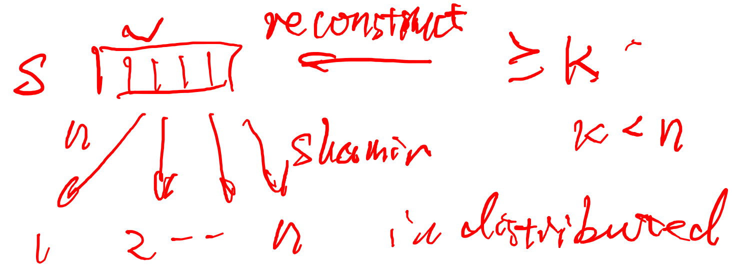
Public-Key Cryptography Algorithm

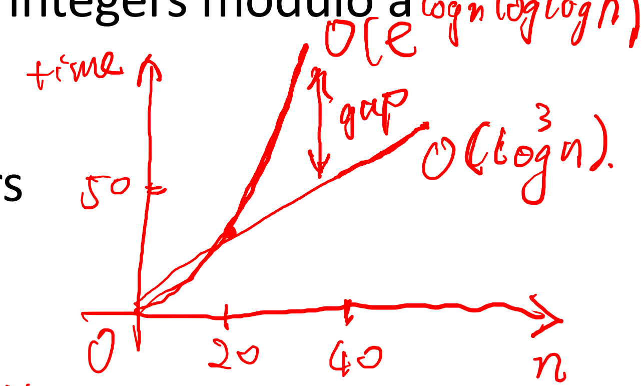
(RSA)

RSA Public-key encryption

→ secret sharing



- 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
  - exponentiation takes $O((\log n)^3)$ operations (easy) $\le O(n^3)$
- 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)



≥ 1024 bits ↑ 4096 bits

RSA key setup

gcd → greatest common divisor

gcd(8, 12) = 4

```
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
  - computing their system modulus n=p·q
    - note ø(n)=(p-1)(q-1) Euler's Totient function
  - selecting at random the encryption key e ← pk.
    - where 1<e<ø(n), gcd(e,ø(n))=1 → coprime
  - solve following equation to find decryption key d relative prime.
    - ed=1 mod ø(n)
  - publish their public encryption key: pk={e,n}
  - keep secret private decryption key: sk={d,p,q}

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 exist?

if Yes, how to get d?

gcd(9, 2) = 1

~~prime~~

gcd(24, 54)

54 ÷ 24 = 2, 54 − 48 = 6

gcd(6, 24)

24 ÷ 6 = 4, 24 − 24 = 0

return 6

$\gcd(e, \phi(n)) = 1 \mod \phi(n)$.

To prove, use contradiction

Assume $\gcd(e, \phi(n)) = \underline{h > 1}$ $\quad h \in \mathbb{Z}$

By definition

$$e = k_1 \cdot h \qquad k_1 \in \mathbb{Z} \qquad (1)$$

$$\phi(n) = k_2 \cdot h \qquad k_2 \in \mathbb{Z} \qquad (2)$$

$\because \quad ed = 1 \mod \phi(n)$

$\Rightarrow \quad ed = 1 + k \cdot \phi(n) \qquad k \in \mathbb{Z}$

By definition of modular

substitute e and $\phi(n)$ with (1), (2)

$$\underbrace{k_1 \cdot h d}_{\text{A multiple of } h} = \underbrace{1}_{\text{A multiple of } h} + \underbrace{k \cdot k_2 \cdot h}_{\text{A multiple of } h}$$

$\Rightarrow$ contradiction

Inverse algorithm

1. Extended Euclidean Algorithm

$$ed + k \cdot \phi(n) = 1 \mod \phi(n),$$

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

for (int d=1; d<ϕ(n); d++)

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

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