Izmir University of Economics, Faculty of Engineering & Computer Science
Project 1: CE 340 Cryptography and Network Security
Term: Fall 2018-2019
Strict deadline: 24.10.2018, 17:00 O'clock
Defined by: Süleyman KONDAKCI

STEPS OF IMPLEMENTATION OF A SINGLE-ROUND ENCRYPTION SCHEME:

You will implement the algorithm given in Fig. 1.

1) Read a plaintext from a file block-by-block with block size of 10 characters. Please note that plaintext file must contain more than 10 lines of text.
2) A permutation of length 10 is chosen to be **IP = 10 8 6 4 2 9 7 5 3 1**. The 10-character text blocks are passed through the initial permutation (**IP)**.
3) Use **Table 1** to encode characters so that each character will be 8 bits of length.

Table 1: Character encoding

A 00000001	**B** 00000010	**C** 00000011	**Ç** 00000100	**D** 00000101	**E** 00000110
F 00000111	**G** 00001000	**Ğ** 00001001	**H** 00001010	**I** 00001011	**İ** 00001100
J 00001101	**K** 00001110	**L** 00001111	**M** 00010000	**N** 00010001	**O** 00010010
Ö 00010011	**P** 00010100	**R** 00010101	**S** 00010110	**Ş** 00010111	**T** 00011000
U 00011001	**Ü** 00011010	**V** 00011011	**Y** 00011100	**Z** 00011101	

4) Preform a Shift-right-rotate operation with 4 positions on the encoded characters.
5) Get 2 characters at a time from the encoded block: Characters at odd positions are placed into the left nibble and characters with even positions are placed into the right nibble.
6) Choose a 16-bit (2 characters) key and convert them to 16 bits, and put the bits in to a 4x4 matrix (table), see **Fig.1**.
7) Now use the column selector 3,1,0, 2 to generate $\mathbf{K_2}$ and use the other column selector 0,1,3,2 to generate $\mathbf{K_3}$**.** selector 3,1,0,2 will be used to select columns in the order of 3,1, 0, and 2 to get 8 bits from the matrix for $\mathbf{K_2}$ and for the input to **LS(3,** $\mathbf{K_2}$**)** function. Likewise, selector 0,1,2,3 will be used to select columns in the order of 0,1, 3, and 2 to get 8 bits for $\mathbf{K_3}$ and for the input to **RS(5,** $\mathbf{K_3}$**)** function.
8) Apply XOR together with $\mathbf{K_2}$ and $\mathbf{K_3}$ to encrypt the first part (stage 6, 8). Swap XOR'd partitions so that left becomes right and right becomes left.
9) Now, generate two new sub-keys ($\mathbf{K_4}$ and $\mathbf{K_5}$) by using $\mathbf{K_2}$ and $\mathbf{K_3}$ and the two rotate functions, see **Fig.1 (part 10).**
10) The swapped partitions are XOR'd with the new sub-keys.
11) Swap the result again and merge the results of swapping.
12) Finally, a cipher block is obtained by passing the resulting 10-character block through the reverse permutation $\mathbf{IP^{-1}}$.
13) Encrypted blocks will be saved in a ciphertext file.
14) Finally, verify your encryption by decrypting the ciphertext file.

Izmir University of Economics, Faculty of Engineering & Computer Science
Project 1: CE 340 Cryptography and Network Security
Term: Fall 2018-2019
Strict deadline: 24.10.2018, 17:00 O'clock
Defined by: Süleyman KONDAKCI

What to deliver:

1) **Source code:** You can choose to implement the algorithm in either of these languages **C, C++, Java, or Python**
2) **Screenshots of a sample run**
3) **Plaintext file**
4) **Ciphertext file**

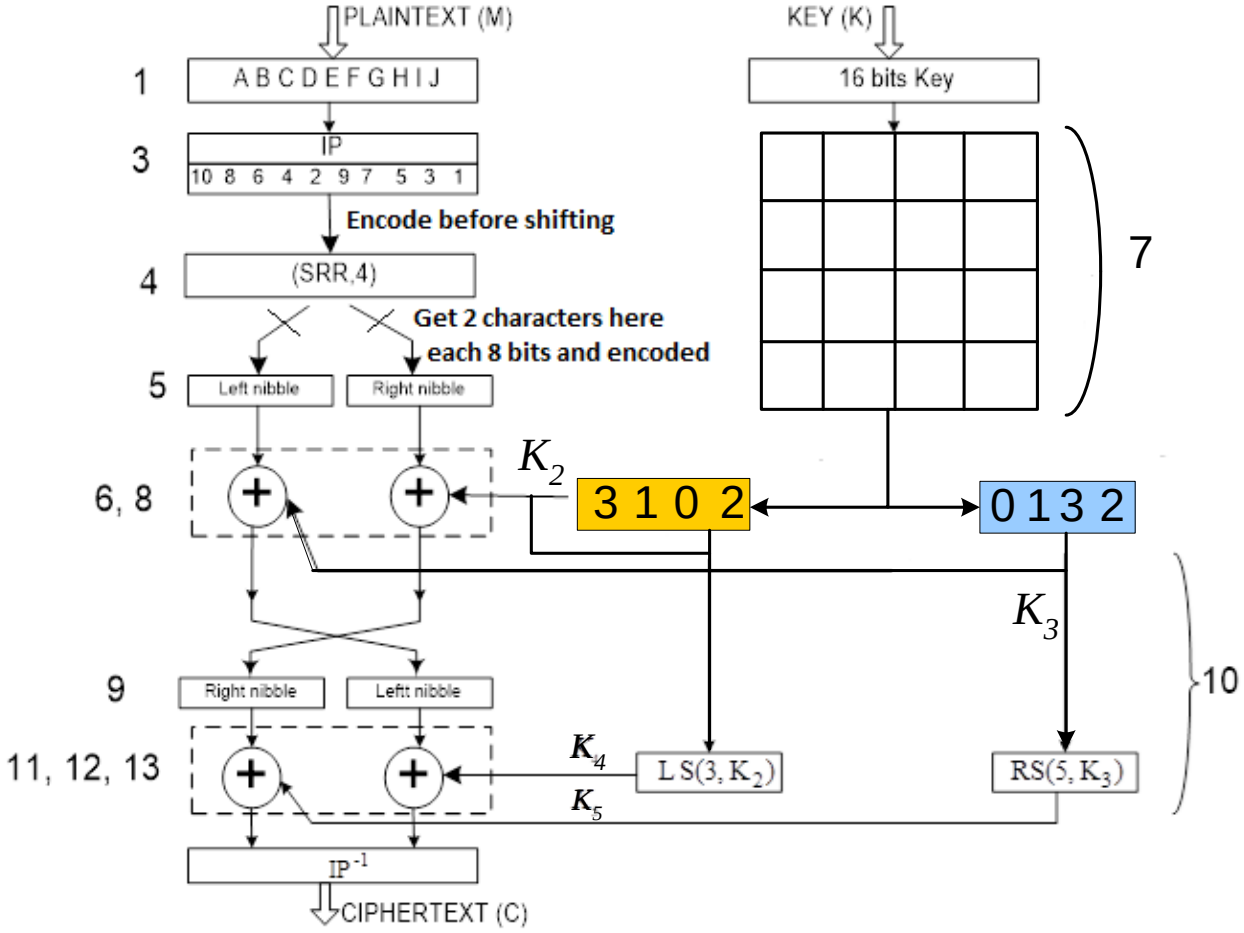

Fig.1 Block Diagram of the encryption process.