Nick Guo

I pledge my honor that I have abide by the Stevens Honor System.

Problem 1

(1) It should be confidentiality and integrity.
    a. For confidentiality, we need to make sure that only authorized users (which is Alice and Bob) can access the data. So no other unauthorized users (especially attackers) can access Alice and Bob's communication and know their keys.
    b. For integrity, we can make sure that only authorized users (Alice and Bob) can modified the asset or the data (which is the key and message here) and give them validation in the long term. No one else can modified the assets they have so that they won't have the wrong result.

(2) This protocol doesn't follow both security properties identified in question (1). Assume that the attacker eavesdrops the encrypted message X and Y since X and Y are the only messages transmitted through the insecure (channel) network. (which is not satisfying confidentiality) Then the attacker can just use XOR operation in between X and Y.

$$X \oplus Y$$

For example, assume random n-bit value r = 10010010
And we assume that Alice and Bob share a key k = 10111010
So X = $k \oplus r = 10010010 \oplus 10111010 = 00101000$
Then Bob receives the X, and compute Y.

$$Y = k \oplus X = 10010010$$

So if the attacker do the XOR operation:
$X \oplus Y = 10111010$ which is exactly the key k. Therefore, this protocol is not satisfied the secure properties.

Problem 2

(1) It is not secure if the cipher period is 1 or 3.
    - When the cipher period is 1, it's just easy Caesar cipher, which is easy to find the key since the attacker only needs to compare the difference of the first two characters then he will know the password. If the difference is only 1, it will be abcd. If the difference is 3, it will be bedg.
    - When the cipher period is 2, it is secure since for both passwords, comparing the first character with third character, and second character with fourth character, ac, bd, eg, the difference between them is always 2. Since cipher period is 2, it means that only the first character and third character, second character and fourth character are in the same system. By using Vigenère cipher, if the attacker doesn't know the key, it would be safe since the attacker can't find out the code according to the difference.

- When the cipher period is 3, similar to the first part, if the attacker compare the fourth character and the first character, since they are in the same system, if the difference is 3, the password is abcd. If the password is 5, the password will be bedg.
- When cipher period is 4, which is also the length of password, it is secure since each character of the password has its own Caesar cipher. In other words, each letter is independent. As long as the attacker doesn't know the key of the Vigenère cipher, it's basically impossible for him to find the password.

(2) When attacker choose to use chosen-plaintext attack, he will know the plaintext and its corresponding ciphertext. Since any plaintext with a distinct 25 letters can show a substitution for each of the 25 letters, by using the process of elimination, the attacker can find the last letter of the key of the mono-alphabetic substitution. And then he can just keep doing the process to get the entire key of this substitution. The shortest possible text that can be found contains 28 letters stating, which is "Jived fix nymph grabs quick waltz". The conditions for mono-alphabetic substitution cipher to be perfectly secure is that the probability of any mapping between the characters and ciphertext is equal.

$$\Pr[\, Enc_k(m) = c \,] = \Pr\,[\, Enc_k(m') = c \,]$$

Problem 3

KEYS: 59 6f 75 66 6f 75 6e 64 74 68 65 6b 65 79 21 63 6f 6e 67 72 61 74 75 6c 61 74 69 6f 6e 73 21 21 21

testing testing can you read this

yep I can read you perfectly fine

awesome one time pad is working

yay we can make fun of Nikos now

i hope no student can read this

that would be quite embarrassing

luckily OTP is perfectly secure

didnt Nikos say there was a catch

maybe but I didnt pay attention

we should really listen to Nikos

nah we are doing fine without him

In order to get the correct answer, I used the frequency analysis and brute-force for this problem. My algorithm is that since there are 33 characters in each message and 66 bits for each ciphertext, it means that every character will be encrypted to 2-bit hex number. Since the eleven messages have the same length, I tried to find the most frequent hex number in each column and put them into a 2d array. Then, I used those most frequent hex number (which is a ciphertext) to xor all possible input characters to get a testing key. Then I used the testing key to xor all other elements in this column. If there's no invalid

input for this column, the testing key will be the key for this column. By looping over all the columns, you can get a possible key for this ciphertext. Then I use the key to xor those ciphertext to get the plaintext.

Formula: $c1 \oplus m1 = k$

$$c_i \oplus k = m_i$$