 bits RC6. Table 4 and figure (5) shows this test.

Conclusions

The proposed algorithm is a secure, compact and simple block cipher. It offers

a good performance and a considerable flexibility.

During the design process, several things can be concluded about cipher design:

1. The design has to remain simple.
2. Using a 64-bit multiplication operation within the proposed algorithm 512 bits RC6 was perhaps the most significant change in moving from RC6 to RC7 and resulted from a careful evaluation of the additional security that might be offered by the multiplication operation. This might attract increasing support in the future.
3. Re-use the key schedule that was used in RC6. This already had an excellent track as a rock-solid key schedule and had been studied widely.
4. The block size of 128 bits makes previous RC6 algorithm vulnerable to the matching ciphertext attack. Where after encryption of 2^{64} blocks, equal ciphertexts can be expected and information is leacked about the plaintext. That, the proposed algorithm with 512 bits block size is resistant to matching ciphertext attacks. It requires 2^{256} ciphertext.
5. As the block size is 128 bits, a dictionary attack will require 2^{128} different plaintexts to allow the attacker to encrypt or decrypt arbitrary message under an unknown key. This attack applies to any deterministic block cipher with 128-bit blocks regardless of its design. So , the proposed algorithm with 512 bits block size requirs 2^{512} different plaintexts.
6. From the results that were obtained in section 5 and after measuring the strength of the proposed algorithm. It is concluded that the proposed algorithm increases the security when compared to the complexity with previous 128 bits RC6 algorithm.

References

1. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. M. Matyas, L. O'Connor, M. Peyravian, D. Safford and N. Zunic, "Mars a candidate cipher for AES", *First Advanced Encryption Standard (AES) Conference*, Ventura, CA, 1998.
2. H. Feistel, "Cryptography and Computer Privacy," Scientific American, v. 228, n. 5, May 73, pp. 15-23.
3. Dieter Schmidt, "*ABC - A Block Cipher*", Wikipedia the free Encycleopedia, May 27, 2002.
4. Bruce Schneier and John Kelsey," *Unbalanced Feistel Networks and Block Cipher Design", Counterpane* Systems, 101 East Minnehaha Parkway, Minneapolis, MN 55419, 2005,*http://f*schneier,kelsey*g*@counterpane.com
5. A. Bogdanov1, L.R. Knudsen, G. Leander1, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe," *PRESENT: An Ultra-Lightweight Block Cipher"*, 2007, www.ist-ubisecsens.org.
6. R. Rivest, M. Robshaw, R. Sidney, and Y. Yin, "The RC6TM Block Cipher," *First Advanced Encryption Standard (AES) Conference*, Ventura, CA, 1998.
7. M.J.B. Robshaw "RC6 and the AES". January 9, 2001, mrobshaw@supanet.com
8. Ross Anderson, Eli Biham, and Lars Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard", *An Internet Survey*, 2000.
9. Shakir M. "A new feedback symmetric block cipher method", *Ph. D, Thesis University of Technology , Baghdad*, 1997.
10. Fred

halsall " Multimedia Communications Applications, Networks, Protocols, And Standards, *ADDISON-WESLEY*, 2001.

11. Thilo Zieschang, " Combinatorial Properties of Basic Encryption Operations", Advances in Cryptology Eurocrypt'97, *International Conference on the Theory And Application of Cryptographic Techniques Konstanz, Germany, May 11-15, 1997 Proceedings, Springer*, 1997.

Table (1) Avalanche Effect of 512 bits RC6 Algorithm: Change one bit in key

Block No.	Ciphertext 512 bits in Hexadecimal	Avalanche
0	**7a4338c0acb5552debbf4291428c5ff65d36252567b6bc6b831a0a0d47dd086ff13afed472150cf44fa25c37bd9c6713961a120aaae11647e5485794d50002d4** **cdae8517005d3d586d8326bec6e4601d72427a03fb5ca0d7740eae1a72f63627ddf40575b05623cf5aea783a9f9680961641347ddf741068dc5fedb3f3bad773**	**261**
1	**93b9d336882658809bf725c53617f42d4272df84d39b5f19a5845ab540eec289e7f662a3f22fdd61b796eeaf4f53914447b0c53552eaaad17e324ef9746619** **7d6173c70188553a9b28db493e638952a4aba91c1f44a4cea6220751a263889c1ec5c1ae77e7ef54af4d119fe12f06459475891e0bad242be837d3516f67ffa1**	**264**
2	**7b9ca425fc90806a82c60a181cadcaf1dcbd9729bad3c8469291ae3f99b7d596e4945d43af20b5f3b159fb618e819751c78e62e8db0a6c9cfe57c4d4ecf52f4** **8008af380eba28d094d92e6a442de7bea5d4805ee2356c8354ce4b9397497fd1d48b3dfe53d511b3853a9b35ecad59c7b8c1a9c429b5eb54f06d4cdb085409f2**	**249**
3	**bfd5903533cce1557d8424dcd419d89286295a3c02033abba976e6a6040376393b73f5a825b48a9f790b2918b08ee95522e15f5d465ff7fa746758e31e3614b** **3bc855817884528c86e1df041c6303254d70bd2cb0b6a9a3109b206c3ee5c8a41c9b285c1f17f7cd372e68892109d2b8acc786fa27f40aaa1fd77e337353b4bc**	**273**
4	**236ede3916eb30dc1f6d740ee753e1431258d09c31031703b14a91f96ef7c141a75af1d8e79744252892b45dda5d7c8740b678aa225250098f73c5be9d25ee65** **134c65e4cb951c76d0c556ad2c2c4aaac91887789d6cab1d6c61b2540b432d97f74f4d62f4485afcd6bb7513bc798634ca5da045ce796fe69d75546535e5921e**	**282**
5	**7c34f2a45473ec25eb9ad94870435a62b1fc0b096ce91a2c4dd392579eb84d3ce1bbd576410b27ce75cc96099fe7b519d9c6e46b15be0aada15cc4bb8d77e0a1** **655557637c541969d1b383867943a5311feaeda5137a06d4546fc4f6080a67cff51f78ef579dbb614bad84233595ff81ca1587a8e544416eb28c6db3cd07e7cb**	**242**
6	**2cb888d439109b15e48eaa39886ef7c2810335bbb14e027c17bfea7cbee4d752f39a08bc40cfa39879d08a4016f1e019520950c5992293b7e51e9aa79f70662d** **a0c5599c91eab0f9da84a91eca6901cf4c97a95a9fea1b35f3507b67418319256e2cf74c4dcabb552ab25b96093f2870bfe6cb5fe4b8e1de68a52d20e8800411**	**275**
7	**a99041b2ce8420ba9d99e2265cc3ae9adeb50c1df7ec01fbc74630592cede9c168fcb9091643d0a71970301dabeb34dd1053a86ba6801e35a8c86bee1ebc316c** **d2a7598d9be76a3edd878d226d76476c2fde4d352ff46461407b571ddabb3d5a84fa866038541c41ac8894a4c8c160069de481a264b65425ea5fb2e7f4947f9**	**253**
8	**249e4f3c8c80295e152e1a6ef6dcfdbbe4a96649b21124fee4720b48f5350bebc1f7e3a73f2c8c8c53c23c40c0e570a9a8684e1a2a2a06e7ba4278477d29d77** **77181a90c23309109b38362654da641b529e437004f174c8fc10df0cb2dbf47ee05ed52a44599ea1c214a932b61aecaa7696ce60eb0a436dcb899f810e92b506**	**231**
9	**84a33ce3ff9864edc51461ca4228dc01c62a7047d244c633e3c869c80dc038dcd6a779d329bd23abf915dfe1414c43c21a2270dd35ea48d928382d26a688404c** **153d4ac5de254a9ef6808d9e257ff64c6273859dd6038c1b9e2ba2ae799956bddf7e7b4b1a1684f1f3c54db180f4770b21fa65cde709cb521bdf513b303255c0**	**245**
10	**de423760e6763ad42effa7f4014adc77cd7b43e606b77b4be3bd7e90fb4b1cf4d76c423fda27c0f9296f949404fb7fb2f8d0da2a01018230c070307b2745a950** **9d3f6a4b785984c995bfc085b2cf2cd977ae914b62c421ab40f1c3513fc9d5d360fd33dbc6d9e716a13d7a9eb7db28a9215bb7b6b12a8209705a9df8a5e8bd79**	**255**
Key1	**ffffffffffffffffffffffffffffffff**	
Key2	**fffffffffffffffffffffffffffffffe**	

Table (2) Avalanche Effect of 256 bits RC6 Algorithm: Change one bit in key

Block No.	Ciphertext 256 bits in Hexadecimal	Avalanche
0	**8f3205a7633944515d873d76e3a735388ff49d67177ff20bd41961fdb3e0daa5 d168ce2bdb069e361900cd9cafd9ff0a403cdac9e2f0145d7a7658695cd61583**	**144**
1	**b4a040ea3ae7b18e1a4881a17a5c64cb10542ccbec53e9859559f087a9d0f2e f9ccc6e591a1d62edeb1b67b871a1dfba8aa47811cd0160a1f930582f5788cb4**	**140**
2	**41b924bccf6e076c4ec565a72735c9fab8337b6714bbe30a95bf0cb3413712ab 92a604bc75951823230690c2ab7bc74ae407440ada9c7f74e212d48342f725e7**	**131**
3	**33f0b29c44de515157c619fec2e8bfac3e6939864fb52cdf99c2ec692552d7c8 953690cb8730e9113d1afa16329aace8eb97b190d2a29b7286fdb8e0c9931**	**127**
4	**31509ba0932aaebd1a967d0592356a804208d687e798c85a399cb20d9c094298 3d7cdadab70cf6cea5b0943e949eac32d11d2898ff4066f6c2fc013a7b26aca**	**129**
5	**b72709d1b86d258900d097d38a3e2c992acbbce1b43b8b4fe77970e9a90e23 8b972ecf7dad39eb53ac2243be4aa928639574ac507bd3a4a7b54100d760c1f1**	**133**
6	**a1f96507bad45afc8954473fedfde86cdcd1e1f1d07f82133627d9feb9a11 6e4276c07eeaaf705d1bb4beac804a5be6fd901469be273b109ac0bb972e5a3**	**133**
7	**5960c78ca60c39f35fe94744c916b823fbf1837a2aa1e17da2c4b7b46ea1a1ff 82967dc37c8ec46e26580d4934877adb38189422607edfee4e1cdec32ec63681**	**145**
8	**ab5d84d076ede311b2555dedc1f7d8ea7838344fa8e07c78f14522e6cb930bca 3940059a19bc9632b50bdb584836399e7da10421b63d92e476ba5c895cb2a2f7**	**132**
9	**583fc10ca1323336412cffbdb20f3d55404e301f64eae40dcf3f0dbc5724997 e5759290cb73257c1720ce6b8fcaf13af4d8778ae731faf55b828df4de57d930**	**130**
10	**d7e7fb89a73769c4f6801f48435bcd19307c118bbed975756cf8b614b49761d5 611ec5595f3247ae9e3aa532e56c9a0f4346ffd4eca1f5a869ead4704917601f**	**127**
Key1	**ffffffffffffffffffffffffffffffff**	
Key2	**fffffffffffffffffffffffffffffffe**	

Table (3) Avalanche Effect of proposed 128 bits RC6 Algorithm: Change one bit in key

Block No.	Ciphertext 128 bits in Hexadecimal	Avalanche
0	cc7ab1aff926ebac1f1a1472af06e9c5 13f8dcd3b867218d3bda92dc892bb521	56
1	a64019e16e073808ec4a7b5ba256294b 784160e0f937bec6b735b94c69847cc8	63
2	95d1998e471ec92047b7d917fddae8f fb127c5ecd55862a82e4e1e51f8afc9	61
3	7d4e16f4725b8517d773f1bafa776b0 5bb799aecb01a9e54e6c76962b80b5e4	54
4	576f5643236f997c505c418f4b165fa2 361fd0d5c90279334889c761f8882ee9	65
5	ee16d7a73430e704b0822cb348524f99 415b08aaec18f838dbef1da4b02117f4	70
6	c33933ec69f0c264046d0b3ceb7eaa8 da7c2cd34b3ee8fd70f7c8a8dedfa55c	60
7	d21ad5efb2c49f868b6545abad8cf36 349309b49b6ff5764d0b587fa1135d62	67
8	51c9d8d98dbae5fea25bbe5584d085e8 ff14d8ecea150abab5d7ca15047274b	59
9	4f893db93ca9c49cc656b37066a28cfb 33ef3520cb9a470bde3ab134192f242	58
10	2a033755e9e02901758cf26d56a629a 5bc4ede838ece1cb7ad1e02c29bf2df	71
Key1	ffffffffffffffffffffffffffffffff	
Key2	fffffffffffffffffffffffffffffffe	

Table (4) Time Comparison Between 512, 256, 128 bits of RC6 Encryption Algorithm

Algorithm	Number of Bytes	Time in Second
256-bits RC6	10000	0.15
	100000	1.512
	1000000	14.911
128-bits RC6	10000	0.15
	100000	1.452
	1000000	13.459

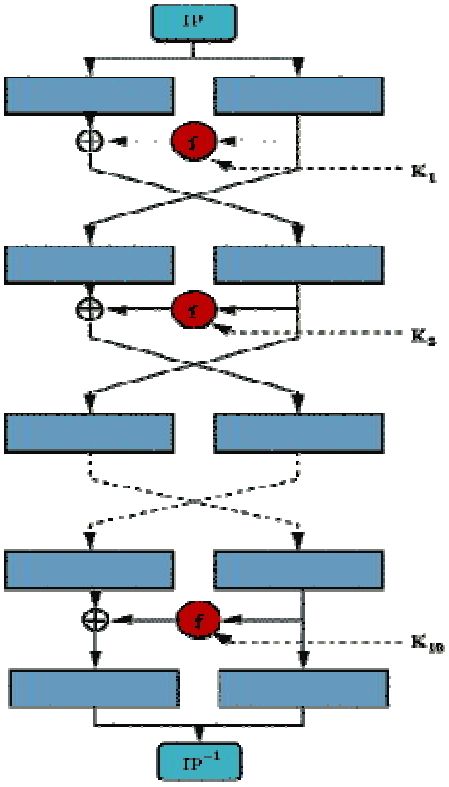

Figure (1) General design of a Feistel cipher

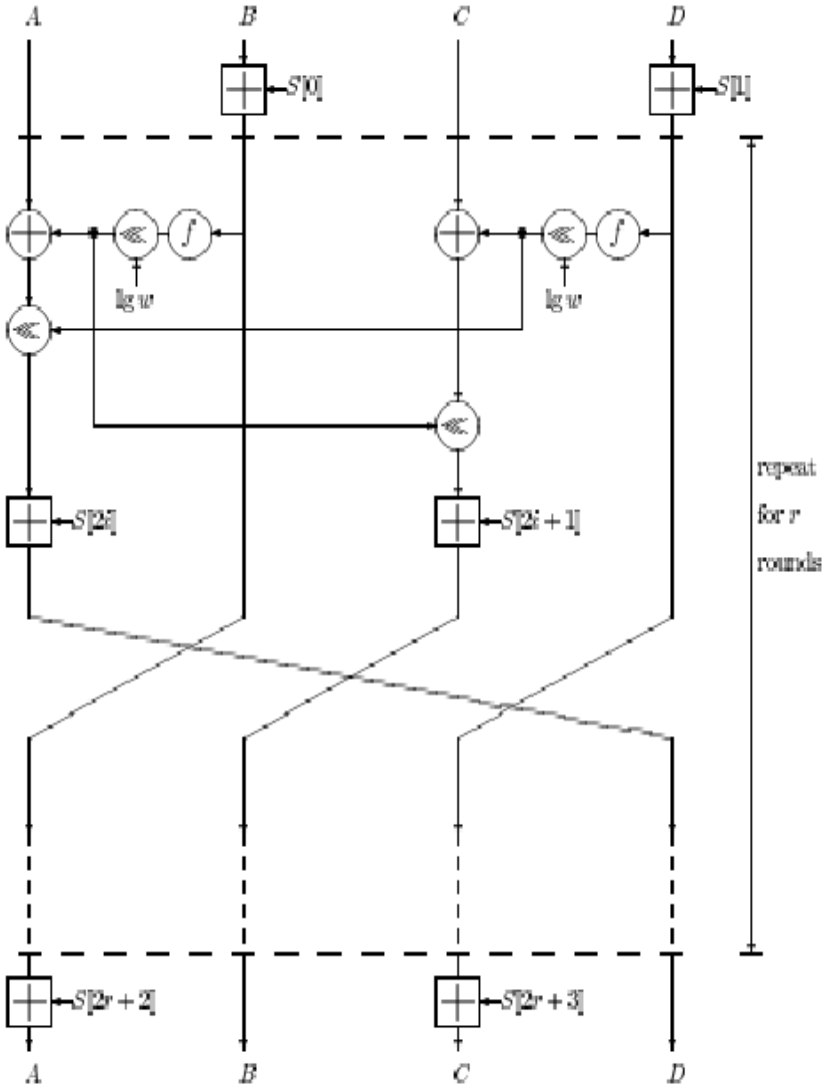

Figure (2) Encryption with RC6-w/r/b . Here f(x) = x(2x +1) . w=32, r=20,b=16,24, or 32

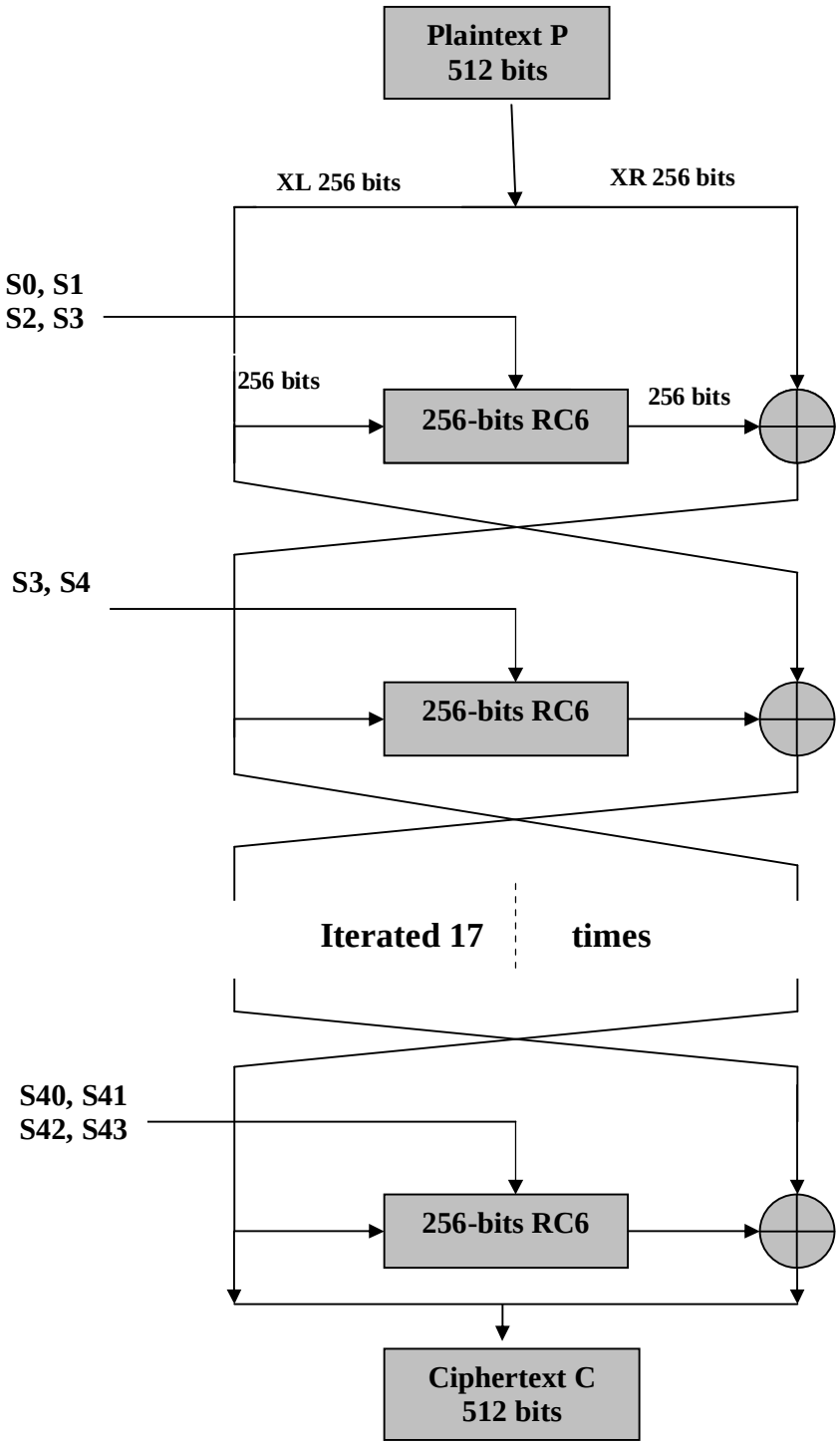

Figure (3) the Proposed 512 RC6 Encryption Algorithm

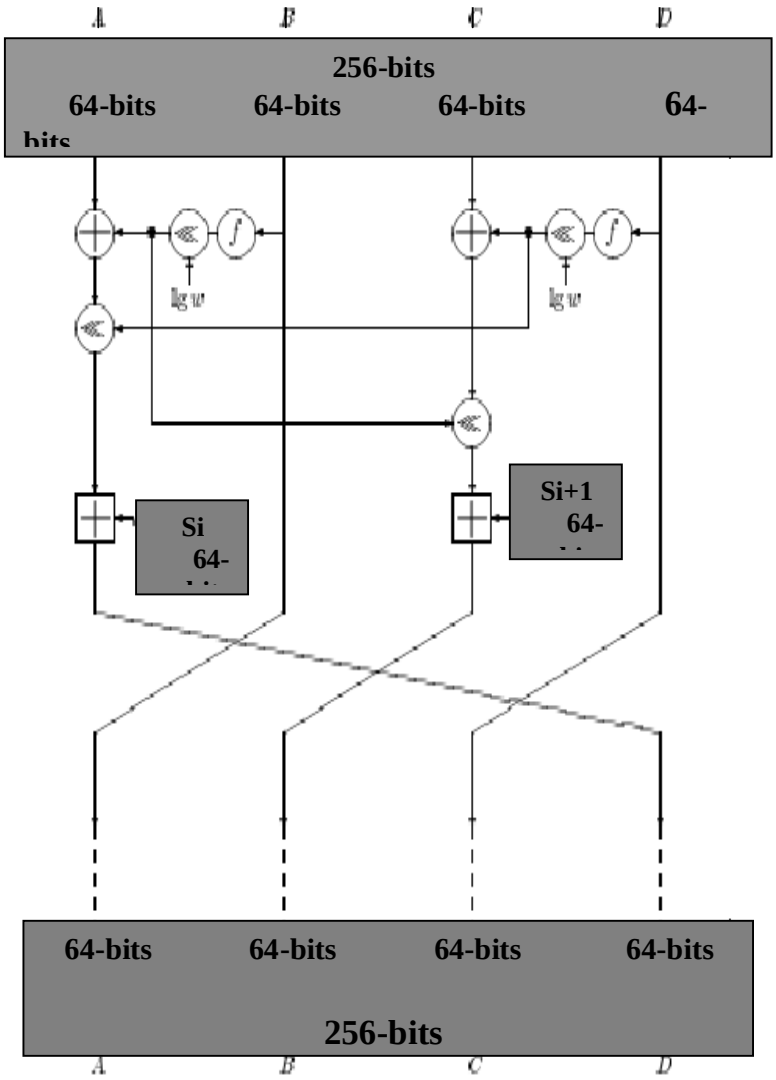

Figure (4) 256-bits RC6-w/r/b . Here f(x) = x(2x +1) w=64

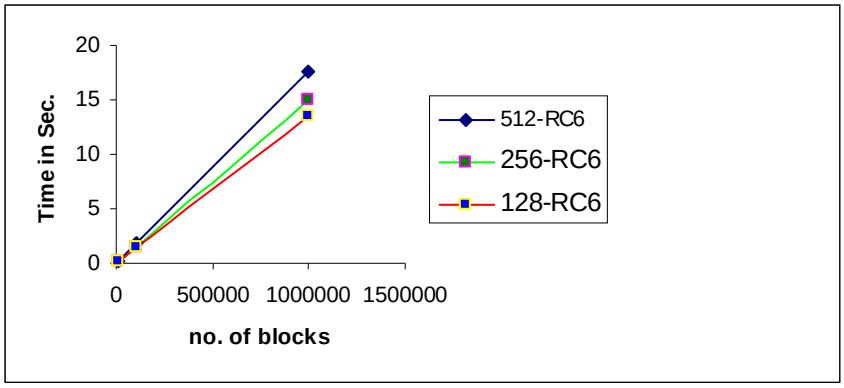

Figure (5) Time Comparison Between 512, 256, 128 bits of RC6 Encryption Algorithm