650.445/600.454: Practical Cryptographic Systems October 24, 2018

Assignment 3, Part 1

Instructor: Matthew Green *Due: 11:59 pm November 7, 2018*

Name: _____

The assignment must be completed individually. You are permitted to use the Internet and any printed references, but your code must be your own!

Please submit the completed assignment via Blackboard.

Problem 1: Implementing the Diffie-Hellman cryptosystem (40 points)

For this problem you will produce a Python or Go program that implements the Diffie-Hellman key exchange protocol at a 1024-bit key strength. This will consist of parameter generation, share exchange, and key derivation. Normally the D-H scheme is an interactive protocol between two parties. However, for this implementation we will use files as our communication channel. As in the previous problems, your implementation should be "from scratch", although you are welcome to re-use any of the code you wrote in the previous parts of this assignment.

Your implementation will consist of three separate programs. The first models Alice's initial message to Bob, and outputs a secret key to be stored for later. The second models Bob's receipt of the message from Alice, and outputs a response message back to Alice. The final program models Alice's receipt of Bob's response. For grading purposes you will hand in the following programs.

`dh-alice1 <filename for message to Bob> <filename to store secret key>`.
Outputs decimal-formatted (p, g, g^a) to Bob, writes (p, g, a) to a second file.

`dh-bob <filename of message from Alice> <filename of message back to Alice>`.
Reads in Alice's message, outputs (g^b) to Alice, prints the shared secret g^{ab}.

`dh-alice2 <filename of message from Bob> <filename to read secret key>`.
Reads in Bob's message and Alice's stored secret, prints the shared secret g^{ab}.

The output of your programs should consist of *decimal formatted* integers separated by commas and parentheses, as in the previous problems. Your shared secret (integers) should be printed to `stdout`. All of the relevant parameters should be generated randomly each time you run the program.