Homework 6 (due 5/21 by midnight, end of day)
CSC 333

We saw our first public-key message exchange using Shamir's Three Pass Protocol, based on just Fermat's Little Theorem (Shamir's protocol is not covered in the book, you'll have to work with the lecture notes/videos). We then covered the math required for RSA, which included Euler's Theorem and the Chinese Remainder Theorem. We then saw RSA, and saw some simple attacks. Next week we will see more attacks on RSA.

Handing it in: Submit to the d2l submission folder. Use some standard format for your homework (Word or pdf, for example), and make sure to include your name and assignment number. Include screenshots for code and test runs. Do not attach the code (or any executables). Please include all work in a single file (and don't use zip). The class information folder contains a sample homework submission (see SampleHWSubmission.docx or SampleHWSubmission.pdf). Note how I cropped and resized the screenshots to make them more readable.

---

1. (Reading Assignment) Read Sections 7.1-7.4 (we already covered 7.4 earlier) and 7.8. If you want to read ahead, continue with chapter 7.

2. (Three Pass Protocol Break, 10pt) You've been listening to Alice and Bob who used the Three Pass Protocol to send a secret message. You saw the following sequence of messages exchanged:

A->B: 28815377349986238948
B->A: 32022638409929718780
A->B: 14438564975518228697

You know that Alice and Bob work with p = 91246234312872996521, and you bribed Chris who works for both of them for more information. You didn't pay enough though, so instead of giving you Alice's secret a or Bob's secret b, you just got (a*b) modulo (p-1) which is 529891231244498430 69. With that determine the message that Alice sent to Bob. *Hint*: Think first. Write down algebraic expressions for the messages that Alice and Bob exchanged (in terms of the secret message m, and the parameters a, b, and their inverses mod p-1). How does knowing (a*b) mod (p-1) help? *Hint 2*: the code for computing modular inverses is in d2l, and you can also use Wolfram Alpha.

3. [Chinese Remainder Theorem, 10pt] Use the method of the Chinese Remainder Theorem to solve the following problems. Show your work.

a) [6pt] Find x (between 0 and 2438 * 4247) such that

x ≡ 211 (2438), and x ≡ 3304 (4247).

b) [4pt] Find x (between 0 and 2438 * 4247 * 7123) such that

x ≡ 211 (2438), x ≡ 3304 (4247) and x ≡ 6614 (7123).

Hint: do a) before b). Be sure to check your solution for x by calculating the remainders. You can use the program we saw for computing inverses.

4. (RSA, 10pt) Working with primes
(p,q) = (556069583727568975173209, 871978615963638608184613 9)

a) [4pt] Set up an RSA system for Alice. Explicitly state $\Phi(n)$ and (e,n) and (d,n). Show how you computed d. *Note*: you can use any of the programs we saw in class.

b) [3pt] Play Bob and send the message m = 122333221 to Alice using her public key.

c) [3pt] Verify that Alice's private key correctly decrypt Bob's message.

5. (RSA, 10pt) Alice published her public key as

(e,n) = (39405811308673791996710325942266660618965628910278434721 5,
79011076080967421418792642560804478402840628852334142531 41)

you observed Bob sending her the ciphertext

ct = 188825782244703357388227158315163319070779972759273610795 6

(encrypted with Alice's public key). You paid an ex-employee of Alice, and got Alice's phi(n) which turns out to be

phi(n) = 79011076080967421418792642555483895014388343621930400843 84 .

1. [5pt] What was the message m? *Hint*: first determine d. You will need modular exponentiation and inverses.
2. [5pt] Break the system entirely, by finding primes p and q so that n = p*q. *Hint*: use the quadratic equation: x^2 - (n - phi(n) + 1) x + n = 0, and show the details of how you solve it. You can use the code for calculating sqrt I showed in class (it's on d2l).