CS 408 – Cryptography & Internet Security (Spring 2015)

Programming Project

For the programming project, you are asked to implement a variant of the DES block cipher (encryption/decryption of one block of data). The project must be completed in `java` and you are given a skeleton in the file `des.java`. The skeleton uses the notation for DES defined in "The Handbook of Applied Cryptography" (`http://www.cacr.math.uwaterloo.ca/hac/about/chap7.pdf`, pages 252-256). The skeleton contains:

- function definitions for the various steps in DES: key schedule (to obtain the round keys), initial permutation, expansion function, XOR function, S-Boxes, the permutation inside function f, one round in DES, the inverse of the initial permutation, encryption, and decryption.
- all the tables needed for the permutations, expansions, S-Boxes, etc. are already hard-coded.
- the `main()` function for testing the program.

Your task is to implement the various steps in DES and you will receive a number of points for implementing each of these steps (the exact number of points for each step is described in the file `des.java`). You should comment your program so that it can be easily read by others and thus facilitate the grading of your project (points are also allocated for proper commenting).

Unlike in the standard DES algorithm which has 16 rounds, you are asked to write a DES variant with 18 rounds.

You are allowed (and encouraged) to work in a team of up to two students. No collaboration between teams is allowed, including sharing portions of the code is not allowed.

The `main()` function will be used by the grader to test the functions you have written. You don't need to fill any code in the `main()` function. The `main()` function takes several arguments as described in the file `des.java`.

You are provided with one test case (file output_1.txt) to check the correctness of your implementation for each of the various steps in DES. For example, if you want to check the correctness of your implementation of the `Initial_Permutation()` function, the test case will provide you with the correct output for a certain input to `Initial_Permutation()`. Similarly, to check the correctness of your implementation of the `One_Round()` function, the test case will provide you with the correct output for a certain input to `One_Round()`.

The first line of the test case file contains the command used to generate the test case. The parameters are required to run the program, and represent inputs for the various functions that you need to implement. The parameters are given as strings of characters that represent binary values (for example, the binary value 01100110 is given as the string "01100110").

For any question related to the programming project, please email the grader at me76@njit.edu and also CC crix@njit.edu. **You need to email your final assignment (the file `des.java`) to me76@njit.edu. You must include the name/s of your team members as a comment in the beginning of the file `des.java` and also include "CS.408" in the Subject of your email. One submission is required per team.**