Public Key Cryptography and RSA

Murat Kantarcioglu

Based on Prof. Ninghui Li's Slides

Review: Number Theory Basics

Definition An integer n > 1 is called a prime number if its positive divisors are 1 and n.

Definition Any integer number n > 1 that is not prime is called a composite number.

Theorem (Fundamental Theorem of Arithmetic)

$$n = p_1^{e_1} p_2^{e2} \ldots p_k^{ek}$$

Definition The greatest common divisor of a and b, denoted by gcd(a, b), is the largest number that divides both a and b.

Definition Two integers a > 0 and b > 0 are relatively prime if gcd(a, b) = 1.

Review: Extended Euclidian Algorithm

Input: a, b
Output: (d,x,y) s.t. d=gcd(a,b) and ax + by = d

```
d=a; t=b; x=1;  y=0; r=0;  s=1;
while (t>0) {
                              q = ⌊d/t⌋
        u=x-qr;  v=y-qs;      w=d-qt
        x=r;     y=s;         d=t
        r=u;     s=v;         t=w
}
return (d, x, y)
```

Euclidian Algorithm

Invariants:

gcd(a,b)=gcd(d,t)

ax + by = d

ar + bs = t

How many times before this loop stops?

3

Review: Chinese Reminder Theorem (CRT)

Let n_1, n_2, ,,, n_k be integers s.t. $\gcd(n_i, n_j) = 1$, $i \neq j$.

$$x \equiv a_1 \bmod n_1$$
$$x \equiv a_2 \bmod n_2$$
$$\ldots$$
$$x \equiv a_k \bmod n_k$$

There exists a unique solution modulo $n = n_1 n_2 \ldots n_k$
The solution is given by

$\rho(a1,a2,\ldots,ak) = (\Sigma\ a_i m_i y_i)$ mod n,

- where $m_i = n / n_i$, and $y_i = m_i^{-1} \bmod n_i$

4

Review: Euler Phi Function

Definition: A reduced set of residues (RSR) modulo m is a set of integers R each relatively prime to m, so that every integer relatively prime to m is congruent to exactly one integer in R.

Definition: Given n, $Z_n^* = \{a \mid 0<a<n \text{ and } \gcd(a,n)=1\}$ is the standard RSR modulo n.

Definition

Given an integer n, $\Phi(n) = |Z_n^*|$ is the size of RSR modulo n.

Theorem: If $\gcd(m,n) = 1$, $\Phi(mn) = \Phi(m)\,\Phi(n)$

Fact: $\Phi(p)=p-1$ for prime p

5

Review: Euler's Theorem

Euler's Theorem

Given integer $n > 1$, such that $\gcd(a, n) = 1$ then

$$a^{\Phi(n)} \equiv 1 \pmod{n}$$

Corollary: Given integer $n > 1$, such that $\gcd(a, n) = 1$ then $a^{\Phi(n)-1} \bmod n$ is a multiplicative inverse of a mod n.

Corollary: Given integer $n > 1$, x, y, and a positive integers with $\gcd(a, n) = 1$. If $x \equiv y \pmod{\Phi(n)}$, then

$$a^x \equiv a^y \pmod{n}.$$

Corollary (Fermat's "Little" Theorem):

$a^{p-1} \equiv 1 \pmod{n}$

6

Lecture Outline

- Why public key cryptography?
- Overview of Public Key Cryptography
- RSA
  - square & multiply algorithm
  - RSA implementation
- Pohlig-Hellman

Limitation of Secret Key (Symmetric) Cryptography

- Secret key cryptography
  - symmetric encryption $\Rightarrow$ confidentiality (privacy)
  - MAC (keyed hash) $\Rightarrow$ authentication (integrity)
- Sender and receiver must share the same key
  - needs secure channel for key distribution
  - impossible for two parties having no prior relationship
- Other limitation of authentication scheme
  - cannot authenticate to multiple receivers
  - does not have non-repudiation

Public Key Cryptography Overview

- Proposed in Diffie and Hellman (1976) "New Directions in Cryptography"
  - public-key encryption schemes
  - public key distribution systems
    - Diffie-Hellman key agreement protocol
  - digital signature
- Public-key encryption was proposed in 1970 by James Ellis
  - in a classified paper made public in 1997 by the British Governmental Communications Headquarters
- Diffie-Hellman key agreement and concept of digital signature are still due to Diffie & Hellman

9

Public Key Encryption

- Public-key encryption
  - each party has a PAIR (K, K^{-1}) of keys: K is the **public** key and K^{-1} is the **secret** key, such that
    $\mathbf{D}_{K^{-1}}[\mathbf{E}_K[M]] = M$
  - Knowing the public-key and the cipher, it is computationally infeasible to compute the private key
  - Public-key crypto system is thus known to be *asymmetric* crypto systems
  - The public-key K may be made publicly available, e.g., in a publicly available directory
  - Many can encrypt, only one can decrypt

10

Public Key Cryptography Overview

- Public key distribution systems
  - two parties who do not share any private information through communications arrive at some secret not known to any eavesdroppers
- Authentication with public keys: Digital Signature
  - the authentication tag of a message can only be computed by one user, but can be verified by many
  - called one-way message authentication in [Diffie & Hellman, 1976]

11

Public-Key Encryption Needs One-way Trapdoor Functions

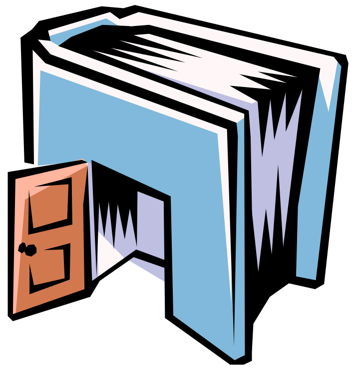
- Given a public-key crypto system,
  - Alice has public key K
  - $\mathbf{E}_K$ must be a one-way function, knowing y= $\mathbf{E}_K[x]$, it should be difficult to find x
  - However, $\mathbf{E}_K$ must not be one-way from Alice's perspective. The function $\mathbf{E}_K$ must have a trapdoor such that knowledge of the trapdoor enables one to invert it

12

Trapdoor One-way Functions

Definition:

A function f: $\{0,1\}^* \rightarrow \{0,1\}^*$ is a trapdoor one-way function iff f(x) is a one-way function; however, given some extra information it becomes feasible to compute f^{-1}: given y, find x s.t. y = f(x)



13

RSA Algorithm

- Invented in **1978** by Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman
  - Published as R L Rivest, A Shamir, L Adleman, "*On Digital Signatures and Public Key Cryptosystems*", Communications of the ACM, vol 21 no 2, pp120-126, Feb 1978
- Security relies on the difficulty of factoring large composite numbers
- Essentially the same algorithm was discovered in 1973 by Clifford Cocks, who works for the British intelligence

14

The Multiplicative Group Z_{pq}^*

- Let p and q be two large primes
- Denote their product n=pq.
- The multiplicative group $Z_n^* = Z_{pq}^*$ contains all integers in the range [1,pq-1] that are relatively prime to both p and q
- The size of the group is
  $\Phi(pq) = (p-1)(q-1) = n-(p+q)+1$
- For every $x \in Z_{pq}^*$, $x^{(p-1)(q-1)} \equiv 1$

15

Exponentiation in Z_{pq}^*

- Motivation: We want to use exponentiation for encryption
- Let e be an integer, $1<e<(p-1)(q-1)$
- When is the function $f(x)=x^e$, a one-to-one function in Z_{pq}^*?
- If x^e is one-to-one, then it is a permutation in Z_{pq}^*.

16

Exponentiation in Z_{pq}^*

- Claim: If e is relatively prime to (p-1)(q-1) then $f(x)=x^e$ is a one-to-one function in Z_{pq}^*
- Proof by constructing the inverse function of f. As gcd(e,(p-1)(q-1))=1, then there exists d and k s.t. ed=1+k(p-1)(q-1)
- Let $y=x^e$, then $y^d=(x^e)^d=x^{1+k(p-1)(q-1)}=x \pmod{pq}$, i.e., $g(y)=y^d$ is the inverse of $f(x)=x^e$.

17

RSA Public Key Crypto System

Key generation:

Select 2 large prime numbers of about the same size, p and q

Compute n = pq, and $\Phi(n) = (q-1)(p-1)$

Select a random integer e, $1 < e < \Phi(n)$, s.t. $\gcd(e, \Phi(n)) = 1$

Compute d, $1< d< \Phi(n)$ s.t. $ed \equiv 1 \bmod \Phi(n)$

Public key: (e, n)

Secret key: d

18

RSA Description (cont.)

Encryption

Given a message M, 0 < M < n $M \in Z_n - \{0\}$

use public key (e, n)

compute $C = M^e \bmod n$ $C \in Z_n - \{0\}$

Decryption

Given a ciphertext C, use private key (d)

Compute

$$C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{ed} \bmod n = M$$

RSA Example

- $p = 11, q = 7, n = 77, \Phi(n) = 60$
- $d = 13, e = 37$ ($ed = 481$; $ed \bmod 60 = 1$)
- Let M = 15. Then $C \equiv M^e \bmod n$
  - $C \equiv 15^{37} \pmod{77} = 71$
- $M \equiv C^d \bmod n$
  - $M \equiv 71^{13} \pmod{77} = 15$

Why does RSA work?

- Need to show that $(M^e)^d \pmod n = M$, $n = pq$
- We have shown that when $M \in Z_{pq}^*$, i.e., $\gcd(M, n) = 1$, then $M^{ed} \equiv M \pmod n$
- What if $M \in Z_{pq} - \{0\} - Z_{pq}^*$, e.g., $\gcd(M, n) = p$.
  - $ed \equiv 1 \pmod{\Phi(n)}$, so $ed = k\Phi(n) + 1$, for some integer k.
  - $M^{ed} \bmod p = (M \bmod p)^{ed} \bmod p = 0$
    so $M^{ed} \equiv M \bmod p$
  - $M^{ed} \bmod q = (M^{k*\Phi(n)} \bmod q)(M \bmod q) = M \bmod q$
    so $M^{ed} \equiv M \bmod q$
  - As p and q are distinct primes, it follows from the CRT that $M^{ed} \equiv M \bmod pq$

21

Square and Multiply Algorithm for Exponentiation

- Computing $(x)^c \bmod n$
  - Example: suppose that c=53=110101
  - $x^{53}=(x^{26})^2 \cdot x=(((x^3)^2)^2 \cdot x)^2)^2 \cdot x =(((x^2 \cdot x)^2)^2 \cdot x)^2)^2 \cdot x \bmod n$

Alg: Square-and-multiply $(x, n, c = c_{k-1}\, c_{k-2} \ldots c_1\, c_0)$

```
z=1
for i ← k-1 downto 0 {
        z ← z^2 mod n
        if c_i = 1 then z ← (z × x) mod n
}
return z
```

22

Efficiency of computation modulo n

- Suppose that n is a k-bit number, and $0 \leq x,y \leq n$
  - computing $(x+y)$ mod n takes time $O(k)$
  - computing $(x-y)$ mod n takes time $O(k)$
  - computing (xy) mod n takes time $O(k^2)$
  - computing (x^{-1}) mod n takes time $O(k^3)$
  - computing $(x)^c$ mod n takes time $O((\log c)\, k^2)$

23

RSA Implementation

n, p, q

- The security of RSA depends on how large n is, which is often measured in the number of bits for n. Current recommendation is 1024 bits for n.
- p and q should have the same bit length, so for 1024 bits RSA, p and q should be about 512 bits.
- P or q should not be small !

24

RSA Implementation

- Select p and q prime numbers
- In general, select numbers, then test for primality
- Many implementations use the Rabin-Miller test, (probabilistic test)

25

Next …

- Finding large prime numbers
- Attacks on RSA
- Factoring

26