# Report : Cryptanalysis of RC4 Encryption Algorithm

Name : Arjun Kumar Gupta
Roll Number : CS20M015

# Introduction

RC4 (Rivest Rivest Cipher 4) is a stream cipher which generates a pseudorandom stream of bits (a keystream) that can be used for encryption  by combining it with the plaintext using bitwise exclusive-OR.But the keystream thus generated by the RC4 algorithm has some flaws and which has been tried to capture by using the cryptanalysis technique in this report.

# Experiment

The RC4 algorithm is implemented using python and is fed with a seed key - a 2048 bit key initialization - which is used by the algorithm to generate the keystream byte by byte.The seed is then toggled at some positions and again fed into the RC4 algorithm which in result generates another keystream byte by byte.The two keystream thus obtained is examined to see the randomness in the key generated.The keystreams are XORed with each other to observe how many bits have been changed in the keystream when the initialization i,e the seed is toggled by a few bits.The XORed is then examined for randomness using a simple frequency counter since the randomness in the XORed result indicates the randomness generated by the RC4 algorithm in the keystream.The randomness is statistically calculated by R = C*D/N where C is the number of counters used for counting the bits changed,D is the standard deviation and N is the number of samples.The closer the randomness R is to zero, the more the random data is.

# Result

The R (randomness) is calculated for each pair - one a random key and the other with some toggling of the first key. Also the toggling positions are taken randomly and an average has been taken over all the values.One by one the number of toggling bits are increased on a fixed length keystream and then the values are analysed.Also fixing the number of toggling bits the length of the keystream are increased for analysing the effects of the R value which turns say that lower the R value the higher the randomness of the keystream would be.

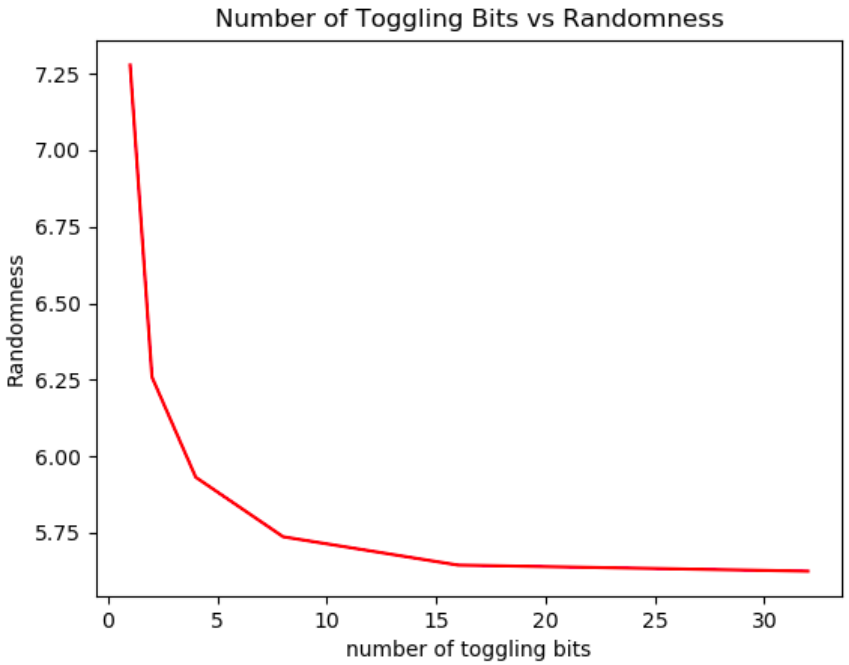



Fig 1. The length of keystream is 2 bytes.

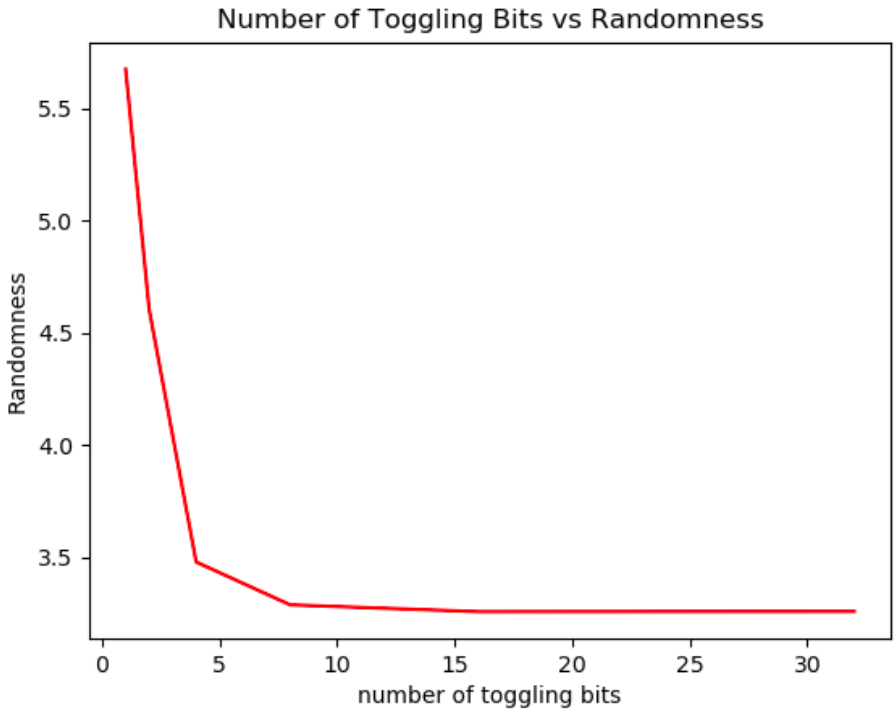



Fig 2. The length of keystream is 4  bytes.

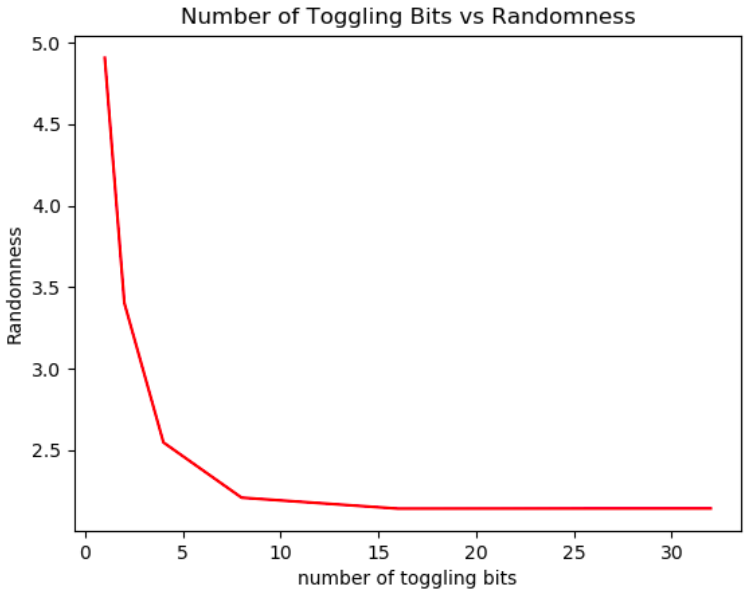



Fig 3. The length of keystream is 8 bytes

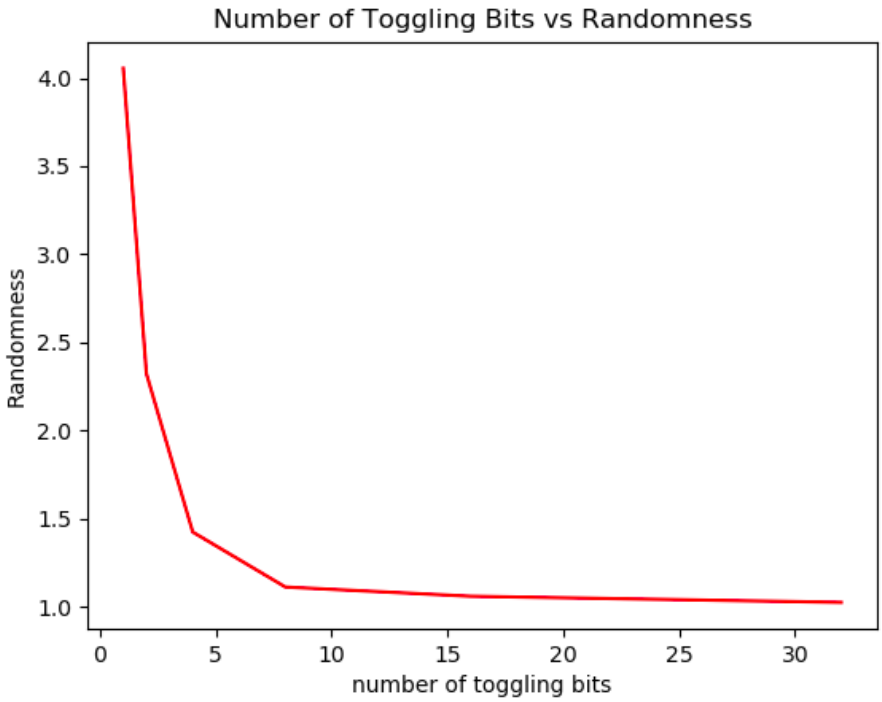



Fig 4. The length of keystream is 32 bytes

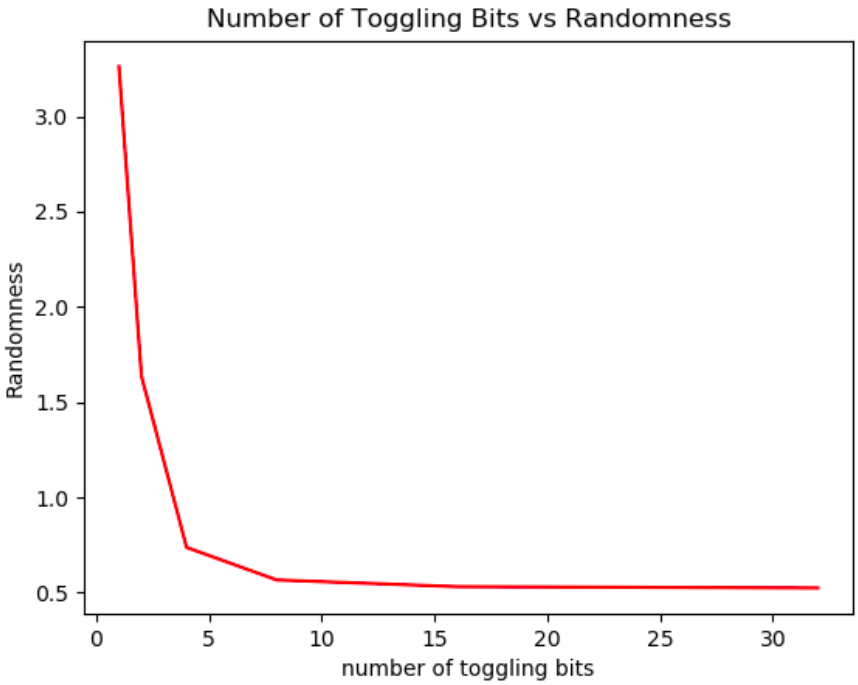



Fig 5. The length of keystream is 128 bytes

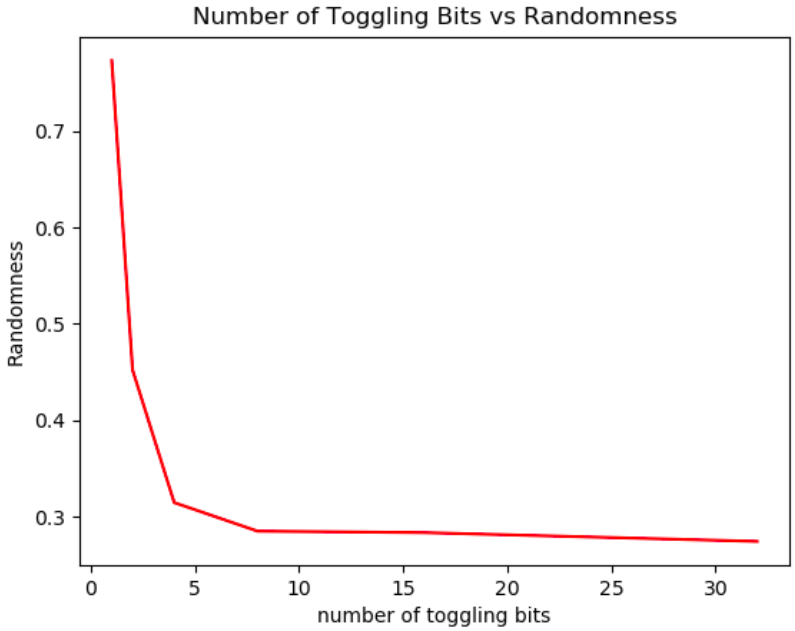
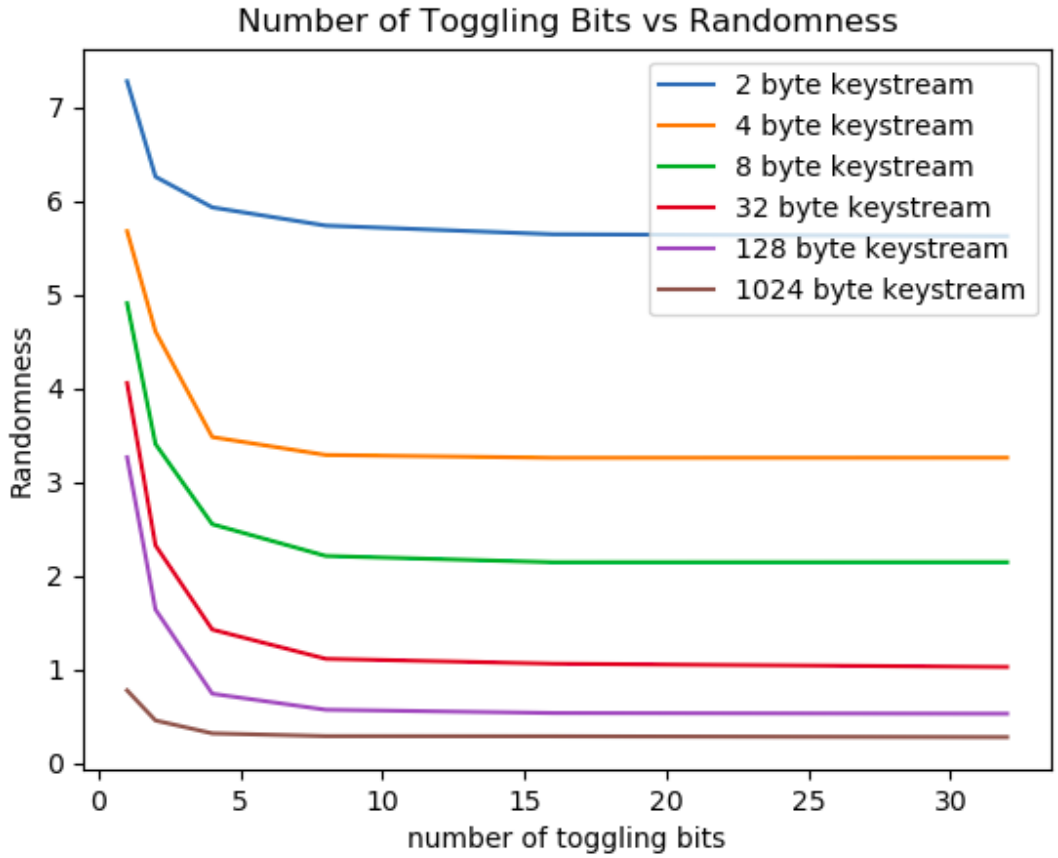



Fig 6. The length of keystream is 1024 bytes




Clearly from the above plots we can inference that the randomness i.e the value R decreases as we increase the number of toggling bits which signifies that the keys generated become more and more random with the number of toggling bits which is required for the RC4 algorithm.Also when we take into account the number of key byte length then we find out that the R is more initially for smaller bytes and as we increase the number of key bytes the R value decreases. Thus we can say that the for few key bytes generated initially by the RC4 algorithm the randomness is less but also we go on the randomness between the keystreams increases.That is we can neglect a few bytes of the keystream initially and then start encrypting the plain text by XORing with the later keystreams.

## Summary

Through this experiment i learnt a lot of things about RC4 especially about the keystreams generated by the algorithm bit by bit.If the initial seed i.e the key is very similar then the starting key stream bytes are less random and as more bytes of keystream are generated a sufficient amount of randomness is noticed.Thus in RC4 algorithm the initial keystream bytes should be neglected and after a certain value only the encryption of the data should be done.