Homework 5

This problem set is new this year, and it's somewhat different from the problem sets you've had so far in this class. Please let your TAs know about any bugs, typos, or autograder issues, and also let us know if you have any feedback (positive or negative) about this assignment!

We've provided starter code for you to use; after reading this PDF, fill in the missing functions in `ps5_submission.py`, and turn in your code to the Gradescope assignment "Problem Set 5".

In this problem set, you'll first review how Diffie-Hellman key exchange works.

Then you'll implement an algorithm that solves the discrete log problem mod p faster than brute-force (but still not fast enough to be practical when p is very large).

Finally, you'll use your discrete log solver to break the security of a Diffie-Hellman handshake that uses an unsafely-small value of p.

Problem 1 [10 points] Warm-up: Finish the Diffie-Hellman handshake!

You and Alice are using Diffie-Hellman key exchange to agree on a shared secret key. The prime modulus p and generator g have already been agreed on.

Alice has sent you $g^a \bmod p$. You've chosen your secret value b at random. What do you send to Alice? And what is your shared secret with Alice?

Problem 2 [50 points] Finding discrete logs mod p in $O(\sqrt{p})$ time

Recall that, given g, p, and $X = g^x \bmod p$, the *discrete log problem* is to recover the value of x. For appropriately chosen p, it's believed to be hard, and an efficient way to find discrete logs would let you break much of modern cryptography.

A brute-force approach for solving the discrete log problem would be to try all possible values of x until you find one where $g^x \equiv X \bmod p$. This would take $O(p)$ time.

In Lecture 13, we covered an algorithm called Baby-Steps-Giant-Steps that solves discrete log in $O(\sqrt{p})$ time and $O(\sqrt{p})$ space. This is much faster than brute force.

Your task is to implement the Baby-Steps-Giant-Steps algorithm. You'll use it in the next question to break a Diffie-Hellman handshake that uses too small a value of p.

Problem 3 [40 points] Breaking Diffie-Hellman by breaking discrete log

You're the attacker Eve, eavesdropping on Alice and Bob's communication. Alice and Bob just ran Diffie-Hellman to agree on a shared secret. Then, using the Diffie-Hellman shared secret to derive a symmetric encryption key, Alice encrypted a message and sent the ciphertext to Bob.

Unfortunately for Alice and Bob, the prime p they used in Diffie-Hellman was much too small.

Your task is to recover the plaintext (i.e., find the shared secret and then decrypt the ciphertext Alice sent to Bob) in $O(\sqrt{p})$ time.

p is large enough that brute force will be too slow to run on the autograder, but your $O(\sqrt{p})$-time discrete log solver from the previous question will be fast enough!
