ĐẠI NAM
UNIVERSITY

Bài 7
MÃ HÓA RSA

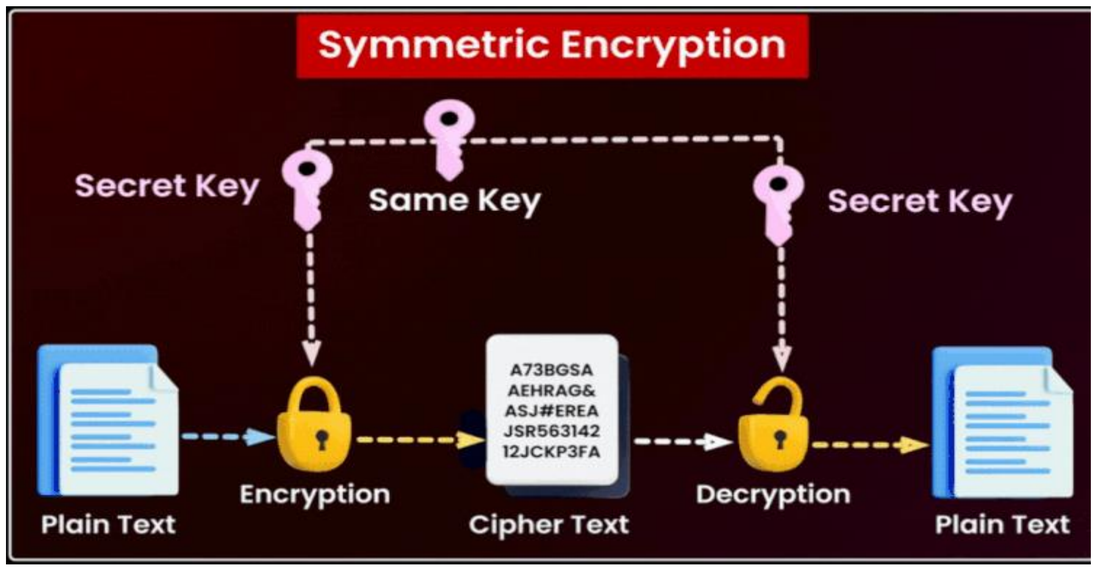
Giảng viên: TS. Trần Quý Nam
(namtq@dainam.edu.vn)

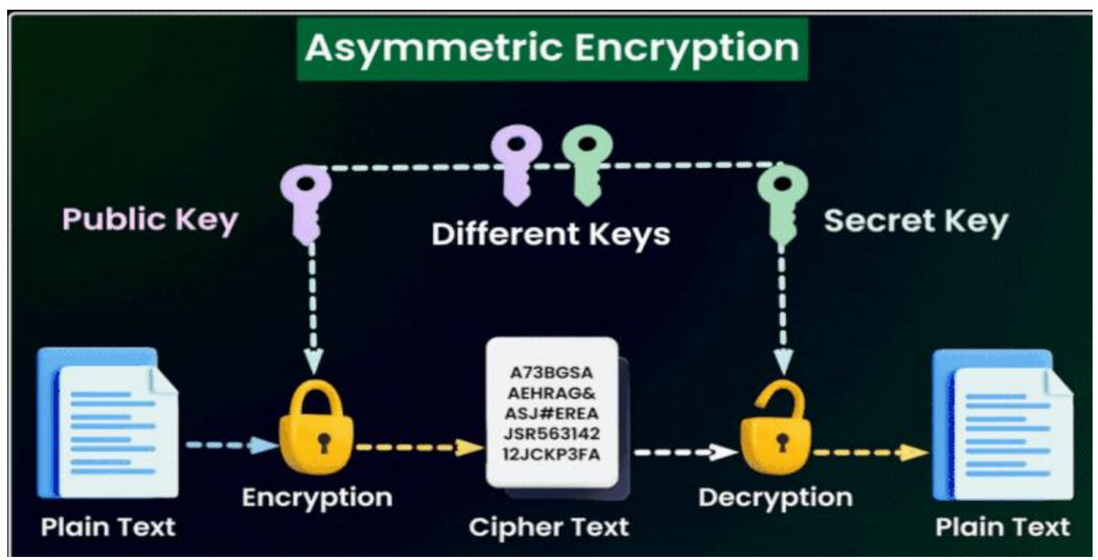
Nội dung

- Giới thiệu về mã khóa công khai
- Thuật toán RSA
- Thực hành lập trình với RSA
- Bài tập Lab ứng dụng với RSA

Nhắc lại: Mã khoá đối xứng và bất đối xứng

- ✓ Bản rõ (ta có thể đọc được)
- ✓ Bản mã (chỉ người giải mã hiểu được)





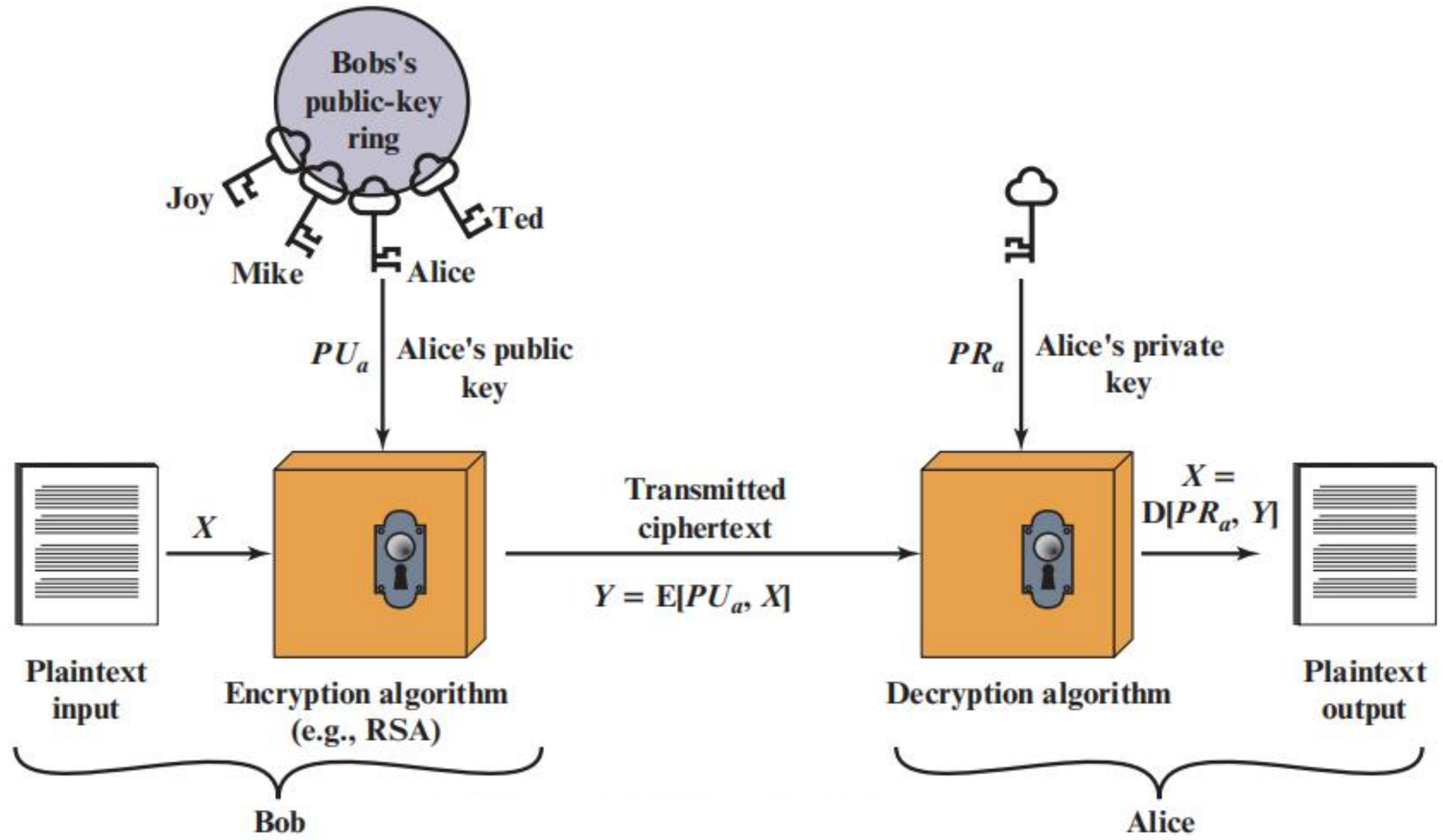
Bảo mật dùng Public Key

PU: Public key

PR: Private key

E(•): Encryption

D(•): Decryption



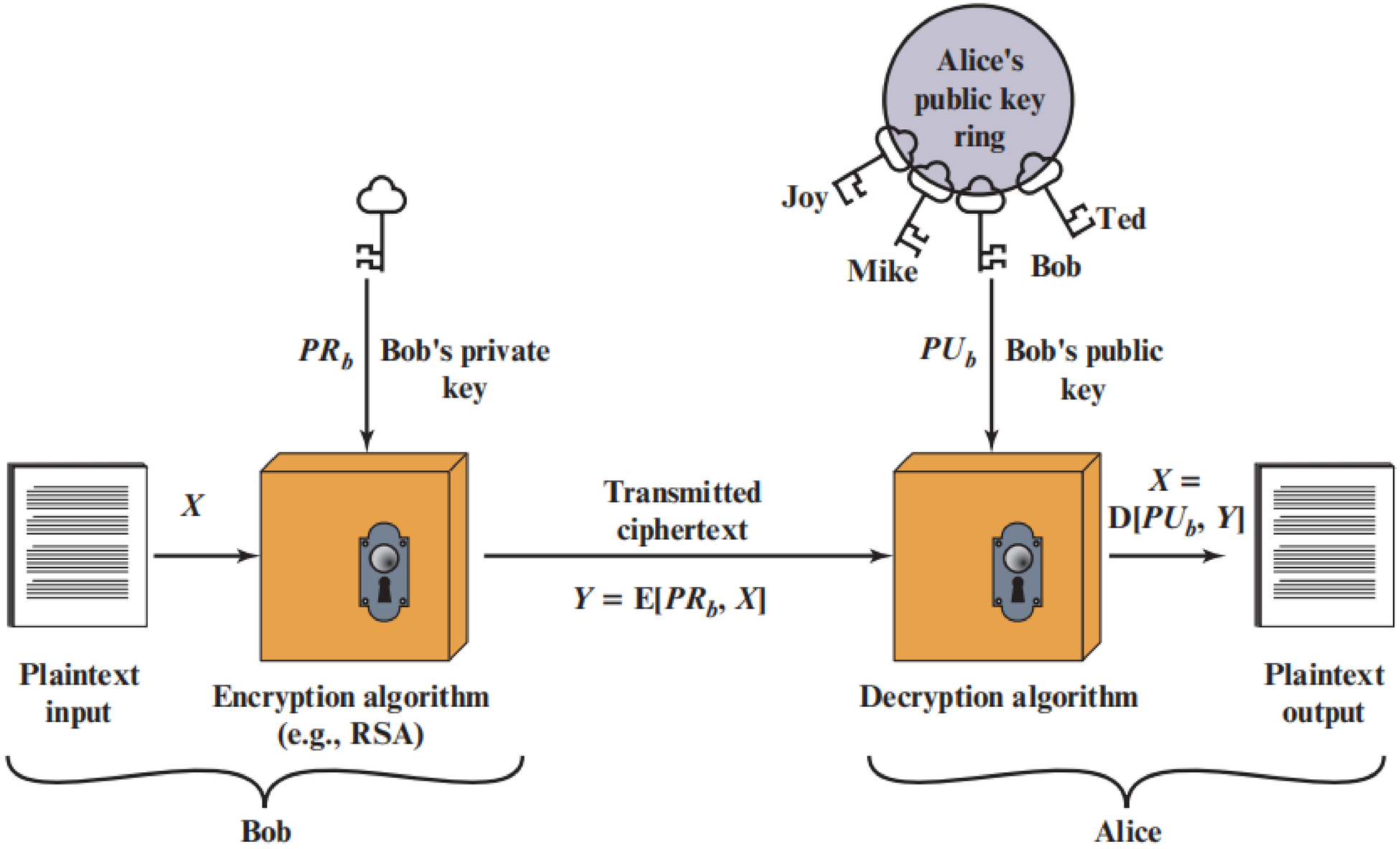
Xác thực dùng Private Key

PU: Public key

PR: Private key

E(•): Encryption

D(•): Decryptio



An adversary, observing Y and having access to PU_b, but not having access to PR_b or X, must attempt to recover X and/or PR_b. It is assumed that the adversary does have knowledge of the encryption (E) and decryption (D) algorithms. If the adversary is interested only in this particular message, then the focus of effort is to recover X by generating a plaintext estimate $\hat{X}$. Often, however, the adversary is interested in being able to read future messages as well, in which case an attempt is made to recover PR_b by generating an estimate $P\hat{R}_b$.

Ứng dụng của mã hoá công khai

Digital Signatures to maintain authenticity of documents

Encrypted browsing sessions for better protection against hackers

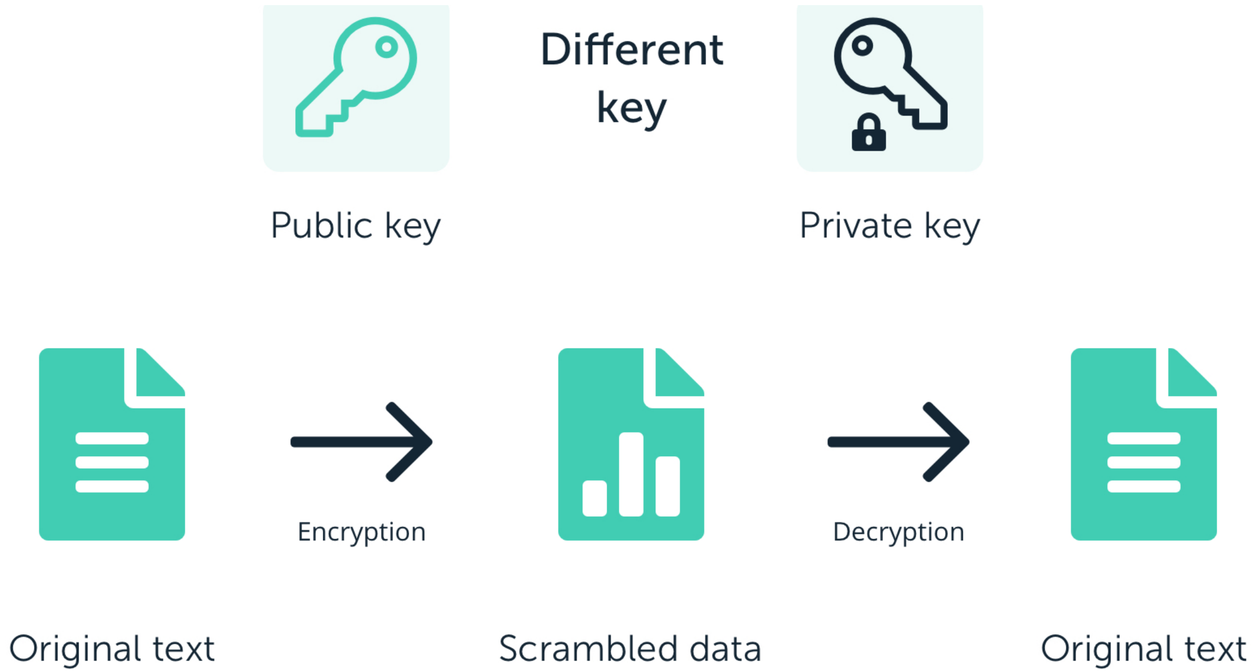
Managing **Crypto-currency transactions** securely

Sharing Keys for Symmetric Key Cryptography

- **Bảo mật truyền thông**
- **Chữ ký số**
- **Quản lý khoá**
- **Tiền mã hoá**

GIỚI THIỆU

- Ý tưởng của hệ mật công khai được Diffie và Hellman đưa ra năm 1976. Rivest, Shamir và Adleman đưa ra thuật toán RSA đầu tiên năm 1977.
- Hệ mã dùng mã công khai nói trên gọi là hệ mã công khai, là hệ mã sử dụng khóa cho mã hóa và giải mã khác nhau.



Hệ mã hóa RSA

- Là hệ mã hóa khóa công khai rất phổ biến
- Là cơ chế mã hóa khối, plaintext và ciphertext là các số nguyên từ 0 đến n-1. Kích cỡ n thường là 1024 bits, hoặc 309 chữ số thập phân (nghĩa là $n < 2^{1024}$)
- Dựa trên hàm mũ (exponentiation) trong trường hữu hạn (finite field)
- Thuật toán mã hóa và giải mã RSA dùng phép lũy thừa modulo của lý thuyết số.
- Bảo mật cao vì chi phí phân tích thừa số của một số nguyên lớn là rất lớn

Nhắc lại nghịch đảo nhân modulo

Nghịch đảo modulo

Nghịch đảo của một số nguyên là số mà khi nhân với nó sẽ có tích là 1. Để tính thương của 2 số, ta nhân một số với nghịch đảo của số kia.

$$\frac{a}{b} = a * b^{-1}$$

Tương tự như vậy, nghịch đảo modulo của một số theo modulo M là số mà khi nhân với nó thì được tích chia M dư 1.

$$\text{b là nghịch đảo modulo M của a} \iff (a * b) \mod M = 1$$

hay viết cách khác:

$$\text{b là nghịch đảo modulo M của a} \iff a * b \equiv 1 \mod M$$

Ví dụ: 7 là nghịch đảo modulo 11 của 8 vì 7*8 = 56 và 56 % 11 = 1

Và phép chia modulo sẽ tương ứng với phép nhân nghịch đảo modulo.

Nhắc lại thuật toán Euclid mở rộng

- Để tính nghịch đảo nhân modulo n, áp dụng giải thuật Euclid mở rộng:

  s * n + t * b = gcd(n,b) = 1 (định lí Bézout)

- Thực hiện phép mod cả 2 vế

  (s * n + t * b) mod n = 1 mod n

  [(s * n) mod n] + [(t*b) mod n] = 1 mod n

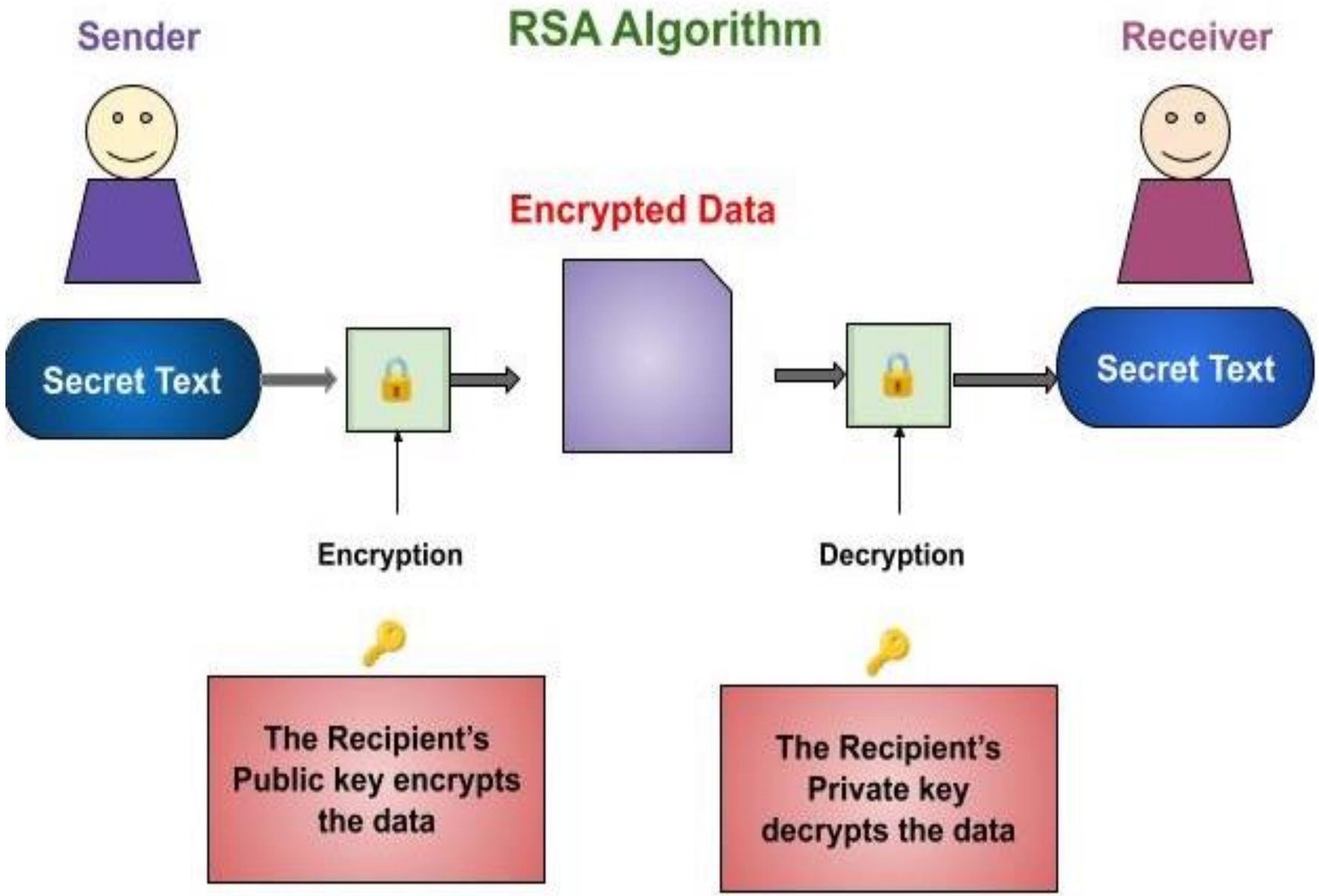
  0 + [(t*b) mod n] = 1

➔ (t * b) mod n = 1 → t chính là nghịch đảo nhân của b

- Thuật toán này vừa có thể tính được gcd(a,b) vừa tính được các giá trị s và t

https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm#:~:text=Extended%20Euclidean%20algorithm%20also%20refers,a%20and%20b%20are%20coprime.
https://laptrinhthidau.wordpress.com/2016/08/23/thuat-toan-euclid-mo-rong/
http://nguyenduccuong.com/nckh/80-thut-toan-euclide-m-rng

Ý tưởng RSA


Sender
RSA Algorithm
Receiver
Encrypted Data
Secret Text
Secret Text
Encryption
Decryption
The Recipient's Public key encrypts the data
The Recipient's Private key decrypts the data

,

Thuật toán RSA

1. Chọn hai số nguyên tố lớn p và q và tính n = p*q
2. Tính $\phi(n) = (p - 1)*(q - 1)$
3. Tìm một số *e* sao cho *e* nguyên tố cùng nhau với *n*

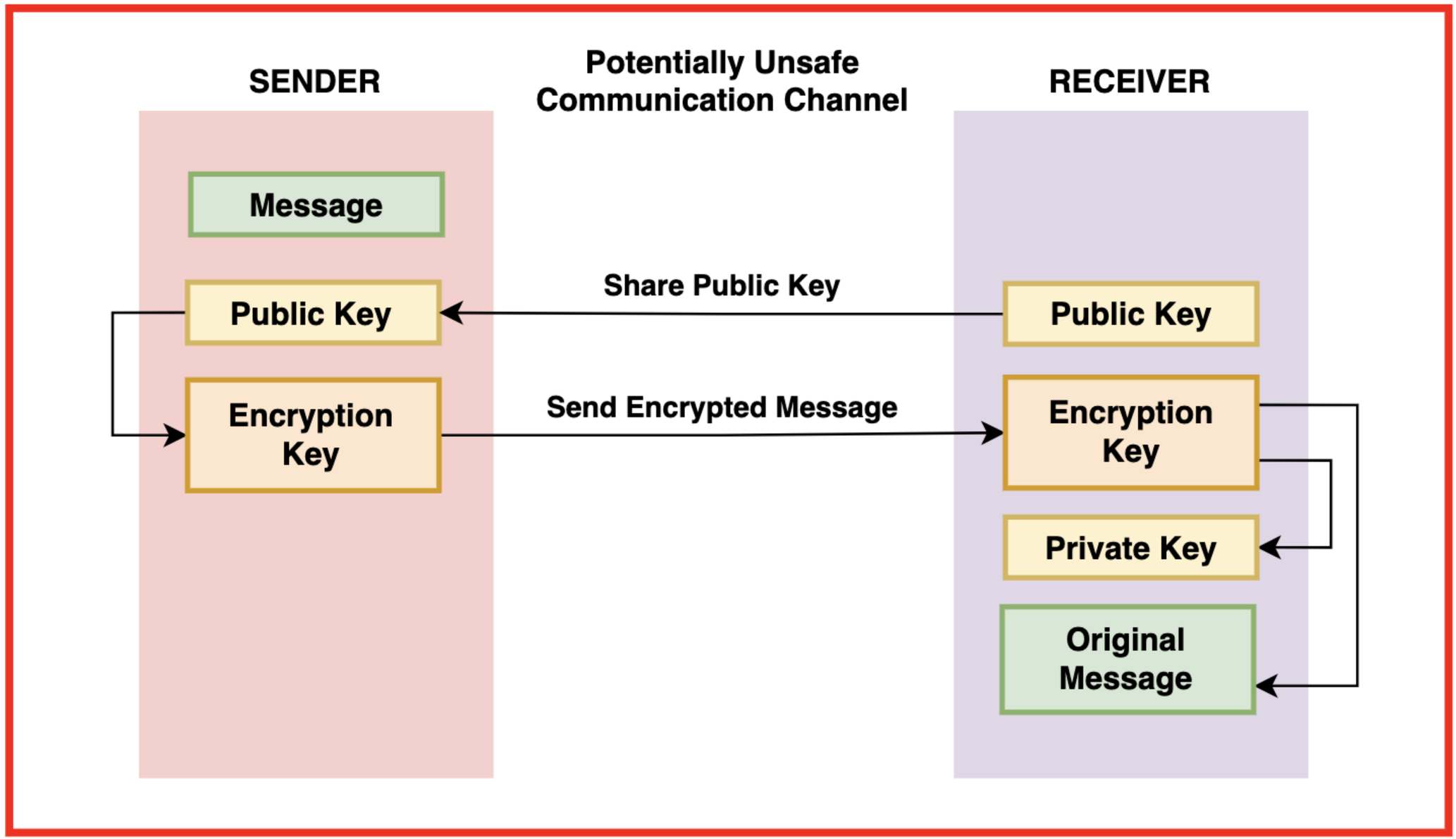
$$UCLN(e, \phi(n)) = 1 \qquad 1 < e < \phi(n)$$

4. Tìm một số *d* sao cho e*d ≡ 1 mod ϕ(n) (*d* là nghịch đảo của *e* trong phép modulo ϕ(*n*)) $d \equiv e^{-1} \bmod \phi(n)$

hay (d * e) % φ(n) = 1

5. Chọn khóa công khai là cặp (*e*, *n*), khóa riêng là cặp (*d*, *n*)

Thuật toán RSA



Thuật toán RSA
Phát sinh khóa

Key Generation Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Thuật toán RSA

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Encryption by Bob with Alice's Public Key

Plaintext: $M < n$

Ciphertext: $C = M^e \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext: C

Plaintext: $M = C^d \bmod n$

Ví dụ RSA

Ví dụ về mã hóa RSA:

1. Chọn p = 11 và q = 3, do đó $n = p*q$ = 33
2. $\phi(n) = (p-1)*(q-1)$ = 20
3. Chọn e = 3 nguyên tố cùng nhau với $\phi(n)$
4. Tính nghịch đảo của e trong phép modulo ϕ(n) được d = 7 (3x7 = 21)
5. Khóa công khai $K_U = (e, n) = (3, 33)$. Khóa bí mật $K_R = (d, n) = (7, 33)$

Ví dụ RSA (tiếp)

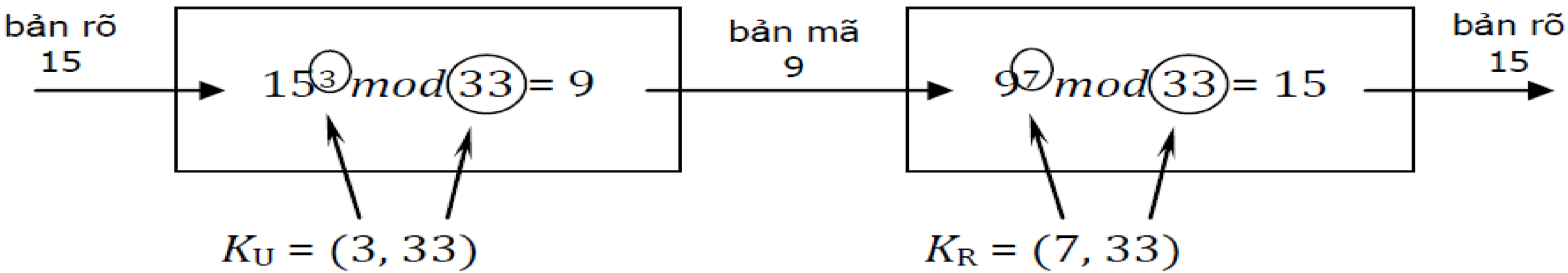
Mã hóa bảo mật:

6. Mã hóa bản rõ *M* = 15:

$$C = M^e \ mod \ N = 15^3 \ mod \ 33 = 9 \quad (\text{vì } 15^3 = 3375 = 102 \times 33 + 9)$$

7. Giải mã bản mã C = 9:

$$\bar{M} = C^d \ mod \ N = 9^7 mod \ 33 = 15 = M \ (\text{vì } 9^7 = 4.782.696 = 144.938 \times 33 + 15)$$



Ví dụ RSA (tiếp)

Mã hóa chứng thực (xác thực điện tử):

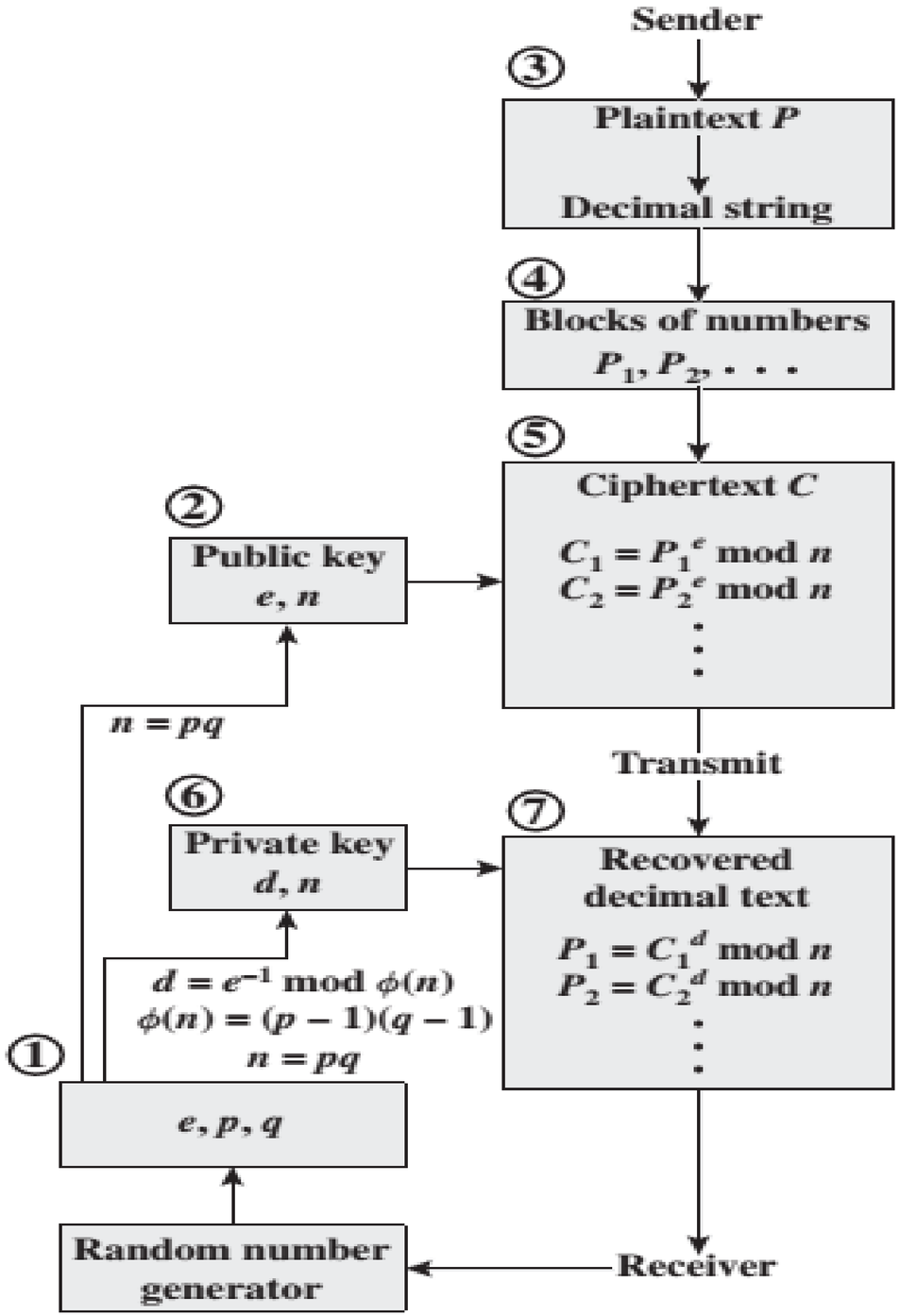
6. Mã hóa bản rõ $M = 15$:

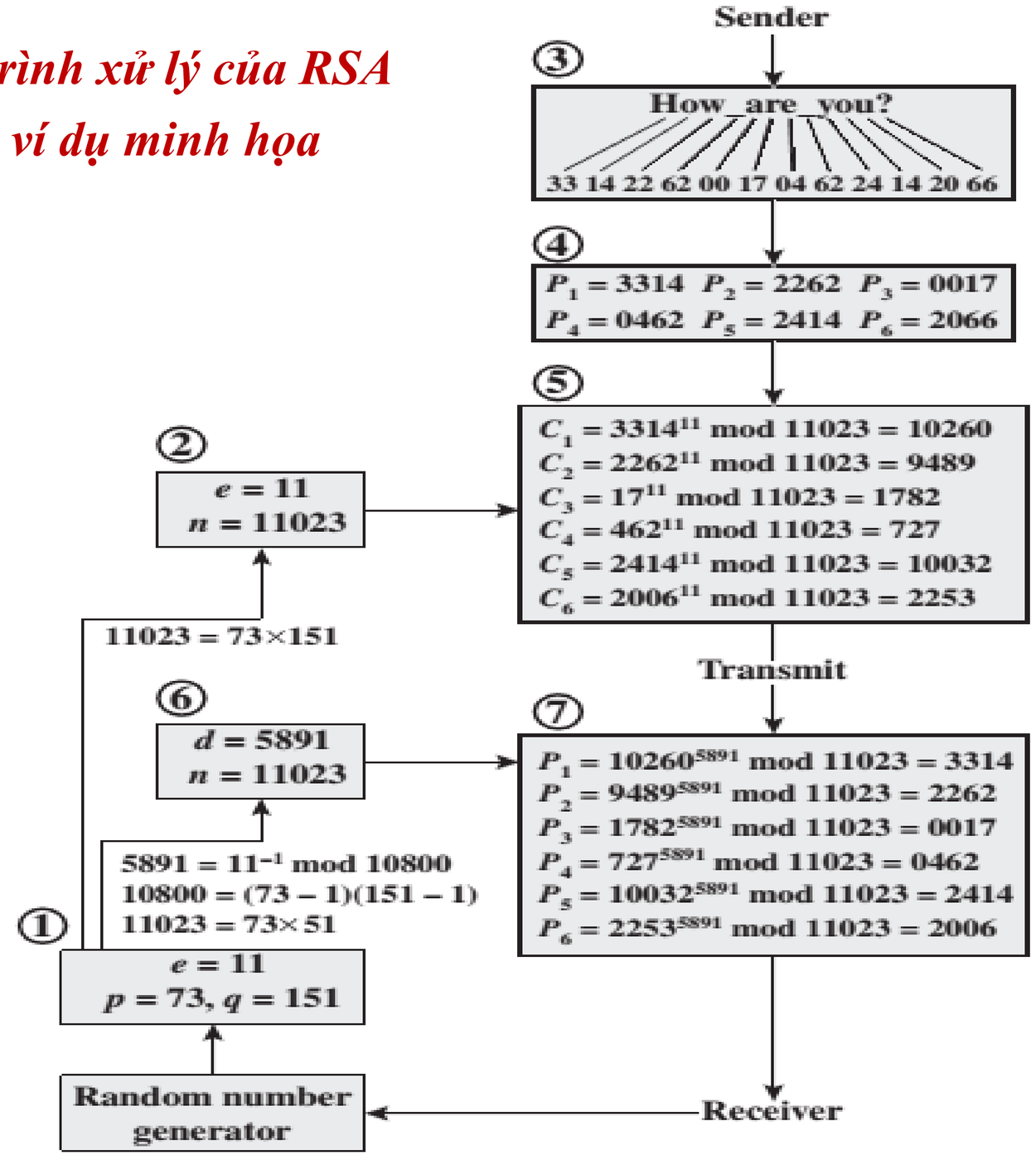
$$C = M^d \ mod \ N = 15^7 \ mod \ 33 = 27 \ (\text{vì } 15^7 = 170.859.375 = 5177.556 \times 33 + 27)$$

7. Giải mã bản mã C = 9:

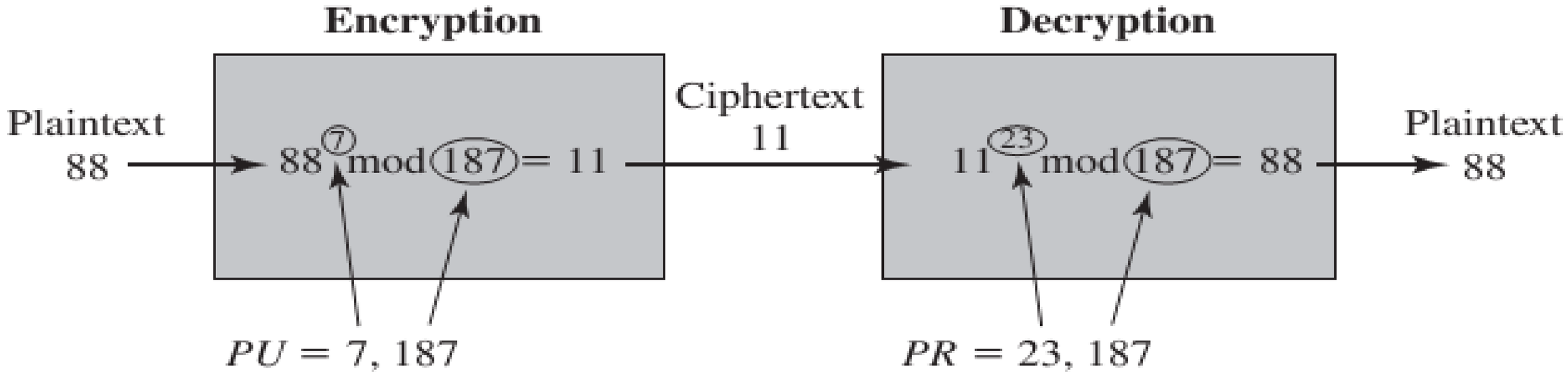
$$\bar{M} = C^e \ mod \ N = 27^3 \ mod \ 33 = 15 = M \ (\text{vì } 27^3 = 19.683 = 596 \times 33 + 15)$$

Quá trình xử lý của RSA và ví dụ minh họa





Ví dụ thực hiện RSA



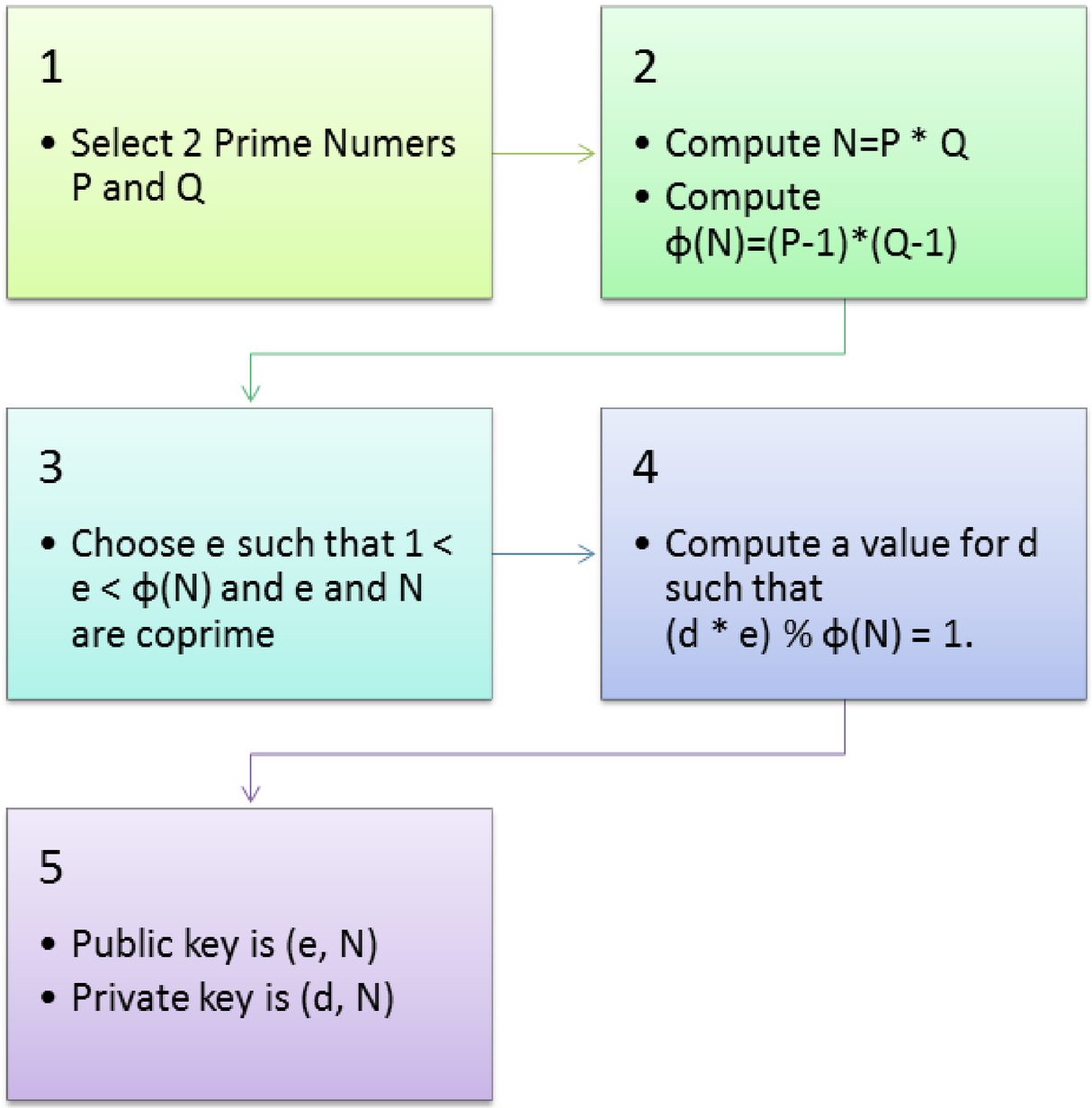
Ghi chú: RSA sử dụng các số nguyên tố rất lớn p, q để việc phân tích n với (n= p*q) là vô cùng khó khăn.

THUẬT TOÁN RSA

- Algorithm:

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

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



The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength lies in the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024-bit keys could be broken shortly. But till now it seems to be an infeasible task.

RSA Example

- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - $07^{17} \bmod 77 = 28$
  - $04^{17} \bmod 77 = 16$
  - $11^{17} \bmod 77 = 44$
  - $11^{17} \bmod 77 = 44$
  - $14^{17} \bmod 77 = 42$
- Bob sends 28 16 44 44 42

RSA Example

- Alice receives 28 16 44 44 42
- Alice uses private key, d = 53, to decrypt message:
  - 28^{53} mod 77 = 07
  - 16^{53} mod 77 = 04
  - 44^{53} mod 77 = 11
  - 44^{53} mod 77 = 11
  - 42^{53} mod 77 = 14
- Alice translates message to letters to read HELLO
  - No one else could read it, as only Alice knows her private key and that is needed for decryption

Example: Integrity/Authentication

- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)
  - $07^{53} \bmod 77 = 35$
  - $04^{53} \bmod 77 = 09$
  - $11^{53} \bmod 77 = 44$
  - $11^{53} \bmod 77 = 44$
  - $14^{53} \bmod 77 = 49$
- Alice sends 35 09 44 44 49

Example: Integrity/Authentication

- Bob receives 35 09 44 44 49
- Bob uses Alice's public key, $e = 17$, $n = 77$, to decrypt message:
  - $35^{17} \bmod 77 = 07$
  - $09^{17} \bmod 77 = 04$
  - $44^{17} \bmod 77 = 11$
  - $49^{17} \bmod 77 = 14$
- Bob translates message to letters to read HELLO
  - Alice sent it as only she knows her private key, so no one else could have enciphered it
  - If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly

Phá mã hệ mã hóa RSA

Bốn hướng có thể để tấn công RSA:

- **Vét cạn (Brute force attacks)**: Thử tất cả các khóa private key có thể. Điều này phụ thuộc vào độ dài khóa. → dùng khóa đủ lớn
- **Phân tích toán học (Mathematical attacks)**: Có vài hướng, nhưng tất cả đều tập trung vào việc phân tích thừa số tích của hai số nguyên tố.
- **Phân tích thời gian (Timing attacks)**: Cách này tùy thuộc vào thời chạy của thuật toán giải mã.
- **Phân tích bản mã được chọn (Chosen ciphertext attacks)**: khám phá các thuộc tính của thuật toán RSA → ngăn ngừa bằng cách làm nhiễu

An ninh của hệ mã hóa RSA

- An ninh của RSA dựa trên độ khó của việc phân tích ra thừa số nguyên tố các số nguyên tố lớn.
- Thời gian cần thiết để phân tích thừa số một số lớn tăng theo hàm mũ với số bit của số đó
  - Mất nhiều năm khi số chữ số thập phân của n vượt quá 100 (giả sử làm 1 phép tính nhị phân mất 1 ηs)
- Kích thước khóa lớn đảm bảo an ninh cho RSA
  - Từ 1024 bit trở lên
  - Khóa RSA 2048 bit được NIST coi là đủ cho đến năm 2030, vì nó cân bằng giữa tính bảo mật và hiệu quả tính toán

The Security of RSA

Five possible approaches to attacking the RSA algorithm are:

- **Brute force:** This involves trying all possible private keys.
- **Mathematical attacks:** There are several approaches, all equivalent in effort to factoring the product of two primes.
- **Timing attacks:** These depend on the running time of the decryption algorithm.
- **Hardware fault-based attack:** This involves inducing hardware faults in the processor that is generating digital signatures.
- **Chosen ciphertext attacks:** This type of attack exploits properties of the RSA algorithm.

Luyện tập thuật toán RSA

- Code:
  - https://github.com/yigitusta/RSA-Implementation
  - https://github.com/antew7/RSA/blob/master/RSA.cpp
  - https://github.com/tanmayrajDTU/RSA-Encryption-Decryption
- Gửi file lớn:
  - https://stackoverflow.com/questions/40243857/how-to-encrypt-large-file-with-rsa

HỎI ĐÁP

ĐẠI NAM
UNIVERSITY

Cảm ơn đã lắng nghe!

TRẦN QUÝ NAM
Giảng viên
namtq@dainam.edu.vn