Outline for May 13, 2004

Reading: Chapter 9.3.2, 9.4, 12.1–12.2.2

Discussion Problem

Microsoft spent February of last year teaching its programmers how to check their code for security vulnerabilities and how to introduce common security flaws. Yet many Microsoft programs still have security vulnerabilities. Wha problems do you think Microsoft encountered, and will encounter, in trying to find and clean up the vulnerabilities in its systems?

Outline for the Day

1. RSA
   a. Provides both authenticity and confidentiality
   b. Go through algorithm:
      Idea: $C = M^e \bmod n$, $M = C^d \bmod n$, with $ed \bmod \phi(n) = 1$.
      Proof: $M^{\phi(n)} \bmod n = 1$ [by Fermat's theorem as generalized by Euler]; follows immediately from $ed \bmod \phi(n) = 1$.
      Public key is (e, n); private key is d. Choose $n = pq$; then $\phi(n) = (p–1)(q–1)$.
   c. Example: $p = 5$, $q = 7$; $n = 35$, $\phi(n) = (5–1)(7–1) = 24$. Pick $d = 11$. Then $de \bmod \phi(n) = 1$, so choose $e = 11$.
      To encipher 2, $C = M^e \bmod n = 2^{11} \bmod 35 = 2048 \bmod 35 = 18$, and $M = C^d \bmod n = 18^{11} \bmod 35 = 2$.
   d. Example: $p = 53$, $q = 61$, $n = 3233$, $\phi(n) = (53–1)(61–1) = 3120$. Take $d = 791$; then $e = 71$. Encipher M = RENAISSANCE: A = 00, B = 01, …, Z = 25, blank = 26. Then:
      M = RE NA IS SA NC Eblank = 1704 1300 0818 1800 1302 0426
      $C = (1704)^{71} \bmod 3233 = 3106$; *etc.* = 3106 0100 0931 2691 1984 2927
2. Cryptographic Checksums
   a. Function $y = h(x)$: easy to compute y given x; computationally infeasible to compute x given y
   b. Variant: given x and y, computationally infeasible to find a second x' such that $y = h(x')$.
   c. Keyed *vs*. keyless
3. Authentication
   a. Basis: what you know/have/are, where you are