CSE 484 (Spring 2012), Homework 2. Due May 25, 5pm.

Please include your name and UWNetID on each page of your submission.

1. Mathematical Fundamentals: Modular Arithmetic and Multiplicative Groups.

(a) Compute 17^{574} (mod 2013) using the simple algorithm 17*17*17*...*17 (you may use a computer). Then answer these questions: What is the result? How many multiplication operations are invoked? How many non-trivial modulus operations were invoked (i.e., how many times did you reduce modulo 2013 in your calculations?

(b) Compute 17^{574} (mod 2013) using the squaring method described in lecture. Answer these questions: How many multiplication operations are invoked? How many non-trivial modulus operations? Please show your work (you can use a computer, but show each step of the calculation.

(c) What are the subgroups generated by 3, 10, and 22 in the multiplicative group of integers modulo p=23? How many elements are in each subgroup?

2. Diffie-Hellman (*Cryptography Engineering*, problem 11.4).

Consider the Diffie-Hellman protocol shown in the Lecture 16 slide deck.

What problems, if any, could arise if Alice uses the same x and g^x for all her communications with Bob, and Bob uses the same y and g^y for *all* his communications with Alice?

3. RSA Improvements (*Cryptography Engineering*, problem 12.6).

To speed up decryption, Bob has chosen to set his private key d = 3 and computes e as the inverse of d modulo phi(n). Is this a good design decision?

4. RSA Key Strength (*Cryptography Engineering*, problem 12.7).

Does a 256-bit RSA key (a key with a 256-bit modulus, i.e., n) provide strength similar to that of a 256-bit AES key?

5. RSA Implementation (*Cryptography Engineering*, problem 12.8).

Consider the RSA primitive. Let p = 71, q = 89, and e = 3.

(a) What is n?
(b) What is phi(n)?
(c) The private exponent d is one of these values: 1103, 4107, 5917. Which is it, and how do you know?
(d) Compute the signature on m_1 = 5416, m_2 = 2397, and $m_3 = m_1m_2$ (mod n) using the basic RSA operation. Show that the third signature is equivalent to the product of the first two signatures. Please show your work. If you use MATLAB, Wolfram|Alpha, Python, or something similar, please show each command you execute and the resulting response.