The University of Newcastle
School of Electrical Engineering and Computer Science

COMP3260 Data Security
Assignment 2

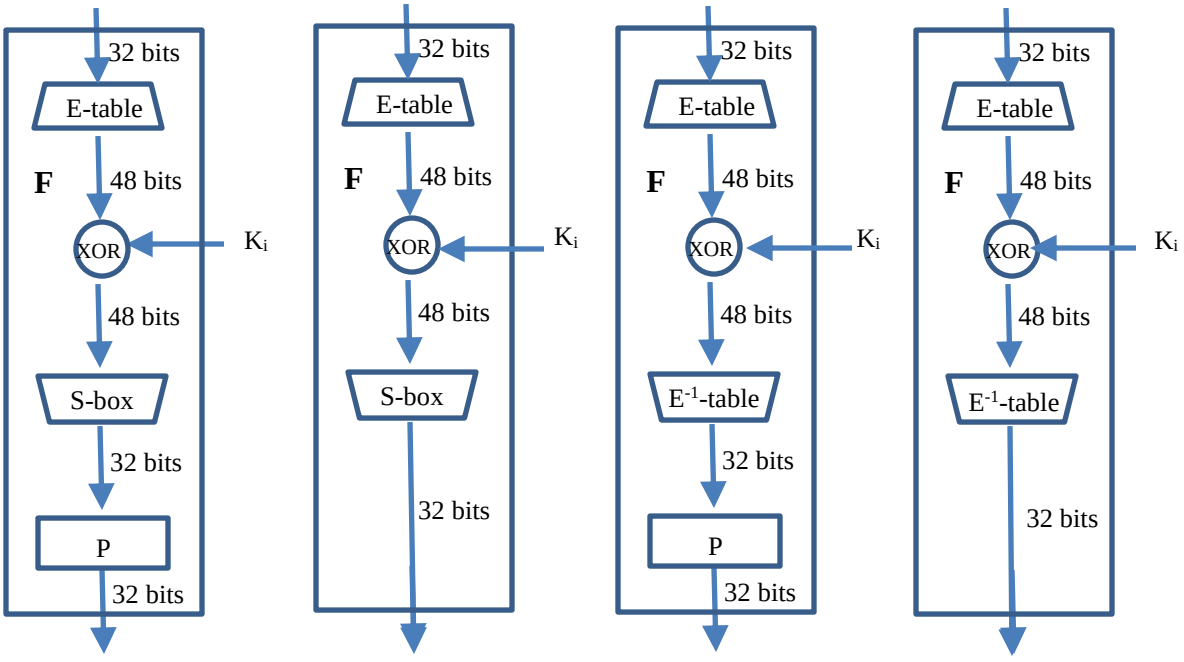
Due on Friday, 23th May 2018, in the Assignment2 in Blackboard.
Total mark: 100

Note:
Before you start working on the Assignment please read the information on academic integrity, which can be found at http://www.newcastle.edu.au/service/academic-integrity/. All available strategies will be used for detecting possible plagiarism and all suspicious cases will be referred to the SACO (Student Academic Conduct Officer).

In this assignment you will implement Data Encryption Standard (DES) 1) encryption and 2) decryption of a single plaintext block. Your program will take as input a 64 bit plaintext block and a 56 bit key and produce as output a 64 bit ciphertext block. You will use your implementation to explore the Avalanche effect of the original as well as three other versions of DES that differ only in the round function F, defined as follows:

1- DES0 – the original version of DES
2- DES1 – the permutation P is missing from the round function F
3- DES2 – S-box in the round function F is replaces with the inverse of the expansion permutation (E-table)
4- DES3 - both the permutation P is missing from the round function F and the S-box is replaced with the inverse of the expansion permutation (E-table)

For additional clarity, the round functions for the four versions of DES are given in the picture bellow.



Inverse E-table (E^{-1}-table) is shown in the figure bellow.

2	3	4	5
8	9	10	11
14	15	16	17
20	21	22	23
26	27	28	29
32	33	34	35
38	39	40	41
44	45	46	47

E^{-1}-table

In addition to the original plaintext block P and the key K, your program should use another plaintext block P_i and key block K_i that differ only in bit i from P and K respectively, and use them to explore the Avalanche effect in DES as follows.

The program will encrypt plaintext P under key K. Then it will encrypt plaintext P_1 under key K and plaintext P under key K_1 and it will find the number of different bits after each of the 16 rounds between
a) P under K, and P_1 under K;
b) P under K, and P under K_1.
Then the above will be repeated for each of the remaining 63 plaintexts P_i that differ from P in a single bit, and 55 key blocks K_i that differ from K in a single bit, and the average results for all 64 bit data and 56 bit key blocks will be presented in the output.

Your program MUST be well commented, include a header stating the authors and purpose of the program, and be easy to understand. You MUST NOT use any available DES code or a portion of it.

1) ENCRYPTION:

INPUT FILE

The following is an example of an input file, where the first row is 0 to denote encryption, the second row is the plaintext P and the third row is key K. Each line is encoded in text with each bit as a separate character.

```
0
000…0
111…0
```

OUTPUT FILE

The following is a format of an output file (note that the numbers provided are sample values and not necessarily what you will obtain for different inputs):

ENCRYPTION

Plaintext P: 000…0

Key K: 111…0

Ciphertext C: 010…0

Avalanche:

P and P_i under K

Round	DES0	DES1	DES2	DES3
0	1	1	1	1
1	2	etc		
2	5			
3	15			
4	33			
5	35			
6	29			
7	31			
8	34			
9	37			
10	31			
11	28			
12	29			
13	33			
14	32			
15	31			
16	33			

P under K and K_i

Round	DES0	DES1	DES2	DES3
0	0	etc		
1	2			
2	13			
3	26			
4	28			
5	29			
6	31			
7	33			
8	35			

9	34
10	29
11	31
12	33
13	30
14	33
15	32
16	32

In the above, 'Round 0' refers to the plain text before the beginning of the encryption. The column DESi contains the number of bits that differ between the original plaintext P, and the intermediate result in each round of the encryption performed by DESi defined above.

2) DECRYPTION: For decryption, the INPUT FILE should contain '1' to denote decryption, the ciphertext and the key, e.g.,

```
1
0000…0
111…0
```

The OUTPUT FILE should contain the ciphertext, the key and the plaintext.

```
DECRYPTION
Ciphertext C: 000…0
Key K: 111…0
Plaintext P: 010…0
```

PROGRAM REQUIREMENTS

The assignment must be completed in either Java or C++, unless you first obtain permission to use another language – please contact your marker Mr. Matt Skerritt to discuss other languages (contact details are on Blackboard).

The program must be able to work entirely from the command line, accepting the input and output file names as arguments. All submissions must compile and run on the University machines and must include a Readme.txt file outlining what each class handles. If implementing in C++ the program must also include a *make* file. Please name the main runnable class Application.

Assessment criteria:

1	DES encryption and decryption – working and correct	55
2	Avalanche analysis, correct	35
3	Comments throughout the program	10
	TOTAL	100

If DES encryption and decryption do not work correctly you can score at most 40 marks in total.