COMP 7402 (FLEX)
Cryptology
Assignment #2

Your submission should be in a **package** `<FirstName_LastName_StudentID.zip>` and include:

- A **PDF** document report containing answers to all questions, tools that you used, **any supporting data (like the Excel files**), and references.
- Ensure your report begins with a table of contents for easy navigation.
- Source **code** files.
- **Screenshots** of the executed code.
- Please note that during the lab, I may ask you questions about the details of your implementation to verify that you completed the assignment yourself. If you are unable to satisfactorily explain your answers, you will not receive any marks for the assignment.

To be completed by **Jan 30, 2025**. This is an **individual** assignment. Late submissions will **not be accepted.** Total mark is 100.

TASK 1 - FEISTEL STRUCTURE CIPHER DESIGN – DES [40 MARKS]

Using **any** programming language of your choice **(preferably python or java)** implement a **Feistel-based Cipher (DES) Encryption and Decryption:**

Here is DES specification as per the NIST standard that helps you with your design and implementation, you can use the constant tables (S-box, P-box, etc.) in your code directly from here:

https://csrc.nist.gov/files/pubs/fips/46-3/final/docs/fips46-3.pdf

This also might be helpful:

https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm

Here is also an example of DES implemented to work with Hex:

https://paymentcardtools.com/basic-calculators/des-calculator

In your task submission, include the following [5 MARKS]

A **README** file, which should explain how to run the code with sample input and output.

Steps:

1. Use **Hex** values for your implementation process.
2. **Initial Permutation (IP):** Apply an initial permutation to the 64-bit plaintext
3. **Feistel Rounds:** For each round:
   - Split the block into left and right halves.
   - Expand the right half from 32 bits to 48 bits (expansion permutation) and XOR with the round subkey.
   - Pass the result through S-boxes to compress it back to 32 bits.
   - Apply a P-box to permute the bits.
   - XOR the result with the left half.
   - Swap the left and right halves (except after the final round).
   - You can review the last couple of pages of the lecture notes diagram is there.
4. **Final Permutation (IP^{-1}):** After all rounds, apply a final permutation (inverse of the initial permutation).
5. **Decryption:** Should be identical to encryption, with round keys applied in reverse order.

Specifications:

- **Number of Feistel Rounds:** 16 rounds
- **Plaintext Block Size:** 64 bits
- **Key Size:** 64-bit key, with 56 bits used, 48 permuted and XORed inside the F-Box

Cipher Components:

- An IP and IP^{-1} (Page 10 of the doc)
- An **Expansion Layer** (E-box) (Page 13 of the doc)
- A **Substitution Layer** (S-box) based on a 6-bit input to a 4-bit output (Page 14 of the doc)
- A **Permutation Layer** (P-box) to ensure diffusion of the bits. (Page 15 of the doc)
- Bit-by-bit addition modulo 2 operations for the 48-bit subkey generated from the key scheduler inside the F-box, and to calculate the RE$_i$ at each round (step). (Slide page 18 and 38)

Key Scheduler:

- The cipher should use a **key schedule** that generates 16 round subkeys (each 48 bits) from the original 64-bit key. (Page 20 of the doc)

TEST [15 MARKS]

Test with the values below, this should generate the example we showed in the class (**page 25 of the slides**), encrypted and then decrypted to verify correctness. Show the value of the output at each round, like the table at page 25 of the slides.

Plaintext:	`02468aceeca86420`
Key:	`0f1571c947d9e859`
Ciphertext:	`da02ce3a89ecac3b`

TASK 2 – DIFFUSION AND CONFUSION – SPAC AND SKAC ANALYSIS [40 MARKS]

Steps:

1. **Strict Plaintext Avalanche Criterion (SPAC): [20 MARKS]**
    - Encrypt two plaintexts that differ by just one bit.
    - For each round, record how many bits differ in the ciphertext after applying each round of encryption. (Table at page 26 of the slides for reference, but try your own hex values)
    - Analyze how changes propagate through each round, looking for the avalanche effect (50% of bits should flip).
    - Create tables in Excel showing:
        - For SPAC: Round-by-round differences in ciphertext between two slightly different plaintexts.
        - Generate graphs showing the number of differing bits over the 16 rounds
        - A short write-up interpreting your SPAC results and explaining how well your cipher satisfies the Avalanche Effect.
2. **Strict Key Avalanche Criterion (SKAC): [20 MARKS]**
    - Perform encryption using the original key and a key with a 1-bit difference (plaintext untouched). (Table at page 27 of the slides for reference, but try your own hex values)
    - For each round, record how many bits in the ciphertext differ between the two encryptions.
    - Track how changes in the key affect the ciphertext, measuring key sensitivity.
    - Create tables in Excel showing:
        - For SKAC: Round-by-round differences in ciphertext between two slightly different keys.
        - Generate graphs showing the number of differing bits over the 16 rounds
        - A short write-up interpreting your SKAC results and explaining how well your cipher satisfies the Avalanche Effect.