ICS 134 – Winter 2015: Homework 2

Name: **Student ID:**

Due date: February 28 @noon (no extensions and no exceptions)

Submission Guidelines: Upload your homework as a **PDF file** to the class dropbox.
Warning: Homework submissions not following guidelines will not be graded.

Problem 1:

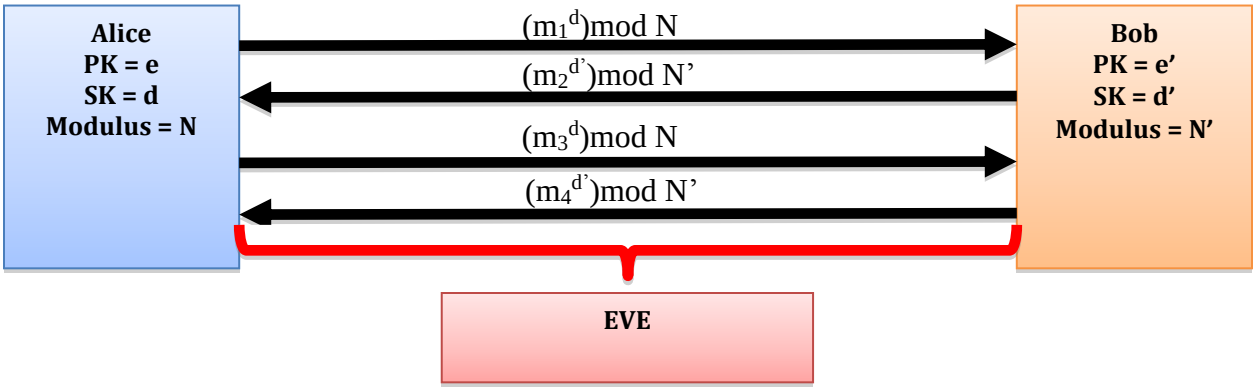
A) Recall the Chinese Reminder Theorem (CRT). It was claimed in class that it helps speed up decryption of RSA ciphertexts. Using the following parameters, show how it's done:

p=11, q=5, e=7, d=?? (find it yourself), C=18

B) Explain whether the same technique (i.e, usage of CRT) can help in more efficient generation or verification of RSA signatures.

Problem 2:

Alice and Bob communicate using RSA signatures for authenticating their messages. Eve eavesdrops on their conversation and observes the messages below. Seeing these messages gives Eve the ability to create some false messages with valid signatures. What kind of "authentic" messages can Eve now forge? What do you suggest to prevent such forgeries?



Problem 3:

Alice and Bob share a secret key K. Consider the following two situations:

A) Alice has no clock and no source of random numbers. How can Alice authenticate Bob? Justify your answer.
B) Alice has a clock. Bob has a source of random numbers but no clock. How does Alice authenticate Bob? How does Bob authenticate Alice? Explain.

Problem 4:

Assuming that A and B already share a long-term key symmetric key K, this protocol has two goals: (1) allow A and B to agree on a new fresh session key Ks and (2) perform mutual authentication. Na is a nonce generated by Alice and Ks is the new session key generated by Bob. Assume that Ks and Na are of the same bit-length and both are always chosen at random. Notation E(K, ….) means that everything is encrypted with K using a strong cipher such as AES-256.

1. A → B: E(K, Na,A,B,"Hello")
2. B → A: E(K, Ks,Na,B)
3. A → B: E(K, Ks,Na,A)

Does this protocols have any security problems? If so, identify them and modify the protocol to avoid them. If not, explain!

Problem 5:

A) Recall the Fiat-Shamir identification protocol. In its description in class, it was claimed that some globally trusted authority (third party) generates the modulus ***n = pq***. What would happen if the Prover (P) learns p and q? What would happen if the Verifier (V) learns them?

B) How would Fiat-Shamir protocol work (if at all) if instead of demonstrating knowledge of a square root of I mod N, (i.e., S), the prover would demonstrate knowledge of a cube root of I mod N? In other words, we replace squaring by cubing in all protocol actions.