Lab 3 – The RC4 Stream Cipher

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable key-size stream cipher with byte-oriented operations which is based on the use of random permutations.
Review the pdf file posted under module 7 in canvas to assist you in working on this lab.

Download and inspect the python file **RC4_1.py**

The Python code in this file includes the following

- Initialization of the S vector
- Generating the T vector
- A key generation algorithm that includes the necessary initial and subsequent permutations on S which in turn provides the keystream necessary to encrypt (and decrypt) a certain message
- An EXOR encryption algorithm that utilizes the generated keystream
- An EXOR decryption algorithm

1) While the RC4 cipher is in essence a byte-oriented stream cipher, the code given in the downloaded program assumes a 3-bit oriented cipher for simplicity. What are the values (show in the octal number system) of the initial S vector?

2) The Python code also provides the following plaintext and key:

   Plaintext: 001010010010111010011000
   Key: 1010010000011101

   What is the value of the T vector that is generated from this key? (Show in octal)

3) What is the keystream that is generated from the permutations? (Show in octal)

4) What is the ciphertext that results from the EXOR operation? ((show in octal)

5) Adjust the code such that it assumes:

   a) a 4-bit oriented cipher with the following plaintext and key

      Plaintext: AABBCCDD
      Key: 456

   b) a 5-bit oriented cipher with the following plaintext and key

      Plaintext: AABBCCDDEEFF
      Key: 456789

   Repeat steps 1-4 above with these values. But display all values in the hexadecimal number system.

6) Adjust the code to function as a true RC4 byte-oriented stream cipher. Using the same plaintext and key as in 5b above, what is T? What is the ciphertext? Is this system formidable against brute force attacks? If not suggest a key that would make it so.