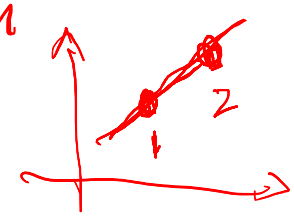
Public-Key Cryptography Algorithm
(RSA)

key reconstruct $k \geq m$

m $k < m$

RSA Public-key encryption

$y = ax + b$



secret sharing

secret

and m

what is k? → reconstruct a

$k = 2$ $m \geq 2$

- by Rivest, Shamir & Adleman of MIT in 1977
- currently the "work horse" of Internet security
  - most public key infrastructure (PKI) products
  - SSL/TLS: certificates and key-exchange
  - secure e-mail: PGP, Outlook, ….
- based on exponentiation in a finite (Galois) field over integers modulo a prime

  m^e
  - exponentiation takes $O((\log n)^3)$ operations (easy)
- security due to cost of factoring large integer numbers
  - factorization takes $O(e^{\log n \log \log n})$ operations (hard)
- uses large integers (eg. 1024 bits)

4096 → key

RSA key setup

gcd is greatest common divisor

gcd(2, 8) = 2

2×4
2×2×2
1×8

int gcd(int a, int b)
{
if (a==0)
return b;
return gcd(b%a, a)
}

remainder

- each user generates a public/private key pair by:
  - selecting two large primes at random - p, q → integer, keep private
  - computing their system modulus n=p·q
    - note ø(n)=(p-1)(q-1) → Euler's totient equation
  - selecting at random the encryption key e
    - where 1<e<ø(n), gcd(e,ø(n))=1 coprime
  - solve following equation to find decryption key d
    - ed=1 mod ø(n)
  - publish their public encryption key: pk={e,n}
  - keep secret private decryption key: sk={d,p,q} (d,n)

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

d?

How to get d?

gcd(10, 3) = 1

gcd(24, 56) = 6
a b

2
24)56
48
8

gcd(6, 24) → 6

4
6)24
24
0
return 6.

$e \cdot d \equiv 1 \mod \phi(n)$. ①

d always exist.

Based on Bezout's identity

For any integers a, b not both zero, there always exist integer x, y

such that

$$ax + by = \gcd(a, b). \quad ②$$

modular definition

$$\underset{a}{e}\,\underset{x}{d} + \underset{y}{k} \cdot \underset{b}{\phi(n)} = \underset{\gcd(e, \phi(n))}{1} \qquad k \geq 0 \quad k \in \mathbb{Z} \quad ③$$

can always find d, k

① Inverse algorithm

Extended Euclidean Algorithm when $\gcd(a, b) = 1$

② $1 \leq d < \phi(n)$ & $d \in \mathbb{Z}$

for (int $d=1$; $d<\phi(n)$; d++)

check $ed \equiv 1 \mod \phi(n)$.

RSA key generation – security

- users of RSA must:
  - determine two primes at random - p, q → secret.
  - select either e or d and compute the other
- primes p, q must not be easily derived from modulus n=p.q
  - means must be sufficiently large
  - typically guess and use probabilistic test
- exponents e, d are inverses, so use Inverse algorithm to compute the other

Attacker has PK, n

factorization

RSA example

Key Generation	
Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

1. Select primes: $p=17$ & $q=11$
2. Compute $n = pq = 17 \times 11 = 187$
3. Compute $\varnothing(n) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e, 160) = 1$; choose $e=7$
5. Determine d: $de = 1 \bmod 160$ and $d < 160$ Value is $d=23$ since $23 \times 7 = 161 = 10 \times 160 + 1$
6. Publish public key $pk = \{7, 187\}$
7. Keep secret private key $sk = \{23, 17, 11\}$

RSA use

- to encrypt a message M the sender:
  - obtains **public key** of recipient `pk={e,n}`
  - computes: $C=M^e \bmod n$, where $0 \le M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key `sk={d,p,q}`
  - computes: $M=C^d \bmod n$
- note that the message M must be smaller than the modulus n (block if needed)

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

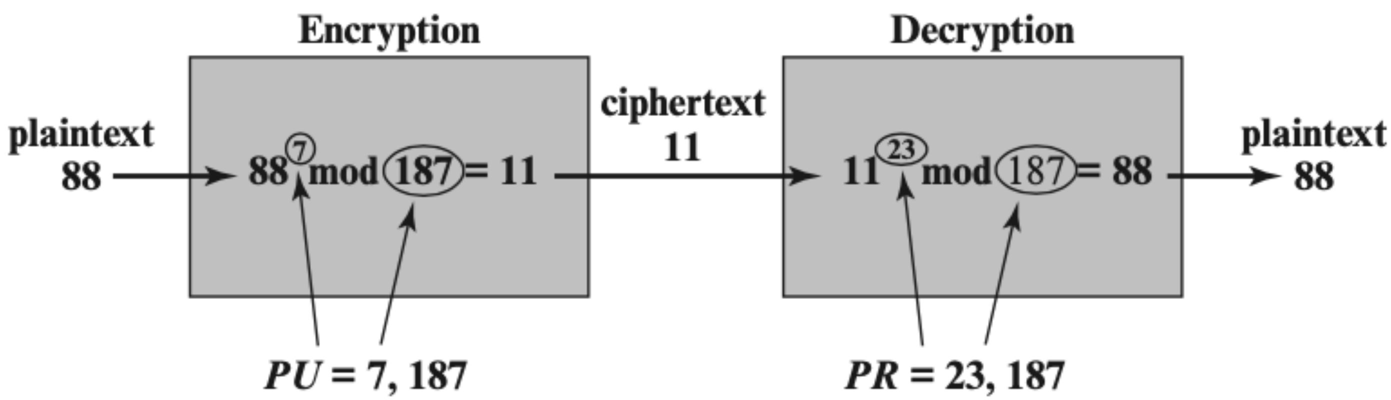
Decryption	
Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

$$\text{Plaintext} \xrightarrow{C = M^e \bmod n} \text{Ciphertext}$$

$$\text{Plaintext} \xleftarrow{M = C^d \bmod n} \text{Ciphertext}$$

`pk={e,n}` `sk={d,p,q}`

Example of RSA algorithm



Correctness of RSA

$y = x^2 \quad x_2$

$M' \stackrel{?}{=} M$

coprime

- Euler's theorem: if gcd (M, n) = 1, then $M^{\phi(n)} = 1$ mod n. Here ϕ(*n*) is Euler's totient function: the number of integers in {1, 2, . . ., *n*-1} which are relatively prime to *n*. When *n* is a prime, this theorem is just Fermat's little theorem

$C = M^e \bmod n$ encryption $\quad (M^e)^d = M^{e \cdot d}$

decryption

$$M' = C^d \bmod n = M^{ed} \bmod n$$

$ed = 1 \bmod \phi(n)$

$$= M^{k\phi(n)+1} \bmod n$$

$$= [M^{\phi(n)}]^k \cdot M \bmod n$$

$[1]^k = 1 \bmod n$

$$= M \bmod n$$

Encryption	
Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

$M \leftarrow$ encode

$M < n$ coprime & padding

Prob = $1/p + 1/q$

$M \to \gcd(M, n) = 1$