CSCI 4331-6331 – Cryptography – Spring 2020
The George Washington University

Homework 1
due 2 March ON BLACKBOARD, midnight

Policy on collaboration: All examinations, papers, and other graded work products and assignments are to be completed in conformance with The George Washington University Code of Academic Integrity.

You may not discuss HWs among yourselves before submission. Each student must work on the HW by himself or herself, with no collaboration whatsoever. You may not refer to any sources other than your class notes, the textbook, and material available on the course website (such as slides, class notes, linked websites, suggested reading).

All code must run on your shell account. The TA will make no attempt to debug your code, or determine why it does not run. You will be graded on the correctness of your output, and the quality of your code: efficiency and documentation. There will be no exceptions. Code must be written in C++, Java or Python. Be aware that Python might be too slow for the project assignment, which will be a cryptanalysis assignment.

Under no circumstances may code be copied from anywhere: classmates, the web, any other source.

Any violations will be treated as violations of the Code of Academic Integrity.

Submit all HW in Blackboard by midnight on due date. Name your files:
CS6331_HW1_LASTNAME_FIRSTNAME.rar or .zip or
CS4331_HW1_LASTNAME_FIRSTNAME.rar or .zip

Implement an SPN with 8 rounds of 4 S-boxes each, each S-box takes as input 8 bits.

Your program takes as input a file with the following information; you may choose the format of your input file, it will be named input_spn.txt:
- a key K of length 36 bytes; 4 bytes being used for each of nine XORs (there is no key schedule as the key bits used are distinct each time)
- the S-box function f taking a byte as input and providing a byte as output
- the permutation function σ permuting 32 bits
- m, an integer that is a multiple of 32; m is the length of the input in bits
- an input x of length m bits
- an integer a of value 0 indicating encryption, and value 1 indicating decryption

Your program produces the following output, in a file of the same format as your input file, named output_spn.txt, with the following information:
- the input key K
- the input S-box function f
- the input permutation function σ
- m, the length of the input in bits
- an encrypted or decrypted output of length m bits; $E_K(x)$ if $a = 0$, and $D_K(x)$ if $a = 1$
- an integer representing $\bar{a}$

Thus, if your output is fed into the code, it should produce the input and vice versa.

You are expected to write the code yourself, and not to get it from a library.

Your code will be tested by running it first on your own input files, and then on ones we will choose. Thus your code should check for errors.

Your code should be submitted with a README file that describes the structure of your input and output files so that we may construct similar files of our own to test your code.