Homework 17, due November 3

(1) (Page 215, problem 8) Suppose you have a random 500-digit prime p. Some people want to store passwords, written as numbers. If x is the password, then the number $2^x \pmod p$ is stored in a file. When y is given as a password, the number $2^y \pmod p$ is compared with the entry for the user in the file. Suppose someone gains access to the file. Why is it hard to deduce the passwords? If instead p is chosen to be a five digit prime, why would the system not be secure?

(2) Use the Baby Step, Giant Step method to compute $L_3(11)$ for $p = 401$. Show your work.

(3) Use the Pohlig-Hellman algorithm to compute $L_2(28)$ for $p = 37$. Show your work.

(4) (Page 216, problem 12) Consider the following Baby Step, Giant Step attack on RSA, with public modulus n. Eve knows a plaintext m and a ciphertext c. She chooses $N^2 \geq n$ and makes two lists: The first list is $c^j \pmod n$ for $0 \leq j < N$. The second list is $mc^{-Nk} \pmod n$ for $0 \leq k < N$.
   (a) Why is there always a match between the two lists, and how does a match allow Eve to find the decryption exponent d?
   (b) Your answer to the first part may be partly false. What Eve has really found is an exponent d such that $c^d \equiv m \pmod n$. Explain why the d you find may not be the decryption exponent. (Usually d is very close to being the correct decryption exponent.)
   (c) Why is this not a useful attack on RSA? (Hint: How long are the lists? Compare to trial division.)