Introduction to Cryptography: Problem Set 2

Name: Student Name
Collaborators: Collaborators

By turning in this assignment, I agree to abide by and uphold the Honor System of Introduction to Cryptography as well as the additional policies outlined on the course website.

Transparency Note: This homework structure is modeled off of David Wu's Advanced Topics in Cryptography course.

Short Answer (3 points each)

1. True or False: Asymmetric cryptography is inherently more secure that symmetric cryptography. Why?

2. Why would someone use public key (asymmetric) over secret key (symmetric) cryptography?

Medium Answer (7 points each)

1. Recall the description of Diffie-Hellman key exchange. It roughly boils down to two messages exchanged: (1) Alice sending g^a to Bob and (2) Bob sending g^b to Alice. Then, they each can compute g^{ab}. This is the shared secret they can then use to for security. How could you modify this protocol to work for three actors?

2. In class we discussed some of the benefits and costs of digital signatures systems. One potential limitation of a digital signature is that for a given signature σ of a message m corresponding a public key pk, one could generate a pk', sk' that produces a signature σ' for m, such that $\sigma' = \sigma$. Given a signing function S, a verification function V, that each take corresponding private and public keys for generating and verifying a signature and are vulnerable to this duplicate keys, please generate a S' and V' that are immune to a duplicate keys attack.

3. In class we discussed the public security semantic security attack game. When constructing the public key encryption for the class, I highlihgted that the encryption function needs to have random outputs. Please describe an attack that would be break semantic security of a deterministic, asymmetric encryption system. (hint: start by defining semantic security, and how public key encryption systems are used in the real world)

4. In class we discussed how to construct a digital signatures using trapdoor functions. It was highlighted that without hashing and then signing the message, an attacker could trivially generate secure signatures. Why is this the case? Please describe an attack an adversary could run against a system that used a trapdoor to construct digital signatures but didn't hash the message beforehand! (hint: think about the properties of trapdoor functions)

Feedback (optional) (0 points each)

1. What was the hardest problem on this problem set?
2. What was your favorite problem on the problem set?
3. How long did you spend on this problem set?
4. Do you have any feedback for how I can improve either this homework set or the course in general?