Network Security and Cryptography, Exercises 1 (unassessed)

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.

1. Decrypt the following ciphertext encrypted with the columnar transposition cipher (you need to find the key by brute-force search):

   AVUEVLETSEISBNACBOOLEOBTILBDLCOBOOE **[1 point]**

2. What is the output of the first round of the DES algorithm when the plaintext and the key are both all zeros? **[1 point]**

3. Remember that it is desirable for good block ciphers that a change in one input bit affects many output bits, a property that is called *diffusion* or *avalanche effect*. We will try to get a feeling for the avalanche property of DES. Let x be all zeros (0x0000000000000000) and y be all zeros except 1 in the 13th bit (0x000*8*000000000000). Let the key be all zeros.

   After just one round, how many bits in the block are different when x is the input, compared to when y is the input? What about after two rounds? Three? Four? (For this exercise, you might like to search for an implementation of DES on the web, and download it and modify it to output the answers.) **[2 points]**

4. Consider AES with 128-bit keys. Assume that the principal key k is all-zeros. Then the initial round key (k_0) is also all-zeros. What is the first round subkey (k_1) and the second round subkey (k_2)? (Again, for this exercise and the following one, you might like to use a computer for help.) **[2 points]**

5. Again using AES-128, assume that the principal key is all-zeros, and that the plaintext is also all-zeros. What is the output of the first round, and what is the output of the second round? **[2 points]**

6. **Programming exercise.** Let MY24SHA a hash function which outputs the first 24 bits (6 nibbles) of SHA-1. For example, SHA-1 of "mark" is

   `f1b5a91d4d6ad523f2610114591c007e75d15084`

   so the MY24SHA of "mark" is `f1b5a9`.

   Find any collision for MY24SHA. Note: you should find two strings such that the unix command

   `echo -n` *str* | `sha1sum - | cut -c1-6`

   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. **[2 points]**