

PC1-RC4 and PC2-RC4 Algorithms: Pragmatic Enrichment Algorithms to Enhance RC4 Stream Cipher Algorithm

¹Pardeep, ²Pushpendra Kumar Pateriya

¹M.Tech Student (Computer Science), Lovely Professional University
Phagwara, India

²Assistant Professor (Computer Science), Lovely Professional University
Phagwara, India

ABSTRACT

This paper is introduced the new self enrichment stream cipher algorithms named as “PC1-RC4” and “PC2-RC4”, to improve the security of the RC4 algorithm and proposed to use over the internet Network and for internet applications like: E-Commerce Application where RC4 algorithm already applied for confidentiality. The RC4 stream cipher algorithm is most used algorithm to provide the confidentiality over the different networks. The different networks like: Sensor, wireless, Internet, Mobile and so on, used this algorithm for making secure the data during transmission. But the issue concern to RC4 algorithm is that RC4 algorithm is not secure. Lots of practical attacks and weaknesses detects over the RC4 stream cipher. So in this concern, this paper is working. And, proposed the new self enrichment algorithms to making secure the RC4 algorithm with breaking all the attacks over the RC4 algorithm and proposed to use over the internet network and internet application. The RC4 algorithm is two stages algorithm. The two stages are PRGA and KSA. Vulnerabilities are found in both the stages. So in these new self enrichment stream cipher algorithms “PC1-RC4” and “PC2-RC4” is based on new algorithm process inside the KSA and PRGA.

KEYWORDS:Stream, Cipher, RC4, PC1-RC4 (PardeepCipher1-RC4), PC2-RC4 (PardeepCipher2-RC4), Confidentiality

1. Introduction

Presently the area of networking and communication of computer science develops too widely. Different types of communication and networks use practically as per the different organizational and industry requirement, like: Wireless, Sensor, Internet and so on. The common concern regards these all the networks are Security. To making secure the secret data and information during transmission from unauthorized access. So, to solve this problem, Cryptography concept is available [12]. The concept of Cryptography provides the various algorithms

to confidentiality purpose. One of the most used algorithms is RC4 stream cipher.

The RC4 algorithm is most used stream cipher algorithm. Different protocol standard used this algorithm to making secure networks. Some of the names of the standard protocols are WPA (Wi-Fi Protected Access), SSL (Secure Socket Layer Protocol), and WEP (Wired Equivalent Privacy) [1]. Common discussion point of these all the standard protocol is that these all used the RC4 algorithm for confidentiality purpose.

RC4 algorithm was introduced by the Rivest in [years]. This is the Symmetric stream cipher Algorithm. This is the most used algorithm. RC4 steam cipher algorithm is providing the fast encryption and decryption, low resources utilization, easy to understand and implement, low time and space complexity as compare to other different algorithms [2]. The RC4 algorithm is two stages process, PRGA and KSA.

Presently, this algorithm is not secure. Number of attacks found over this algorithm from last decades. Lots of research announces this that the RC4 algorithm is not secure today and east to break [3, 4]. Second side the standard protocol is using this algorithm to confidentiality purpose. So in this concern, this paper has worked. This paper is introduces new self enrichment algorithms to making secure the RC4 stream cipher named as “PC1-RC4” and “PC2-RC4” stream cipher algorithms and proposed to used for internet network and internet applications like: E-Commerce internet application where RC4 algorithm already used. This algorithm is based in new encryption and decryption process, this paper is discussing this in the later section of the paper. This self

enrichment algorithm is based on different algorithm process for KSA and PRGA Stages.

2. RC4 Stream Cipher Algorithm

Pardeep and Pushpendra [13], RC4 stream cipher most preferred Stream cipher algorithm. In the RC4 algorithm, there are two stages process during encryption as well as decryption. The algorithm is dividing into the two parts KSA (Key scheduling Algorithm) and PRGA (Pseudo Random Generator Algorithm). KSA as the first stage of algorithm also known as initialization of S (s is state vector) and PRGA known as stream generation in the RC4 whole process of algorithm, mean RC4 basically two stages process.

In the first stages of RC4 Stream Cipher algorithm on the bases of variable sized key from 1 to 256 a State Vector (State Table) of fixed length 256 bytes is generated, after on the base of State Table, we generate the key stream that XOR with plaintext and cipher text during encryption and decryption. During encryption the key stream is XOR with the plaintext and during decryption the cipher text XOR with key stream then convert into the plaintext. In the description of RC4, first we discussing the first stage of the algorithm known as KSA, in this stage

Following steps are done

1. inputting the variable length key of size from 1 to 256
2. initialize the key matrix as per the size of the input key
3. Initialize the State table of fixed size 256 bytes from the value 0 to 255 in ascending order.
4. using the key matrix of variable size done the permutation on the S table
5. Output of the KSA, the final prepare S table after shuffling operation.

In this manner the KSA generate the State Table (State Matrix) of 256 bytes.

Now let's discuss the algorithm of the KSA as following

KSA

1. for $i=0$ to $N-1$
2. $s[i]=i$
3. $j=0$
4. for $i=0$ to $N-1$
5. $j=(j+s[i]+k[i]) \bmod N$
6. swap($s[i],s[j]$)

Now we going to discuss the second stager of the algorithm known as PRGA

These stages basically used to generate the output key stream that used to encrypt and decrypt the data by XORing operation.

The algorithm description the algorithm as following

PRGA

1. $i=j=0$
2. Loop
3. $i=(i+1) \bmod N$
4. $j=(j + s[i]) \bmod N$
5. swap($s[i],s[j]$)
6. output= $s[s[i] + s[j]] \bmod N$

From the last long time there are lots of weaknesses and attack to be found over the RC4 algorithm, some of the weaknesses detect in the KSA and some of the weakness is detect in PRGA of the algorithm.

3. Attacks on the RC4 Stream Cipher

In 1994, the RC4 algorithm was disclosed in to the market and then experts start to analyze the RC4 algorithm and find out the lots of weaknesses in both the stages of the algorithm KSA and PRGA. Many cryptanalysis of the algorithm was divided into the two parts, analysis of the initialization of RC4 which focuses on the initialization of KSA and analysis of the output key stream generation which focuses on the internal state and the round operation of PRGA [11].

Mantin and Shamir [7], was find out the weakness in the second round the probability of Zero output bytes as the major weakness of the algorithm.

Fluher et al. [8], was discovered the big weakness in the RC4, if anyone now the portion of the secret key than possible to attack fully over RC4.

Paul and Maitra [9], was discovered the secret key by using the initial state table. They generated some equation on the bases of initial state table and they select some of the bytes of secret key on the bases of guess and remain secret key find out by using the equation.

So we know that the security of RC4 depends on the security of the secret key and the internal states of S-box, so many attacks focus on resumming the secret key of the internal states of the S-box[10].

And also there is lot of other weaknesses and attacks are to be found over the RC4 algorithm. To making secure the RC4 that capable to stand against the attack, lot of research done over RC4 to enhancing the security of RC4.

4. Related Work

In this section, this paper describes the some proposed enhancement by the some researchers over the RC4 standard stream ciphers. This over all discussion

Pardeep and Pushendra [13], describes some enhanced Algorithm to making secure the RC4 standard Stream Ciphers.

Some of the proposed approaches as following:

FJ-RC4 [5]: The new developed approach based on the RC4 for the purpose to making strong the RC4 approach against the attacks. In this study shows the new KSA stage of the RC4 having vulnerable stage against the attack so in this study introduced new approach named as FJ-RC4 on the bases of the new developed KSA algorithm to making strong the stage against the attacks and also to the RC4 stream cipher. In this self developed new algorithm, FJ-RC4 is built from the new KSA, which uses the key stream in three stages process and shares the PRGA structure same as based on the previous structure of the PRGA of RC4, just one difference in PRGA stage of the FJ-RC4 algorithm, it is three stages encryption and decryption process but in the PRGA of RC4 having one stage encryption and decryption process.

Let's discuss the algorithm in detail

Key Scheduling Algorithm

In this KSA process of FJ-RC4 is introducing the new KSA algorithm to making strong the stage against the attacks. In this stage of FJ-RC4, the main key is further dividing in to the three equal portions in form of three sub keys. If the length of the main key is not divisible by the three for the purpose to divide it into the three different equal portions than Zero padding have done over the key to making it divisible by 3.

Above is giving the description of the

KSA Algorithm:

```
String key;  
String [] array = new String[3];  
int remain = 3 - (key.length() % 3);  
if(remain != 3)  
{  
    for(int i=0; i<remain ; i++)  
    {  
        key = key + "0";  
        Repeating 0 as necessary;  
    }  
}  
int temp = Key.length() / 3;  
array[0] = key.subsrting ( 0, temp);
```

```
Fill the first string array;  
array[1] = Key.substring( temp, temp+temp);  
Fill another array of the same size with the Key  
Array[2] = Key.substring9 temp +temp, Key.length());
```

During the encryption process, in the first stage, the plaintext XOR with the key stream generated on the basis of first key than in the second stage, the encrypted output of the of the first stage XOR with the key stream generated with the help of second sub key and then in the final stage, the second double encrypted message then again encrypted with the key stream generated on the basis of the third sub key.

In this study shows the new developed algorithm on the basis of RC4 is stronger than the previous RC4 against the attack, it takes more time to find out the key as compare to the RC4.

These approaches also require more resources than the RC4 and Key scheduling in FJ-RC4 slower then the RC4 because of the three key processes. But this algorithm proves to be more secure than RC4.

2. Improved RC4 Stream Cipher [6]: This Algorithm is work on the PRGA the second stage of the RC4 algorithm which is also detected to be vulnerable. To making secure the RC4 as well as to increase the security of the PRGA against the various attacks and to protect against the attack, in this study is introducing the new PRGA algorithm and shows the new enhancement over the RC4 to increase the security of the algorithm.

In this study basically discuss the one of the weakness of the PRGA is the relations between the S-boxes in different time. Many attacks tried to resume the initial state of the PRGA and achieved good efficiency. In this paper, mainly focus on the weakness of the PRGA and introduced the new improved RC4 to protect PRGA against the attack named as "An improved RC4 stream cipher".

Algorithm Description

In this improved rc4 algorithm researcher in the KSA stages generate 2 S-Boxes on the based of the two secret key secret key1 and secret key2.

KSA;

```
For i=0 to N-1  
{  
    s1 [i] = i;  
    s2 [i] = i;  
}  
j1=j2=0;  
for i=0 to N-1  
{  
    j1=(j1 + s1[i] + k1[i] ) mod N;  
    swap( s1 [i], s1 [j1] );
```

```

        j2=(j2 + s2[i] + k2[i] ) mod N;
        swap( s2 [i], s2 [j2] );
    }

PRGA;

I=j1=j2=0;
Loop
{
    i=i+1;
    j1=j1+s1[i];
    swap ( s1[i], s1[j1] );

    j2=j2+S2[i];
    swap (s2[i], s2[j2]);

    output= s1[ (s1 [i] + s1 [j1] ) mod N ];
    output= s2[ (s2 [i] + s2 [j2] ) mod N ];

    swap ( s1[s2[j1] ] , s1[s2 [j2]] );
    swap ( s1[s2[j1] ] , s1[s2 [j2]] );
}
    
```

and in the PRGA stage, two state boxes s1 and s2 are used to getting the output random key stream that used for the encryption and decryption. However, the elements of S-box are swapped only by a public pointer i and a secret pointer j+ s [i] in RC4, which leads to the relations between the states of S box. Therefore, our efforts focus on destroying the relations. In the improved RC4, we can get two secret parameters j 1 and j2 from the S boxes s 1 and s2 at the end of every loop. By the secret parameters j1 and j2, we can calculate four secret index-pointers s2 [j1], s [j2], sI [j1], s [j2]. The elements of s1 are swapped by the pointers s2 [j1] and s2 [j2] and the elements of s2 are swapped by the pointers sI [j1] and s [j2] at the end of every loop. As the index-pointers s2 [j1], s2 [j2], s1 [j1] and s1 [j2] are secret, which elements of sI and s2 are swapped is unknown. This is one of the innovations in this paper. More elements are swapped increases more confusion in the S-box. By this, the relations between the S-boxes are no longer existence in the improved RC4. The new algorithm enhances the security of the RC4 and it is faster than RC4.

RC4A Stream Cipher [14, 15]: This enhanced algorithm attempts to increase security without decreasing efficiency. RC4A stream cipher works in two phases, KSA (Key Scheduling Algorithm) phase and PRGA (Pseudo Random number Generation Algorithm) phase. During PRGA two successive output bytes are generated. The goal behind RC4A was to increase security primarily by increasing the internal complexity of the algorithm [9].

RC4A is made through improvement on the RC4, i.e., providing 2 S arrays (S1 and S2) that are Independent from each other and so that RC4A should not have bias in consecutive output byte. RC4A uses three counter i, j1, and j2. Variable j1 and j2 are introduced, corresponding to S1 and S2. In KSA for RC4A, like KSA of RC4, the array S1 is initialized, using the secret key K, Key stream, WK, are generated from the array S1 like PRGA (Pseudo Random number Generation Algorithm) of RC4. Then, like S1, the array S2 is initialized using WK. Unlike RC4 in PRGA of RC4A two successive output bytes are generated. All the arithmetic operations are computed Modulo N (N=256) [11].

Algorithm

KSA (K)

RC4_KSA(K,S1)

For i=0.....I-1

WK[i]=RC4_PRGA(S1)

RC4_KSA(WK,S2)

PRGA(S1,S2)

Initialization

i=0

j1=j2=0

Generation loop;

I=i+1

J1=j1+s1[i]

Swap(s1[i],s1[j1])

Output z1=s2[s1[i] + s1[j1]]

J2=j2+ s2[i]

Swap(s2[i] , s2[j2])

Output z12= s1[s2[i] + s2 [j2]]

5. PC1-RC4 Algorithm

5.1 Introduction

This is the first self designed algorithm, “PC1-RC4”, to enhance the security of the RC4 and PC-RC4”, stream cipher algorithm. This approach basically designed to making strong to the RC4 algorithm against the various attacks and weaknesses.

This algorithm basically based on the two way encryptions as well as decryptions. This algorithm provides the new algorithm process for both KSA and PRGA. This algorithm is basically design to making secure the large sized secret document during transmission in the E-Commerce application and mailing application.

5.2 Algorithm:

PC1_RC4

N=256

Shuffle function for swapping

KSA:

```

1.      Input single Key (Key Length)(base
key)
2.      Generate two sub keys
if(k1%2==1)
    {
        k1=k1+1;
    }
sub1=sub2=k1/2;
3.      initialize the two Key[length] //
generate on the bases of two sub keys
For i=0 to sub1
Random1[i]=random value;// (secret random1)
End for
For i=0 to sub2
Random2 [i] =random value ;/( secret random2)
End for
4.      Initialize the Two Temporary Matrix
For i=0 to N
Random_temp1[i]= value
Random_temp2[i]= value
STATE1[i]=i;
STATE2[i]=i;
End for
5.      Permutation on State Matrix
j1=j2=j3=0
For i=0 to N

J1=(j1+state1[i]+state1[j1]+random_temp2[i]+
random_temp2[j1]+random_temp2[j2])%N;

J2=(j2+state2[i]+state2[j2]+random_temp1[i]+
random_temp1[j1]+random_temp1[j2])%N;

```

shuffle(state1[i],state1[j1])

shuffle(state2[i],state2[j2])

End for

PRGA:

```

1.      Generate the random values used for
encryption
i=j1=j2=0
while(True)
i=i+1 % N
j1=j1+state1[i]+state2[j1]+
state2[j2] % N
j2=j2+state2[i]+state1[j1]+
state1[j2] % N

shuffle(state1[i],state1[j1])
shuffle(state2[i],state2[j2])

indx1=(state1[i]+state1[j1]) % N
indx2=(state2[i]+state2[j2]) % N
byte1= state1[indx1]

```

byte2= state2[indx2]

CT= PT1 XOR byte1

CT1= CT XOR byte2

Shuffle(state1[state2[i]],state1[state2[j1]])

Shuffle(state2[state1[i]],state2[state1[j2]])

Wend(End While)

5.3 Description

In this algorithm, in the KSA, first inputs one keylen (basic key) in between the size of 1 to 256. Then, this algorithm divides the single keylen into the two keylens (two basic keys). After this process, this algorithm generates the two secret keys as equal the length of the two subkeys which contain the random values. After generating the two secret keys, end for step is to generate the two random_temp1 [] and random_temp2 [] matrices of size 256 that contain random values on the bases of secret random1 and secret random2. End for step to initialize the two state matrix state1[] and state2[] of size 256 with inserting the values from 0 to 255 bytes in ascending order.

End for step is to done the shuffling on the both state matrix, where different process to be follow.

Let's take a look over these lines,

J1=(j1+state1[i]+state1[j1]+random_temp2[i]+

random_temp2[j1]+random_temp2[j2])%N;

J2=(j2+state2[i]+state2[j2]+random_temp1[i]+

random_temp1[j1]+random_temp1[j2])%N;

In this lines, this algorithm generates the confusion by generating j1 random index location pointer on the bases of random_temp2[256] that used to do shuffling over the state1[256] state matrix and provides the randomness on the index level by used the j1 and j2 at the index level. Same in this manner, shuffling process has done over the state2[256] state matrix. In this algorithm, used state[256] matrix to provide the randomness through using the state1[j1], state2[j1], state1[j2] and state1[j2] values. The output of the KSA is the random two state matrix state1 and state2.

This algorithm is also providing the different process for the PRGA. In this stage, the PRGA stage is having 2 state matrices of random values from 0 to 255 bytes values. Then In this stage, on the bases of this two state matrices, generated j1 and j2, two index location indicators, on the based attempt the shuffling and further output byte generated byte1 and byte2, use to two times encryption on the plaintext.

In this PRGA, this algorithm provides the tough level of randomness by using the state2[j1], state2[j1] with relate to the state1[] and using state1[j1], state2[j2] with relate to the state2[].

And also in the end of the PRGA algorithm, is adding additional shuffling in inside the PRGA to create the more confusion inside the PRGA algorithm. By using the

following lines have to done shuffling over the state matrix.

Let us take a look on these lines

Shuffle(state1[state2[i]],state1[state2[j1]])

Shuffle(state2[state1[i]],state2[state1[j2]])

5.4 Encryption/ Decryption Process of PC1-RC4 algorithm

This algorithm is based on the two stages encryption and decryption.

Let us understand through diagram

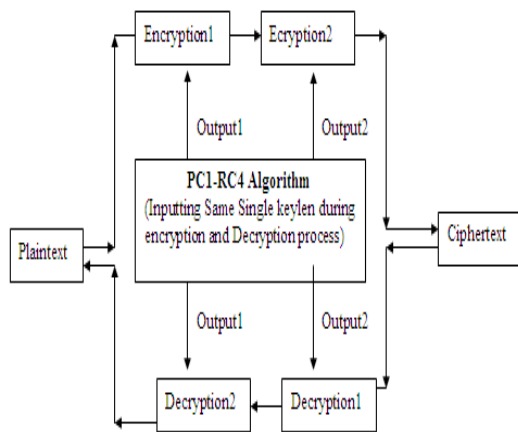


Figure 1 PC1-RC4 Encryption and Decryption process

In Figure1, show the encryption and decryption process does by the PC1-RC4 algorithm. During encryption and decryption process, same input keylen (base key) inputs to PC1-RC4 algorithm. Then, on the bases of the keylen algorithm does its corresponding working and generated the two output bytes, output1 and output2, as given in the algorithm. In the encryption process first data encrypted by the first generated output key stream value via XOR operation then encrypted data in the end for second encryption process again encrypt by the second output key stream bytes. Then finally, we have the cipher text, same process to be follow in decryption of data as shown in the diagram.

5.5 Weakness

The biggest weakness in this algorithm PC1-RC4, is generate the sub keylens subkey1 & subkney2 (two sub basic keys) from the main Keylen (main basic key), his point is become the weak point of this algorithm. Because in this case, the two sub keylens always having maximum

size just 128, because the mean keylen is input in between the size of 1 to 256.

For example: let us suppose the main keylen inputs 128, then we have to divide into two lengths to generate the two sub keylens, then after division, the resulting sub keylens are 62 & 62.

So this case, the size of two sub keylens (base subkeys) always generate in between the 1 to 128, mean when after generated the secret keys and state matrix contain no more randomness than required. So this point becomes the weak discussion point of the algorithm.

6. PC2-RC4 Algorithm

6.1 Introduction

This self design algorithm basically has designed to remove the weakness of the PC1-RC4 self design algorithm.

Basic discussion of this algorithm same as the PC1-RC4 algorithm, it inherits all the properties and process of encryption and decryption almost same as the PC1-RC4 algorithm.

6.2 Algorithm:

PC2-RC4

N=256

Shuffle function using for Swapping

KSA:

1. Inputting two Keys (k1 & k2) (Key Lengths)(base keys)
2. initialize the two Key[length] // generate on the bases of two sub keys
 For i=0 to k1
 Random1[i]=random value;// (secret random1)
 End for
 For i=0 to k2
 Random2 [i] =random value ;//(secret random2)
 End for
3. Initialize the Two Temporary Matrix and State Matrix
 For i=0 to N
 Random_temp1[i]= value
 Random_temp2[i]= value
 STATE1[i]=i;
 STATE2[i]=i;
 End for
4. Permutation on State Matrix
 j1=j2=j3=0
 For i=0 to N
J1=(j1+state1[i]+state1[j1]+random_temp2[i]+random_temp2[j1]+random_temp2[j2])%N;

```
J2=(j2+state2[i]+state2[j2]+random_temp1[i]+
random_temp1[j1]+random_temp1[j2])%N;
```

```
shuffle(state1[i],state1[j1])
shuffle(state2[i],state2[j2])
```

End for

PRGA:

```
5. Generate the random values used for
encryption
i=j1=j2=0
while(True)
i=i+1 % N
j1=j1+state1[i]+state2[j1]+state2[j2] % N
j2=j2+state2[i]+state1[j1]+state1[j2] % N
shuffle(state1[i],state1[j1])
shuffle(state2[i],state2[j2])
indx1=(state1[i]+state1[j1]) % N
indx2=(state2[i]+state2[j2]) % N
byte1= state1[indx1]
byte2= state2[indx2]
CT= PT1 XOR byte1
CT1= CT XOR byte2
Shuffle(state1[state2[i]],state1[state2[j1]])
Shuffle(state2[state1[i]],state2[state1[j2]])
Wend( End While)
```

6.3 Description

The description of this algorithm is same as the PC1-RC4 algorithm. In this algorithm, just based on the two keylens (two basic keys) is input from I to 256 sizes. On the bases, further algorithm generates the two output bytes and done two stages encryption and decryption.

Remain of the discussion, how generates the secret key, randomize the state matrix and the algorithm process of KSA and PRGA also do in same manner as PC1-RC4 algorithm.

6.4 Encryption/Decryption Process

The encryption and decryption in this algorithm does same as PC1-RC4 algorithm. Data processes two times during encryption and decryption.

But in this algorithm, the encryption and decryption has done on the bases of two key2.

Let's understand through diagram

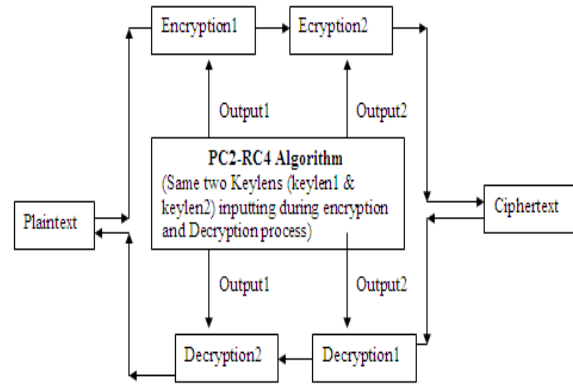


Figure 2: PC2-RC4 Encryption and Decryption Process

In figure2, shown the encryption and decryption process of PC2-RC4 algorithm. In this algorithm, two different keylen1 and keylen2 as base key inputs, on the bases encryption does in two stages.

6.5 Strong and Weak point

This algorithm is providing more security then the standard RC4 and other two self define algorithms, RC4, PC1-RC4. This algorithm is also removes the weakness of the PC1-RC4 algorithm. This algorithm uses two keys mean two keylen as basic key. At this point, when data encrypted, then receiver side during decryption requires the two keys mean keylens, which used during encryption at sender side, mean needs to distributing the two keys. So in this algorithm, we have to distributing the 2 keys, mean to say some task increases at key management process. We have to manage and distribute the two keys in this algorithm. If we just have to concern with the security and having good key management system, then this algorithm is a good option to use then RC4 and PC1-RC4.

7 Simulation Result and Discussion

In this section of the paper, we go to discuss the result and simulation of the Standard RC4 Algorithm and the two proposed algorithm. The implementation of all the algorithms has performed in the java programming. Java code writes in the Eclipse java editor and java application implementation tool.

7.1 Implementation Output Result

In this Section, describes the Encryption and decryption output of RC4 and Both the Proposed Algorithms the algorithm. The output image of all the algorithms are showing the inputting plaintext and keylen in between 1 to 256 and then plaintext convert into the ciphertext during

encryption process and ciphertext convert back in to plaintext during decryption process.
 Let's see the implementation output of all the Algorithms:

7.1 1. RC4 Algorithm Output:

```

enter key b/w 1 to 256 100
plaintext is =pardeppushpendrapramodkumarji
cipher text is =
?*J?+ dB $vãf0j%0x0?0zf{áWú±;
nano time 96799

Used memory is bytes: 209856
plaintext is pardeppushpendrapramodkumarji
    
```

Figure3: RC4 Output Snapshot

7.1.2 PC1-RC4 Algorithm Output:

```

input the key : 100
Plaintext is =pardeppushpendaraparmodkumarji
Ciphertext is =)A%lx@5?Áðu0eQ@
[=¥]l&Gkðü0Û=
nano time 208804

Plaintext is =pardeppushpendaraparmodkumarji
    
```

Figure4: PC1-RC4 Output Snapshot

7.1.3 PC2-RC4 Algorithm Output

```

inputting two keys: 100 100
Plaintext is =pardeppushpendrapramodkumarji
Ciphertext is =?g0%q5@)Rè"j8RÛèç?^00d 0;?9
nano time 206058

Plaintext is =pardeppushpendrapramodkumarji
    
```

Figure5: PC2-RC4 Snapshot

7.2 Analysis Result

In this section, discuss analysis of RC4 and both the algorithms on the based of below define parameters. We collecting the data and performs the analysis on the collected data and final the resulted data and then prepare the graphical simulation result on each parameter analysis. This analysis has done on the following parameter bases.

7.2.1 Encryption Time of Algorithms

7.2.2 Memory Utilization of Algorithms

7.2.1 Encryption Time of Algorithms

In this analysis, we describe the encryption time in Nano Second taken by all the algorithms during encrypt the different sized data. To done this analysis, first write the code of the entire algorithm in java, where eclipse java editor tool used. And to calculate the time, write the code in java and calculate the encryption time in Nano Second.

7.2.1.1 RC4 Algorithm

Keylen=100

Table 1: RC4 Encryption Time

Data Size	Encryption Time
100	132139
200	202508
300	227928
400	333952
500	390177

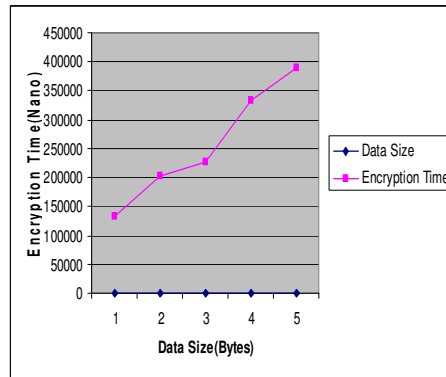


Figure 6: RC4 Encryption Time

7.2.1.2 PC1-RC4 Algorithm:

Keylen=100

Table 2: PC1-RC4 Encryption Time

Data Size	Encryption Time
100	293290
200	441560
300	566785

400	666360
500	896660

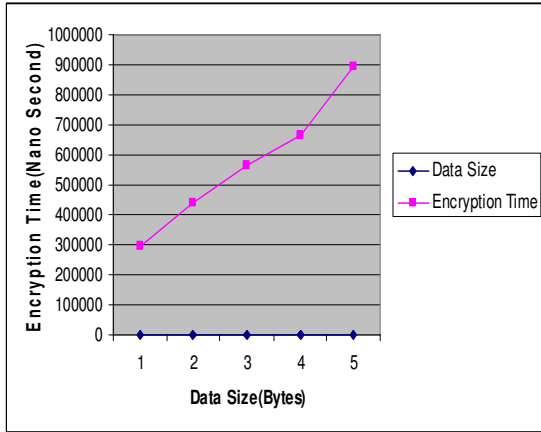


Figure 7: PC1-RC4 Encryption Time

7.2.1.3 PC2-RC4 Algorithm

Keylen1=Keylen2=100

Table 3: PC2-RC4 Encryption time

Data Size	Encryption Time
100	281360
200	453344
300	577590
400	686855
500	905750

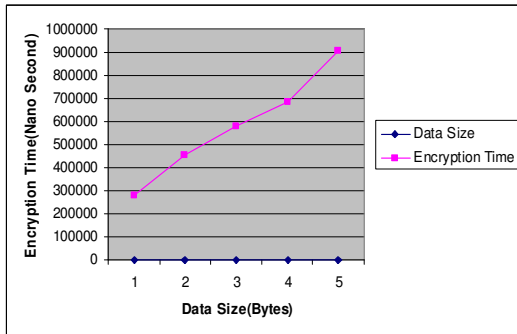


Figure 8 PC2-RC4 Encryption Algorithms

7.2.2 Memory utilization of Algorithms

Now we go to discuss another analysis parameter to analyze RC4 and both the algorithms. Now we discuss the

memory utilization of all the algorithms during encryption of data.

To calculate the memory utilization of the algorithms, we take the keylen=100 and size of the data is 100 bytes.

Table 4: Analyze data

Name of algorithm	Memory(Bytes)
RC4	210064
PC1-RC4	211160
PC2-RC4	211304

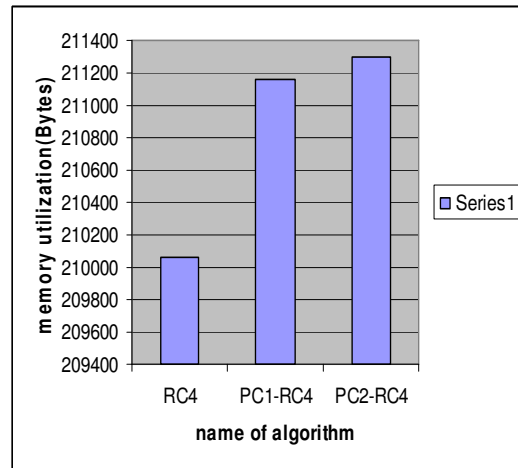


Figure 9: Memory Utilization of the Algorithms

Analysis Conclusion: In this analysis, this paper describes the encryption time and memory utilization during encryption of different data size of the algorithms. As per the result, still both the proposed algorithms taking more time and more memory utilization than standard RC4 algorithm. But second side the main concern of this paper just to improve the security of the RC4 algorithm and remove all the weak point of the algorithm, this paper work in this manner. Second side RC4 used over the different types of networks previously discussed, this proposed algorithm has proposed from the Internet application point like E-Commerce application, where just demanding the security and second side no any lack of resources, just having demand of security. In this manner, this paper try to enhance RC4 algorithm to complete the demand of this types of application domain where already use the RC4 algorithm.

Second side, obvious this also the strong fact, if we want to improve the security of any algorithm, then we have to use some additional rules and changes, on the based we enhance any algorithm. And also this additional rules and

changes can be effect performance of the focused algorithm.

8. Conclusion

So in this manner, this paper introduced the two enhanced RC4 approaches PC1-RC4 and PC2-RC4, which making strong the RC4 algorithm and remove all the vulnerable point of the RC4 algorithm and proposed to use over the internet application where RC4 standard algorithm already used. This paper describes the proposed algorithm in the explored manner and also showing the output of the algorithms and simulation result. As per the result, the proposed algorithm utilizing more time and memory then RC4, but the purpose of this paper, to making secure the RC4 algorithm and design to proposed over the internet network and internet applications concern where no any lacking of resources just the main security demanded like: E-Commerce Application over the internet. The proposed algorithm encryption, decryption and working almost same but the difference is the key. PC1-RC4 based on single keylen and PC2-RC4 based on two keylen. But PC1-RC4 having some weakness, which removes by the second proposed algorithm, but in the second algorithm, we have to manage the two keylens on the application network. This make key distribution complex but if we have no any problem with key distribution, we have well key distribution system then PC2-RC4 best to use then PC1-RC4. But both the approaches make the RC4 algorithm strong.

REFERENCES

- [1]A.A. Noman, Dr. Roslina b. Mohd. Sidek, Dr. A.R.b. Ramli, Dr. L. Ali, "RC4 Stream Cipher for WLAN Security: A Hardware Approach", 5th International Conference on Electrical and Computer Engineering, ICECE, 2008.
- [2] Chuan-Chin Pu, Wan- Young Chung, "Group Key Update Method for Improving RC4 Stream Cipher in Wireless Sensor Network", International Conference on Convergence Information Technology, 2007.
- [3] Suhaila Omer Sharif, S.P. Mansoor, "Performance analysis of Stream Cipher algorithms", 3rd international conference on Advanced Computer Theory and Engineering (ICATE), 2010.
- [4] C.S Lamba, "Design and Alnalysis of Stream Cipher for Network Security ", Second International Conference on Communication Software and Networks, 2010.
- [5]Fahime Javdan Kherad, Mohammad V. Malakooti, Hamid R. Naji, Payman Haghinghat, "A New Symmetric Crptographic Algorithm to Secure E-Commerce Tracsactions", International Conference on Financial Theory and Engineering, 2010.
- [6]Jian Xie, Xiaozhong Pan, "An Improved RC4 Stream Cipher ", International Conference on Computer Application and System Modeling (ICCSM), 2010.
- [7] I. Mantin, A. Shamir, "A practical Attackon Broadcast RC4", FastSoftware Encryption 2001 (M.Matsui,ed.), pp. 152-164, Springer-Verlag, 2010.
- [8] S.Fluthrer, I. Mantin, A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", SAC2001 (S. Vaudenay, A. Youssef,eds.), col. 2259 of LNCS, pp. 1-24, springer-Verlag, 2001.
- [9]G. Paul, S.Maitra, "RC4 state in formation at Any Stage Reveals the Secret Key ", In proceedings of SAC2007,2007, <http://eprint.iacr.org/2007/208.pdf>.
- [10]A. Klein, Attacks on the RC4 Stream Cipher, <http://cage.ugent.be/-klein/RC4/RC4-rn>.
- [11] S. Paul, and B. Preneel, "A New Weakness in the RC4 Keystream Generator", Fast Software Encryotion, FSE 2004, LNCS 3017, 245-259, Springer- Verlag, 2004.
- [12] Atul Kahate ,"Cryptography and Network Security" , pp- 123-125, 2008.
- [13] **Pardeep and Pushpendra**, "A Pragmatic Study on the Different Stream cipher and Different Flavor of RC4 Stream Cipher", March 2012.
- [14]A.A. Noman, Dr. Roslina b. Mohd. Sidek, Dr. A.R.b. Ramli, Dr. L. Ali, "RC4 Stream Cipher for WLAN Security: A Hardware Approach", 5th International Conference on Electrical and Computer Engineering, ICECE 2008.
- [15] S. Paul, and B. Preneel, "A New Weakness in the RC4 Keystream Generator", Fast Software Encryotion, FSE 2004, LNCS 3017, 245-259, Springer- Verlag, 2004.

AUTHORS



Pardeep is pursuing his Master of Technology in Computer Science and Engineering degree, from Lovely Professional University Phagwara.

Pardeep obtained his Master of Science in Information Technology degree, from Punjab Technical University Jalandhar in 2008 and later obtained the Master of Computer Applications degree, from Punjab Technical University Jalandhar in 2009. Pardeep's Research interests include Network security and Cryptography, Grid and Cloud Computing, Selective Image Encryption.



Pushpendra Kumar Pateriya is an Assistant Professor in the Department of Computer Science at Lovely Professional University, Phagwara. Pushpendra's Research interests include Cloud Computing, Grid Computing, Network Security and Cryptography, Image Processing, Selective Image Encryption, Software Engineering.