EECS 482: Homework 9.
Due date: Wed. Apr. 12th, 11:59 PM

This homework is a makeup homework (like the previous one). $\{msg\}_k$ is same as *encrypt(msg, k)* in class.

Alice often sends Bob "to do" reminders about tasks he should do that day; e.g. "Go to the bank", "Send a check to the cable company", etc. They would like for Alice to be able to send these reminders securely over the Internet: in particular, they would like to guarantee that her communication is authentic, confidential, and fresh. They share a symmetric key, K, that was securely exchanged at some prior time. Alice uses the following protocol to send a reminder to Bob:

First, Alice sends Bob a randomly-generated nonce, N, encrypted with their shared key. The checksum is over the text in the message (not including the checksum).

message 1: (Alice to Bob): $\{$Here is a nonce: N, cksum$\}_K$

Second, Bob replies to Alice with the incremented nonce encrypted with their shared key. The checksum is over the text in the message (not including the checksum).

message 2: (Bob to Alice): $\{$the next number is: N+1, cksum$\}_K$

Finally, Alice sends Bob a reminder, with an incremented nonce, encrypted with the shared key. The checksum is over the text in the message (not including the checksum).

message 3: (Alice to Bob): $\{$N+2, send a check to the cable company, cksum$\}_K$

Assume that Alice and Bob keep some state between these three messages, but that they keep no state after the last message for a reminder.

A. The problem Briefly describe the most serious problem with the above protocol.

B. The solution Briefly describe how you would fix the problem you identified. Your solution must not require Bob or Alice to keep any state after the last message for a reminder.