Written problems:

1. Every day, Alice and Bob perform Diffie-Hellman key exchange, using public parameters p and g (the textbook's Diffie-Hellman reference table is provided at the back of the exam packet). Unfortunately, Alice and Bob do not randomize their secret numbers well.

   On Monday, Alice sends Bob the number A, Bob sends Alice the number B, and they establish a shared secret S. On Tuesday, Alice sends A', Bob sends B', and they establish a shared secret S'. Eve examines the numbers A, B, A', B', and discovers the following facts (resulting from poor random number generation).

$$\begin{aligned} A' &\equiv Ag \pmod{p} \\ B' &= B^2 \pmod{p} \end{aligned}$$

   Show that if Eve manages to learn Monday's shared secret S, then she can quickly determine Tuesday's shared secret S' as well.

   More precisely, describe a procedure Eve could follow to efficiently compute the number S'. You may assume that Eve knows p, g, A, B, A', B', and S. Do not assume that Eve knows (or can learn) Alice and Bob's secret numbers a or b. You do not need to write your solution as a program, but be clear about any algorithms Eve will require in her computation, and explain why your method will work.

2. Textbook exercise 2.10, parts (a), (b), and (c). (On a three-transmission cryptosystem)

3. Write a function that reduces the problem of breaking the cryptosystem in the previous problem to the Diffie-Hellman problem. That is, assumed that you have an efficient function `dhOracle(p,g,A,B)` that extracts the shared secret from the public parameters and transmitted values in Diffie-Hellman, and use it to write a function `analyze210(p,u,v,w)` that would efficiently find the plaintext m in the system from exercise 2.10. It is fine to write the code by hand. (Obviously I cannot autograde it because I am unwilling to confirm or deny that I have a Diffie-Hellman oracle at this time.)

   Hint. The reduction is a little tricky to find; think about all the different ways you could match up the information you know with the g, A, B from Diffie-Hellman.

4. Let $p \geq 3$ be a prime number, and let g be a primitive root modulo p. For any $h \in (\mathbb{Z}/p\mathbb{Z})^\times$, denote by $\log_g h$ the solution $x \in \{0, 1, \cdots, p-2\}$ to the congruence $g^x \equiv h \pmod{p}$.

   (a) Prove that this notation is well-defined, i.e. prove that for each choice of h there is a *unique* $x \in \{0, 1, \cdots, p-2\}$ solving this discrete logarithm problem.

   (b) Prove that $\log_g h$ is even if and only if h has a squre root modulo p (a "square root modulo p" is a solution to the congruence $x^2 \equiv h \pmod{p}$).

   (c) Prove that for all $h_1, h_2 \in (\mathbb{Z}/p\mathbb{Z})^\times$,

$$\log_g(h_1 h_2) \equiv \log_g h_1 + \log_g h_2 \pmod{p-1}.$$

5. Use the babystep-giantstep algorithm to solve each of the following discrete logarithm problems. Show your calculations, e.g. in the form of the table on page 83 of the textbook.
   (a) $10^x \equiv 13 \pmod{17}$
   (b) $15^x \equiv 16 \pmod{37}$
   (c) $5^x \equiv 72 \pmod{97}$

Programming problems:

Note There are no programming problems this week, since you are preparing for the midterm exam on Friday 3/10.