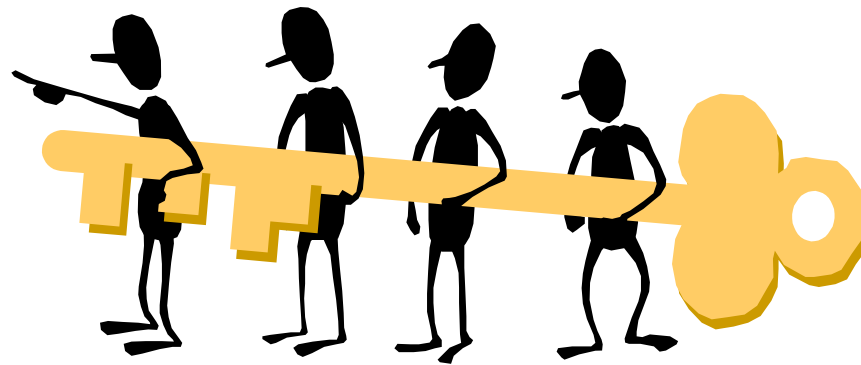# Public Key Cryptography and RSA



Raj Jain

Washington University in Saint Louis

Saint Louis, MO 63130

Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

http://www.cse.wustl.edu/~jain/cse571-17/

# Overview

1. Public Key Encryption
2. Symmetric vs. Public-Key
3. RSA Public Key Encryption
4. RSA Key Construction
5. Optimizing Private Key Operations
6. RSA Security

These slides are based partly on Lawrie Brown's slides supplied with William Stallings's book "Cryptography and Network Security: Principles and Practice," 7th Ed, 2017.

# Public Key Encryption

- Invented in 1975 by Diffie and Hellman at Stanford
- Encrypted_Message = Encrypt(Key1, Message)
- Message = Decrypt(Key2, Encrypted_Message)

Key1        Key2

Text $\longrightarrow$ Ciphertext $\longrightarrow$ Text

- Keys are **interchangeable**:

Key2        Key1

Text $\longrightarrow$ Ciphertext $\longrightarrow$ Text

- One key is made **public** while the other is kept **private**
- Sender knows only public key of the receiver $\Rightarrow$ **Asymmetric**

Ref: http://en.wikipedia.org/wiki/Public-key_cryptography

# Public Key Encryption Example

- Rivest, Shamir, and Adleman at MIT
- RSA: Encrypted_Message = $m^3$ mod 187
- Message = Encrypted_Message$^{107}$ mod 187
- Key1 = <3,187>, Key2 = <107,187>
- Message = 5
- Encrypted Message = $5^3$ = 125
- Message = $125^{107}$ mod 187 = 5
  $= 125^{(64+32+8+2+1)}$ mod 187
  $= \{(125^{64} \bmod 187)(125^{32} \bmod 187)...$
  $(125^2 \bmod 187)(125 \bmod 187)\}$ mod 187

Ref: http://en.wikipedia.org/wiki/RSA

# Symmetric vs. Public-Key

| Conventional Encryption | Public-Key Encryption |
|---|---|
| *Needed to Work:*<br><br>1. The same algorithm with the same key is used for encryption and decryption.<br><br>2. The sender and receiver must share the algorithm and the key.<br><br>*Needed for Security:*<br><br>1. The key must be kept secret.<br><br>2. It must be impossible or at least impractical to decipher a message if no other information is available.<br><br>3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key. | *Needed to Work:*<br><br>1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.<br><br>2. The sender and receiver must each have one of the matched pair of keys (not the same one).<br><br>*Needed for Security:*<br><br>1. One of the two keys must be kept secret.<br><br>2. It must be impossible or at least impractical to decipher a message if no other information is available.<br><br>3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key. |

# Public-Key Authentication and Secrecy

| B's Public Key | A's Private Key | Message |
|---|---|---|

A → B
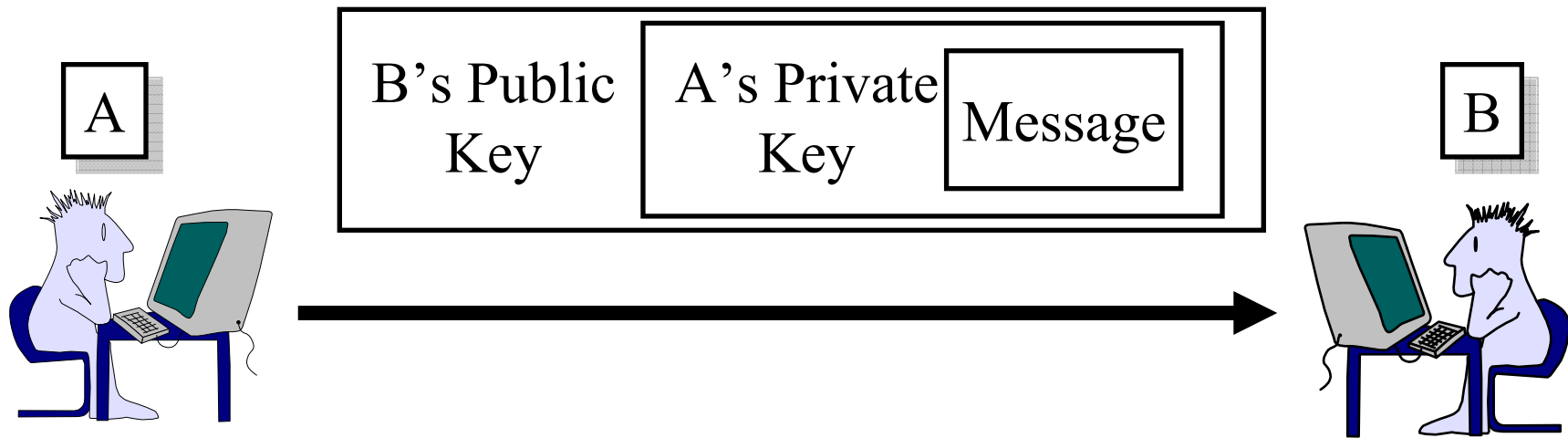
- A encrypts the message with its private key and then with B's public key
- B can decrypt it with its private key and A's public key
- No one else can decrypt $\Rightarrow$ Secrecy
- No one else can send such a message
  $\Rightarrow$ B is assured that the message was sent by A
  $\Rightarrow$ Authentication

# Public-Key Applications

- ❑ 3 Categories:
    - ➢ **Encryption/decryption** (provide secrecy)
    - ➢ **Digital signatures** (provide authentication)
    - ➢ **Key exchange** (of session keys)
- ❑ Some algorithms are suitable for all uses, others are specific to one

| Algorithm | Encryption/Decryption | Digital Signature | Key Exchange |
|---|---|---|---|
| RSA | Yes | Yes | Yes |
| Elliptic Curve | Yes | Yes | Yes |
| Diffie-Hellman | No | No | Yes |
| DSS | No | Yes | No |

# Public-Key Requirements

❑ Need a trapdoor one-way function

❑ One-way function has

  ➢ $Y = f(X)$ easy

  ➢ $X = f^{-1}(Y)$ infeasible

❑ A trap-door one-way function has

  ➢ $Y = f_k(X)$ easy, if k and X are known

  ➢ $X = f_k^{-1}(Y)$ easy, if k and Y are known

  ➢ $X = f_k^{-1}(Y)$ infeasible, if Y known but k not known

❑ A practical public-key scheme depends on a suitable trap-door one-way function

# Security of Public Key Schemes

- Like private key schemes brute force **exhaustive search** attack is always theoretically possible
- But keys used are too large (>512bits)
- Security relies on a **large enough** difference in difficulty between **easy** (en/decrypt) and **hard** (cryptanalyse) problems
- More generally the **hard** problem is known, but is made hard enough to be impractical to break
- Requires the use of **very large numbers**
- Hence is **slow** compared to private key schemes

# RSA Public Key Encryption

❑ Ron Rivest, Adi Shamir, and Len Adleman at MIT 1978

❑ Exponentiation in a Galois field over integers modulo a prime
   ➢ Exponentiation takes $O((\log n)^3)$ operations (easy)

❑ Security due to cost of factoring large numbers
   ➢ Factorization takes $O(e^{\log n \log \log n})$ operations (hard)

❑ Plain text M and ciphertext C are integers between 0 and n-1.

❑ Key 1 = {e, n},
   Key 2 = {d, n}

❑ $C = M^e \bmod n$
   $M = C^d \bmod n$

❑ How to construct keys:
   ➢ Select two large primes: p, q, p ≠ q
   ➢ n = p×q
   ➢ Calculate Euler's Totient Fn Φ(n) = (p-1)(q-1)
   ➢ Select e relatively prime to Φ ⟹ gcd(Φ, e) = 1; 0 < e < Φ
   ➢ Calculate d = inverse of e mod Φ ⟹ de mod Φ = 1
   ➢ Euler's Theorem: $x^{ed} = x^{k\Phi(n)+1} = x \bmod n$

# Finding d and e

- de = 1 mod $\Phi(n)$
- Select e first, e.g., e=$2^1+1$, $2^4+1$ or $2^{16}+1$
  $\Rightarrow$ Exponentiation is easy.
- Find inverse of e using Euclid's algorithm
- The public key can be small.
- The private key should be large $\Rightarrow$ Don't select d=3.
  - Can be attacked using Chinese remainder theorem & 3 messages with different modulii
- Both d and n are 512 bit (150 digits) numbers.

# RSA Key Construction: Example

- Select two large primes: p, q, p ≠ q
  p = 17, q = 11
- n = p×q = 17×11 = 187
- Calculate $\Phi$ = (p-1)(q-1) = 16x10 = 160
- Select e, such that gcd($\Phi$, e) = 1; 0 < e < $\Phi$
  say, e = 7
- Calculate d such that de mod $\Phi$ = 1
  - Use Euclid's algorithm to find d=$e^{-1}$ mod $\Phi$
  - 160k+1 = 161, 321, 481, 641
  - Check which of these is divisible by 7
  - 161 is divisible by 7 giving d = 161/7 = 23
- Key 1 = {7, 187}, Key 2 = {23, 187}

# Exponentiation

- ❑ Can use the Square and Multiply Algorithm
- ❑ `E.g.,` $3^{129} = 3^{128} \times 3^1 = 5 \times 3 = 4 \bmod 11$
- ❑ Takes log (b) operations for $a^b$
- ❑ To compute $a^b \bmod n$:

Expand b as a binary number: $b_k b_{k-1} \ldots b_2 b_1 b_0$

k= Number of bits in b

c = 0; f = 1

**for** i = k **downto** 0

   **do** c = 2 × c

       f = (f × f) mod n

       **if** $b_i$ == 1 **then**

           c = c + 1

           f = (f × a) mod n

**return** f

### Excel

a=  125
b=  107
n=  187

| j | i=2^j | a^i | a^i mod n | bi | c | a^c mod n |
|---|---|---|---|---|---|---|
| 0 | 1 | 125 | 125 | 1 | 1 | 125 |
| 1 | 2 | 15625 | 104 | 1 | 3 | 97 |
| 2 | 4 | 10816 | 157 | 0 | 3 | 97 |
| 3 | 8 | 24649 | 152 | 1 | 11 | 158 |
| 4 | 16 | 23104 | 103 | 0 | 11 | 158 |
| 5 | 32 | 10609 | 137 | 1 | 43 | 141 |
| 6 | 64 | 18769 | 69 | 1 | 107 | 5 |
| 7 | 128 | 4761 | 86 | 0 | 107 | 5 |
| 8 | 256 | 7396 | 103 | 0 | 107 | 5 |
| 9 | 512 | 10609 | 137 | 0 | 107 | 5 |
| 10 | 1024 | 18769 | 69 | 0 | 107 | 5 |

# Optimizing Private Key Operations

1. $c^d \bmod n = c^d \bmod pq$
   - Compute $c^d \bmod p$ and $c^d \bmod q$
   - Use Chinese remainder theorem to compute $c^d \bmod pq$
2. Chinese remainder theorem requires $p^{-1} \bmod q$ and $q^{-1} \bmod p$. Compute them once and store.
3. Since d is much bigger than p, $c^d \bmod p = c^r \bmod p$ where $r = d \bmod (p-1)$
   - $d = k(p-1) + r$
   - Mod p: $a^d = a^{k(p-1)+r} = a^{k\Phi(p)} a^r = a^r$ [Euler's Theorem]
- ❑ Only owner of the private key knows p and q and can optimize

# Optimization Example

- $\langle e, n \rangle = \langle 7, 187 \rangle$, $\langle d, n \rangle = \langle 23, 187 \rangle$
- $17^{-1} \bmod 11 = 2$ and $11^{-1} \bmod 17 = 14$
- $c^d \bmod pq = [(c^d \bmod p)\, q\, (q^{-1} \bmod p)$
  $+ (c^d \bmod q)\, p\, (p^{-1} \bmod q)]$ **mod n**
- $101^{23} \bmod 11 = 101^3 \bmod 11 = 2\times4=8$
- $101^{23} \bmod 17 = 101^7 \bmod 17 = 101^{4+2+1} \bmod 17$
  $= 16 \times 16^2 \times 16^4 \bmod 17 = 16 \times 1 \times 1 = 16$
- $101^{23} \bmod 187 = [(101^{23} \bmod 11)17(2)] + (101^{23} \bmod 17)11(14)] \bmod 187$
- $= [8 \times 34 + (16^7 \bmod 17)\times 154] \bmod 187$
  $= [8 \times 34 + 16 \times 154] \bmod 187 = [272+2464] \bmod 187$
  $= 2736 \bmod 187 = 118$
- **Verification**:
  - $101^2 \bmod 187 = 10201 \bmod 187 = 103$
  - $101^4 \bmod 187 = 103^2 \bmod 187 = 10609 \bmod 187 = 137$
  - $101^8 \bmod 187 = 137^2 \bmod 187 = 18769 \bmod 187 = 69$
  - $101^{16} \bmod 187 = 69^2 \bmod 187 = 4761 \bmod 187 = 86$
- $101^{23} \bmod 187 = 101^{16}\times 101^4 \times 101^2 \times 101 = 86 \times 137 \times 103 \times 101 \bmod 187$
  $= 86 \times 137 \times 10403 \bmod 187 = 86 \times 137 \times 118 \bmod 187 = 86 \times 16166 \bmod 187$
  $= 86 \times 84 \bmod 187 = 7224 \bmod 187 = 118$

# RSA Issues

- RSA is computationally intense.
- Commonly used key lengths are 1024 bits
- The plain text should be smaller than the key length
- The encrypted text is same size as the key length
- Generally used to encrypt secret keys.
- Potential Attacks:
  1. Brute force key search - infeasible given size of numbers
  2. Timing attacks - on running of decryption
     Can Infer operand size based on time taken
     $\Rightarrow$ Use constant time
  3. Mathematical attacks - based on difficulty of computing ø(n), by factoring modulus n
  4. Chosen ciphertext attacks

# Progress in Factoring

| Number of Decimal Digits | Approximate Number of Bits | Date Achieved | MIPS-years | Algorithm |
|---|---|---|---|---|
| 100 | 332 | April 1991 | 7 | quadratic sieve |
| 110 | 365 | April 1992 | 75 | quadratic sieve |
| 120 | 398 | June 1993 | 830 | quadratic sieve |
| 129 | 428 | April 1994 | 5000 | quadratic sieve |
| 130 | 431 | April 1996 | 1000 | generalized number field sieve |
| 140 | 465 | February 1999 | 2000 | generalized number field sieve |
| 155 | 512 | August 1999 | 8000 | generalized number field sieve |
| 160 | 530 | April 2003 | — | Lattice sieve |
| 174 | 576 | December 2003 | — | Lattice sieve |
| 200 | 663 | May 2005 | — | Lattice sieve |

Ref: The RSA Factoring Challenge FAQ, http://www.rsa.com/rsalabs/node.asp?id=2094

# Optimal Asymmetric Encryption Padding (OAEP)

- ❑ RSA is susceptible to "Chosen Ciphertext Attack"

$$E(PU, M) = M^e \bmod n \qquad PU = \text{Public Key}$$
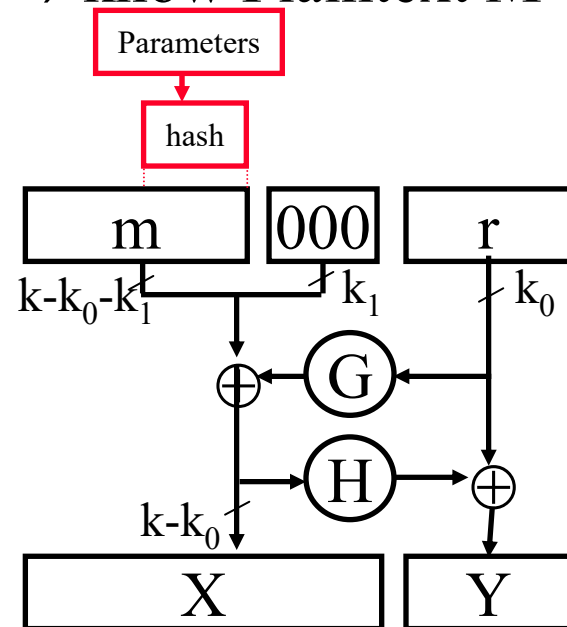
$$E(PU, M_1) \times E(PU, M_2) = E(PU, M_1 \times M_2)$$

$$E(PU, 2M) = 2^e\, E(PU, M)$$

- ❑ Submit $2^e \times$ Ciphertext and get back $2M \Rightarrow$ know Plaintext M

- ❑ OAEP: Let k =# bits in RSA modulus

  - ➢ Plaintext m is $k-k_0-k_1$ bit string

  - ➢ G and H are Cryptographic fn
    G expands $k_0$ bits to $k-k_0$ bits
    H reduces $k-k_0$ bits to $k_0$ bits

  - ➢ $r$ is a random $k_0$ bit seed

- ❑ Need to recover entire X and Y

Ref: http://en.wikipedia.org/wiki/Optimal_asymmetric_encryption_padding

# Summary

1. Public key encryption uses two keys: one to encrypt and the other to decrypt. The keys are interchangeable. One key is public. Other is private.

2. RSA uses exponentiation in GF($n$) for a large $n$.
   $n$ is a product of two large primes.

3. RSA keys are $<e, n>$ and $<d, n>$ where $ed$ mod $\Phi(n)=1$

4. Given the keys, both encryption and decryption are easy. But given one key finding the other key is hard.

5. The message size should be less than the key size. Use large keys 512 bits and larger.

# Homework 9

A. In an RSA system, the public key is {31, 3599}. What is the private Key?

B. Perform encryption and decryption using RSA algorithm using the following parameters:

1. p=3; q=11; e=7; M=5
2. p=5; q=11; e=3; M=9
3. p=7; q=11; e=17; M=8
4. p=11; q=13; e=11; M=7
5. p=17; q=31; e=7; M=2

   Try all. Submit answer to 1 only.

C. In a public key system using RSA, you intercept the ciphertext C=10 set to a user whose public key is {5,35}. What is the private key and the plaintext M?

D. Compute $5^{596}$ mod 1234

# Lab 9: Key Logger

❑ Download the key logger from:
  https://kidlogger.net

❑ Use the version for Mac or Windows.

❑ Install the key logger

❑ Go to any financial/bank web site and try to login with a fake user name and password.

❑ Print the key log and underline the password you used and submit.

❑ Note that this works even though the site uses https.

❑ Remember to uninstall the key logger

# Acronyms

- DB          Data block
- EM          Encrypted Message
- FAQ         Frequently Asked Questions
- GF           Galois Field
- GHz         Giga Hertz
- MGF        Moment generating function
- MIPS       Millions of Instructions per Second
- OAEP     Optimal Asymmetric Encryption Padding (OAEP)
- PR           Private Key
- PU           Public Key
- RSA         Rivest, Samir, and Adleman

# Scan This to Download These Slides



Raj Jain
http://rajjain.com

# Related Modules

CSE571S: Network Security (Spring 2017),
http://www.cse.wustl.edu/~jain/cse571-17/index.html

CSE473S: Introduction to Computer Networks (Fall 2016),
http://www.cse.wustl.edu/~jain/cse473-16/index.html

Wireless and Mobile Networking (Spring 2016),
http://www.cse.wustl.edu/~jain/cse574-16/index.html

CSE571S: Network Security (Fall 2014),
http://www.cse.wustl.edu/~jain/cse571-14/index.html

Audio/Video Recordings and Podcasts of
Professor Raj Jain's Lectures,
https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw