TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

Bài giảng môn học: AN TOÀN THÔNG TIN

Chương 5:

HỆ MẬT MÃ KHÓA BẤT ĐỐI XỨNG

Số tín chỉ: 3
Số tiết: 60 tiết
(30 LT + 30 TH)

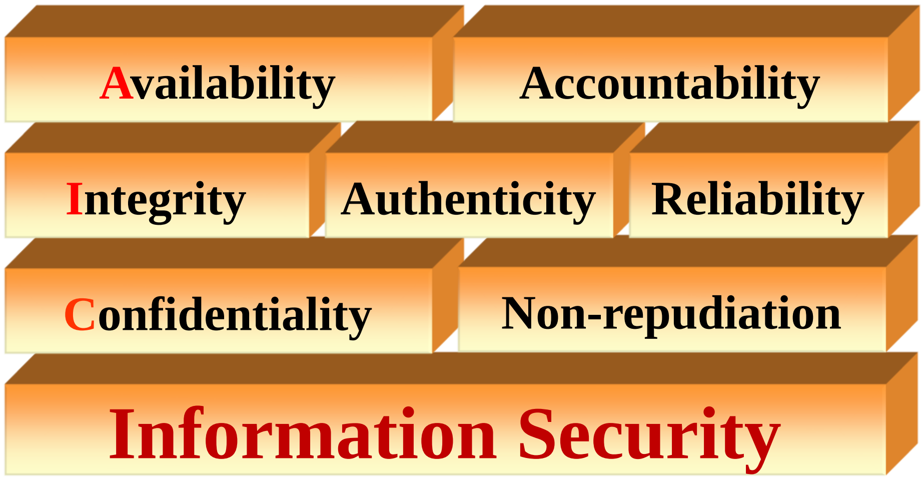
Biên soạn: ThS. **Nguyễn Thị Phong Dung**
Email : ntpdung@ntt.edu.vn

Bài 5: MẬT MÃ KHÓA BẤT ĐỐI XỨNG

- Dẫn nhập về Mã hóa khóa bất đối xứng
- Hệ mật mã khóa bất đối xứng
- Toán học trong thuật toán RSA
- Thuật toán mã hóa RSA
- Ứng dụng thuật toán mã hóa RSA
- Bài tập

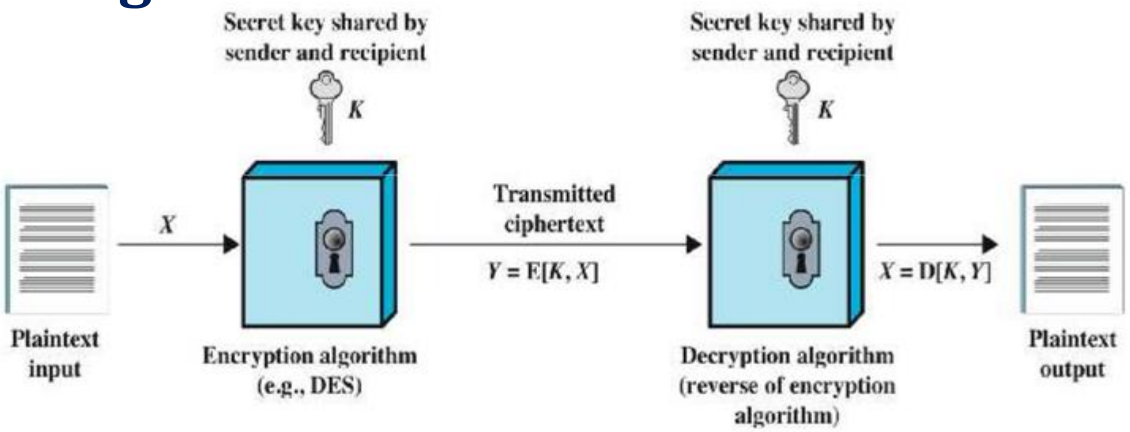
Dẫn nhập về Mật mã khóa công khai

- **Tiêu chuẩn an toàn thông tin:**
  - ***Confidentiality*** (*tính bí mật*): thông tin là bí mật với người không có thẩm quyền.
  - ***Integrity*** (*tính toàn vẹn*): bên nhận xác minh được dữ liệu toàn vẹn .
  - ***Authenticity*** (*tính xác thực*): bên nhận xác minh được nguồn gốc của thông tin.
  - ***Non-repudiation*** (*tính chống thoái thác*): bên tạo ra thông tin không thể phủ nhận thông tin mình đã tạo.
  - ***Reliability*** (*tính ổn định / tin cậy*): độ an toàn của thuật toán cao.



- Kỳ vọng đối với hệ mã hóa:
  - Đảm bảo *tính bí mật*, tính *chống thoái thác*, *tinh xác thực* và *ổn định*.

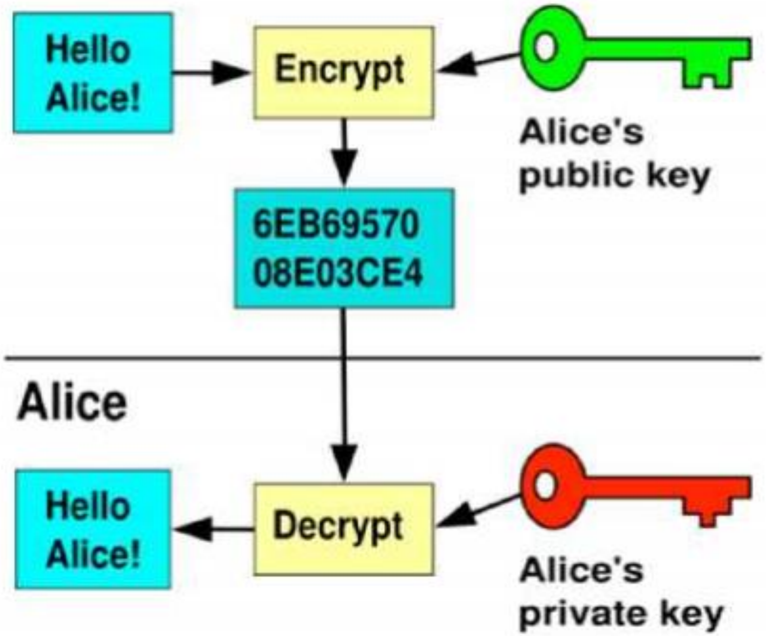
- **Hệ mã hóa Khóa đối xứng:**
  - Nguyên lý:
    - Mã hóa: **Y = E [X, K]**
    - Giải mã: **X = D [Y, K]**



  - Ưu điểm:
    - Tạo được *tính bí mật* cho thông tin.
  - Yếu điểm:
    - Phải cung cấp khóa giải mã cho đối tác => không an toàn.
    - Bên nhận không xác thực được nguồn gốc thông tin.
    - Không có cơ sở để "chống thoái thác".
  - Cần giải thuật mã hóa thỏa mãn nhiều yêu cầu hơn, an toàn hơn.

- **Nguyên lý:**
  - Dùng thuật toán **RSA** (*Rivest – Shamir – Adleman*)
  - Bộ khóa bao gồm **2 khóa**:
    - **K_r**: Khóa riêng (*private*) – giữ trong máy, không *public* ra ngoài.
    - **K_p**: Khóa chung (*public*) – không giữ trong máy, *public* ra ngoài.

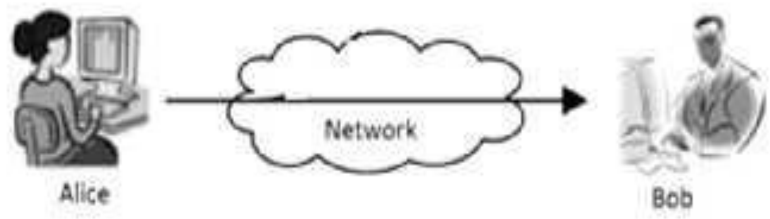
    => Mã hóa *khóa bất đối xứng* còn gọi là mã hóa *khóa công khai*.
  - Nguyên tắc mã hóa và giải mã:
    - Dữ liệu mã hóa bằng *khóa riêng* **K_r** => giải mã bằng *khóa chung* **K_p**
    - Dữ liệu mã hóa bằng khóa chung **K_p** => giải mã bằng khóa riêng **K_r**



Mật mã khóa bất đối xứng

- **Các trường hợp mã hóa và giải mã:**

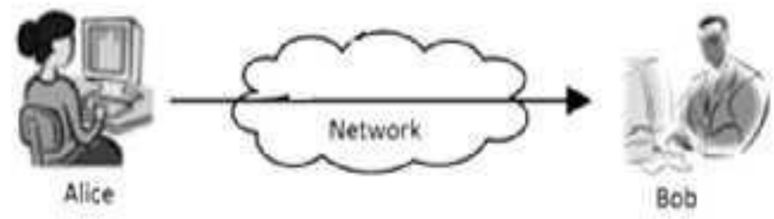
  - *Alice* muốn truyền thông tin cho *Bob*



  - **Trường hợp 1:**
    - *Bob* công khai khóa $\mathbf{K_{pB}}$ ra ngoài
    - *Alice* mã hóa bằng khóa chung $\mathbf{K_{pB}}$ của *Bob* => $\mathbf{C = e(M, K_{pB})}$
    - *Bob* giải mã bằng khóa riêng $\mathbf{K_{rB}}$ của *Bob* => $\mathbf{M = d(C, K_{rB})}$
  - Nhận xét trường hợp 1:
    - Nhận xét về *tính bí mật*: nếu *Trudy* bắt được thông tin **C**, *Trudy* có giải mã được không?
    - Nhận xét về *tính xác thực*: nếu *Trudy* tự tạo thông tin **C**, giả mạo là của *Alice*, gởi cho *Bob*, *Bob* có biết không?

Mật mã khóa bất đối xứng

- **Các trường hợp mã hóa và giải mã:**
  - *Alice* muốn truyền thông tin cho *Bob*



  - **Trường hợp 2:**
    - *Alice* công khai khóa K_{pA} ra ngoài.
    - *Alice* mã hóa bằng khóa riêng K_{rA} của *Alice* => **$C = e(M, K_{rA})$**
    - *Bob* giải mã bằng khóa chung K_{pA} của *Alice* => **$M = d(C, K_{pA})$**
  - Nhận xét trường hợp 2:
    - Nhận xét về *tính bí mật*: nếu *Trudy* bắt được thông tin **C**, *Trudy* có giải mã được không?
    - Nhận xét về *tính xác thực*: nếu *Trudy* tự tạo thông tin **C**, giả mạo là của *Alice*, gởi cho *Bob*, *Bob* có biết không?
    - Nhận xét về *tính chống từ chối*: *Alice* gởi thông tin **C** cho *Bob*, sau đó Alice có thoái thác rằng gói **C** đó không phải của cô ấy được không?

Toán học trong thuật toán RSA

- **Ước số chung lớn nhất:**
  - Định nghĩa:
    - Là số lớn nhất mà 2 số **a** và **b** có thể chia hết (dư số = **0**)
  - Ký hiệu và biểu thức:
    - Tiếng Việt: **ƯSCLN(a,b)** - Ước số chung lớn nhất của 2 số a và b.
    - Tiếng Anh: **GCD (a,b)** - *Greatest Common Divisor of a, b*.
  - Ví dụ: ước chung lớn nhất của **6** và **15** là **3** vì:
    - **6** và **15** cùng chia hết cho: **1**, **3**. Trong đó: 3 là số lớn nhất.
  - Ví dụ: tìm **GCD(27, 45)**
    - Các ước số của **27** là: **1, 3, 9, 27**.
    - Các ước số của **45** là: **1, 3, 5, 9, 15, 45**.
    - Ước chung lớn nhất của 2 số **27** và **45** là **9**.

Toán học trong thuật toán RSA

- **Giải thuật Euclid:** tìm ƯSCLN của 2 số nguyên.
  - Nguyên lý:
    - ƯSCLN của 2 số nguyên không đổi khi *thay số lớn* bằng *hiệu của chúng*
  - Ví dụ: tìm ƯSCLN của 2 số: **252** và **105**.
    - Thay **252** bằng (252-105= **147**) => cặp số mới: **147** và **105**.
    - Thay **147** bằng (147-105= **42**) => cặp số mới: **42** và **105**.
    - Thay **105** bằng (105-42= **63**) => cặp số mới: **42** và **63**.
    - Thay **63** bằng (63-42= **21**) => cặp số mới: **42** và **21**.
    - Thay **42** bằng (42-21= **21**) => cặp số mới: **21** và **21** <= ƯSCLN

Toán học trong thuật toán RSA

- **Cặp số nguyên tố cùng nhau:**
  - 2 số **a** và **b** gọi là ***số nguyên tố cùng nhau*** nếu giữa chúng chỉ có duy nhất ƯSCLN là **1**.
    - Ký hiệu biểu thức: **GCD (a,b) = 1**
  - Ví dụ:
    - Cặp số **9** và **28** là *số nguyên tố cùng nhau.*
      - Ước số của **9** gồm: **1, 3, 9**
      - Ước số của **28** gồm: **1, 2, 4, 7, 14, 28**
      - Cặp số **9** và **28** chỉ có ƯSCLN là **1**
  - Lưu ý:
    - Cặp ***số nguyên tố cùng nhau*** không hẳn là 2 *số nguyên tố*.

Phép toán Modulo (hay Modulus)

- Định nghĩa: *modulo* là lấy dư số của phép toán: **a** chia **n**.
  - Ký hiệu: **a mod n** hoặc: **a % n**
- Ví dụ: 27 mod 8 = 3, 35 mod 9 = 8.
- Các tính chất của *Modulo*:

$$(a + b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$$
$$(a - b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$$
$$(a \times b) \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$$

- Nhận xét tính chất của *Modulo*:
  - Gần tương tự như tính phân phối của phép nhân.
  - Không đúng nếu biểu thức bên trái là phép chia.

Toán học trong thuật toán RSA

- **Đồng dư (có cùng số dư):**
  - ***Định nghĩa***:
    - 2 số **a** và **b** gọi là **đồng dư** *mod* **n** nếu *a* và *b* **có cùng dư số** khi *mod* **n**.
    - Ký hiệu: $a \equiv b \ (mod \ \ n)$ hay viết tắt là $a \equiv b \ mod \ \ n$
  - Ví dụ: 2 số **10** và **13** là *đồng dư mod **3**.*
    - Vì: 10 % 3 =1;   13 % 3 = 1
  - ***Ứng dụng***: dùng thay thế khi tính *modulo của lũy thừa số lớn*
    - Ví dụ: tính $\mathbf{5^6 \ mod \ 7}$
    - $\mathbf{5^6 \ mod \ 7} = (\mathbf{5^2} \times \mathbf{5^2} \times \mathbf{5^2}) \ \mathbf{mod \ 7}$
      $= (\mathbf{5^2 \ mod \ 7} \times \mathbf{5^2 \ mod \ 7} \times \mathbf{5^2 \ mod \ 7}) \ \mathbf{mod \ 7}$ (*)
    - Thay $\mathbf{5^2 \ mod \ 7} =$ **25 mod 7** = **4** vào (*), ta được:
      $= (\mathbf{4} \times \mathbf{4} \times \mathbf{4}) \ \mathbf{mod \ 7}$ biểu thức này **đồng dư** với (*)
    - Vậy: $\mathbf{5^6 \ mod \ 7} \equiv (\mathbf{64}) \ \mathbf{mod \ 7} = \mathbf{1}$

Toán học trong thuật toán RSA

- **Giải thuật tính "modulo của lũy thừa số lớn"**
  - *Modular Exponentiation* (*ModExp*).
  - Ví dụ: tính **y = 5^{20} mod 11**
    - Thành lập bảng *bình phương liên tiếp* của cơ số **5**, **mod** với **11** (chọn các số mũ là số **2^n**)

5^1 mod 11 = 5 mod 11 = 5
5^2 mod 11 = 25 mod 11 = 3
5^4 mod 11 = $(5^2)^2$ mod 11 ≡ $(5^2$ mod $11)^2$ mod 11 ≡ 3^2 mod 11 = 9
5^8 mod 11 = $(5^4)^2$ mod 11 ≡ $(5^4$ mod $11)^2$ mod 11 ≡ 9^2 mod 11 = 4
5^{16} mod 11 = $(5^8)^2$ mod 11 ≡ $(5^8$ mod $11)^2$ mod 11 ≡ 4^2 mod 11 = 5

    - Từ yêu cầu: thay số mũ **20** thành (**16** + **4**) - tương đương **2^4** + **2^2**
    - Ta có: **5^{20} mod 11** = **5^{16+4} mod 11**

$$= (5^{16} * 5^4) \bmod 11$$

$$\equiv (5 * 9) \bmod 11 \equiv 45 \bmod 11 = 1$$

- **Giải thuật tính "modulo của lũy thừa số lớn"**
  - Ví dụ: tính **8^{17} mod 15**
    - Thành lập bảng *bình phương liên tiếp* của cơ số **8**, **mod** với **15** (chọn các số mũ là số **2^n**)

8^1 mod 15 = 8 mod 15 = 8
8^2 mod 15 = 64 mod 15 = 4
8^4 mod 15 = $(8^2)^2$ mod 15 ≡
8^8 mod 15 = $(8^4)^2$ mod 15 ≡
8^{16} mod 15 = $(8^8)^2$ mod 15 ≡

    - Từ yêu cầu: thay số mũ **17** thành (**16** + **1**)
    - Ta có: **8^{17} mod 15** = (**8^{16+1}**) **mod 15**
      = (**8^{16}** * **8^1**) **mod 15**
      ≡ (**x** * **y**) **mod 15** ≡ **?**

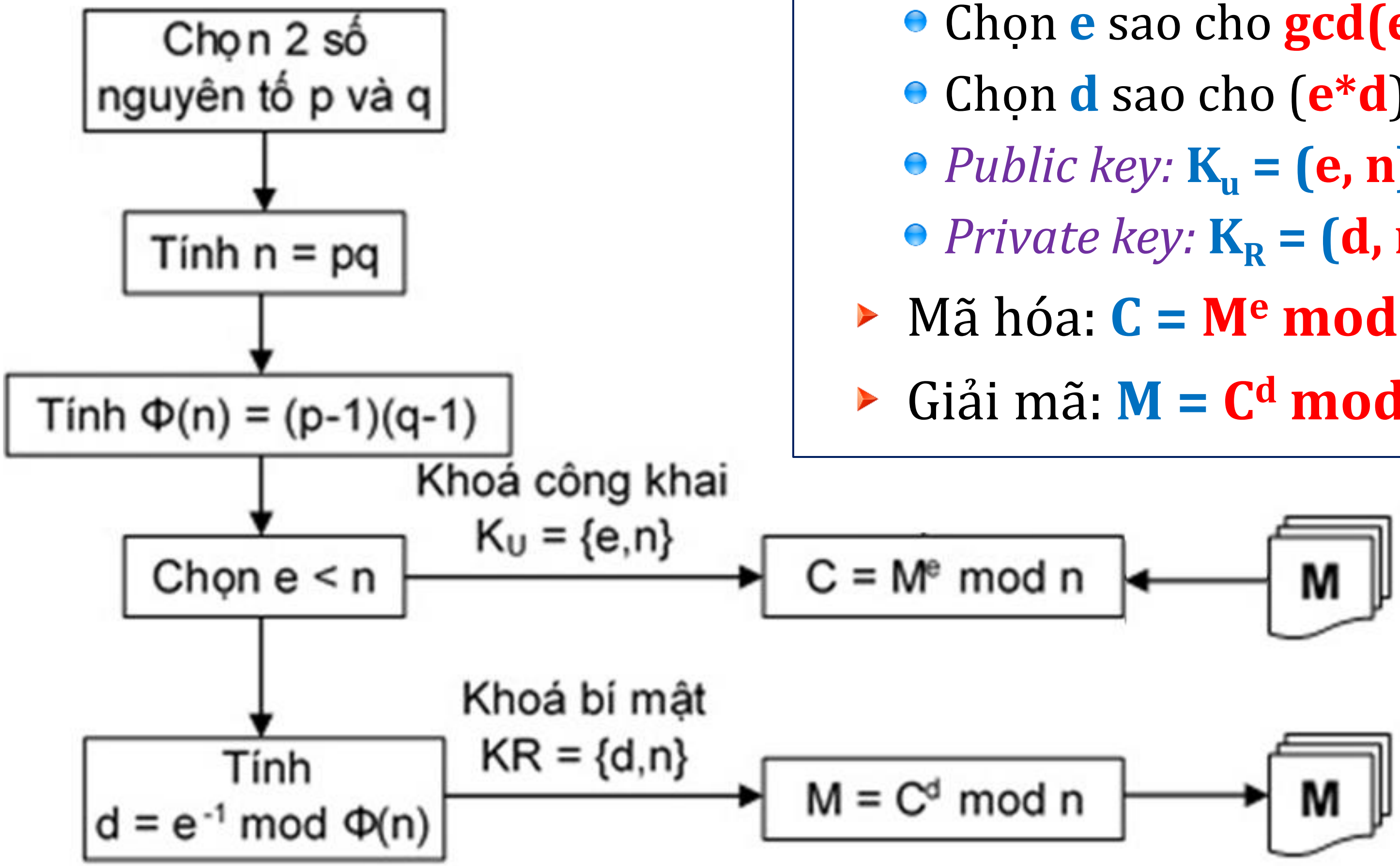
Thuật toán mã hóa RSA

- **Tổng quan về RSA:**
  - Đề xuất bởi *Rivest*, *Shamir* và *Adleman* (1977).
  - Là hệ mã công khai được sử dụng nhiều nhất cho đến nay.
  - Dựa trên các phép tính *modulo của lũy thừa số lớn* .
  - Độ an toàn phụ thuộc vào độ lớn của cặp *số nguyên tố* được chọn.

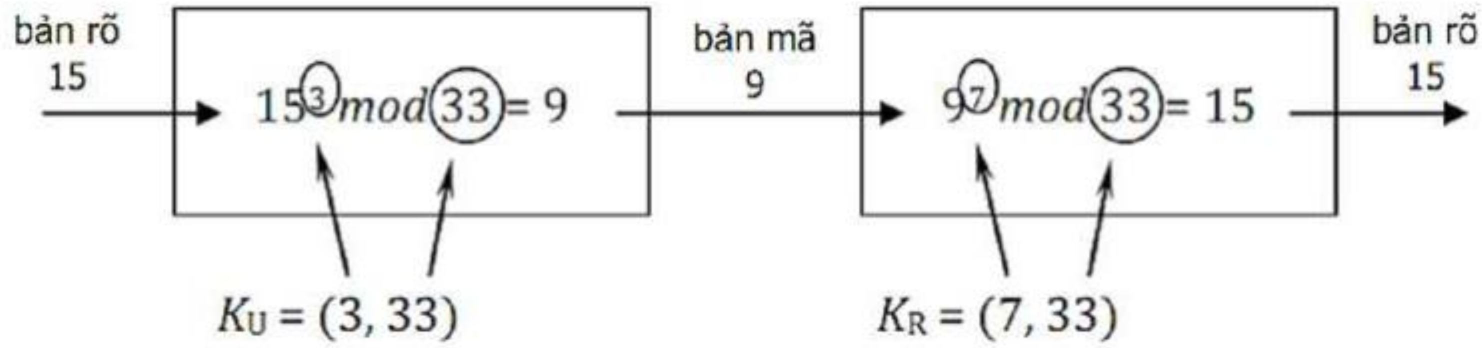
Thuật toán mã hóa RSA

- **Giải thuật RSA:**

> ➤ Qui trình sinh khóa:
> - Chọn 2 ***số nguyên tố lớn*** **p**, **q**
> - Tính **n = p.q** và **φ(n) = (p-1)(q-1)**
> - Chọn **e** sao cho **gcd(e, φ(n)) = 1**
> - Chọn **d** sao cho (**e*d**) **% φ(n) = 1**
> - *Public key:* $\mathbf{K_u = (e, n)}$
> - *Private key:* $\mathbf{K_R = (d, n)}$
>
> ➤ Mã hóa: $\mathbf{C = M^e \bmod n}$
>
> ➤ Giải mã: $\mathbf{M = C^d \bmod n}$



- **Minh họa RSA:**
  - Chọn 2 số nguyên tố: **p=11**, **q=3** => **n** = p.q = **33**
  - Tính **φ(n)** = (p-1)(q-1) = **20**
  - Chọn **e**: gcd(e, 20) = 1 => **e** = **3**
  - Chọn **d**: (e*d) % 20 = 1 => **d** = **7**
  - Khóa công khai K_u = (**3**, **33**), khóa bí mật K_R = (**7**, **33**)
  - **Cho M = 15, mã hóa: C = M^e mod n = 15^3 mod 33 => C = 9**
  - **Từ C = 9, giải mã: M = C^d mod n = 9^7 mod 33 => M = 15**

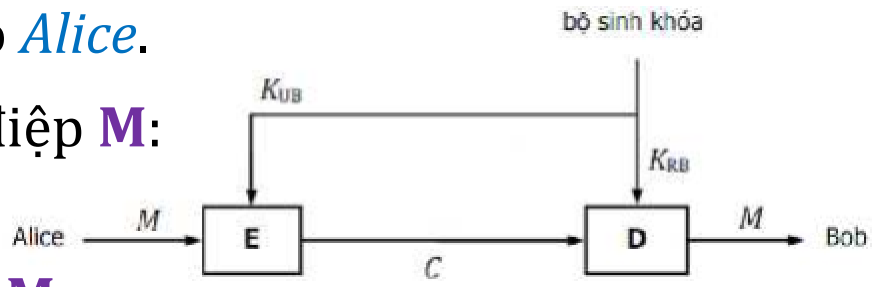


Thuật toán mã hóa RSA

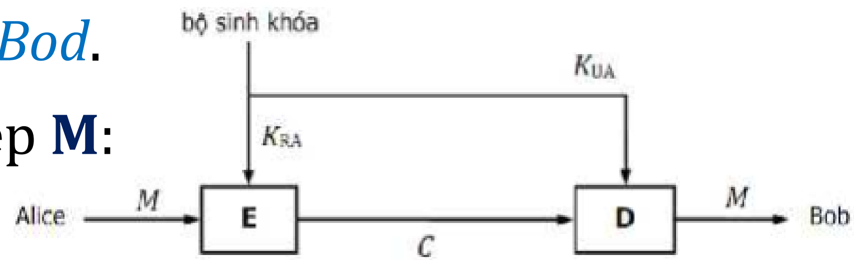
- **Nhận xét về độ phức tạp RSA:**
  - Phức tạp khi sinh khóa:
    - Khóa sẽ càng an toàn nếu chọn được ***cặp số nguyên tố đủ lớn***.
    - => vấn đề kiểm tra tính nguyên tố của 1 số lớn.
      (thuật toán *Miller-Rabin* hoặc *Solovay-Strassen*)
  - Phức tạp khi mã hóa và giả mã:
    - Phải thực hiện *modulo* cho lũy thừa số lớn (**M^e mod n** hoặc **C^d mod n**)
    - => giải thuật tính "*modulo của lũy thừa*" (*Modular Exponentiation*)
  - Khả năng chống phá mã bằng vét cạn (*Brute force*):
    - **n = p*q** (p và q là 2 số nguyên tố lớn).
    - Nếu **n > 1024** bit (tương đương **309** chữ số *Decimal*) => vô phương vét cạn.

Ứng dụng thuật toán mã hóa RSA

- **Tạo tính bí mật (*Confidentiality*)**
  - *Bob* sinh cặp khóa, gởi $\mathbf{K_{uB}}$ cho *Alice*.
  - *Alice* tạo bản mã **C** cho thông điệp **M**:
    $\mathbf{C = e(M, K_{uB})}$
  - Khi nhận, *Bod* giải mã **C** thành **M**:
    $\mathbf{M = d(C, K_{rB})}$



- **Tạo tính xác thực (*Authenticity*)**
  - *Alice* sinh cặp khóa, gởi $\mathbf{K_{uA}}$ cho *Bod*.
  - *Alice* tạo bản mã **C** cho thông điệp **M**:
    $\mathbf{C = e(M, K_{rA})}$
  - *Bod* dùng $\mathbf{K_{uA}}$ để giải mã $\mathbf{C = d(C, K_{uA})}$.
  - Nếu thành **M** => **C** là của *Alice*

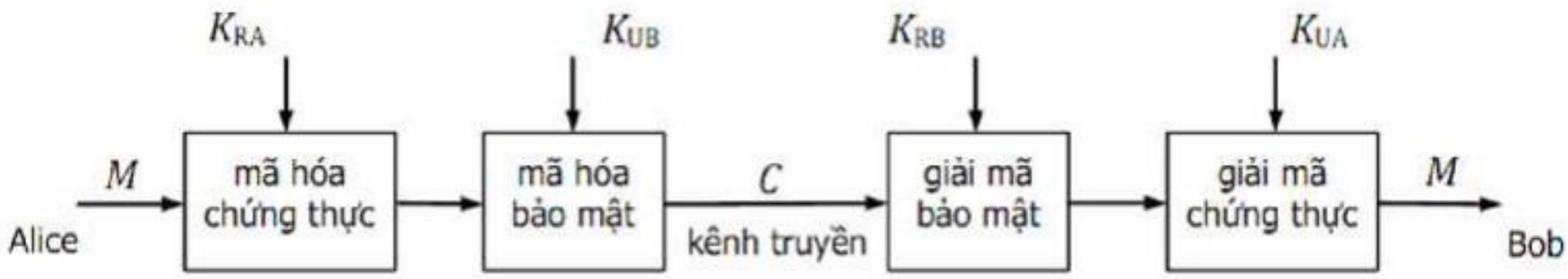


Ứng dụng thuật toán mã hóa RSA

- **Mô hình kết hợp *Confidentiality* và *Authenticity***

$$C = E(E(M, K_{RA}), K_{UB})$$

$$M = D(D(C, K_{RB}), K_{UA})$$



- **Bài tập:**
  - Sinh khóa RSA:
    - Chọn 2 số nguyên tố: **p=7**, **q=2**
    - Tính **n** =
    - Tính **φ(n)** =
    - Chọn **e**:
    - Chọn **d**:
    - *Public Key* là:
    - *Private key* là:
  - Bản rõ **M = 5**
    - **Mã hóa M** thành **C** =
    - **Giải mã C** thành **M** =

> - Qui trình sinh khóa:
>   - Chọn 2 ***số nguyên tố lớn*** **p**, **q**
>   - Tính **n = p.q** và **φ(n) = (p-1)(q-1)**
>   - Chọn **e** sao cho **gcd(e, φ(n))=1**
>   - Chọn **d** sao cho (**e*d**) **% φ(n) = 1**
>   - *Public key:* $K_u = (e, n)$
>   - *Private key:* $K_R = (d, n)$
> - Mã hóa: $C = M^e \bmod n$
> - Giải mã: $M = C^d \bmod n$

Cám ơn !