RC4c : A Secured Way to View Data Transmission in Wireless Communication Networks

O. O Olakanmi
Department of Electrical & Electronic Engineering
Office 6, Engineering Annex, off Engineering Drive
University of Ibadan
Nigeria
olakanmi.oladayo@mail.ui.edu.ng

ABSTRACT

In spite of several crypto analytical criticisms on RC4 data encryption algorithm, its simplicity, speed has made it one of the popular encryption techniques employ in wireless communication networks. However, in recent time many experts no longer consider RC4 secure against attacks. This is due to some vulnerability detected through repetition of keys over a period of time. This exposes the weakness of the exclusive OR operator on which the RC4 technique is anchored.

In recent time, another variant of RC4 was developed called RC42s. This encryption algorithm innovatively solves the problem of mutual exclusive of XOR operator in RC4; however, the experimental result shows the improvement rate of 68% over RC4. That is, 32% of encrypted messages are susceptible to hacking whenever there is key collision.

In this paper, a novel method is developed to counter the negative effect of the mutual exclusiveness of the functional operator on RC4 and RC42s. The new technique called RC4c uses 2s complement and shifting operation to give a perfect and secured data encryption technique for wireless network. Performance analysis on speed and effect of mutual exclusiveness of XOR on RC4, RC42s and RC4c security was done. It was discovered that RC4c not only completely nullifies the weakness introduced by mutual exclusiveness of XOR in RC4 and RC42s but maintains the simplicity and throughput of the RC4 and RC42s.

KEYWORDS: *Encryption, RC4, Mutual Exclusive, Wireless Network, Key Collision, Data Encryption*

1. INTRODUCTION

A secured system has been defined has any system whose users do not feel any apprehension or anxiety while using [3]. Encryption algorithm can be better defined has any method that makes data in a system or over a communication channel secured. Data encryption algorithms can be categorized using either the type of input data they operate on or the types of key they use for encryption. Using the input type, encryption algorithms can be categorized as stream cipher and block cipher. Block encryption algorithm encrypts data in block form. In its simplest mode, the plain text P is divided into blocks P_1, P_2……P_n, which are fed into the data encryption system. Stream encryption algorithm encrypts stream of data in such a way that, individual bit of the data is fed into the encryption system. Stream cipher contains two major components; a key stream generator and a mixing encryption function. This mixing encryption function is exclusive OR function that performs two modulus addition on both the key stream and the plaintext.

DOI : 10.5121/ijcnc.2012.4208

Also data encryption may be categorized as Symmetric and Asymmetric based on the types of key used for encryption and decryption. In Symmetric algorithm, a private key is agreed upon by the sender and receiver node. This is used to encrypt the message by the sender and decrypt by the receiver. However, its main set back is how to securely share the secret key between the sender and the receiver. That is, if the hacker gets the knowledge of the key the entire encryption system collapses. Asymmetric algorithm uses two keys, the private and public key, for encryption and decryption. Asymmetric algorithm encrypts with the public key and decrypts with the private key. After agreeing on the type of encryption to be used in the communication, the receiver node sends its public key; this key is used by sender node to encrypt the messages. Then, when the encrypted messages arrive, the receiver uses its private to decrypts it. Some of the algorithms inherit the features of these two categories, that is, may be symmetric- stream cipher e.g. RC4, symmetric-block cipher e.g. RC5, asymmetric-block algorithm e.g. 3DES and asymmetric-stream cipher algorithm. Some of these algorithms are briefly described below.

Rivest Shamir Adleman (RSA), uses a private and a public key to encrypt. Two large prime numbers are selected and then multiplied together, that is, n=p*q. If f(n) = (p-1) (q-1), and e>1 such that GCD(e, f(n))=1. Here will have a fairly large probability of being co-prime to f(n), if n is large enough and e will be part of the encryption key. If the Linear Diophantine equation is solved; ed congruent 1 (mod f(n)), for d. The pair of integers (e, n) is the public key and (d, n) form the private key. Encryption of M can be accomplished by the following expression; $M_e = q_n + C$ where $0<= C < n$. Decryption would be the inverse of the encryption and could be expressed as; Cd congruent R (mod n) where $0<= R < n$. RSA is the most popular method for public key encryption and digital signatures today [15]. The security of the system depends on the prime numbers p and q. It had been shown that the larger these primes, the more secure the system. The implementation of RSA uses 256-bit p and q. One of the weaknesses of RSA is that it is not easy to generate such large primes. And, it is very difficult to generate large prime numbers that most implementations of RSA use probable-primes, which are defined as numbers which are “probably prime [3, 15].

Advanced Encryption Standard (AES) is also an example of block data encryption algorithm with variable key size (128, 192 and 256). It performs data encryption on data blocks of 128bits in 10, 12 and 14 depending on the key size. Though, AES is fast and efficient its energy consumption is high especially if the key size is increased [7].

Another data encryption algorithm is RC6 which evolves from RC5. It is a block cipher, supports variable key (128,192 and 256 bits) and has a block size of 128 bits. However, its energy consumption is very high.

Blowfish is a 64 bits block data encryption algorithm with variable key size. It is one of the commonest public domain encryption algorithms. This algorithm is vulnerable to key attack using 234 chosen plain text [4, 7, 13].

Data Encryption Standard (DES) is an encryption algorithm which is based on IBM proposed algorithm called Lucifer. It is the first encryption standard to be recommended by National Institute of Standard and Technology (NIST).

International Data Encryption Algorithm (IDEA) was developed by Dr. X. Lai and Prof. J. Massey in Switzerland in the early 1990s to replace the DES standard. It uses the same key for encryption and decryption, like DES operating on 8 bytes at a time but it uses a 128 bit key. This key length makes it impossible to break by simply trying every key, and no other means of attack is known. It is a fast algorithm, and has also been implemented in hardware chipsets, making it even faster [15].

RC4 is a cipher invented by Ron Rivest. It is used in a number of commercial systems and wireless LAN. RC4 is most popular data encryption algorithm. It is used in Wired Equivalent Privacy (WEP) used by IEEE 802.11x standard in Wireless LAN. It is a symmetric stream cipher algorithm, which generate a pseudorandom stream of bits as a key. This is combined with the plaintext by finding the 2 modulus addition of the key stream and the plaintext. RC4 is simple and impressively fast. It is a cipher of up to 256bytes key size and creates a stream of random bytes and Xoring those bytes with the text. However, recent works had proved that RC4 is not highly secured whenever there is key repetition which can rarely be avoided.

This paper looks into the effect of mutual exclusiveness of XOR operator on the performance of RC4 and provides a novel method, called RC4c, which removes the vulnerabilities introduce by the mutual exclusiveness of the XOR functional operator. The remaining part of the paper is arranged as follows; section two contains the review of the related works on the performance evaluation of some of data encryption techniques. The crypto analysis of the effect of key repetition and mutual exclusiveness of XOR on RC4 are shown in section three. Section 4 contains the effects of key repetition and mutual exclusions of XOR on the new method and how the design space of the new method solves this vulnerability. In section 5, experimental evaluation of the RC4c and some other techniques is done. Finally the conclusion is drawn in section 6.

2. RELATED WORK

Many cryptoanalysis had been done on effect of different variables on the performance of data encryption algorithms such as the key length and file size as a function of the execution time, energy consumption. Some of these cryptoanalysis are described in this section.

RC4 is the fastest, commonest and simplest data encryption algorithm. It is used in some of the popular protocols such as secured socket layer which protects internet traffic and in WEP to secure wireless networks. Therefore, there is need for RC4 to be secured in all manners against unauthorized user of the system is protecting. Some authors have analysed methods of attacking RC4 [1, 7, 10, 12]. None of these attacks is practical against RC4 with a reasonable key length, such as 128 bits. A more serious attack is reported in [8, 12,14]. In [8], it was affirmed that the WEP protocol intended to provide confidentiality on 802.11wireless LAN networksis vulnerable to a particular attack caused by exclusive nature of the mixing operator (XOR). This is further shown in [5, 15]. The negative effect of the exclusiveness affects RC4 in WEP due to the way the keys are generated in the algorithm. This particular problem does not appear to be applicable to other applications using RC4 and can be remedied in WEP by countering the mutual exclusive nature of the XOR. The problem is well explained in the next section.

Omar and Adegoke in [11] describes RC5 as a parameterised symmetric block algorithm. It parameters are variable key size (k), variable block size (w) and a variable number of round (r). The RC5 algorithm uses three primitive operations and the inverse for encryption and decryption.

1 Addition/Subtraction of words modulo 2w, w is the word size

2 Bit-wise exclusive-OR.

3 Rotation; the rotation of word x left by y.

RC5 uses expanded key along with segments of the input message to produce its output. The RC5 is more secured than RC4 but is slower in operation which is due to key expansion. The possibility of key collision as it is on RC4 is drastically reduced in RC5 due to increased key size [11]. However, RC5 needs to read the expanded key, in a sequential way. This allows unauthorised users to tap and read the memory content if they have access to the system [11].

In [3] analysis of the effect of file size and key length on the speed of RC4 was done. It was discovered that the speed of RC4 during encryption and decryption is proportional to key length, data type and file size. It was shown that image encryption is slower than audio, text encryption and decryption. The result of their analysis was interpreted as mathematical equations which show the relationship between the examined parameters and the encryption and decryption time.

Meanwhile, in[14] a variant of RC4 called RC42s which engages 2's complement to solve the problem caused by the mutual exclusive nature of XOR was proposed. RC42's uses the same concept as the RC4. It is a stream key cipher which uses 2's complement to encrypted messages whenever there is key collision. The key stream of RC42's is absolutely independent of the plaintext used. It employs a variable length from 1 to 256 bit in order to initialize a 256-bit initialization vector (IV). The IV is used for generating subsequent pseudo-random stream key which is XORed with the plaintext in order to generate RC4 equivalent ciphertext. The 2's complement of the RC4 equivalent ciphertext produces RC42's encrypted text. However, the experimental result shows an average improvement rate of 68% over RC4. That is, 32% of encrypted messages may still be affected by key collision. Due to this 32% imperfection in RC42s, there is need for a more perfect technique which would be 100% secured in the cases of key collisions.

2.1 Effect of Key Collision and Mutual Exclusiveness of XOR Operator on the Security and Efficiency of RC4

The earlier researches on RC4 algorithm has shown that the use of 24-bit initializationvector(IV) is not adequate because the same initialisation vector will be reused over a period of time [9, 10, 12,14]. This is called collision of key; the hackers rely on this in order get the cipher key. Considering a network runing at 300 Mbps and 2000B packets.

$$Transmitted\ Packets\ per\ second = \frac{300Mbps}{2000bytes * 8} = 19661\ \ Packets\ per\ second$$

This shows that the network transmits 19661P/s. Since different initialisation vector must be appended to each packet, then time to exhaust all the generated IV is:

$$= \frac{2^{24}\ Possible\ IV}{19661\ packet\ per\ second}$$

$$= \frac{16777216}{19661\ packet\ per\ second} = 853\ seconds = 14\ minutes$$

This analysis shows that in every 14 minutes key collision occcurs. That is same key is used for the encryption. With this, it is assumed that only one device is connected. If more devices are connected using the same initialization vector IV, the time will be reduced.This means collision will occur in smallest interval and once there is collision the hacker is having two different plain texts both encrypted with the same key stream. Then, it is possible for the hacker to XOR these two encrypted plain text. The XOR of these plain texts will nullify the key stream thereby decrypt the encrypted text as briefly described below.

E_1 = Encrypted text$_1$
E_2 = Encrypted text$_2$
T_1 = Text$_1$
T_2 = Text$_2$
IV = Initialization Vector
K = Secret Key

$\oplus$ = XOR

$$if$$

$$E_1 = T_1 \oplus \text{RC4}(IV, K) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (1)$$

$$E_2 = T_2 \oplus \text{RC4}(IV, K) \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (2)$$

Then

$$E_1 \oplus E_2 = \left(T_1 \oplus \text{RC4}(IV, K)\right) \oplus \left(T_2 \oplus \text{RC4}(IV, K)\right) \dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

$$E_1 \oplus E_2 = \left(\text{RC4}(IV, K) \oplus \text{RC4}(IV, K)\right) \oplus \left(T_1 \oplus \text{T}_2\right)$$

$$since,\ \ \text{RC4}(IV, K) \oplus \text{RC4}(IV, K) = 0$$

$$Then, \quad E_1 \oplus E_2 = T_1 \oplus T_2 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

It is observed from equations (3) and (4) that if there is collision of key, the exclusive OR of the two encrypted messages will knock off the key leaving the exclusive OR of the two messages. Therefore, if a hacker has the inkling of one of the messages, he can decipher the other message. The proof from equation (1)-(4) is further illustrated in table I-III.

Table III shows that the exclusive OR of the two plaintexts is the same with exclusive OR of the two encrypted messages. This further establishes the equation 3 and 4. That is, if there is key collision and packets loss, hacker would have two encrypted texts both encrypted with the same secret key. If these encrypted messages are exclusive-Ored the secret key will be knock off leaving the XOR of the two plaintexts.Therefore, if a hacker has the knowledge of the content of one of the text messages, when collision occur the hacker could then decrypt the other encrypted messages, thereby, defeating the reason for encryption. This is the greatest weakness of RC4 as an encrypting algorithm. Most past cryptoanalysis had attributed this weakness to key collision which is brought by limited key size. On the contrary, key collision only exposes the weakness of the XOR functional operator used in stream cipher which is caused by its mutual exclusiveness.

Although, RC4 is fast, easy to implement but being easily susceptible to key collision makes RC4 ciphers to be more susceptible to attacks. Once the exclusive OR of two texts is obtained, the partial knowledge of one of the text leads attacker to decipher the other texts. This vulnerability alone is enough to look for replacement or improvement. RC5 uses added features like expanded key and rotation features to outsmart the mutual exclusiveness of the exclusive-OR thereby making RC5 more secured than RC4 but slower than RC4.

Table I: Encrypting Message 'A' with RC4 Technique

	Data
Letter "A" text$_1$ (T$_1$)	01000001
Letter "z" RC4 Key	01111010
XOR "A" with RC4 key (E$_2$)	00111011

Table II:Encrypting Message 'D' with RC4 Technique

	Data
Letter “D” text$_2$ (T$_2$)	01100010
Letter “z” RC4 Key	01111010
XOR “D” with RC4 key (E$_2$)	00011000

Table III: Effect of Mutual Exclusiveness of XOR
On RC4 Encrpted Message

	Data
$T_1 \oplus T_2$ **(D XOR A)**	00100011
$E_1 \oplus E_2$	00100011
$\mathbf{E1 \oplus E2 = T1 \oplus T2}$	Key Knock off

3.0 RC4c: THE IMPROVED STREAM DATA ENCRYPTION TECHNIQUE

The previous section has shown weakness of RC4 through stream key collision and the effect of exclusive nature of XOR which is the latent weakness of this algorithm.However, this section proposes anew approach to improve RC4 such that it overcomes the weakness caused by key collision and exclusive nature of the XOR operator without increasing the size of the initialization vector field, complexity of the algorithm or affecting the efficiency of the algorithm. The improved encryption algorithm is called RC4c, it uses the same principle as the RC42 proposed in [14] except that it incorporates shift left to further protect the key from being nullify by the mutual exclusiveness of the XOR operator anytime there is key collision. With RC4c, the message is exclusively OR with the stream key and two complement of the resulting encrypted message is taken.The encrypted message is shifted left once in order to shroud the secret key whenever there is key collision. The result is the RC4c encrypted message which will be transmitted wirelessly.

The RC4c technique is divided into four stages: Initialisation, Operation, De-exclusive Level 1 and De-exclusive Level 2 stage. In the initialization stage, the initialization vector is populated using the key as randomize seed, similar to RC4 and RC42s. The IV continues to modify in a regular pattern as the data encryption progresses as shown in the pseudo code below;

```
n = 0;
for (m=0; m<= 255; m++)
S[m] = n;
for (m=0; m<= 255; m++)
 n = (m + S[m] + K[m]) mod 256;
swap S[i] and S[j];
```

At the operation stage, similar to RC4, a stream of pseudo-random values are generated. The plaintext is XORed with these values bit by bit which gives the RC4 equivalent ciphertext. The operation section is summarized with this pseudo code;

```
n= m= 0;
for (k =0; k<= $P_n$-1; k++)
n = (n+1) mod 256;
m = (m + S[n]mod 256);
swap S[n] and S[j];
streamkey = S[(S[n] + S[m])mod 256]
RC4ciphertext = (P[k]) XOR (streamkey);
```

De-exclusive Level 1 section is the first level of improvement on RC4 algorithm. It nullifies the effect of the exclusive nature of XOR operator through collided stream key by generating the 2's complement of the output operation section. This resulted to RC42's ciphertext as summarized below.

RC42s_Ciphertext = (P[k] XOR (skey))' + 1;

De-exclusive Level 2 section is the second level of improvement on RC42s algorithm. This level further nullifies the effect of the exclusive nature of XOR operator on collided stream key by shifting right the RC42s ciphertext. This introduces multiplication operator which mop up the effect of mutual exclusiveness of XOR operator which De-exclusive could not do. This resulted to RC4c ciphertext as summarized below.

RC4c_Ciphertext = RC42's_Ciphertext **shiftleft** 1;

The resulted RC4c_Ciphertext will be transmitted wirelessly. At the receiving node, decryption takes place. The RC4c_Ciphertext is first shifted right, decreased by one, and de-complemented by inverting the result. Then the secret key stream will be exclusively OR with de-complemented message this will give the plain message. The added features neither incur extra cost nor affect the speed due to the fact that most hardware chipsets have in built shift right, shift left and 2s complement features.

3.1 Effect of Key Collision and Mutual Exclusiveness of XOR Operator on the Security and Efficiency of RC4c

Exclusive-Oring of RC4c encrypted messages cannot knock off the key even if an attacker has the partial knowledge of the message. This can be proved by using the previous procedure which shows thisweaknessin RC4 algorithm.

E_1 = Encrypted text_1
E_2 = Encrypted text_2
T_1 = Text_1
T_2 = Text_2
IV = Initialization Vector
K = Secret Key
$\oplus$ = XOR
[]'+1 = 2's complement

$$If\ \ E_1 = 2 * (\left[T_1 \oplus \mathrm{RC4(IV,K)}\right]' + 1) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$$E_2 = 2 * \left(\left[T_2 \oplus \mathrm{RC4(IV,K)}\right]' + 1\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Proof
Since

$$2 * (\left[T_1 \oplus \mathrm{RC4(IV,K)}\right]' + 1) \oplus 2 * \left(\left[T_2 \oplus \mathrm{RC4(IV,K)}\right]' + 1\right)$$

$$= 2 * \left[T_1 \oplus \mathrm{RC4(IV,K)}\right]' + 2 \oplus 2 * \left[T_2 \oplus \mathrm{RC4(IV,K)}\right]' + 2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(7)$$

This implies that,

$$2 * \left[(T_1 \oplus \mathrm{RC4(IV,K)})\right]' + 2 \oplus 2 * \left[(T_2 \oplus \mathrm{RC4(IV,K)})\right]' + 2$$
$$= 2 * \left[(T_1 \oplus \mathrm{RC4(IV,K)})\right]' \oplus 2 * \left[(T_2 \oplus \mathrm{RC4(IV,K)})\right]' + 2 \oplus 2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(8)$$

Therefore,

$$E_1 \oplus E_2 = 2 * \left[(T_1 \oplus \mathrm{RC4(IV,K)})\right]' \oplus 2 * \left[(T_2 \oplus \mathrm{RC4(IV,K)})\right]' \ldots\ldots\ldots\ldots\ldots.(9)$$

That is, $E_1 + E_1 \neq T_1 + T_2$

The result of exclusive-Oring of equation 5 and 6 shows that when there is key collision exclusive-oring of two encrypted messages will not knock off the secret key.This is further established in the next section; by using the RC4ctechnique on two messages ‘A’ and ‘D’ with the secret key is ‘z’.

4. RESULT AND DISCUSSION

The result of the analysis carried out in previous section, that is equations [3-4], shows that mutual exclusiveness of the XOR operator affects the security and performance of the RC4. However, equation 9 shows that the inclusion of 2s complement and shifting left operations in the new method (RC4c) completely counters the effect of the XOR mutual exclusiveness.

The way RC4c counters the effect of mutual exclusiveness is demonstrated further in Table [IV-VI]. Table IV (row 6) and table V (row 6) generate the RC4cE_1andE_2encrypted messages respectively. Table VI (row 2-4) diagrammatically explains equation (9). From these tables it could be observed that if there is secret key collision and exclusive OR of the encrypted messages are found, the secret key would not be nullified. Therefore, keep the encrypted message intact even if the hacker has the fore knowledge of any of the sent messages or texts.

Table IV: Encrypting Message 'D' with RC4c Technique

	Data
Letter “D” text$_2$	01100010
Letter “z” RC4 Key	01111010
XOR “D” with RC4 key	00011000
2’s of (D Xor z)	11101000
Shift Right once 2’s of (D XOR z) **(E$_1$)**	111010000

Table V: Encrypting Message 'A' with RC4c Technique

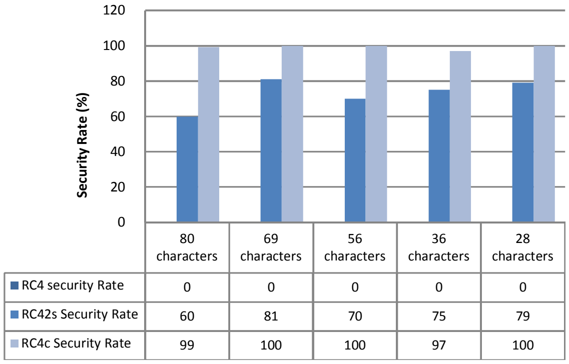
	Data
Letter "A" text$_1$	01000001
Letter "z" RC4 Key	01111010
XOR "A" with RC4c key	00111011
2's of (A Xor z)	11000101
Shift Right once 2's of (D XOR z) **(E$_2$)**	110001010

Table VI:Testing for Effect of Mutual Exclusiveness of XOR on Encrypted Messages

	Data
$T_1 \oplus T_2$ (D XOR A)	00100011
$E_1 \oplus E_2$	001011010
$E1 \oplus E2 \neq T1 \oplus T2$	Key Substained

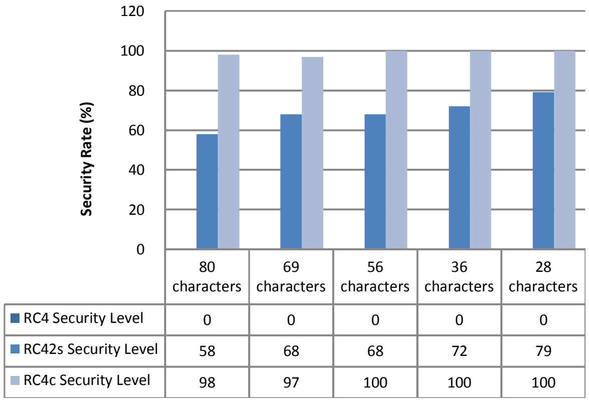
A simulator was developed using Borland delphi 7 to simulate the effect of key collision and mutual exclusion of XOR on RC4c, RC42s and RC4 encryption using all the 128 available ASCII characters. Different set of ASCII data were encrypted with the same key and mutual exclusion was tested by exclusive-oring two different encrypted messages. The percentage of encrypted characters not affected by key collision for RC4, RC42s and RC4c were plotted for the three different keys as shown in figure[1-3]. The results shown in figure [1-3] show that the inclusion of 2s complement makes average of 68% of RC42s encrypted messages secured. Leaving an average of 32% of the encrypted messages susceptible to unauthorised deciphering. Also the result shows that almost 100% of RC4c encrypted messages are secured, which coroborates equation 9. This is as a result of the added shift-left operation on RC4c which introduces multiplication operator. The multiplication operator eventually cancels the reminant effect of exclusiveness of XOR which 2s complement could not clear. The result also shows that 0% of RC4 encrypted messages are secured. That is, the exclusive nature of XOR operator through the key collision knocks off the encryption key in all the encryptions done using RC4 algorithm, thereby, leaving the messages unprotected. That is, equation (3) holds for any character encrypted with RC4 algorithm.

Also, six encryption algorithms were simulated and their throughputs were compared with that of the RC4c. From the result of the simulation shown in figure 4, RC4c has the highest throughput than DES, 3DES, and RC2 but almost the same throughput withRC4 and RC42s. This shows that the new encryption algorithm apart from solving the vulnerabilities of RC4 and RC42s still maintains their speed and throughput but with better efficiency.



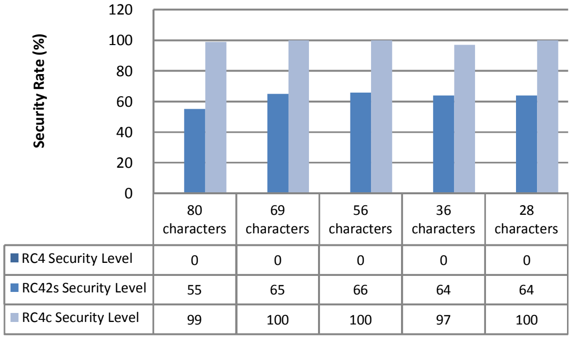
	80 characters	69 characters	56 characters	36 characters	28 characters
■ RC4 security Rate	0	0	0	0	0
■ RC42s Security Rate	60	81	70	75	79
■ RC4c Security Rate	99	100	100	97	100

Figure 1: Effect of Key Collision and Mutual Exclusiveness of XOR on Security Level of RC4,RC42 and RC4c with secret Key 'Y'



	80 characters	69 characters	56 characters	36 characters	28 characters
■ RC4 Security Level	0	0	0	0	0
■ RC42s Security Level	58	68	68	72	79
■ RC4c Security Level	98	97	100	100	100

Figure 2: Effect of Key Collision and Mutual Exclusiveness of XOR on Security Level of RC4,RC42 and RC4c with secret Key 'M'



	80 characters	69 characters	56 characters	36 characters	28 characters
■ RC4 Security Level	0	0	0	0	0
■ RC42s Security Level	55	65	66	64	64
■ RC4c Security Level	99	100	100	97	100

Figure 3: Effect of Key Collision and Mutual Exclusiveness of XOR on Security Level of RC4,RC42 and RC4c with secret Key 'c'

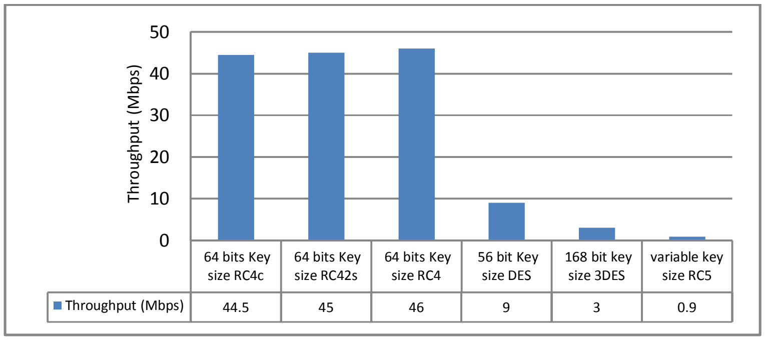


Figure 4: Throughputs of DES, 3DES, RC4, RC42'S and RC4c Data Encryption Algorithms

5. CONCLUSION

This paper presented RC4c, which is a modification to RC4 and RC42s data encryption algorithms for wireless data communication. RC4c is a stream cipher and performs encryption of data by finding the 2 modulus addition of data and the stream key. The 2s complement of the resulted 2 modulus addition becomes the encrypted data which is further shifted left once. The 2s complement partly nullifies the effect of the exclusive nature of the mixing operator (XOR) thereby solving the negative effect of key collision. The left shifting of the encrypted message mop up the reminant effect of the XOR exclusive nature, thereby, leaving a tamper proof encrypted messages.It solves the vulnerabilities of the RC4 and RC42s without compromising their efficiency and simplicity. It adds more security features to newly developed RC42's and increases its security level from an average of 68% to 100%.

It can be concluded that the RC4c maintains the throughput of RC4 and RC42s with no extra cost That is, the added 2s complement and shift left operations neither reduces the RC4c speed nor increase the cost because these two operations are inbuilt in the processing element of the most systems. Therefore, no need for additional hardware before RC4c can be implemented.

Also, the RC4c offers a perfect solution to vulnerability of RC4 thereby making it a more perfect and seamless replacement to RC4 than RC42s. This is corroborated in the first simulation result which shows that RC4c encrypted messages are 100% protected against the key collision and mutual exclusiveness of XOR functional operator compare to RC4 and RC42s which only gives 68% and 0% respectively. Therefore RC4c is key-collision-proof irrespective of the nature of the characters and the key in the encrypted messages.

REFERENCES

1. Abdel-Karim, A. (2006) 'Performance analysis of data encryption algorithms', available at http://www.cse.wustl.edu/~jain/cse567 06/encryptionperf.htm.
2. Allam Mousa and Ahmad Hamad (2006) 'Evaluation of the RC4 Algorithm for Data Encryption' , International Journal of Computer Science & Application.
3. Chanra, P. (2005) *Bulletproof Wireless Security*, pp.150–176, Elsevier Inc., 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA.
4. Coppersmith, D. (1994) 'The data encryption standard (DES) and its strength against attacks', *IBM Journal of Research & Development*, pp.243–250.
5. Dawson, E. and Nelson, I. (1996) 'Automated cryptoanalysis of XOR plaintext strings', *Proc. Cryptologia.*

6. Diaa Salama, A.E., Hatem, M.A. and Mohie, M.H. (2008) ‘Performance evaluation of symmetric encryption algorithms’, *International Journal of Computer Science & Network Security(IJCSNS)*, Vol. 8, No. 12.
7. Diaa Salama, A.E., Hatem, M.A. and Mohie, M.H. (2009) ‘Performance evaluation of symmetric encryption algorithms on power consumption for wireless devices’, *International Journal of Computer Theory & Engineering*, Vol. 1, No. 4, pp.343–351.
8. Kerry, S.J. (2005) ‘Chair of IEEE 802.11 respond to WEP security flaws’, available at http://slashdot.org/articles/01/02/15/1745204.shtm.
9. Lars, R.K., Willi, M., Bart, P., Vincent, R. and Sven, V. (1998) ‘Analysis method for alleged RC4’, *Conference Proceeding of ASIACRYPT*, pp.327–341.
10. Michael, S. (2002) ‘Hacking the invisible network insecurities in 802.11x’, iAlert White paper, pp.1–35, available at http://www.net-security.org/dl/articles/Wireless.pdf.
11. Omar, S.E. and Adegoke, O. (2008) ‘Performance comparisons, design and implementation of RC5symmetric encryption core using reconfigurable hardware’, *Journal of Computers*, Vol. 3,No. 3, pp.49–55.
12. Scott, R.F., Itsik, M. and Adi, S. (2001) ‘Weakness in the key scheduling algorithm of RC4’, *8th Annual Workshop in Selected Areas of Cryptography*.
13. Syed Zulkanain, I., Syed Alwee, A., Salina, M., Suhizaz, S. and Rbadlishan, A. (2008) ‘Performance analysis of encryption algorithms’ text length size in web browser’, *International Journal of Computer Science and Network Security (IJCSNS)*, Vol. 8,No. 1.
14. O.O Olakanmi, (2011) "RC42’s innovative way for data security in wireless data communication",*Int. Journal. of Information and Computer Security*, Vol. 4, No. 3, pp.264–275.
15. MyCrypto.net ' Encryption Algorithms' internet article available at http://www.MyCrypto.net

Biographical notes: Olakanmi Olufemi Oladayo received his B.Tech in Computer Engineering from Ladoke Akintola University of Technology, Nigeria in 2000. He started his career as a Lecturer in Computer Engineering Department, Federal Polytechnic Oko, Nigeria (2000 to 2007) before he later pursued his Master of Science in Computer Science from University of Ibadan, Nigeria. He consulted for RCM of Power Holding Company Nigeria (PHCN), Ijora District on Geographic Information System. Later, he moved to Electrical and Electronic Engineering Department, Federal Polytechnic Nekede, Nigeria as a Lecturer (2007). He is presently a Lecturer in Electrical and Electronic Engineering Department, University of Ibadan, Nigeria where he currently pursuing his Ph.D. His research areas include Data and Information Security, Distributed and Parallel Computing.