**Homework 00**

Question 1 (20pts)

1. Explain the difference between message secrecy and message integrity? How are they related? Justify your answer.

Message secrecy (aka. Confidentiality) deals with keeping outside people/groups from gaining access to internal information. Message integrity deals with preserving the content of the internal information.

Generally, in a secure system, you would want both. You want to keep people from reading into your personal files AND keep them from altering your information, such as when trying to keep internal company communications secret and correct. However, information can also maintain data integrity without secrecy, such as when communicating with others in public message forums in which the previous messages are not altered from their original message and anyone can gain access to them.

1. Give an example of where a symmetric key system would be preferable to a public key system, and vice versa. Justify your answer.

The main benefits/cons of key systems deal with two aspects: key distribution and key management. Symmetric key systems are ideal for systems few private users because key management/distribution is relatively easy, keys are simple to generate, and keeps communication strictly to the intended parties. Asymmetric keys, on the other hand, are more ideal for systems that have a bigger user base (such as organizations and communities) because key management becomes exponentially harder with a larger user base on symmetric keys with each user added whereas asymmetric keys only become more complex in a constant rate.

Question 2 (20pts)

1. Compute: gcd(25 \* 33 \* 56 \* 112 , 24 \* 33 \* 72 \* 111) = 4752
2. Using the Euclidean Algorithm, compute: gcd(161733, 234175) = 29

234175 = 1\*161733 +72442  
161733 = 2\*72442 + 16849  
72442 = 4\*16849 + 5046  
16849 = 3\*5046 + 1711  
5046 = 2\*1711 + 1624  
1711 = 1\*1624 + 87  
1624 = 18\*87 + 58  
87 = 1\*58 + 29  
58 = 2\*29 + 0

Question 3 (30pts)

This exercise shows that the Euclidean algorithm computes the gcd.  
Let a, b, qi, ri be as in Section 3.1.

1. Let d be a common divisor of a, b. Show that d | r1, and use this to show that d | r2.

Because

r1 = a – bq1   
d|a,b

Therefore

d|r1

Because

r2 = b – q2r1  
d|b,r1

Therefore

d|r2

1. Let d be as in 1. Use induction to show that d | ri for alli. In particular, that d | r2.

Given

d|r1,…,rj

Because

rj+1 = rj-1 – qj+1rj  
d|rj-1,rj

Therefore

d|rj+1

Therefore

d|ri for all i

1. Use induction to show that rk | r1 for 1 <= i <= k.

Because

rk-1 = qk+1rk

Therefore

rk|rk-1

Given

rk|rk-1 for i = 1,2,…,j  
Because

rk-j-1 = qk-j+1rk-j + rk-j+1  
rk|rk-j,rk-j+1

Therefore

rk|rk-j-1

Therefore

rk|ri for all i

1. Using the facts that rk | r1 and rk | r2, show that rk | b and then rk | a. Therefore, rk is a common divisor of a, b.

Because

b = q2r1 + r2  
rk|r1,r2

Therefore

rk|b

Because

a = q1b + r1  
rk|a,r1

Therefore

r|a

1. Use 2. to show that rk >= d for all common divisors d, and therefore rk is the greatest common divisor.

Because

d|rk for each common divisor

Therefore

rk >= d

Question 4 (30pts)

1. Find the smallest integer x >= 50 for which  
    < 0.06  
   where π(x) is the prime-counting function.

Using both brute force and Mathematica I was able to find the result of x = 142595394.

e.g. Brute Force:

The answer is between the range…

(100000000, 200000000)  
 (140000000, 150000000)  
 (142000000, 143000000)  
 (142500000, 142600000)  
 (142590000, 142600000)

Once finding a moderately small range of numbers to test I started testing from 142590000 until a suitable answer was found by Mathematica using the following code:

x = 142590000  
primeCount = PrimePi[x]  
result = N[Abs[1 - primeCount/(x/Log[x])]]  
  
For[i = x, (result = N[(Abs[1 - PrimePi[i]/(i/Log[i])])]) >= 0.06, i++, Print["Integer: ", i, " => ", result]]  
Print["Smallest integer: ", i]