# CS 5319
# Advanced Discrete Structure

## Lecture 10:
## Introduction to Number Theory III

# Outline

- Divisibility
- Greatest Common Divisor
- Fundamental Theorem of Arithmetic
- Modular Arithmetic
- Euler Phi Function
- RSA Cryptosystem

Reference: Course Notes of MIT 6.042J (Fall 05)
by Prof. Meyer and Prof. Rubinfeld

# RSA Cryptosystem

- A cryptosystem allows a sender to encrypt a message *M* into some form *C* so that only the intended receiver can decrypt *C* back to *M*
- Most cryptosystems are symmetric, where the sender and the receiver have to *share* a secret key in order to perform the encoding and decoding
  - we can encrypt if and only if we can decrypt
- Major problem : How can the receiver and sender agree on the secret key in the first place ?

# RSA Cryptosystem

- In 1977, Rivest, Shamir, and Adleman announced a scheme which does not need a shared secret key
- This is widely known as the RSA cryptosystem
  - Indeed, a similar scheme was invented earlier in 1973 by Ellis and Cocks
  - Since these schemes do not need shared secret keys, they are called public key cryptosystems
- The following describes how RSA works

# RSA Cryptosystem

**Setup.** Receiver performs the following :

- Choose two distinct primes $p$ and $q$. Let $n = p \cdot q$
- Select an integer $e$ coprime to $\varphi(n)$.
  - The pair $(e, n)$ is the *public key* and the receiver tells all the others
- Find the unique $d$ such that $ed \equiv 1 \pmod{\varphi(n)}$.
  - The pair $(d, n)$ is the *secret key*, and the receiver keeps this hidden

# RSA Cryptosystem

**Encryption.** Sender performs the following :

- Get the public key ($e$, $n$) of the receiver
- Given a message $M$, with $0 < M < n$, encrypt $M$ by computing

$$C = M^e \text{ rem } n$$

- Send $C$ to the receiver

# RSA Cryptosystem

**Decryption.** Receiver performs the following :

- Receive $C$ from sender
- Decrypt $C$ by computing

$$M = C^d \text{ rem } n$$

Question : Why does RSA work ??

# RSA Cryptosystem

Theorem 9:

> Decryption of RSA works.

Proof: When $M$ is coprime to $n$.

Since $ed \equiv 1 \pmod{\varphi(n)}$, there is an integer $t$ such that $ed = 1 + t\,\varphi(n)$ . Thus

$$C^d \equiv M^{ed} \equiv M^{1+t\,\varphi(n)} \equiv M \pmod{n}$$

➔ $$M = C^d \text{ rem } n$$

# RSA Cryptosystem

Proof (cont) : When $M$ is not coprime to $n$.

Suppose $M$ is a multiple of $p$ (but not $q$). Then

$$C^{d} \equiv M^{ed} \equiv 0 \equiv M \quad (\text{mod } p)$$

$$C^{d} \equiv M^{ed} \equiv M^{1 + t\varphi(n)}$$

$$\equiv M\,(M^{q-1})^{t(p-1)} \equiv M \quad (\text{mod } q)$$

➔ $C^{d} - M$ is a multiple of both $p$ and $q$

➔ $C^{d} \equiv M \ (\text{mod } n)$ [why?]

# RSA Cryptosystem

Ex : Finding Public and Secret Keys

- Suppose receiver chooses primes $p = 7$ and $q = 11$
- Then $n = 77$, with $\varphi(n) = (7 - 1)(11 - 1) = 60$
- Suppose the receiver choose $e = 7$, since 7 is coprime to 60
- The corresponding $d$ becomes 43, since

$$43 \times 7 = 301 \equiv 1 \pmod{60}$$

➔ Public key = (7, 77) ; Secret key = (43, 77)

# RSA Cryptosystem

Ex : Encryption and Decryption

- If sender wants to send $M = 4$, she encrypts it as

$$C = 4^7 \text{ rem } 77$$

$$= 16384 \text{ rem } 77 = 60$$

- When receiver receives $C = 60$, he decrypts it as

$$M = 60^{43} \text{ rem } 77 = 4$$

Note: $60^2 \equiv 58$, $60^4 \equiv 53$, $60^8 \equiv 37$, $60^{16} \equiv 60$, $60^{32} \equiv 58 \pmod{77}$

$$\Rightarrow 60^{43} \equiv 60^{32} \times 60^8 \times 60^2 \times 60$$

$$\equiv 58 \times 37 \times 58 \times 60 \equiv 4 \pmod{77}$$

# Security of RSA

- Security of RSA relies on the assumption below :

  Given the public key ($e$, $n$) and $C$ , it is difficult to compute the message $M$

  ➔ This relies on the assumption that given the public key ($e$, $n$), it is difficult to compute $d$

  ➔ This further relies on the assumption that it is difficult to factor $n$ into $p$ and $q$

- It is recommended that $n$ is at least 2048 bits long

# Security of RSA

- Because RSA is now widely used, many people wants to break RSA
- Some weaknesses in RSA are known.

  Example :

  - If the prime factors of either $p - 1$ or $q - 1$ are all small, the technique by Pollard (1974) can factor $n$ quickly
  - Also true if the prime factors of either $p + 1$ or $q + 1$ are all small (Williams (1982))

# Security of RSA

Theorem 10:

| |
|---|
| If $p$ and $q$ are 'close', then RSA is insecure. |

Proof:

If $p$ and $q$ are 'close', then $(p + q) / 2$ is not much larger than $\sqrt{n}$ (we know that it is at least as big)

Now, suppose $p > q$ and we set

$$x = (p + q) / 2, \quad y = (p - q) / 2$$

# Security of RSA

Proof (cont) :

Thus $n = p \cdot q$

$$= x^2 - y^2 = (x + y)(x - y)$$

Hence, if an attacker can express $n$ as a difference of two squares, she can factor $n$

To do this, the attacker tests the numbers

$$\lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil + 1, \lceil \sqrt{n} \rceil + 2, \ldots$$

until finding $s$ such that $s^2 - n$ is a square number

# Security of RSA

Proof (cont) :

The number of tests is equal to

$$x - \lceil \sqrt{n} \rceil = (p+q)/2 - \lceil \sqrt{n} \rceil$$

which is small

More precisely, if $p = (1+\varepsilon)\sqrt{n}$, then the number of tests is approximately :

$$\left( \frac{(1+\varepsilon) + (1+\varepsilon)^{-1}}{2} - 1 \right) \sqrt{n} = \frac{\varepsilon^2 \sqrt{n}}{2(1+\varepsilon)}$$

# Security of RSA

Ex : Primes $p$ and $q$ are too close

If $n = 56759$ ,

then the ceiling of its square root is 239.

By testing $s = 239, 240, \ldots$ we find that

$$240^2 - 56759 = 841 = 29^2$$

➔ Thus we have

$$n = 56759 = 240^2 - 29^2 = 269 \times 211$$