Problem Set 4

Due: Lesson 24

(50 pts)

Help Policy

AUTHORIZED RESOURCES: Any, except another cadet's program.

NOTE:

- Never copy another person's work and submit it as your own.
- Do not jointly create a program.
- You must document all help received from sources other than your instructor or instructor-provided course materials (including your textbook).
- **DFCS will recommend a course grade of F for any cadet who egregiously violates this Help Policy or contributes to a violation by others.**

1. [5] Is 5 a primitive root of 37? Show your work using modular exponentiation. (HINT: Don't raise 5 to all possible positive values in the mod space – take the shortcut!)

2. [5] Suppose the key generation algorithm in your implementation of RSA sometimes generates a large *p* or *q* that is composite. What effect will this have? Be specific.

3. [5] Your friend is using an RSA like encryption scheme in which they encrypt a message *m* by computing $c \equiv m^3 (mod\ 101)$. What is your decryption exponent such that $m \equiv c^d (mod\ 101)$? (Note: Your *n*, 101, is prime). Is this as strong as RSA? Why or why not?

4. [5] Suppose two users, Alice and Bob, have the same RSA modulus *n* and suppose their encryption exponents, e_A and e_B, are relatively prime. Charles wants to send the same message, *m*, to both Alice and Bob so he encrypts to get $c_A \equiv m^{e_A} (mod\ n)$ and $c_B \equiv m^{e_B} (mod\ n)$. Show how Eve can determine *m* if she intercepts c_A and c_B.

5. [5] Show all the steps of key generation, encrypting and decrypting with RSA, when the primes you choose are 17 and 11, the public exponent is 7, and the message is 4. Clearly identify the public key, private key, ciphertext, etc.

6. [5] How many prime divisors do you need to check to determine whether or not 149 is prime? Use them to determine the primality of 149, showing your work.

7. [10] Use Pollard's Rho Algorithm to factor 1961. Show all the (a,b) pairs that led to a successful factorization, the corresponding values of gcd(|a-b|,n), and provide the final factorization. (Note: Copy/paste from Excel is acceptable to show your work)

8. [10] (Recommend program or Excel) Use the Miller-Rabin primality test algorithm to check the following numbers *n* for primality: {19, 41}. Show all the steps. Just run the algorithm once for each n, with a witness of your choice (ie: t = 1).

4.24 Algorithm Miller-Rabin probabilistic primality test

MILLER-RABIN(n,t)
INPUT: an odd integer $n \geq 3$ and security parameter $t \geq 1$.
OUTPUT: an answer "prime" or "composite" to the question: "Is n prime?"

1. Write $n-1 = 2^s r$ such that r is odd.
2. For i from 1 to t do the following:
   2.1 Choose a random integer a, $2 \leq a \leq n-2$.
   2.2 Compute $y = a^r \bmod n$ using Algorithm 2.143.
   2.3 If $y \neq 1$ and $y \neq n-1$ then do the following:
       $j \leftarrow 1$.
       While $j \leq s-1$ and $y \neq n-1$ do the following:
           Compute $y \leftarrow y^2 \bmod n$.
           If $y = 1$ then return("composite").
           $j \leftarrow j+1$.
       If $y \neq n-1$ then return ("composite").
3. Return("prime").