ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Robustness of RC4 against Differential attack

Bhargvi H. Kamble[*1], Dr. B. B. Meshram[2]
[1]*Computer Department,V.J.T.I, Matunga, Maharashtra, India*
[2]*Computer Department,V.J.T.I, Matunga, Maharashtra, India*
bhargavihkamble@gmail.com[1]

Abstract: **Cryptanalysis of ciphers is a technique used to find flaws in the cryptographic algorithm and try to overcome these flaws to give much better security. There are many cryptographic algorithms such as DES, AES, RSA etc. Rc4 is a stream cipher that is used on a large scale in wireless networks and IEEE standards. Many attacks have been tried over RC4. In this paper we will see what is RC4, how encryption is done, a differential attack on rc4 and a method or the changes required to the cryptographic algorithm to secure this cryptographic algorithm against differential attack.**

***KeyWord*: Cryptography, Encryption, Cryptanalysis, Robustness**

I. Introduction

Cryptography is a technique where plain text is converted into a scrambled code and sent over the network so that even if an anonymous user gets the scrambled data he cannot retrieve the meaning of that data. There are people who are interested in the transferred data for using the data for some wrong reasons. There are many type of attacks that take place. Some are just to retrieve the data and some even change the data and forward the wrong data. These attacks are defined as passive attacks and active attacks respectively. Passive attacks are hard to identify. In case of active attacks these attacks try to change the data and these attacks can be identified easily using hamming code.
Identification of attacks is not sufficient. There must be steps taken to avoid the attacks on the algorithm. That means the algorithm should be made robust against the identified attack.
The paper is organized as Section I consists of the basics of RC4 algorithm that is how Rc4 algorithm works. Section II consists of Differential attack on Rc4. And Section III consists of how the Rc4 algorithm is made robust against Differential attack.

II. RC4 Encryption

RC4 is a stream cipher which comes under additive stream ciphers[4]. It uses a variable-sized key ranging from 1 to 256 bytes. The parameter 'n' is the word size for the algorithm. In most applications, 'n' is chosen to be 8. The internal state of RC4 consists of a table, S-box i.e. referred 'S', of size 2^n words and two word-sized counters, i and j.
The encryption process consists of two functions namely *Key Scheduling Algorithm*(KSA) and *Pseudo Random Generation Algorithm*(PRGA).

Key Scheduling Algorithm (KSA)
It consists of two phases: initialization phase in which S is set to the identity permutation, and mixing phase in which it uses a key (K) with L bytes long to continuously swap values of S to produce new unknown key dependent permutations. As the only action on S is value swapping, S always contains a permutation.

```
j=0,
For i=0 to 255
S[i]=i;
For i=0 to 255
{
j=j+S[i]+K[g]…g=i mod L;
Swap(S[i],S[j]);
}
```

Pseudo Random Generation Algorithm (PRGA)
It continuously shuffles the permutation stored in S and picks up a different value from the S permutation as output. One round of the cipher outputs an n-bit word as the key stream.

```
i=0,j=0
i=(i+1)mod 255,
j=(j+S[i])mod 255
Swap(S[i],S[j])
t=(S[i]+S[j]) mod L
Output S[t]
```

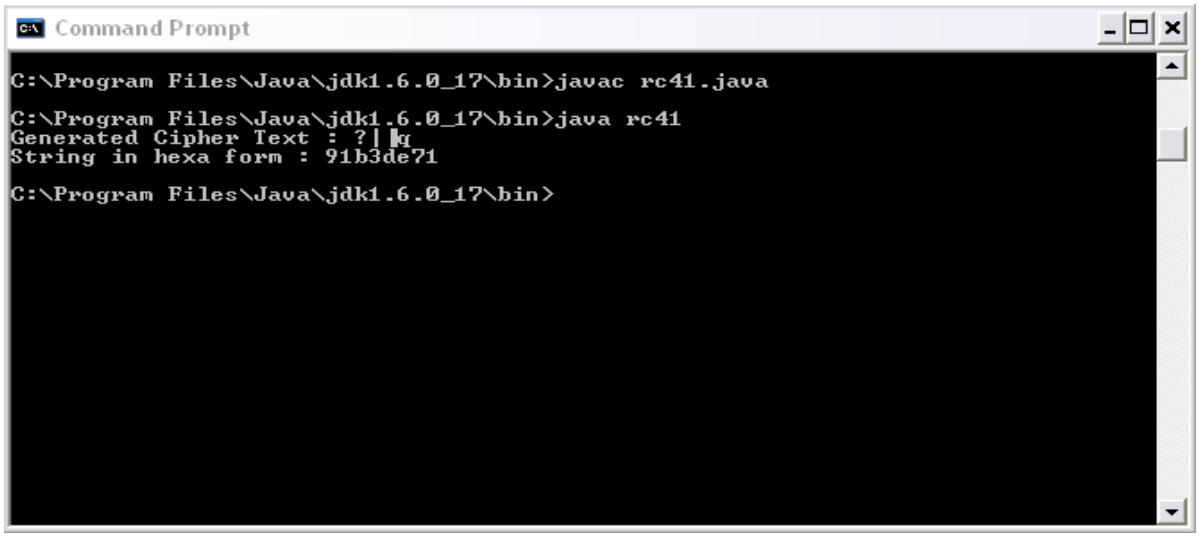

Fig 1. Encryption

The above key scheduling and key stream generation algorithms,
produces following characteristics.
• S is an identity permutation initially, of the 2^n possible words and remains a permutation throughout.
• Knowing the internal state of the cipher at a given time is sufficient to predict all the key stream bits in the future and so to break the cipher.
• As the permutation of S depends completely on K, knowing K can break the cipher.
• The period of the output key stream depends only on K, which is normally very long and hard to predict.

III. Cryptanalysis of RC4

Cryptanalysis is a technique where plain text is retrieved from the cipher text without actually knowing the necessary key to decrypt the cipher text. This is the way how attacker try to deduce the plain text. But in cryptanalysis a cryptanalyst tries to deduce the plain text to find the flaws in the cryptographic algorithm and to improve the cryptographic algorithm

In cryptanalysis of stream ciphers, it is common to assume either that:-

• some part of plaintext is known, (known plaintext attack), or
• plaintext has redundancy (e.g., has ASCII format).

For additive stream cipher, a known part of plaintext is equivalent to a known part of key stream. Some of the various attacks are as follows:-

1. Key recovery attack: Attempt to recover secret key *K* out of observed key stream
2. Distinguishing attack: Try to distinguish observed key stream from being a purely random sequence
3. Fast correlation attack: Significantly faster than exhaustive search over all initial states of target LFSR. Based on using certain parity check equations created from feedback polynomial of LFSR.

Differential Attack on RC4

Differential attacks on stream ciphers has gained popularity as stream ciphers are considered more powerful than block ciphers. This section explains the differential cryptanalysis of rc4 using a simple 8-byte representation. Consider two keys which have good differentials.
The difference between the two keys is in the last byte, so that after the initialization, the two

internal states differ in three bytes. So in this case, the output streams are expected to be the same in the first few bytes[2]. In this type, the attacker has a knowledge of frequently used

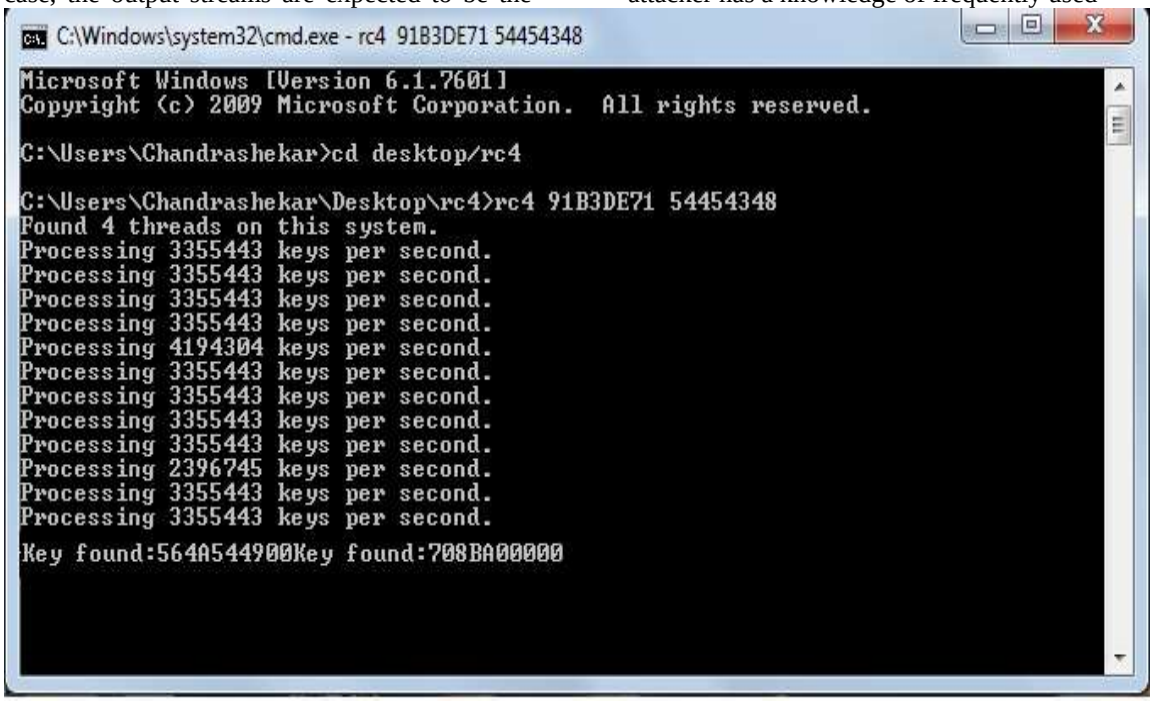

Fig 2. Cryptanalysis

plaintexts. But, it is seen that if 256-bit keys are used , the probability of same internal state being generated is $1/2^{112}$ [2].

Consider an 8-byte state and 8-byte key for convenience. Here L=4.

Let a random key chosen be {4,5,7,1,3,0,2,6}.Also consider a plain text "HI". The steps for cryptanalysis is as follows:-

Step1:After applying the RC4 encryption algorithm, the 1st key stream generated is "2(0000 0010)" and the 2nd key stream byte generated is "0(0000 0000)".The respective key stream bytes are XOR' ed with 'H' and 'I' respectively producing the cipher text. The 2 output bytes can be shown as follows:-
i=1,j=0
s=[0 4 7 6 2 5 3 1]
s[4]=2 …..**key stream1**
i=2,j=7
s=[0 4 1 6 2 5 3 7]
s[0]=0…...**key stream2**

Step2: Now, it is known that the difference in the last two bytes of key produces the same key stream in the first few bytes. The keys are guessed randomly to an unsuccessful attack. When the key is chosen to be {4,5,7,1,3,0,3,5},we get the following PRGA output:-

i=1,j=0
s*=[0 4 7 6 2 3 5 1]
s*[4]=2….key stream1(new)
i=2,j=7
s*=[0 4 1 6 2 3 5 7]
s*[0]=0…..key stream2 (new)

Step 3: From the known plain texts, the attacker uses them to XOR them with new key streams. Once the match is found, attacker can then guess the original internal states as s and s* differ by two bytes. The rest of the internal states can be calculated as it is algorithmic.
Hence with an 8-byte RC4,we have recovered the plain text using differentials in the key.
In real time applications,256-bit keys are used. For a 256-bit key, we consider two 256-bit(32-byte) keys K and K^1.
Now, K^1[i]=K[i]..for 0 to 29
=K[i]+1..i=30
=K[i]-1..i=31

The above equations was suggested as in[2,3]. K^1 is chosen to reflect the changes in K. The initial 30 iterations of the initialization remain the same. If j is directly updated to 30 when i=30

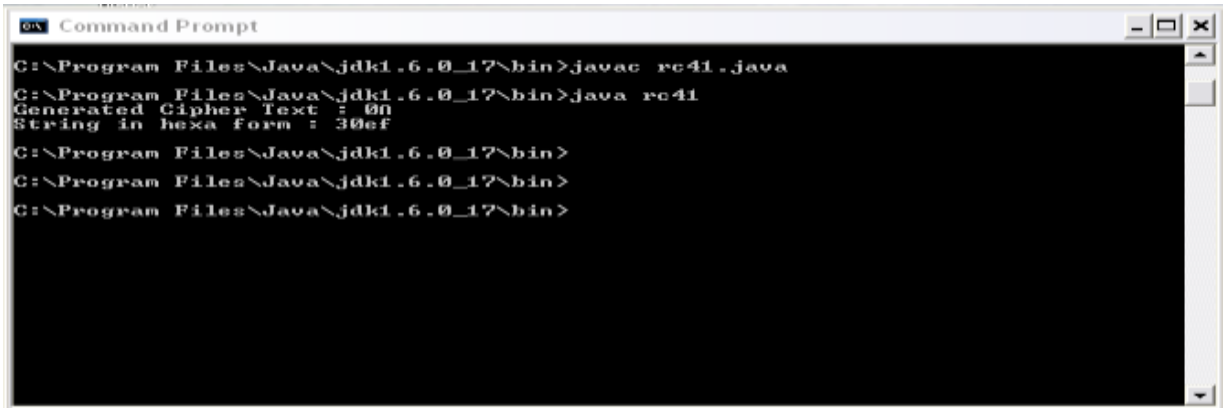

Fig 3. Encryption

Fig 4. Robust RC4

instead of the algorithmic value, j^1 is updated to 31. This difference causes changes to internal states S and S^1 as they differ by bytes 30 and 31. S is not swapped but S^1 30 and S^1 31are swapped. Secondly if i=31 and j=31, j^1 is updated to 30 which causes the second swap operation between S^1 30 and S^1 31.This leads to the same internal state again. It can be said that for a 256-bit key,RC4 has two states same after 32 iterations. The probability is given by $1/2^{16}$ [2].

Once the keys have been guessed, the output key stream bytes can be found out if the number of iterations are known. Each byte of key stream can be XOR'ed with the cipher text to find the plaintext, which is the motive of this attack

IV. Robustness of RC4

Using Differential attack we were able deduce the plain text. In the above attack we can see that though we have got a key that differ slightly (by two bytes) we are able to retrieve data that is 90% approximate using differential attack. Due to this the security provided by RC4 reduces. So to increase the robustness of RC4 we have to make some changes in the algorithm. In this case rather than making any changes in the actual algorithm we are going to put some restrictions on the key usage. When the differential attack was implemented on different key size it was observed that plain text that is

derived completely matches for 32 bit key size. It partially matches to 24 bit key size.

So to avoid Differential attack on RC4 we will have to put restrictions on the key size. We can set key size to greater than 32 bit or less than 24 bit. This improves our RC4 algorithm against differential attack

In fig 4 we can see that the plain text that is derived doesnot match the entered plain text in fig 3. Thus our RC4 has become ribust against differential attack.

V. Conclusions

In this paper we have seen how RC4 is implemented and how the plain text can be retrieved using differential attack. We have also seen how we can take steps against differential attack. We have made RC4 robust against Differential attack. Future work is to see different attacks that can be implemented on RC4 with lager key size.

VI. Refrences

1. Scott Fluhrer, Itsik Mantin, and Adi Shamir "*Weaknesses in the Key Scheduling Algorithm of RC4"*
2. Eli Biham, Orr Dunkelman "*Differential Cryptanalysis in Stream Ciphers*"
3. Alexander L. Grosul, Dan S. Wallach, "*A Related-Key Analysis of RC4*",Rice University technical report TR00-358, 2000.
4. "*Stream Ciphers*", Found online at www.nku.edu/~christensen/**Stream**%20**ciphers**.pdf
5. Effective uses of fpgas for brute-force attack on rc4 ciphers. Sammy h. M. Kwok and Edmund y. Lam
6. Di?erential cryptanalysis in stream ciphers. Eli biham and Dunke lman
7. Impossible fault analysis of rc4 and di?erential fault analysis of rc4. Eli Biham, Louis Granboulan and Phong Q. Nguy
8. Modern optimisation algorithms for cryptanalysis Andrew clark
9. A distinguishing attack on a fast software-implemented rc4-like stream cipher. Yukiyasu Tsunoo, Teruo Saito, Hiroyasu Kubo, and Tomoyasu Suzaki
10. Weaknesses in the key scheduling algorithm of rc4. Scott Fluhrer Itsik Mantin , and Adi Shami
11. Cryptography and security by William Stallings

Authors Profilre

Dr. B. B. Meshram is working as Professor in Computer Technology Dept., VJTI, Matunga, Mumbai. He is Ph.D. in Computer Engineering and has published international journal is 25, National journal is 1, international conference is 70 and national conference 39 papers to his credit. He has taught various subjects such as Object Oriented Software Engg., Network Security, Advanced Databases, Advanced Computer Network (TCP/IP), Data warehouse and Data mining, etc at Post Graduate Level. He has guided several projects at graduate and post graduate level. He is the life member of CSI and Institute of Engineers etc

Bhargavi H. Kamble is a student of VJTI Matunga, Mumbai. She did her B.E. computer degree from Datta Meghe college of Engineering. She has published three papers. She was a lecturer in Vidyalankar.