Public Key Cryptosystems

Guevara Noubir
http://www.ccs.neu.edu/home/noubir/Courses/CSG252/F05

Textbook: "Cryptography: Theory and Applications",

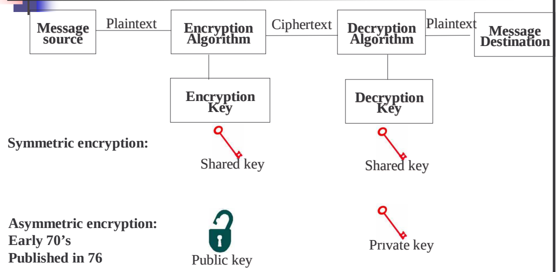
Douglas Stinson, Chapman & Hall/CRC Press, 2002

Reading: Chapter 5 upto section 5.7

Outline

- Concepts behind public key crypto
- Some number theory
- RSA cryptosystem
- Primality testing
- Factoring numbers and other attacks

Encryption Models



Cannot provide unconditional security

Applications

- Symmetric algorithms vs. asymmetric algorithms (public-key crypto systems)
  - About 1000 times faster!
  - However, require a shared key!
- Practice:
  - Use public key crypto to establish a shared key
  - Examples
    - Email:
      - Choose a key for the symmetric algorithm *K*, encrypt it with the public key of the destination
      - Use the key K to encrypt the message and integrity protect it
    - IPSec/IKE:
      - IKE: establish a session key (using either public-key cryptosystem or shared secrets)
      - IPSec uses the session key to provide confidentiality and integrity

Number Theory

- Z_n^*: abelian group of numbers < n, relatively prime to n
- Euclidean Algorithm (a, b):
  - Computes the gcd(a, b)
- Extended Euclidean Algorithm(a, b):
  - Computes r, s, t s.t. $sa + bt = r = \gcd(a, b)$
  - If $r = 1 \Rightarrow s = a^{-1} \bmod b$
    - If $r \neq 1 \Rightarrow$?
- Time complexity less than $O(k^3)$ if a and b are encoded in less than k bits.

Chinese Remainder Theorem

- Assume that $m_1, \ldots, m_r$ are pairwise relatively prime positive integers
- Chinese Remainder Theorem (CRT):
  - Suppose $a_1, \ldots, a_r$ are integers s.t.
    - $x \equiv a_1 \pmod{m_1}$
    - $x \equiv a_2 \pmod{m_2}$
    - ...
    - $x \equiv a_r \pmod{m_r}$
  - There exists a unique x mod $m_1 m_2 \ldots m_r$ that satisfies all previous equations

$$x = \sum_{i=1}^{r} a_i M_i y_i \bmod M \qquad M_i = M / m_i;\ y_i = M_i^{-1}$$

Other Known Results

- If G is a multiplicative group of order n then the order of any element of G divides n
- Order of $Z_n^* = \phi(n)$
- If $b \in Z_n^*$, then $b^{\phi(n)} \equiv 1 \pmod{n}$
- How about when n is prime?
- If p is prime then Z_p^* is a cyclic group

RSA Cryptosystem

- Due to Rivest-Shamir-Adleman in 1977
- Let $n = pq$, where p and q are primes
- $P = C = Z_n$
- $K = \{(n, p, q, a, b) : ab \equiv 1 \pmod{\phi(n)}\}$
- Encryption:
  - $e_k(x) = x^b \bmod n$
- Decryption:
  - $d_k(y) = y^a \bmod n$
- Public key: n and b
- Private key: p, q, a

Example

- $p = 101;\ q = 113 \Rightarrow n = 11413$
- $\phi(n) = 11200 = 2^6 5^2 7$
- Let $b = 3533 \Rightarrow b^{-1} = 6597$
  - How is b chosen?
- Encrypt plaintext: 9726
  - Ciphertext = $9726^{3533} \bmod 11413 = 5761$
- Decryption ciphertext: 5761
  - Plaintext = $5761^{6597} \bmod 11413 = 9726$

Use of RSA

- Encryption (A wants to send a message M to B):
  - A uses the public key of B and encrypts M (i.e., $e_{kB}(M)$)
  - Since only B has the private key, only B can decrypt M (i.e., $M = d_{kB}(M)$)
- Digital signature (A wants to send a signed message to B):
  - Based on the fact that $e_{kA}(d_{kA}(M)) = d_{kA}(e_{kA}(M))$
  - A encrypts M using its private key (i.e., $d_{kA}(M)$) and sends it to B
  - B can check that $e_{kA}(d_{kA}(M)) = M$
  - Since only A has the decryption key, only him can generate this message

Security of RSA

- Security of RSA is based on the belief that:
  - x^b mod n is a one-way function
- The trapdoor is the knowledge of the factorization of n into pq
- Conjecture:
  - RSA is as difficult as factoring numbers

RSA Implementation

- RSA Parameters Generation
  - Generate two large primes: p, q
  - $n \leftarrow pq$, and $\phi(n) \leftarrow (p-1)(q-1)$;
  - Choose a random b ($1 < b < \phi(n)$) s.t. $\gcd(b, \phi(n)) = 1$
  - $a \leftarrow b^{1} \bmod \phi(n)$
  - Public key is (n, b) and private key is (p, q, a)
- p and q should be **at least 512 bits long each**
  - $\Rightarrow n$ is at least 1024 bits long
- Computation Complexity:
  - Exponentiation cost: SQUARE-AND-MULTIPLY
    - $(m_1)^c \bmod n$ can be computed in $O(\log(c) \times k^2)$
  - Modular inverse: Extended Euclidean Alg.
    - $(m_1)^{-1} \bmod n$ can be computed in $O(k^3)$
  - Modular Muliplication:
    - $(m_1 m_2) \bmod n$ can be computed in $O(k^2)$

Prime Numbers Generation

- Density of primes (prime number theorem):
  - $\pi(x) \sim x/\ln(x)$
  - E.g., a random number of 512 bits has probability: 1/ln(512) = 1/355 to be prime
- Sieve of Erathostène
  - Try if any number less than SQRT(n) divides n
- Fermat's Little Theorem does not detect Carmichael numbers
  - $b^{n-1} \equiv 1 \bmod n$
  - E.g., 561 is the smallest Carmichael number
- Solovay-Strassen primality test
  - If n is not prime at least 50% of b fail to satisfy the following:

$$b^{(n-1)/2} \bmod n = \left(\frac{b}{n}\right)$$

  - Jacobi symbol can be computed in less than $O((\log n)^3)$
  - Jacobi symbol is a generalization of the Legendre symbol:

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a \equiv 0 \bmod p \\ 1 & \text{if } a \text{ is a quadratic residue} \bmod p \\ -1 & \text{if } a \text{ is a quadratic non-residue} \bmod p \end{cases}$$

Computing Jacobi Symbol

- Definition: $n = \prod_{i=1}^{k} p_i^{ei} \Rightarrow \left(\frac{a}{n}\right) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{ei}$
- No need to factor n to compute the Jacobi symbol
- Use the following rules [n is positive odd]:
  - $m_1 \equiv m_2 \pmod{n} \Rightarrow \left(\frac{m_1}{n}\right) = \left(\frac{m_2}{n}\right)$
  - $\left(\frac{2}{n}\right) = \begin{cases} 1 & \text{if } n \equiv \pm 1 \pmod 8 \\ -1 & \text{if } n \equiv \pm 3 \pmod 8 \end{cases}$
  - $\left(\frac{m_1 m_2}{n}\right) = \left(\frac{m_1}{n}\right)\left(\frac{m_2}{n}\right)$
  - $\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{if } m \equiv n \equiv 3 \pmod 4 \\ \left(\frac{n}{m}\right) & \text{otherwise} \end{cases}$

- Rabin-Miller primality test
  - If n is not prime then it is not pseudoprime to at least 75% of random $a < n$:
    - n-1 = $2^k m$,
    - $b \leftarrow a^m \bmod n$;
    - **If** $b \equiv 1 \bmod n$ **then return**(n prime)
    - **For** i=0 to k-1 **do**
      - **If** $b \equiv -1 \bmod n$ **then return**(n prime)
      - **Else** $b \leftarrow b^2$;
    - **return(**n composite)
  - Probabilistic test, deterministic if the Generalized Riemann Hypothesis is true
- Deterministic polynomial time primality test [Agrawal, Kayal, Saxena'2002]

Attacks on RSA

- Factoring
  - Many factoring algorithms were proposed: quadratic sieve, elliptic curve factoring, number field sieve, Pollard's rho-method
  - Capable of factoring a 512 bits modulus ≈ 155 digits in 1999 using 8400 MIPS-years
- Other attacks:
  - Computing $\phi(n)$
  - Decryption exponent: if a is known!
    - Las Vegas algorithm (5.10) that will factor n with probability ½
- Semantic Security

Rabin Cryptosystem

- Motivation:
  - The difficulty of factoring does not necessarily prove RSA security
  - Hardness of factoring leads to security proof of Rabin's cryptosystem against chosen-plaintext attack
- Scheme:
  - $n = pq$ (p and q are two primes and $p \equiv q \equiv 3 \bmod 4$)
  - $P = C = Z_n^*$; $K = \{(n, p, q)\}$
  - $e_K(x) = x^2 \bmod n$
  - $d_K(y) = \sqrt{y} \bmod n$
- Note:
  - Conditions: $p \equiv q \equiv 3 \bmod 4$ and Z_n^* is for simplification of decryption and security proof purpose

Rabin Cryptosystem

- Observation:
  - Is the encryption function injective?
    - Solution?
- How can we decrypt?
  - Solution: CRT
  - Consider x s.t.:

$$x \equiv \pm y^{(p+1)/4} \bmod p$$

$$x \equiv \pm y^{(q+1)/4} \bmod q$$

  - $x^2 \equiv y \bmod n$
  - When can we use this technique of decoding?
  - Example:
    - $n = 7\text{x}11$
    - Decrypt $y = 23$

Security of Rabin Cryptosystem

- If Rabin cryptosystem can be broken then we can build a Las Vegas probabilistic algorithm with success probability ½
- Rabin Oracle Factoring(n)
  - External RabinDecrypt
  - Choose a random r;
  - Let $y \leftarrow r^2$;
  - $x \leftarrow$ RabinDecrypt(y);
  - **If** $x = \pm r \bmod n$ **return(**failure**)**
  - **Else return(**p=gcd(x+r, n) ; q=n/p**);**
- Conclusion:
  - Rabin cryptosystem is secure against a chosen plaintext attack
- Additional security results:
  - Rabin cryptosystem is insecure against a chosen ciphertext attack