CS 5490/6490: Network Security – Fall 2015
Homework 2
Due by 1:25 PM MT, Wednesday September 23rd 2015

- cs5490 = 23 points, cs6490 = 32 points
- Programming Questions (for both cs5490 and cs6490) = 10 points
- No cheating will be tolerated. No copying from the Internet is allowed.
- No extensions will be granted.
- The code for programming questions and the output files must be submitted using the *handin* utility.

Question 1:
(a) (2 points) Suppose Alice sends a message to Bob by representing each alphabetic character as an integer between 0 and 25 (A -> 0, B -> 1,..., Z -> 25) and then encrypting each number separately using RSA with a large e and a large n. Describe an efficient attack against this encryption method.

(b) (3 points) Consider a simple authentication protocol using hash functions where Bob sends a random number $r_A$ to Alice and Alice responds with $\text{SHA512}(K_{AB}|r_A)$. Here $K_{AB}$ is the secret key shared by Alice and Bob and $K_{AB}|r_A$ means $K_{AB}$ concatenated with $r_A$. Bob authenticates Alice by computing $\text{SHA512}(K_{AB}|r_A)$ and comparing his result with the response of Alice. Suppose instead of $\text{SHA512}(K_{AB}|r_A)$ we used $\text{SHA512}(K_{AB} \text{OR } r_A)$. Would that be secure? What if we use $\text{SHA512}(K_{AB} \text{XOR } r_A)$?

Question 2 (Diffie Hellman Key Establishment):
(a) (2 points) Let there be n people in a group. Each person in the group wishes to establish a secret with every other person in the group. Let us assume that each person can send broadcast messages to reach all the other members of the group. Show an efficient Diffie-Hellman exchange that allows each member of the group to establish a secret with every other member of the group. How many broadcast messages does your scheme use?

(b) (2 points) Encrypting the Diffie-Hellman value with the other side’s public key prevents the person-in-the-middle attack. Why is this the case, given that the attacker can encrypt whatever it wants with the other side’s public key?

Question 3:
(a) (3 points) Use the Euclid’s algorithm described in Section 7.4 of the textbook to show that the numbers 173 and 901 are relatively prime. Find the values of $u$ and $v$ by drawing a table similar to the one in Section 7.4. What is the multiplicative inverse of 173 modulo 901? (Note: Remember to include your table in your homework.)

(b) (2 points) Consider the following protocol: Alice sends all messages for Bob encrypted with Bob’s RSA public key, and Bob sends all messages for Alice encrypted with Alice’s RSA public key. Does this protocol have perfect forward...
secrecy? Explain briefly (no credit without explanation). [Here, the private keys of Alice and Bob are their longer-term secrets.]

**Question 4 (Zero Knowledge) 9 points:**
(a) (6 points) Design your own zero knowledge proof system for interactive authentication using the ideas presented in Section 6.8 of the textbook. You must present arguments to show that your scheme is secure. (You can find a long list of NP-complete problems in the book by Michael Garey and David Johnson.)
(b) (3 points) Transform your scheme into a zero knowledge signature scheme and also show that your signature scheme is secure.

**Question 5 (Required for cs6490 students, extra-credit for cs5958 students) 9 points:**
(a) (2 points) Briefly describe the properties of the wireless channel between a pair of wireless nodes that enable these nodes to extract a symmetric/secret bit sequence?
(b) (2 points) What is the similarity between the secret key extraction presented in this paper and the Diffie-Hellman cryptosystem?
(c) (3 points) In this paper, the authors have evaluated secret key extraction in different types of environments. Which type of environment is best suited for secret key extraction? Why?
(d) (2 points) What is predictable channel attack? Which settings are more vulnerable to predictable channel attack? Why?

**Programming Question 10 points:** Write a program (in C, C++, C#, Python, or Java), call it EXPO, to efficiently exponentiate big numbers modulo n as described in Section 6.3.4.1 of your textbook. Your program should take three positive numbers as input, one representing the number m, another representing the exponent d to which m is raised to and the third being n. Write client server programs (in C, C++, C#, Python, or Java) using TCP sockets where the client (Alice) and the server (Bob) perform a Diffie Hellman exchange. Use your EXPO program (or function call) to compute the Diffie Hellman numbers. Let \( g = 19070, p = 784313, S_A = 160009 \) (Alice’s secret), and \( S_B = 12007 \) (Bob’s secret). Turn in your programs and the output using handin (directory HW2). Your output must show the numbers sent by Alice and Bob for the Diffie Hellman exchange as well the shared key after the exchange.