DOI: http://dx.doi.org/10.26483/ijarcs.v10i4.6458

ISSN No. 0976-5697

Volume 10, No. 4, July – August 2019

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

FORMATION OF AN IMPROVED RC6 (IRC6) CRYPTOGRAPHIC ALGORITHM

Agu,Edward O.,Ogar Michael O., Okwori Anthony O.
Computer Science Department
Federal University Wukari
Wukari Taraba State, Nigeria

Abstract: IRC6 cryptosystem is an improvement on RC6 which was developed in the course of this research to guard against crypto-analytical attack. This is achieved by doubling its security at little or no computational cost. RC6 is an improvement on RC5, and RC5 was an improvement on RC4. IRC6 is designed to meet the requirements of increased security and better performance.

Object-oriented analysis and design methodology (OOADM) together with Java programming and Development technologies were used to implement both Normal RC6 algorithm and improved RC6 (IRC6) cryptographic algorithm in this research in order to achieve the research goal which is to show the significant improvement of IRC6 over normal RC6 by carrying out performance evaluations of the two algorithms. The developed IRC6 was also evaluated to have transparent cipher-text and thereby found to survive any crypto-analytical attack.

Keywords: NRC6, IRC6, Cryptography, Algorithm, Data Security, cipher text

I. INTRODUCTION

Security of data has always being the greatest challenge in IT world today. Revester code version 6 (RC6) among other cryptographic algorithms has been tipped for security inversion which necessitate this review and improvement of the existing RC6 cryptographic algorithm [1].

RC6 is an improvement over RC5, and RC5 is an improvement over RC4. RC6 was designed to meet the requirements of increased security and better performance [2]. RC6 makes use of data dependent rotations. One new feature of RC6 is the use of four working registers instead of two. While RC5 is a fast block cipher, extending it to act on 128-bit blocks using two 64-bit working registers [3]. RC6 modified its design to use four 32-bit registers rather than two 64-bit registers. This has the advantage that it can be done two rotations per round rather than the one found in a half-round of RC5 [4].

Two components of RC6 that were absent from RC5 are a quadratic function to mix bits in a word more effectively and a mixed rotation that is used both to hinder the construction of good differentials and linear approximations and also to ensure that subsequent data dependent rotation amounts are more likely to be affected by any ongoing avalanche of change [5]. The research illustrates the features of enhanced RC6 techniques.

II. LITERATURE REVIEW

An initial analysis of the security of RC6 and its resistance to the basic forms of differential and linear cryptanalysis was given in [2].

[6] in their research titled the block encryption algorithm combined with the logistic mapping and SPN structure shows that S-boxes have good confusion effects and P-boxes have good diffusion effects. But it limitations implies that it has to compromise and perform a balancing act between S and P boxes as well as balancing the security as well.

Existing RC6 block Cipher cryptographic algorithm was analyzed in a research carried out by [7] in which it was established that it is a secured, compact and simple block cipher. It offers good performance and considerable flexibility.

Hash RC6 - Variable length hash algorithm using RC6 was equally studied by [8] which show that it is possible to generate hash algorithm using symmetric block cipher without any limitation.

[9] in their research on a comparative survey on symmetric key encryption algorithms shows that RC6 cryptography provides number of security goals to ensure the privacy of data, on-alteration of data and so on. Due to the great security advantages of cryptography it is widely used today but with its limitations which shows that it is impossible for a hacker to decrypt RC6 algorithm.

An enhanced RC6 algorithm with the same structure of encryption and decryption was carried out by [10] which yielded to Feistel structure that has the same algorithm between encryption and decryption. The features of the SPN structure in their research has a different algorithm between encryption and decryption. The limitation of this research lies in the area which increases twice when compared with the Feistel one when SPN structure in implemented via hardware.

III. SYSTEM ANALYSIS AND DESIGN

A. Description of Normal RC6 (NRC6) Cryptographic Algorithm

RC6 is a fully parameterized family of encryption algorithms. A version of RC6 is also specified as RC6-*w/r/b* where the word size is *w* bits, encryption consists of a number of rounds *r*, and *b* denotes the encryption key length in bytes.
RC6 is targeted at *w* = 32 and *r* = 20, the parameter values specified as RC6-*w/r* are used as shorthand to refer to such versions. For all variants, RC6-*w/r/b* operates on four *w*-bit words using the following six basic operations:

$a + b$: Integer addition modulo 2*w*
$a - b$: Integer subtraction modulo 2*w*
$a \oplus b$: Bitwise exclusive-OR of *w*-bit words
$a \times b$: Integer multiplication modulo 2*w*
$a <<< b$: Rotate the *w*-bit word *a* to the left by the amount given by the least significant lg *w* bits of *b*
$a >>> b$: Rotate the *w*-bit word *a* to the right by the amount given by the least significant lg *w* bits of *b* (where lg *w* denotes the base-two logarithm of *w*).

RC6 exploits data-dependent operations such that 32-bit integer multiplication is efficiently implemented on most processors. Integer multiplication is a very effective diffusion, and is used in RC6 to compute rotation amounts so that these amounts are dependent on all of the bits of another register. As a result, RC6 has much faster diffusion than RC5 and RC4.

- **Key Schedule**

The key schedule of RC6-w/r/b is practically identical to that of RC5-w/r/b. In fact, the only difference is that in RC6-w/r/b, more words are derived from the user-supplied key for use during encryption and decryption.

The user supplies a key of b bytes, where $0 \le b \le 255$. 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 into an array of c w-bit words L[0], L[1], . . . , L[c − 1]. The number of w-bit words generated for additive round keys is 2r + 4, and these are stored in the array S[0, 1, . . . , 2r + 3].
The key schedule algorithm is as shown as follows:

- **Key Schedule for RC6-w/r/b**

Input: User-supplied *b* byte key preloaded into the *c*-word array *L*[0, 1, . . . , *c* − 1] Number of rounds, *r*
Output: *w*-bit round keys *S*[0, 1, . . . , 2*r* + 3]

- **Key expansion:**

Definition of the magic constants:
P_w= Odd((e − 2)2^w)
Q_w= Odd((φ − 2)2^w)
Where:
e = 2.71828182 . . . (base of natural logarithms)
φ = 1.618033988 . . . (golden ratio)
Converting the secret key from bytes to words:
for i = b − 1 down to 0 do
L[i/u] = (L[i/u] <<<8 + K[i]

Initializing the array S
S[0] = Pw
for i = 1 to 2r + 3 do
S[i] = S[i − 1] + Q_w

Mixing in the secret key S
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) mod c
}

- **Encryption**

RC6 encryption works with four w-bit registers A, B, C and D which contain the initial input plaintext. The first byte of plaintext is placed in the least significant byte of A. The last byte of plaintext is placed into the most significant byte of D. The arrangement of (A,B,C,D) = (B,C,D,A) is like that of the parallel assignment of values (bytes) on the right to the registers on the left, as shown in Figure 2.12. The RC6 encryption algorithm is shown as follows:

- **Encryption with RC6-w/r/b**

Input: Plaintext stored in four w-bit input registers A,B,C,D
Number of rounds, r
w-bit round keys S[0, 1, . . . , 2r + 3]
Output: Cipher text stored in A,B,C,D
Procedure:
B = B + S[0]
D = D + S[1]
for i = 1 to r do
{
t = (B × (2B + 1)) <<<1g w
u = (D × (2D + 1)) <<<1g 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]

- **Decryption**

RC6 decryption works with four w-bit registers A,B,C,D which contain the initial output ciphertext at the end of encryption. The first byte of ciphertext is placed into the least significant byte of A. The last byte of ciphertext is placed into the most significant byte of D. The RC6 decryption algorithm is illustrated below:

- **Decryption with RC6-w/r/b**

Input: Ciphertext stored in four w-bit input registers A,B,C,D
Number of rounds, r
w-bit round keys S[0, 1, . . . , 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
{
(A, B,C,D) = (D,A, B,C)
u = (D × (2D + 1)) <<<1g w
t = (B × (2B + 1)) <<<1g w
C = ((C − S[2i + 1] >>> t) ⊕u
A = ((A − S[2i]) >>> u) ⊕t
}

D = D − S[1]
B = B − S[0]

B. Description of the Improved Rivester Code Version 6 (IRC6) Cryptographic Algorithm

Encryption/Decryption with IRC6-w/r/b
Input: Plaintext stored in four w-bit input registers A, B, C & D
r is the number of rounds
w-bit round keys S[0, ... , 2r + 3]
Output: Ciphertext stored in A, B, C, D

- Encryption Procedure:

```
B = B + S[0]
D = D + S[1]
for k = 1 to r do
{
        i = hash(hash(k))
        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]
```

- Decryption Procedure:

```
C = C - S[2r + 3]
A = A - S[2r + 2]

for k = r downto 1 do
{
   i = hash(hash(k))
  (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]
```

The strength of the improved version of the techniques is based on the double cryptographic hashing of the key which is denoted by “**key = hash(i)”** to sustain crypto-analytical attack on cypher text during transmission of data to the cloud.

IV. PERFORMANCE EVALUATION OF RESULT AND DISCUSION

The performance of the IRC6 implemented in this research is evaluated against the normal RC6 (NRC6) and the overall research goal is equally evaluated.
Table 1 is used to evaluate the efficiency of both normal and improved RC6 (IRC6) algorithm.

Table 1: Performance Evaluation of Normal RC6 and Improved RC6

S/ No	Plaintext	Encrypted Cyphertext	Compare	Modules	Start Time	End Time	Time Difference (Secs)	Total Time Lag (Secs)
1	Eddy improving RC6	vKsK671WBPtleBTZjz/zqHGK0z05vMuLwTDp0mPGfXA=	NRC6	KeyGen Time	31.05.2018 07:22:41.224	31.05.2018 07:22:41.224	0.0	0.438
				Encrpt Time	31.05.2018 07:22:41.224	31.05.2018 07:22:41.662	0.438	
				Decript Time	31.05.2018 07:22:41.662	31.05.2018 07:22:41.662	0.0	
		MsZ+yXwhBB41TQZ3CMBqaNlzBgmbVy7ZD3fZkf5gHOY=	IRC6	KeyGen Time	31.05.2018 12:04:23.474	31.05.2018 12:04:23.474	0.0	0.609
				Encrpt Time	31.05.2018 12:04:23.474	31.05.2018 12:04:24.068	0.594	
				Decript Time	31.05.2018 12:04:24.068	31.05.2018 12:04:24.083	0.015	
2	This is the time to show improvement over existing knowledge	AZ3dnuTJ8N/lAQSC8sugFsnTnlksqxD5NQFEKTtlEejvFVVMNCDrzHETCe69rW4Vn87hlbXE2Jq+n61GY1epOg==	NRC6	KeyGen Time	31.05.2018 10:54:22.623	31.05.2018 10:54:22.623	0.0	0.344
				Encrpt Time	31.05.2018 10:54:22.623	31.05.2018 10:54:22.967	0.344	
				Decript Time	31.05.2018 10:54:22.983	31.05.2018 10:54:22.983	0.0	
		wlvEW9PBgGKBt4mWTXyFMigIVUUsev4whq3oIYv2qEGz6nFlhRkM8FXFrMCvjac5TBJcRD4Lvgs+e6DDhqiUhQ==	IRC6	KeyGen Time	31.05.2018 11:19:25.961	31.05.2018 11:19:25.976	0.015	0.578
				Encrpt Time	31.05.2018 11:19:25.976	31.05.2018 11:19:26.539	0.563	
				Decript Time	31.05.2018 11:19:26.539	31.05.2018 11:19:26.539	0.0	

3	Hello Computer Science department, Nnamdi Azikiwe University of standards and values. It’s time to secure our data.	XEMSyDDE17C x5EMixiaV8Iqw3 sbZmsZu6rAAw1 cvbD2AQg0691t Tp3nXACZfbITjb DYYNb0IpVO4 aWtsO/wuuEl9Z U+U04zg5Pj2TiQ CZpcDeItFAtG3 OFzwgP1SXRD5 vr8Iqcn/Rdn5zmz i5qkGsEB5 DDt3KiclOt6MYs mEIkI=	NRC6	KeyGen Time	31.05.2018 11:25:15.520	31.05.2018 11:25:15.520	0.0	0.609
				Encrpt Time	31.05.2018 11:25:15.520	31.05.2018 11:25:16.129	0.609	
				Decript Time	31.05.2018 11:25:16.129	31.05.2018 11:25:16.129	0.0	
		7DS8EHchIyTEx 8xMZ3Lj1AYlxp We19ZkKCfRHP KL2Til9U0h52Tu qxALa+uP3woe WREZAluETMU E nC1QG7EOVmsl JTwSk22+bZ94n UAIMkR4drlOH8 FSyvKp0+C2PL+ GJRq+f3hHzG/Z7 TtsQixQp3Iu LZX3rVUAeQCx Rn44Kkg=	IRC6	KeyGen Time	31.05.2018 11:31:50.179	31.05.2018 11:31:50.179	0.0	0.563
				Encrpt Time	31.05.2018 11:31:50.179	31.05.2018 11:31:50.742	0.563	
				Decript Time	31.05.2018 11:31:50.757	31.05.2018 11:31:50.757	0.0	
4	Cryptograp hic strength in RC6 Vs IRC6. The difference In time complexity.	ReSkmxa2H4oUn i5gRrZtyTUZFpL WunEvsHhXNJG agFX5172VBr0xg SB+Bf6POos+Hi UraRpOuEbt hS4Ne5hLGn8dU QvxXljsQluypSH WVec=	NRC6	KeyGen Time	31.05.2018 11:47:26.269	31.05.2018 11:47:26.269	0.0	0.703
				Encrpt Time	31.05.2018 11:47:26.269	31.05.2018 11:47:26.972	0.703	
				Decript Time	31.05.2018 11:47:26.972	31.05.2018 11:47:26.972	0.0	
		dQiwW54qDVA D0/DJSel4uFk4H cDdw3mUsesJ04 HRBht2ovOL3qi Nyr6E1M7/q0OB vNQ7w5EMFgHI 8cXheqH0AEJ6 WHoWBzENEny aDLHhe4w=	IRC6	KeyGen Time	31.05.2018 11:56:24.783	31.05.2018 11:56:24.783	0.0	0.438
				Encrpt Time	31.05.2018 11:56:24.783	31.05.2018 11:56:25.221	0.438	
				Decript Time	31.05.2018 11:56:25.237	31.05.2018 11:56:25.237	0.0	

Data contained in tables 1 were obtained by running the NRC6 algorithm against IRC6 version which were designed and implemented in this research to achieve the research goal. The modules considered include the key generation, encryption and decryption modules of the new system. Parameters considered in the system evaluation are the start time, end time and time lag of each module of the new system. Key generation time lags were observed to be negligible in both NRC6 and IRC6 due to small character length in key size. Also decryption time lags were equally observed to be negligible due to data have being read into buffer memory during encryption, close to system core readily for processing. Time lags in encryption modules in both NRC6 and IRC6 shows that the difference between the algorithm in terms of memory, time and computational complexity is negligible which implies that the security of the system is improved with little or no cost. The graphical representation of this improvement is shown in figure 1 below.

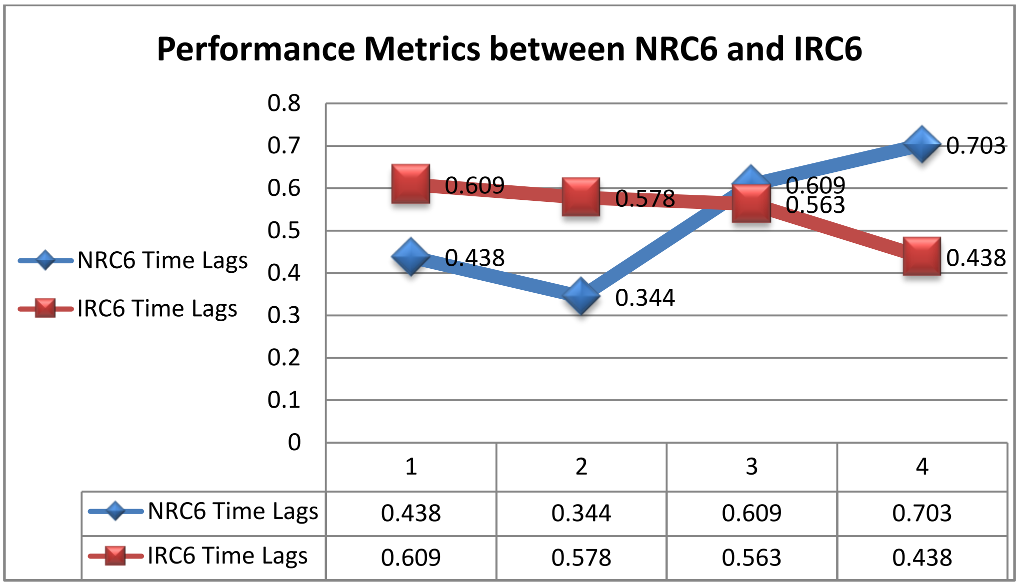

Figure 1: Performance Metrics between NRC6 and IRC6

Again, the encrypted cipher of IRC6 is quite difference from that of NRC6. It is not transparent and survives crypto-analytical attacks during data transmission.

V. CONCLUSION

Since it has been established that data security in IT world today is one of emerging and greatest challenging treat faced by IT industries. The best method of mitigating these challenges is to review and improve the existing solutions. This was the motivation into this research and improvement of the existing RC6 cryptographic algorithm was achieved with great success.

The significant security improvement achieved in this research at little or no cost is mind bugging and can be adapted as a security model or subsystem to secure many other IT applications.

VI. REFERENCES

[1] E. Biham and A. Shamir, “Differential ryptanalysis of the Data Encryption Standard”. Springer-Verlag, New York, zbMATH/Google Scholar. 1993

[2] S. Contini, R. L. Rivest, M. J. B. Robshaw and Y. L. Yin, “The Security of the RC6 Block Cipher”. v1.0. Available at www.rsa.com/rsalabs/aes/.Google Scholar, 1998.

[3] A. Biryukov and E. Kushilevitz, “Improved cryptanalysis of RC5”. In K. Nyberg, editor, *Advances in Cryptology-Eurocrypt*, vol. 1403 Lecture Notes in Computer Science, pages 85–99, Springer Verlag.Google Scholar, 1998

[4] B. S. Kaliski and Y. L. Yin, “On differential and linear cryptanalysis of the RC5 encryption Algorithm”. In D. Coppersmith, editor, *Advances in Cryptology-Crypto*, Vol. 963 of *Lecture Notes in Computer Science*, pages 171–184, Springer Verlag.Google Scholar, 1995.

[5] R. L. Rivest, “The RC5 encryption algorithm. In B. Preneel, editor, Fast Software Encryption”, Vol. 1008 of Lecture Notes in Computer Science, pages 86-96. Springer Verlag. 1995.

[6] H. H. Jian and L. Yang, “A block encryption algorithm combined with the Logistic mapping and SPN structure”. 2nd International Conference on Industrial and Information Systems (IIS), Vol. 2. 2010.

[7] L. R. Ronald, M.J.B Robshaw, R. Sidney and Y.L. Yin, “The RC6 Block Cipher, Version 1.1. 1998.

[8] K. Aggarwal and H. K. Verma “Hash RC6 Variable length Hash algorithm using RC6”, International Conference on Computer Engineering and Applications (ICACEA). 2015

[9] T. Gunasundari and K. Elangovan, “A Comparative Survey on Symmetric Key Encryption Algorithms”, International Journal of Computer Science and Mobile Applications, Vol.2 Issue. 2, 2014.

[10] H. K. Gil, N. K. Jong, Y. C. Gyeong, “An improved RC6 algorithm with the same structure of encryption and decryption, 11th International Conference on Advanced Communication Technology, ICACT2009, Vol. 2. 2009.

