# CS 478/513: Computer Security
# Spring 2022
## Programming Assignment 2
Due: Wed., 3/23/22, 11:59 pm

**Overview:** All code must be written modularly in C or C++, and must successfully compile and run on the Linux machines in the CS department's labs.

1. (50 points) Implement the A5/1 algorithm we talked about in class (see "CS3.pdf" slide set). Assume the initial fill of the three registers to be:

$$X = 1010010011000011100$$

$$Y = 00110111001000011110 11$$

$$Z = 11101010001110111000 10$$

   (a) (40 points) List the next 32 keystream bits and give the contents of X, Y, and Z registers after these 32 bits have been generated. Note that this is similar to Chapter 3, Problem 5, but the initial fill is different, so outputs will be different.

   (b) (10 points) Encrypt the message `7e5d7fff` and output the ciphertext. Then decrypt the ciphertext and verify you get the message back.

2. (50 points) In this problem, you will implement the various steps of the DES algorithm (see slide 20 in "CS3.pdf"). For testing your implementation, please choose a random message of size not equal to a multiple of block length (which will require you to account for padding). Show the following implementations and their outputs on your test message.

   (a) (10 points) Implement and show the output of the expansion function (see slide 21 in CS3.pdf).

   (b) (10 points) Implement the S-box and show the output (see slide 22).

   (c) (10 points) Implement the P-box and show the output (see slide 23).

   (d) (15 points) Implement the key schedule, and show the output (see 24, 25, 26).

   (e) (5 points) Rest of the stuff in DES (xors's, etc. You can implement the $IP$ and $IP^{-1}$ if you'd like to, but it is not really required.)

   Note that your implementation should work on any input message. We will test it on a random message of arbitrary length to check if it works, we will also generate the 56-bit key ourselves. The size of the message we test it on is not guaranteed to be a multiple of the block length – please check for padding, if needed. Per good coding practices, please also include reasonable error checks (bad key length, null key, etc.)

**Submission requirements**: Upload a tarball (.tar or .tar.gz) to Canvas. The submission should include the following material:

1. Your C/C++ code, with a Makefile to compile it.
2. The 32 bytes of the keystream in A5/1 cipher.
3. The key, ciphertext, and verified decrypted plaintext in A5/1 cipher.
4. For DES, the output of each step of the implementation as given in the point-wise breakup, for your test message.
5. A text file (Readme) explaining how to use your programs for each of the following tasks:
   (a) Instructions on how to test your DES program on an arbitrary message/key of our choice.