The University of Newcastle
School of Information and Physical Sciences

COMP3260 Data Security

Assignment 2

This assignment is to be done in pairs.

*Due on **Friday 11:59 PM of Week 12**, electronically via the “Assignment 2” submission link in Canvas.*

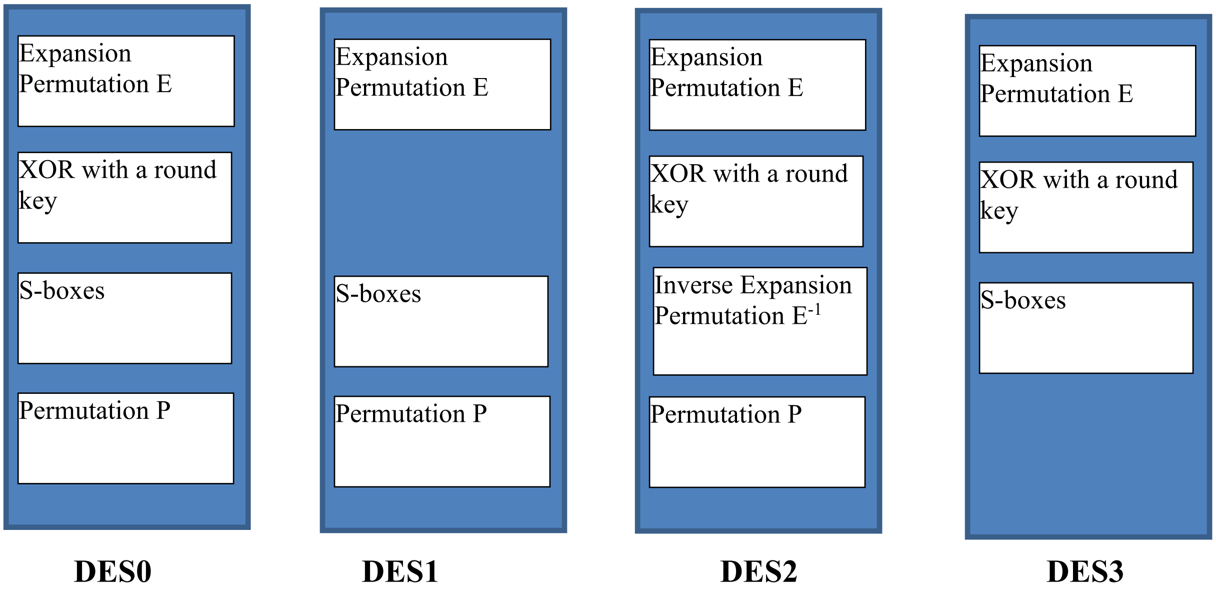
Total 100 marks

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 DES encryption and decryption of a single plaintext block and use this to analyse the avalanche effect of DES. You will first write DES code that will take as input a 64-bit plaintext block and a 64-bit key block (note that only 56 bits of those will be selected by PC-1, since every 8th bit represents odd-parity) and produce as output a 64-bit ciphertext block. You will use your implementation to explore the Avalanche effect of the original DES denoted as DES0, as well as DES1, DES2, and DES3, where in each version an operation is missing in each round as follows:

0. DES0 - the original version of DES.
1. DES1 – XOR with a round key is missing from F function in all rounds.
2. DES2 – S-boxes are missing from F function in all rounds; instead, inverse $E^{-1}$of the Expansion Permutation E is used for contraction from 48 bits down to 32 bits.
3. DES3 – Permutation P is missing from F function in all rounds.

For additional clarity, the encryption algorithm for the four versions of DES is given in the picture bellow.



In addition to the original plaintext block P and the key K, your program should use another plaintext blocks P', such that P'differs from P only in (any of) **one bit** and another K' differs from K only in (any of) **one bit** 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' under key K and it will find the number of different bits after each of the 16 rounds between P under K, and P' under K.

Similarly, your program will encrypt plaintext P under key K' and it will find the number of different bits after each of the 16 rounds between P under K, and P under K'.

Your program MUST be well commented, include a header stating the authors and purpose of the program, and be easy to understand. As a rule of thumb, it is good practice to give descriptive header comments to functions/methods that are longer than 10 lines, and to any classes or structures.
You MUST NOT use any DES code that is not your own work, or a portion of it. For example, you cannot import a publicly available DES function.

Encryption

INPUT FILE

The following is an example of an input file, where

- the first row is the plaintext P
- the second row is the plaintext P'
- the third row is key K
- the last row is key K'

```
000…0
010…0
111…0
110…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):

```
Avalanche Demonstration
Plaintext P:  000…0
Plaintext P’: 010…0
Key K:  111…0
Key K’: 110…0
Total running time: XXX (second)

P and P’ under K
Ciphertext C:  010…0
Ciphertext C’: 101…1
```

Round	DES0	DES1	DES2	DES3
0	1	1	1	1
1	5	etc		
2	20			
3	30			
4	31			
5	34			
6	32			
7	29			
8	36			
9	41			
10	38			
11	29			
12	33			
13	39			
14	36			
15	40			
16	37			

P under K and K’

Ciphertext C: 110…1

Ciphertext C’: 001…0

Round	DES0	DES1	DES2	DES3
0	0	etc		
1	2			
2	18			
3	27			
4	33			
5	41			
6	30			
7	34			
8	37			
9	29			
10	33			
11	40			
12	37			
13	43			
14	38			
15	29			
16	35			

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 (resp. the original key K), and the intermediate result in each round of the encryption performed by DESi defined above.

Decryption

For decryption, the INPUT FILE should contain the ciphertext and the key, and the OUTPUT FILE should contain the ciphertext, the key and the plaintext.

The following is an example of an input file, where

- the first row is the ciphertext C
- the second row is the original key K

```
000…0
111…0
```

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):

```

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

Program Requirements

This assignment may be completed in Python, Java, or C++.

Documentation Requirements

In addition to the code, you will include a text document in pdf format (i.e. word document, LaTeX document, etc.) with a single instance of output for each of the encryption and decryption using any valid input of your choice. You should briefly discuss (~250 words) the impact of the different types of DESi upon the avalanche effect, including observations/proof of significance of each operation within standard DES for preserving this effect. Afterwards you should include a 300–500-word reflections section in this document, which discusses how various resources (such as lecture slides, web resources, textbooks, etc.) helped to complete this assignment. The reflections should also discuss any notable problems you encountered and how you solved them. This document should be concluded with a bibliography following the IEEE, APA, or Harvard citation standard.

Alongside your submission, you should submit a document stating how each group member contributed to each part of the assignment. This document should be concluded with signatures from each group member. A template of this document is available on Canvas on the Assignment 2 page.

Submission

All assignments must be submitted via Canvas. If you submit more than once, then only the latest will be graded. Your submission should be one ZIP file containing:

- A PDF file that contains outputs of your program, and your reflections.
- All source code files.
- A text README file that contains instructions to execute your code.
- A signed document confirming the contributions of the group members.

The mark for an assessment item submitted after the designated time on the due date, without an approved extension of time, will be reduced by 10% of the possible maximum mark for that assessment item for each day or part day that the assessment item is late. Note: this applies equally to week and weekend days.

Marking Rubric

Task	Criterion				
Expansion Permutation E (8 marks)	(3) The expansion function is applied in the correct location.	(6) A variable is correctly assigned for the expansion box.	(8) The expansion permutation is correctly applied.		
Round Keys (10 marks)	(2) The correct key length and odd parity are checked. The program throws an error if incorrect.	(4) Fresh 42-bit subkeys are generated for each round. Prior criteria are also correctly applied.	(6) Subkeys are applied correctly to a block during each round. Prior criteria are also correctly applied.	(8) PC1 is correctly implemented. Prior criteria are also correctly applied.	(10) PC2 is correctly implemented. Prior criteria are also correctly applied.
S-boxes (10 marks)	(2) S-boxes are correctly assigned a usable variable.	(4) A s-box is selected each round.	(6) The correct s-box is selected for each round.	(8) The correct s-box is applied to the block.	(10) The correct s-box permutes the block.
Permutation P (8 marks)	(2) 1 of Initial Permutation, Permutation, or Inverse Permutation are correctly assigned a usable variable.	(4) 2 of Initial Permutation, Permutation, or Inverse Permutation are correctly assigned a usable variable.	(6) Initial Permutation, Permutation, and Inverse Permutation are correctly assigned a usable variable.	(8) Initial Permutation, Permutation, and Inverse Permutation are correctly assigned a variable which is applied to the blocks using a function.	
Encryption and Decryption (8 marks)	(3) A Feistel structure is implemented.	(6) Encryption or decryption is correctly implemented.	(8) Encryption and decryption are correctly implemented.		
File IO (8 marks)	(2) File input works correctly for either encryption or decryption.	(4) File input works correctly for both encryption and decryption.	(6) File input works correctly and file output works correctly for either encryption or decryption.	(8) File input works correctly and file output works correctly for both encryption and decryption.	
Commenting throughout the Program (6 marks)	(2) Header comments are provided for each file.	(4) Header comments are provided for each file, complicated/long functions are also given header comments.	(6) Header comments are provided for each file, complicated/long functions are given header comments, and complicated lines are given inline comments.		
Avalanche Analysis Code (15 marks)	(3) One of DES {1,2,3} are implemented.	(6) Two of DES {1,2,3} are implemented.	(9) All of DES {1,2,3} are implemented.	(12) All of DES {1,2,3} are implemented, and are used in encryption.	(15) All of DES {1,2,3} are implemented, are used in encryption, and in round differences are output.
Avalanche Analysis Documentation (15 marks)	(5) A correct avalanche analysis output is provided.	(10) Alongside the correct output, the general trends are noted and summarised.	(15) Alongside the correct output, the general trends are noted and summarised, additionally, the importance/impact of each operated is discussed.		
Reflections (10 marks)	(2) What is learnt from external resources is only listed	(5) What is learnt from external resources is discussed	(8) What is learnt from external resources is discussed, tasks are related back to the lecture and workshop contents	(10) External resource lessons are discussed, tasks are related back to this course, and discussions of your problem-solving approach and hardships are provided	
Bibliography (2 marks)	(1) A bibliography is provided in the correct format.	(2) A bibliography is provided in the correct format, and citations are used throughout the text where relevant.			