Public Key Cryptosystems

Guevara Noubir
http://www.cs.ncl.ac.uk/home/node/CSC230V64

Reading: Chapter 5 optics section 5.7

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

Encryption Models

Symmetric encryption:
- Shared key
- Cannot provide unconditional security

Asymmetric encryption:
- Public key
- Private key
- Published in 76
- Early 70's
- Cannot provide unconditional security
Applications

- Symmetric algorithms vs. asymmetric algorithms (public-key cryptography)
  - 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 recipient
      - Use the key $K$ to encrypt the message and integrity protect it
    - IPSec (IKE):
      - Initial establish a session key (using either public-key cryptosystem or shared secrets)
      - IPSec uses the session key to provide confidentiality and integrity

Number Theory

- $\mathbb{Z}_n^*$: abelian group of numbers $< n$, relatively prime to $n$
- Euclidean Algorithm $(a, b)$:
  - Computes the gcd$(a, b)$
- Extended Euclidean Algorithm $(a, b)$:
  - Computes $r, c, s$ s.t. $sa + cb = r = \gcd(a, b)$
  - If $r = 1 \Rightarrow s = a \mod b$
  - If $r = 1 \Rightarrow 0$
- Time complexity less than $O(k^2)$ if $a$ and $b$ are encoded in less than $k$ bits.

Chinese Remainder Theorem

- Assume that $m_1, \ldots, m_i$ are pairwise relatively prime positive integers
- Chinese Remainder Theorem (CRT):
  - Suppose $a_1, \ldots, a_i$ are integers s.t.
    - $x = a_i \mod m_i$
    - $x = a_{i+1} \mod m_{i+1}$
    - $\ldots$
    - $x = a_i \mod m_i$
  - There exists a unique $x \mod m_1m_2...m_i$ that satisfies all previous equations
    - $x = \sum_{j=1}^{i} a_j y_j \mod \prod_{j=1}^{i} m_j$
    - $M_j = M \mod m_j$
    - $y_j = M_j^{-1}$
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 $b^{\phi(n)} = 1$ (mod $n$)
- 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
- $P = \mathbb{Z}_n^*$
- $K = \{ (n, p, q, a, b) : ab = 1 \ (mod \ \phi(n)) \}$
- Encryption:
  - $e(x) = x^a \mod n$
- Decryption:
  - $d(y) = y^b \mod n$
- Public key: $n$ and $b$
- Private key: $p$, $q$, $a$

Example

- $p = 101$; $q = 113 \Rightarrow n = 11413$
- $\phi(n) = 11200 = 2^5 \times 7$
- Let $b = 3533 \Rightarrow b^4 = 6597$
  - How is $b$ chosen?
- Encrypt plaintext: 9726
  - Ciphertext = 9726$^{1023}$ mod 11413 = 5761
- Decryption ciphertext: 5761
  - Plaintext = 5761$^{1023}$ mod 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., \(e_B(M)\)).
  - Since only B has the private key, only B can decrypt \(M\) (i.e., \(M = d_B(e_B(M))\)).

- Digital signature (A wants to send a signed message to B):
  - Based on the fact that \(e_A(d_B(M)) = d_A(e_A(M))\).
  - A encrypts \(M\) using its private key (i.e., \(d_A(M)\)) and sends it to B.
  - B can check that \(e_A(d_B(M)) = M\).
  - Since only A has the decryption key, only he can generate this message.

Security of RSA

- Security of RSA is based on the belief that:
  - \(x^e \mod n\) 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: \(p, q\).
  - \(n = pq\), and \(\Phi(n) = (p-1)(q-1)\).
  - Choose a random \(e\) (\(1 < e < \Phi(n)\)) such that \(\text{gcd}(e, \Phi(n)) = 1\).
  - \(d = e^{-1}\) mod \(\Phi(n)\).
  - Public key is \((n, e)\) and private key is \((n, d)\).

- \(p\) and \(q\) should be at least 512 bits long each.
  - \(n\) is at least 1024 bits long.

- Computation Complexity:
  - Exponentiation cost: \(\text{SQUARE-AND-MULTIPLY}\)
  - \((x^n) \mod n\) can be computed in \(O(\log(n)\log^2 \log(n))\) steps.
  - Modular Inversion: Extended Euclidean Algorithm.
  - \((x^{-1}) \mod n\) can be computed in \(O(n)\) steps.
  - Modular Multiplication:
    - \((xy) \mod n\) can be computed in \(O(n)\) steps.
Prime Numbers Generation

- Density of primes (prime number theorem):
  - \( \pi(x) = x / \ln(x) \)
  - For, \( n \) random number of 512 bits has probability: \( L(n(512) = 1/205 \) to be prime
- Sieve of Eratosthenes
- Try if any number less than \( \sqrt{n} \) divides \( n \)
- Fermat's Little Theorem does not detect Carmichael numbers
  - \( a^{\phi(n)} = 1 \mod n \)
  - \( \phi(n) \) is the smallest Carmichael number
- Solovay-Strassen primality test
  - If \( n \) is not prime at least 50% of \( a \) fail to satisfy the following: \( a^{(n-1)/2} \equiv (\frac{a}{n}) \mod n \)
  - Jacobi symbol can be computed in less than \( O(\log(n)^2) \)
  - Jacobi is a generalization of the Legendre symbol
  - Probability of the Solovay-Strassen primality test failing to detect a composite number is less than: \( (\ln n / 2) / n \cdot 2^{-k} \)

Rabin-Miller primality test

- If \( n \) is not prime then it is not pseudoprime to at least 75% of random \( a \):
  - \( a^m = 2\mod m \)
  - \( b = a^{2^r} \mod n \)
  - For \( r \) to \( k \) do
    - If \( b = 1 \mod n \) then return(n prime)
    - Else if \( b = n - 1 \mod n \) then return(n prime)
    - Else \( b = b^2 \mod n \)
  - 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 8460 MIPS-years
- Other attacks:
  - Computing \( \phi(n) \)
  - Decryption exponent: if \( a \) is known!
    - Las Vegas algorithm (5.10) that will factor \( n \) with probability \( 1/2 \)
  - 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 \( p \equiv q \equiv 3 \mod 4 \))
- \( P = C = \mathbb{Z}_n^* ; \ K = \{ n, p, q \} \)
- \( e(x) = x^2 \mod n \)
- \( d(y) = y^{(n-1)/2} \mod n \)

Note:
- Conditions: \( p \equiv q \equiv 3 \mod 4 \) and \( \mathbb{Z}_n^* \) 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 \( x \) such that:
    - \( x = \pm y^{(n-1)/4} \mod p \)
    - \( x = \pm y^{(n-1)/4} \mod q \)
- \( x \mod n \)
- When can we use this technique of decoding?
- Example:
  - \( n = 7 \times 11 \)
  - 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 \( \frac{1}{2} \)
- Rabin Oracle Factoring(n)
  - External RabinDecrpyt
  - Choose a random \( r \)
  - Let \( y \leftarrow r^2 \)
  - \( x \leftarrow \text{RabinDecrpyt}(y) \)
  - \( x = \text{return}(\text{failure}) \)
  - \( \text{else return}(\gcd(x+i, n) : q \mid p) \)
- Conclusion:
  - Rabin cryptosystem is secure against a chosen plaintext attack
- Additional security results:
  - Rabin cryptosystem is insecure against a chosen ciphertext attack