Cryptography module, Exercises 2 (assessed)

You must type your answers in a word processing system, and create a PDF. We will not accept handwritten and scanned or photographed answers. Please submit your PDF on Canvas. The maximum mark you can get is 8, which will be scaled to a mark out of 10. The deadline for these exercises is 6 November 2019.

You can work on your own, or you can work in pairs. If you work in a pair, both of you should submit the answers (both will get the credit). Please note on your answers that you worked as a pair, and please mention both ID numbers.

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

1. Consider AES with 128-bit keys. Assume that the initial subkey (k_0) and the round 1 subkey (k_1) are both all-zero, and that the plaintext block is also all-zero. What is the output of the first round? **[2%]**

2. Consider AES with 128-bit keys. Suppose the encryption key is all-one (i.e., 128 ones). Compute the initial subkey (k_0) and the round 1 subkey (k_1). **[2%]**

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: You should not write the code for SHA-1; you should use an existing library. Also, it's a good idea to find shorter collisions first. For example, start off finding a collision for the first 24 bits (6 nibbles); 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 computer that has 4GB RAM. **[2%]**

4. Add a photo of yourself to Canvas. This helps staff members match their verbal discussions with you with their Canvas interactions. If you object to adding a photo of yourself to Canvas, you can instead earn the marks allocated to this exercise by writing a one-paragraph summary of why you object. In your summary, please focus on carefully articulating the potential harms that could occur if you added your photograph. **[2%]**