CS 70 Discrete Mathematics for CS Fall 2003 Wagner — HW 6

Due Thursday October 16

You do not need to give formal proofs on this homework. However, you should explain your answers.

1. **(5 pts.) Any questions?**
   Is there anything you'd like to see explained better in lecture or discussion sections?

2. **(25 pts.) Chinese remainder theorem**
   Solve the following equations for x modulo the indicated modulus, or show that no solution exists.

   (a) Suppose $x \equiv 3 \pmod 5$, $x \equiv 4 \pmod 7$. Find x mod 35.

   (b) Suppose $x \equiv 3 \pmod 6$, $x \equiv 6 \pmod 7$, $x \equiv 2 \pmod 8$. Find x mod 504.

   (c) Suppose $x \equiv 1 \pmod p$, $x \equiv 0 \pmod q$, and $\gcd(p,q) = 1$. Describe an efficient algorithm $\text{UNIT}(p,q)$ to compute x mod pq given p and q as input.

   (d) Suppose $x \equiv a \pmod p$, $x \equiv b \pmod q$, and $\gcd(p,q) = 1$. Describe an efficient algorithm $\text{CRT}(p,q,a,b)$ to compute x mod pq given p,q,a,b as input.

3. **(15 pts.) Roots**
   Here is a basic fact about the roots of polynomials in modular arithmetic (which you may assume without proof):

   Theorem 0.1: *Let p be prime, and $f(X)$ be a non-zero polynomial of degree d. Then the equation $f(x) \equiv 0 \pmod p$ has at most d solutions for x* mod *p.*

   In other words, any non-zero polynomial of degree d has at most d roots modulo p. Moreover, there are efficient algorithms to find the solutions of such equations, i.e., the roots modulo p of any given polynomial.

   With that background, do the following two problems:

   (a) Let p be prime. Argue that every value z has at most 2 square roots modulo p.

   (b) If n is a product of k distinct primes and z is relatively prime to n, how many different square roots can z have modulo n (at most)? Explain why.

4. **(20 pts.) Factoring**
   Some factoring methods are based on the following recipe: (1) first, using deep number theoretic magic, find two numbers x, y so that $x^2 \equiv y^2 \pmod n$; (2) then, use x and y to derive a factor of n. This problem is intended to show you how step (2) works. (Step (1) is usually much harder.)

   In the following, assume that $n = pq$, where p, q are two different odd primes, and assume that x, y are relatively prime to n, $x \not\equiv \pm y \pmod n$, and $x^2 \equiv y^2 \pmod n$.

   (a) Argue that x^2 has exactly 4 square roots modulo n.

(b) Argue that, if x, y are as above, then either: (i) p divides $x-y$ but not $x+y$; or, (ii) p divides $x+y$ but not $x-y$. *Hint:* $x^2-y^2=(x-y)(x+y)$.

(c) Argue that, if x, y are as above, then either $\gcd(x-y,n)$ or $\gcd(x+y,n)$ discloses a prime factor of n and thus enables one to factor n.

5. (15 pts.) Squaring

In practical implementations of RSA, it is common to use $e=3$ as the public exponent, because this provides performance enhancements.

Could we use $e=2$ for RSA encryption? Why or why not?

6. (20 pts.) Insecure RSA

Suppose we modified RSA to use $n=2^k$ as our public modulus. Explain why this variation on RSA is insecure. In other words, describe how an adversary could decrypt encrypted ciphertexts without knowing the private key.