Laboratory: Data Encryption Algorithm

1 Objective

Objective of the lab: The objective of this lab is to give the student some practical exposure to the DES concepts and algorithm presented in class.

2 Requirements

Note: This lab requires some preparations, in terms of theoretical background as well as the use of the tools (use of Matlab, the m-files, etc.). Matlab is the only programming language allowed to be used for this lab. Students are not allowed to use the standard libraries for the DES algorithm and they should only use the basic functions.

Instructions, source material and preparation required:

- Each student is required to do all the preparation needed to implement the algorithms beforehand. All students need to have access to all the permutation tables (Initial Permutation, Inverse Initial Permutation, S-boxes, E-boxes, P-boxes, Key Permutations, etc.) as well as all the detailed workings of each step.
- Each student is required to complete tasks independently.

Report: Each student must submit:

- a single report (Maximum 5 double-column pages excluding the appendices).
- All the files used in the lab.
- All files should be compressed into one file (.zip) with the file name in the format 'LS_XXXXXXXX.zip' (where XXXXXXXX is the student number).

3 Outcomes

1. Implement a function that will generate the specific subkey K_i when the following parameters are passed as inputs to the function: a 56 bit key and the index i.
2. Calculate the number of unique subkeys for the following 64-bit keys (ignore the parity bits in your calculations) and classify the keys:
   - $1F1F\ 1F1F\ 0E0E\ 0E0E$,
   - $1FFE\ 1FFE\ 0EFE\ 0EFE$,
   - $1FFE\ FE1F\ 0EFE\ FE0E$.
3. Implement a function that will produce two 32-bit output blocks, given a 64-bit input block, the index of the round ($i \in \{1, 2, ..., 16\}$) and the 48-bit subkey K_i. (The whole round must be implemented.)

4. Using the functions of 1 and 3, implement the Data Encryption Algorithm (DEA).

5. Implement the decryption algorithm as well.

To verify if a student's implementation is working correctly, arbitrary inputs will be tested. A student is required to provide examples to test the tasks mentioned above. All input files for demonstration should be included in a submission.

Note that the keys are stated in Little-Endian format, that is, the least significant nibble is stated first. Converting the nibble to binary is done using the conventional way, i.e., $MSB, \ldots, LSB$. As an example, $1FFE$ is converted to binary as 0001 1111 1111 1110, where 0001 constitutes bit 0 to 3.