Coding and Cryptography Fall 2016
Mission #5
Due Friday, September 30

Katherine E. Stange

September 23, 2016

Note: This project consists of implementing various things on Sage. Please, to the extent possible, each team member should *individually* successfully implement the same things, with help from one another to write working code. Code needn't be identical, but should perform the same thing. Don't just copy code: the goal is to learn some coding if you are new to it.

1. Your group should, as a group, create a public and private key pair. Rules:
   (a) Use primes which are about 50 bits in size.
   (b) You may NOT use Sage's inbuilt primality testing, but you MAY use Sage's inbuilt modular exponentiation.
   (c) Therefore you should implement Fermat primality testing by writing a small loop.
   (d) Remember to use several bases to be sure you actually have a prime.
   (e) Use good practices in choosing primes.
2. Now, please
   (a) Trade your public keys with another group
   (b) Create a message to send to the other group. This *isn't* best practice, but for today just use the usual numbers 0 through 26 for letters, and concatenate to create a moderate message (a few words), e.g. ABC would be 000102 (which is in fact just 102... leading 'A's may get lost from messages).
   (c) Send each other encrypted messages.
   (d) Decrypt the message received.
3. Now, try to attack the other group's private key. Try using Fermat factorization, giving the computer five minutes runtime to search for squares. You must write a simple loop to execute this, you may use Sage's inbuilt function:

   ```
   is_square()
   ```

   Record whether it worked.
4. Next, use Sage's inbuilt factoring capability to factor your partner team's public modulus, and then use this to determine their private key. Check if you are right by using it to decrypt the message you sent to them.
5. For your writeup, please include your Sage code and the results (screenshots are fine), with just enough surrounding info for me to be able to follow what is going on.