Cryptography module, Exercises 2 (unassessed)

You can't get credit for these exercises, but you can still submit them on Canvas and you will get feedback. You must type your answers in a word processing system, and create a PDF. I cannot accept handwritten and scanned or photographed answers. The deadline (if you want to get comments) is 6 November 2017.

Extra incentive: there is a prize for the first person that manages to email me two strings satisfying the conditions of Q3.

1. Let L be a 9 bit LFSR with taps on the 1st, 4th and 8th bits, and seed $s = [1, 0, 1, 1, 1, 0, 0, 1, 0]$. The 9th bit (initially 0) is the next keystream bit. Compute 20 bits of the keystream.

2. Consider a modified version of the cipher block chaining mode, where the initialisation vector is not chosen randomly for each encryption, but instead chosen randomly once (like the key) and then increased by one for each encryption. Show that this modified version is not IND-CPA-secure. Note that the initialisaton vector is always part of the ciphertext.

3. **Programming exercise.** Let MY60SHA be a hash function which outputs the first 60 bits (15 nibbles) of SHA-1. For example, SHA-1 of "mark" is

   ```
   f1b5a91d4d6ad523f2610114591c007e75d15084
   ```

   so MY60SHA of "mark" is `f1b5a91d4d6ad52`. Find any collision for MY60SHA. (Note: you should find two strings such that the unix command

   ```
   echo -n str | sha1sum - | cut -c1-15
   ```

   produces the same answer when *str* is replaced by each string. To enable me to verify your answer, please make sure the two strings are typable on a regular keyboard!

   Hint: It's a good idea to find shorter collisions first. For example, start off finding a collision for the first 30 bits; that's a lot easier. The challenge that 60 bits gives you is that you probably can't store all the intermediate hashes you generate in memory (unless you have a lot of memory). Nevertheless, you should be able to write a program which finds a 60 bit collision in a few hours on a regular desktop or laptop computer. My program is about 50 lines and runs in a few hours on my desktop that has 4GB RAM.

4. Exam 2016, Q2 (about MAC functions).

5. Exam 2017, Q1 (about DES, 2DES and 3DES).