Key Exchanges, Digital Signatures and Public Key Cryptography

MA 410

April 11, 2011

Where It All Began

- "New Directions in Cryptography", by Whitfield Diffie and Martin Hellman (November 1976)
- Defined public key cryptosystem: a pair of families of algorithms, $\{E_K\}$ and $\{D_K\}$ (representing invertible transformations on a "message space"), such that
  1. For each K, E_K is the inverse of D_K
  2. For each K and each M (message), E_K and D_K are easy to compute
  3. For almost all K, any equivalent to D_K is computationally infeasible to derive from E_K
  4. For each K, it is feasible to compute inverse pairs E_K and D_K from K.
- Note: Item (3) implies that E_K may be made public *without* compromising the security of D_K
- Had the setup, but no instantiation

Where It All Began

A "suggestive, although unfortunately useless, example" (using linear algebra)

- Represent the "plaintext" message as a binary n-vector m
- Multiply by an invertible binary $n \times n$ matrix E, so $E_K(m) = Em = c$ ("ciphertext")
- Letting $D = E^{-1}$, decrypt via $D_K(c) = Dc = E^{-1}Em = m$
- Easy to generate E and D (from identity matrix)
- Downside: matrix-vector multiplication takes about $\sim n^2$ operations, and matrix inversion takes about n^3 operations (not a good ratio)

Public Key Distribution System

"Diffie-Hellman Key Exchange"

Alice wants to send Bob a message, using a secret key that only she and Bob know.

- Alice and Bob agree on a prime p and a primitive root α in $\mathbb{Z}_p$
- Alice picks a secret $x \in \{1, 2, \ldots, p-1\}$ and computes $\alpha^x \mod p$
- Bob picks a secret $y \in \{1, 2, \ldots, p-1\}$ and computes $\alpha^y \mod p$
- Exchange: $\text{Alice} \overset{\alpha^y}{\underset{\alpha^x}{\rightleftarrows}} \text{Bob}$
- Alice computes $(\alpha^y)^x$ mod p, Bob computes $(\alpha^x)^y$ mod p

  Fermat's Little Theorem: $\alpha^{p-1} \equiv 1 \bmod p$
- Only α^x, α^y are transmitted
- Security relies on *discrete log problem*

Digital Signatures

Easy to recognize, difficult to forge

Can use a public key cryptosystem:

- Alice has $E_A : M \mapsto C$ (public), $D_A : C \mapsto M$ (private).
  Bob has E_B, D_B.
- Alice sends Bob $D_A(M)$, as opposed to $E_B(M)$
- Bob computes $E_A(D_A(M)) = M$ (E_A is public)
- Only Alice knows D_A (forgery is difficult)
- Everyone knows E_A (recognition is easy)
- Note: D_A is private, but *examples* of $D_A(M)$ are public
  "known plaintext attack"

RSA Cryptosystem

An instantiation of a public key cryptosystem

- Rivest, Shamir, Adelman (1978)

  [Also: Cocks, Ellis, Williamson (1973) with GCHQ, UK's equivalent of NSA]
- Uses Euler's (Generalization of Fermat's Little) Theorem:
  If $\gcd(a, n) = 1$, then $a^{\phi(n)} \equiv 1 \bmod n$, where
  $\phi(n) = \{m \in \mathbb{Z}_n : \gcd(m, n) = 1\}$. (Theorem 7.5 in ENT, 7th ed.)
- $\phi(n) = n \cdot \prod_{p|n} \left(1 - \frac{1}{p}\right)$ (Theorem 7.3 in ENT, 7th ed.)

  For distinct primes p and q,

$$\phi(pq) = pq\left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right) = (p-1)(q-1)$$

RSA Cryptosystem

How it works

Alice wants to send a secret message (encoded as a number M) to Bob.

- Bob picks two (large) primes, p and q, and sets $n = pq$
- Bob picks e ("encoding exponent") such that $\gcd(e, \phi(n)) = 1$
- Bob computes d ("decoding exponent") such that $de \equiv 1 \bmod \phi(n)$
- Bob publishes (e, n), keeps (p, q) secret
- Alice computes $c \equiv M^e \bmod n$, sends c to Bob
  (If $M \geq n$, then break into blocks smaller than n)
- Bob computes (use "$\equiv_n$" for "congruent modulo n")

$$c^d \equiv_n (M^e)^d \equiv_n M^{t\cdot\phi(n)+1} \equiv_n (M^{\phi(n)})^t \cdot M$$
$$\overset{\text{Euler}}{\equiv_n} 1^t \cdot M \equiv_n M \qquad\qquad M < n \Rightarrow c^d = M$$

- One catch: Euler's Theorem assumes $\gcd(M, n) = 1$

RSA Cryptosystem

What if $\gcd(M, n) > 1$?

- Suppose $\gcd(M, n) = \gcd(M, pq) > 1$. Then either

$$p \mid M \text{ and } q \mid M, \quad \text{or (WLOG) } p \mid M \text{ but } q \nmid M.$$

- Suppose $p \mid M$ but $q \nmid M$, so $M^{ed} \equiv_p 0$ and $\gcd(M, q) = 1$.

$$M^{ed} = M^{\phi(n)\cdot t+1} = (M^{(p-1)(q-1)})^t \cdot M$$
$$= (M^{q-1})^{t(p-1)} \cdot M \overset{\text{Euler}}{\equiv_q} M$$

- Set $x = M^{ed}$. Then $x \equiv_p 0$, $x \equiv_q M$, and $\gcd(p, q) = 1$.

Recall: Chinese Remainder Theorem

Theorem 4.8 in ENT, 7th ed.

Chinese Remainder Theorem

Let $n_1, n_2, \dots, n_r$ be positive integers such that $\gcd(n_i, n_j) = 1$ for $i \neq j$. Then the system of linear congruences

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ &\vdots \\ x &\equiv a_r \pmod{n_r} \end{aligned}$$

has a simultaneous solution, which is unique modulo the integer $n_1 n_2 \cdots n_r$.

RSA Cryptosystem

What if $\gcd(M, n) > 1$?

- Suppose $\gcd(M, n) = \gcd(M, pq) > 1$. Then either

$$p \mid M \text{ and } q \mid M, \quad \text{or (WLOG) } p \mid M \text{ but } q \nmid M.$$

- Suppose $p \mid M$ but $q \nmid M$, so $M^{ed} \equiv_p 0$ and $\gcd(M, q) = 1$.

$$\begin{aligned} M^{ed} &= M^{\phi(n)\cdot t+1} = (M^{(p-1)(q-1)})^t \cdot M \\ &= (M^{q-1})^{t(p-1)} \cdot M \overset{\text{Euler}}{\equiv_q} M \end{aligned}$$

- Set $x = M^{ed}$. Then $x \equiv_p 0$, $x \equiv_q M$, and $\gcd(p, q) = 1$.
- By CRT, there is unique $\bar{x}$ mod pq such that $\bar{x} \equiv_p 0$, $\bar{x} \equiv_q M$.
- $\bar{x} \equiv M$ mod n is a solution, hence *the* solution. $(M < n \Rightarrow \bar{x} = M)$
- If $p \mid M$ and $q \mid M$, then $M \equiv_n 0$ (contradicting $0 \leq M < n$)

RSA Cryptosystem
How secure is it?

- Security/efficiency depends on ease of exponentiation and difficulty of factoring $n = pq$

  With p and q, can find d ($de \equiv_{\phi(n)} 1$) via Euclidean Algorithm
- (ENT, 7th ed.) A 200-digit number can be tested for primality in 20 seconds, but the quickest factoring algorithm takes about 1.2×10^{23} operations for the same size number.
  - At 10^{-9} operations per second (1 GHz), it would take about 3.8×10^6 years. "...appears to be quite safe."
  - RSA-129: $100 prize offered by R, S, and A; 129-digit encoding modulus; factored in 1994 by 600 volunteers running over 1600 computers for 8 months; "The magic words are squeamish ossifrage."
  - RSA Challenge List (42 numbers, posted in 1991); most recent, 193-digit factorization (two primes, 95 digits each); inactive as of 2007

RSA Cryptosystem
Malleability

- Say M itself starts as a number (e.g., a bid on a product)
- Eve hears $C \equiv_n M^e$
- Suppose $\gcd(100, n) = 1$

  $[n = pq$, so if $\gcd(100, pq) > 1$, then $p \in \{2, 5\}]$

  Then there exists $100^{-1} \bmod n$
- Eve sends

$$C \cdot (101 \cdot [100^{-1} \bmod n])^e \equiv_n M^e \cdot 101^e \cdot 100^{-e} \equiv_n \left(M \cdot \frac{101}{100}\right)^e$$

- Outbids by 1%!

Attacking the RSA

- Suppose p, q, $p^{-1} \bmod q$, $q^{-1} \bmod p$ are stored on a microchip, and suppose $M^e \bmod n$ is computed in a particular way:

- After the computation of $q(q^{-1}C \bmod p)$, toss the micro*chip* in the micro*wave* at the "$p^{-1}C \bmod q$" step:

$$\widetilde{C} \equiv_n q(q^{-1}C \bmod p) + p(G \bmod q)$$

- $C - \widetilde{C} = p[(p^{-1}C - G) \bmod q]$ (divisible by p, but not q)
- $\gcd(C - \widetilde{C}, pq) = p$
- "Differential Fault Analysis"; Boneh, DeMillo, Lipton (Sep 1996)

Recall: Chinese Remainder Theorem (Proof)

Theorem 4.8 in ENT, 7th ed.

Setup: $\gcd(n_i, n_j) = 1$ for $i \neq j$. Then $x \equiv_{n_1} a_1$, $x \equiv_{n_2} a_2$, ..., $x \equiv_{n_r} a_r$ has unique solution $\bar{x} \bmod n_1 n_2 \cdots n_r$.

- Let $n = n_1 n_2 \cdots n_r$, and let $N_k = \dfrac{n}{n_k}$, so that $\gcd(N_k, n_k) = 1$.
- Then there exists x_k such that $N_k x_k \equiv_{n_k} 1$. $[x_k = N_k^{-1} \bmod n_k]$
- Let $\bar{x} = a_1 N_1 x_1 + a_2 N_2 x_2 + \cdots + a_r N_r x_r$.
  Note that $N_i \equiv_{n_k} 0$ for $i \neq k$, but $N_k x_k \equiv_{n_k} 1$.
- $\bar{x} \equiv_{n_k} a_k N_k x_k \equiv_{n_k} a_k \cdot 1 \equiv_{n_k} a_k$ for each k
- For RSA, $n = pq$, $N_p = \dfrac{n}{p} = q$, $N_q = \dfrac{n}{q} = p$.
- Then $x \equiv_p C$, $x \equiv_q C$ has unique solution $\bar{x} \bmod pq$:

$$C \cdot q \cdot (q^{-1} \bmod p) + C \cdot p \cdot (p^{-1} \bmod q)$$

Attacking the RSA

- Suppose p, q, $p^{-1} \bmod q$, $q^{-1} \bmod p$ are stored on a microchip, and suppose $C \equiv M^e \bmod n$ is computed in a particular way:

$$C \equiv_n q(q^{-1}C \bmod p) + p(p^{-1}C \bmod q).$$

- After the computation of $q(q^{-1}C \bmod p)$, toss the micro*chip* in the micro*wave* at the "$p^{-1}C \bmod q$" step:

$$\widetilde{C} \equiv_n q(q^{-1}C \bmod p) + p(G \bmod q)$$

- $C - \widetilde{C} = p[(p^{-1}C - G) \bmod q]$ (divisible by p, but not q)
- $\gcd(C - \widetilde{C}, pq) = p$
- "Differential Fault Analysis"; Boneh, DeMillo, Lipton (Sep 1996)

The ElGamal Cryptosystem

Taher ElGamal (1985)

- RSA security: difficult to factor large numbers
- ElGamal security: difficult to solve *discrete log problem*:
  Find x, $0 < x < \phi(n)$, such that $r^x \equiv_n y$
  ("`log()`" button won't work)
- RSA: public exponent, private (factored) modulus
- ElGamal: public (prime) modulus, private exponent(s)

The ElGamal Cryptosystem
How it works

Alice wants to send a secret message (encoded as a number M) to Bob.

- Bob picks a prime p and a *primitive root* r (so that $r^x \equiv_p y$ has a solution for all $y \in \mathbb{Z}_p$)
- Bob picks (random) $k \in \{2, 3, \ldots, p-2\}$ and computes $a \equiv_p r^k$, where $a \in \{0, 1, \ldots, p-1\}$
- Bob publishes (a, r, p), keeps k secret
- Alice picks (random) $j \in \{2, 3, \ldots, p-2\}$ and computes

  $$C_1 \equiv_p r^j, \quad C_2 \equiv_p Ma^j \equiv_p M(r^k)^j,$$

  and sends C_1, C_2 to Bob
- Bob computes

  $$\begin{aligned} C_2 C_1^{p-1-k} &\equiv_p M(r^k)^j (r^j)^{p-1-k} \equiv_p M r^{kj} r^{j(p-1)-kj} \\ &\equiv_p M r^{kj} r^{-kj} (r^{p-1})^j \equiv_p M(r^{p-1})^j \overset{\text{Fermat}}{\equiv_p} M \end{aligned}$$

The ElGamal Cryptosystem
Features

- Can use same k, j (hence, C_1) for each block, or change for each block (no need to tell other party)
- Bob never announces k, Alice never announces j

  Two private exponents, one public modulus
- Capitalizes on difficulty of discrete log problem
- Can be used for digital signatures as well (ENT §10.3, 7th ed.)

The ElGamal Cryptosystem

Malleability

- Alice (rightfully) sends $C_1 \equiv_p r^j$, $C_2 \equiv_p Ma^j$
- Eve hears C_1 and C_2, then sends

$$C_1' \equiv_p r^{j'} C_1 \equiv_p r^{j'} r^j \equiv_p r^{j'+j}$$
$$C_2' \equiv_p \lambda a^{j'} C_2 \equiv_p \lambda a^{j'} Ma^j \equiv_p \lambda Ma^{j'+j}$$

- Properly decrypts as λM

Happy Encrypting/Decrypting!