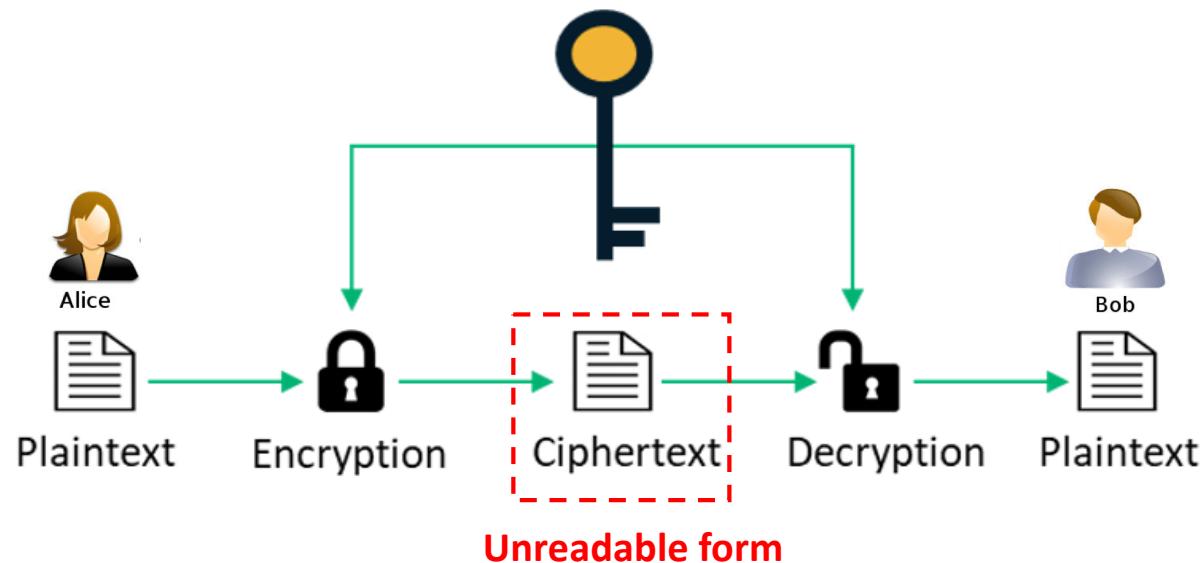


Hệ mã hóa đối xứng

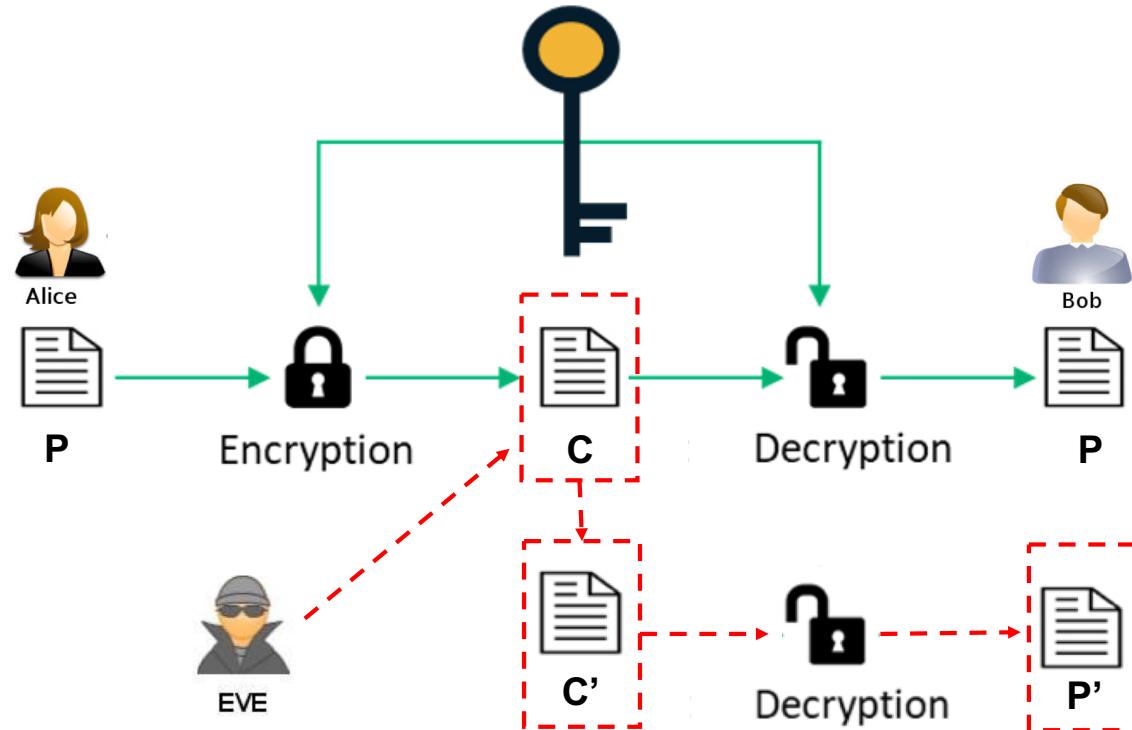
Tính bảo mật (**confidentiality**):

Data is **unreadable** until it is decrypted using the correct cipher and key.



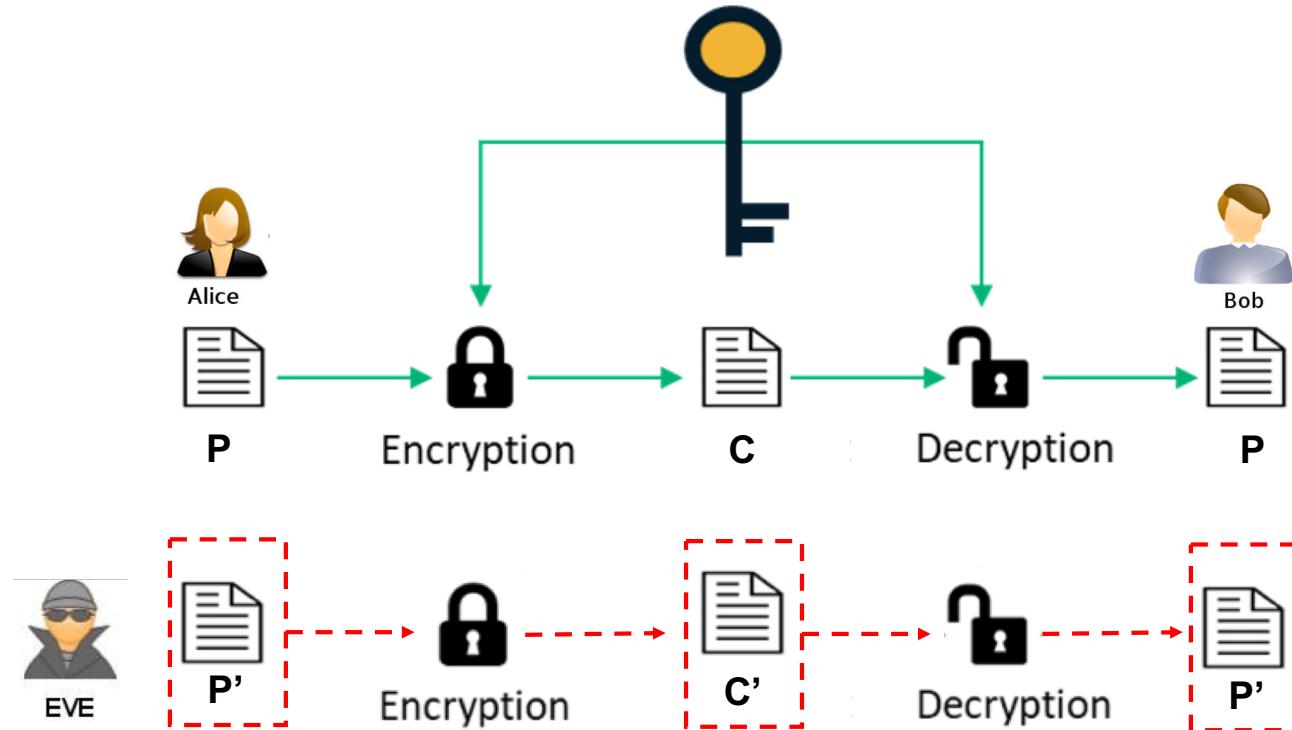
Hệ mã hoá đối xứng

Tính chứng thực (authentication)



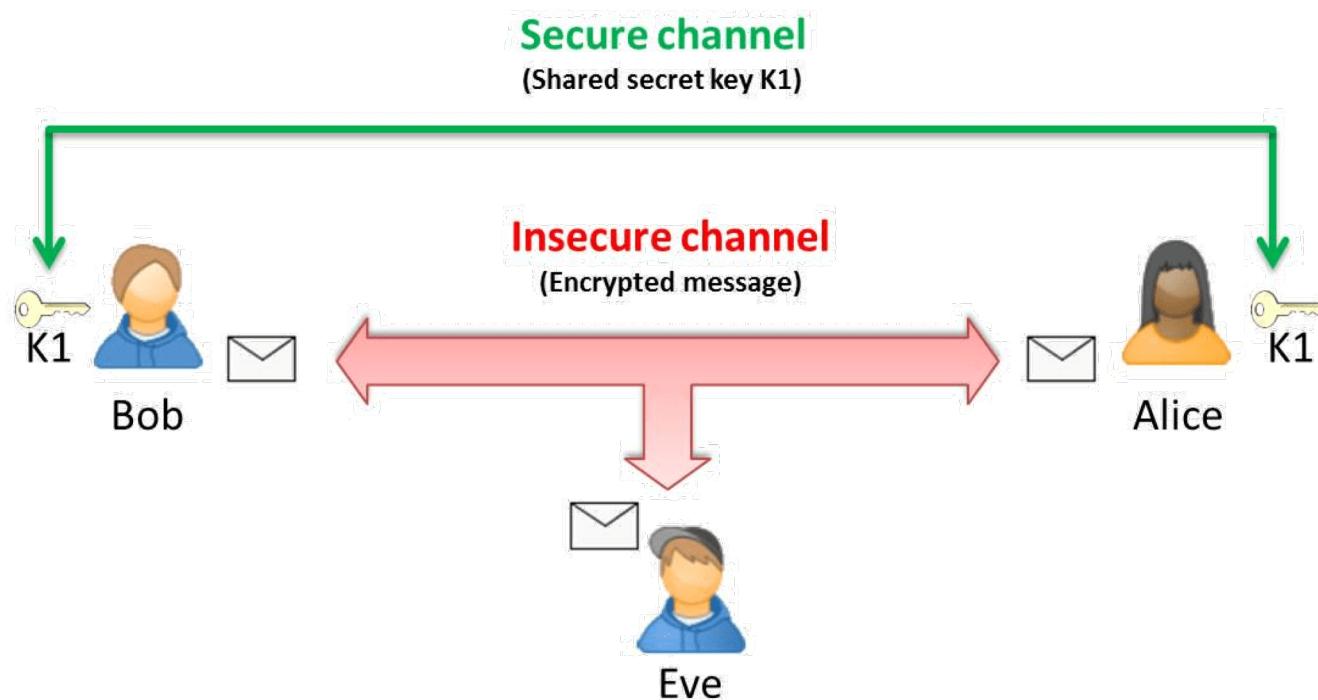
Hệ mã hóa đối xứng

Tính không từ chối (**non-repudiation**)



- Eve somehow knew the Key, and sent a message P' to Bob
- How can Bob certify that P' was sent by Alice?

Hệ mã hóa đối xứng



1. Trao đổi khoá an toàn
2. Tính bí mật của khoá



Khoá lập mã và khoá giải mã là khác nhau

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.1. Một số kiến thức về Lý thuyết số

Cho a, b và n là các số nguyên, phép modulo có các tính chất:

- a) $(a + b) \text{ mod } n = [(a \text{ mod } n) + (b \text{ mod } n)] \text{ mod } n$
- b) $(a - b) \text{ mod } n = [(a \text{ mod } n) - (b \text{ mod } n)] \text{ mod } n$
- c) $(a \times b) \text{ mod } n = [(a \text{ mod } n) \times (b \text{ mod } n)] \text{ mod } n$

Nếu $a \text{ mod } n = 0$ (viết cách khác $a \equiv 0 \text{ mod } n$) thì có nghĩa là a chia hết cho n , hay n là ước số của a .

Uớc số chung lớn nhất của hai số: ký hiệu $\text{gcd}(a, b)$.

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.1. Một số kiến thức về Lý thuyết số

Thuật toán Euclid dùng để tìm ước số chung lớn nhất của hai số nguyên a và b . Ta ký hiệu ước số chung lớn nhất này là $\gcd(a, b)$. Thuật toán này dựa trên định lý sau:

Định lý: với mọi số nguyên $a \geq 0$ và $b > 0$ thì:

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

```
/* Thuật toán Euclid tính gcd(a,b) */
```

```
EUCLID (a,b)
```

```
    A = a; B = b;  
    while B<>0 do  
        R = A mod B;  
        A = B;  
        B = R;  
    end while  
    return A;
```

Thuật toán được minh họa qua hình sau:

Ví dụ: $a=57, b=42$

$$\begin{array}{l} A_1 = B_1q + R_1 \\ A_2 = B_2q + R_2 \\ A_3 = B_3q + R_3 \\ \dots \\ A_n = B_nq + 0 \end{array}$$

$57 = 42 \times 1 + 15$
 $42 = 15 \times 2 + 12$
 $15 = 12 \times 1 + 3$
 $12 = 3 \times 4 + 0$
 $3 \quad 0$

$gcd(a,b) \leftarrow A_{n+1} \quad 0$

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.1. Một số kiến thức về Lý thuyết số

Một số p được gọi là số nguyên tố nếu p chỉ chia hết cho 1 và chính nó, ngoài ra không chia hết cho số nào khác từ 2 đến $p - 1$.

Hai số nguyên a, b được gọi là nguyên tố cùng nhau nếu USCLN của a và b là 1. Ký hiệu: $a \perp b$. Ví dụ: $3 \perp 8, 7 \perp 9, 4 \perp 15$. Hai số 20 và 15 không nguyên tố cùng nhau vì có USCLN là 5.

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.1. Một số kiến thức về Lý thuyết số

Nếu hai số nguyên a và n nguyên tố cùng nhau, thì tồn tại số nguyên w sao cho:

$$a \cdot w \equiv 1 \pmod{n}$$

Ta gọi w là phần tử nghịch đảo của a trong phép modulo cho n và ký hiệu là a^{-1}

Ví dụ:

- $n = 10, a = 7$ là hai số nguyên tố cùng nhau, do đó tìm được $a^{-1} = 3$ ($21 \equiv 1 \pmod{10}$)

a^{-1}	0	1	2	3	4	5	6	7	8	9
$a^{-1} \times 7$	0	7	4	1	8	5	2	9	6	3

- $n = 10, a = 2$ không phải là hai số nguyên tố cùng nhau, ta có bảng phép nhân sau:

a^{-1}	0	1	2	3	4	5	6	7	8	9
$a^{-1} \times 2$	0	2	4	6	8	0	2	4	6	8

Trong bảng trên không tồn tại số a^{-1} nào sao cho $a \cdot a^{-1} \equiv 1 \pmod{10}$. Vậy không tồn tại phần tử nghịch đảo.

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.1. Một số kiến thức về Lý thuyết số

Định lý Fermat:

Dinh ly:

Nếu p là số nguyên tố và a là số nguyên không chia hết cho p thì $a^{p-1} \equiv 1 \pmod{p}$

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.2. Mã hóa khóa công khai

Nhược điểm của mã hóa đối xứng (**symmetric cryptography**) :



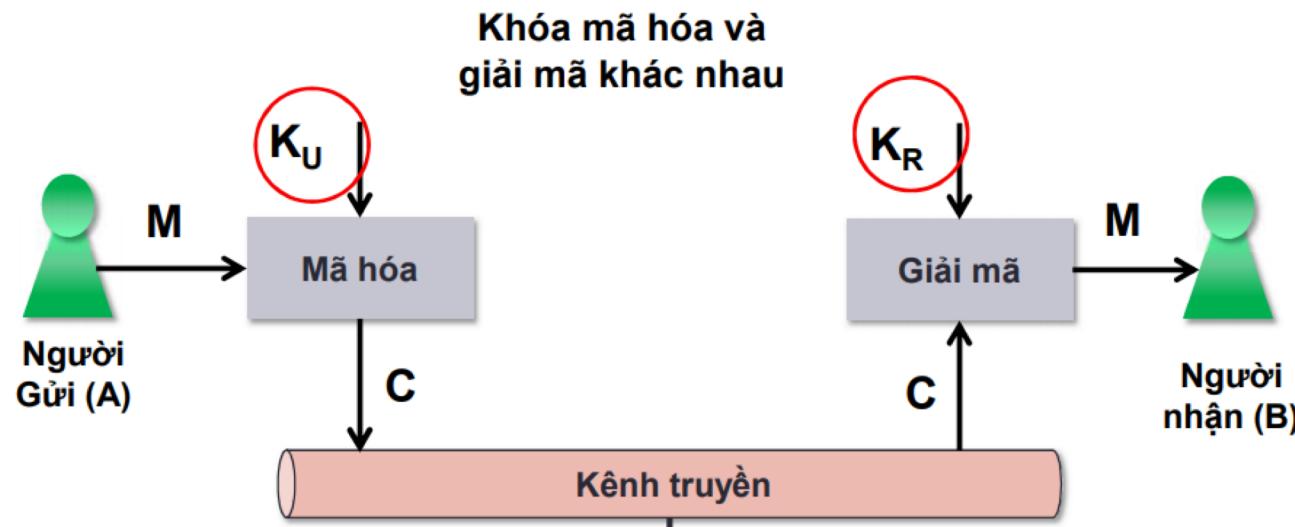
- Vấn đề trao đổi khóa giữa người gửi và người nhận: Cần phải có một kênh an toàn để trao đổi khóa sao cho khóa phải được giữ bí mật chỉ có người gửi và người nhận biết. Điều này tỏ ra không hợp lý khi mà ngày nay, khối lượng thông tin luân chuyển trên khắp thế giới là rất lớn. Việc thiết lập một kênh an toàn như vậy sẽ tốn kém về mặt chi phí và chậm trễ về mặt thời gian.
- Tính bí mật của khóa: không có cơ sở quy trách nhiệm nếu khóa bị tiết lộ.

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.2. Mã hóa khóa công khai

Hệ mã hóa bất đối xứng (asymmetric cryptography) :

1976: Diffie và Hellman đã tìm ra một phương pháp mã hóa khác mà có thể giải quyết được hai vấn đề trên, đó là **mã hóa khóa công khai (public key cryptography)** hay còn gọi là **mã hóa bất đối xứng**



Một khóa được giữ bí mật chỉ một người biết, còn khóa kia được công khai. Do đó mô hình mã hóa trên được gọi là **mã hóa khóa công khai**.

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.2. Mã hóa khóa công khai

Hai phương án mã hóa và giải mã:

- Phương án 1: khóa lập mã là công khai, khóa giải mã là bí mật
 - Phương án 2: Khóa lập mã là bí mật, khóa giải mã là công khai
- Để tránh nhầm lẫn với khóa bí mật của mã đối xứng, khóa bí mật trong mô hình trên được gọi là khóa riêng (private key) và ký hiệu là K_R .
 - Khóa công khai (public key) được ký hiệu là K_U .
 - Bản rõ được ký hiệu là M , còn bản mã giữ nguyên ký hiệu là C
 - Phương án 1 viết lại thành:

$$C = E(M, K_U)$$

$$M = D(C, K_R)$$

- Phương án 2 viết lại thành:

$$C = E(M, K_R)$$

$$M = D(C, K_U)$$

CHƯƠNG 4: Hệ mã hóa khóa công khai

4.2. Mã hóa khóa công khai

Mối quan hệ giữa khóa công khai và khóa bí mật:

- Mối quan hệ toán học: $K_R = f(K_U)$.
- Yêu cầu: việc tính toán hàm f phải là bất khả thi về mặt thời gian.

→ **Hàm một chiều (one-way function): việc tính nghịch đảo rất khó thực hiện**

Cho p và q suy ra $N=p \cdot q$
Dễ

Cho N suy ra p và q
Khó

→ **Hệ mã hóa RSA** (Rivest, Shamir và Adleman (MIT 1977))

Một số hệ mã hóa khóa công khai khác: phương pháp Knapsack, RSA, Elgaman, và phương pháp đường cong elliptic ECC, ...

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.3. Mã hóa RSA

- RSA là một phương pháp mã hóa theo khôi
- bản rõ M và bản mã C là các số nguyên từ 0 đến 2^i với i số bít của khôi
- Kích thước thường dùng của i là 1024 bít
- Hàm một chiều là vấn đề phân tích một số thành thừa số nguyên tố

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.3. Mã hóa RSA

- Nguyên tắc thực hiện:

- 1) Chọn hai số nguyên tố lớn p và q và tính $N = pq$. Cần chọn p và q sao cho:
 $M < 2^{i-1} < N < 2^i$. Với $i = 1024$ thì N là một số nguyên dài khoảng 309 chữ số.
- 2) Tính $n = (p - 1)(q - 1)$
- 3) Tìm một số e sao cho e nguyên tố cùng nhau với n
- 4) Tìm một số d sao cho $e \cdot d \equiv 1 \pmod{n}$ (d là nghịch đảo của e trong phép modulo n)
- 5) Hủy bỏ n, p và q . Chọn khóa công khai K_U là cặp (e, N) , khóa riêng K_R là cặp (d, N)
- 6) Việc mã hóa thực hiện theo công thức:
 - Theo phương án 1, mã hóa bảo mật: $C = E(M, K_U) = M^e \pmod{N}$
 - Theo phương án 2, mã hóa chứng thực: $C = E(M, K_R) = M^d \pmod{N}$
- 7) Việc giải mã thực hiện theo công thức:
 - Theo phương án 1, mã hóa bảo mật: $\bar{M} = D(C, K_R) = C^d \pmod{N}$
 - Theo phương án 2, mã hóa chứng thực: $\bar{M} = D(C, K_U) = C^e \pmod{N}$

Bản rõ M có kích thước $i-1$ bít, bản mã C có kích thước i bít.

CHƯƠNG 4: Hệ mã hóa khoá công khai

4.3. Mã hóa RSA

Tính đúng đắn của RSA:

Bản giải mã chính là bản rõ ban đầu: $\bar{M} = M$

Không thể suy ra K_R từ K_U , nghĩa là tìm cặp (d, N) từ cặp (e, N) :

Có e và N , muốn tìm d , ta phải dựa vào công thức: $e \cdot d \equiv 1 \pmod{n}$. Do đó phải tính được n . Vì $n = (p - 1)(q - 1)$ nên suy ra phải tính được p và q . Vì $N = pq$ nên ta chỉ có thể tính được p và q từ N . Tuy nhiên điều này là bất khả thi vì $N = pq$ là hàm một chiều. Vậy không thể tính được K_R từ K_U .

CHƯƠNG 4: Hệ mã hóa khoá công khai

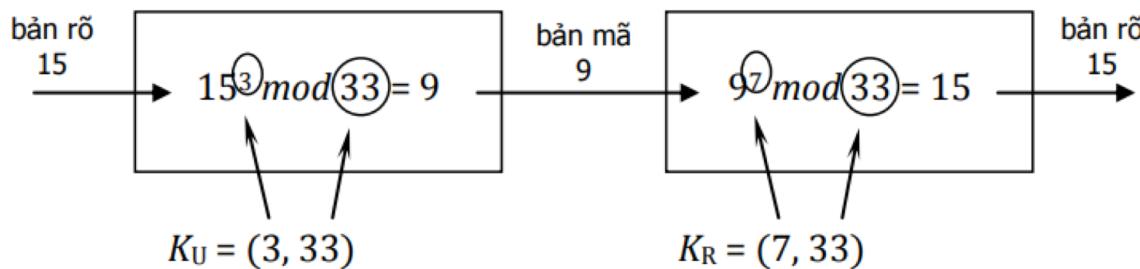
Ví dụ RSA với kích thước khóa là 6 bit:

- 1) Chọn $p = 11$ và $q = 3$, do đó $N = pq = 33$ ($2^5 = 32 < 33 < 64 = 2^6$)
- 2) $n = (p-1)(q-1) = 20$
- 3) Chọn $e = 3$ nguyên tố cùng nhau với n
- 4) Tính nghịch đảo của e trong phép modulo n được $d = 7$ ($3 \times 7 = 21$)
- 5) Khóa công khai $K_U = (e, N) = (3, 33)$. Khóa bí mật $K_R = (d, N) = (7, 33)$

Theo phương án 1 (mã hóa bảo mật):

$p=11; q=13, e =11, M =15.$ Tính $C = ?$

- 6) Mã hóa bản rõ $M = 15$:
$$C = M^e \text{ mod } N = 15^3 \text{ mod } 33 = 9 \quad (\text{vì } 15^3 = 3375 = 102 \times 33 + 9)$$
- 7) Giải mã bản mã $C = 9$:
$$\bar{M} = C^d \text{ mod } N = 9^7 \text{ mod } 33 = 15 = M \quad (\text{vì } 9^7 = 4.782.696 = 144.938 \times 33 + 15)$$



CHƯƠNG 4: Hệ mã hóa khoá công khai

Ví dụ RSA với kích thước khóa là 6 bit:

- 1) Chọn $p = 11$ và $q = 3$, do đó $N = pq = 33$ ($2^5 = 32 < 33 < 64 = 2^6$)
- 2) $n = (p-1)(q-1) = 20$
- 3) Chọn $e = 3$ nguyên tố cùng nhau với n
- 4) Tính nghịch đảo của e trong phép modulo n được $d = 7$ ($3 \times 7 = 21$)
- 5) Khóa công khai $K_U = (e, N) = (3, 33)$. Khóa bí mật $K_R = (d, N) = (7, 33)$

Theo phương án 2 (mã hóa chứng thực):

- 6) Mã hóa bản rõ $M = 15$:

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

- 7) Giải mã bản mã $C = 9$:

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

CHƯƠNG 4: Hệ mã hoá khoá công khai

Độ an toàn của RSA

- 1) Vết cạn khóa: cách tấn công này thử tất cả các khóa d có thể có để tìm ra bản giải mã có ý nghĩa, tương tự như cách thử khóa K của mã hóa đối xứng. Với N lớn, việc tấn công là bất khả thi.
- 2) Phân tích N thành thừa số nguyên tố $N = pq$: Chúng ta đã nói rằng việc phân tích phải là bất khả thi thì mới là hàm một chiều, là nguyên tắc hoạt động của RSA. Tuy nhiên, nhiều thuật toán phân tích mới đã được đề xuất, cùng với tốc độ xử lý của máy tính ngày càng nhanh, đã làm cho việc phân tích N không còn quá khó khăn như trước đây. Năm 1977, các tác giả của RSA đã treo giải thưởng cho ai phá được RSA có kích thước của N vào khoảng 428 bít, tức 129 chữ số. Các tác giả này ước đoán phải mất 40 nghìn triệu năm mới có thể giải được.

CHƯƠNG 4: Hệ mã hoá khoá công khai

Độ an toàn của RSA

<i>Số chữ số của N</i>	<i>Số bít</i>	<i>Năm phá mã</i>	<i>Thuật toán</i>
100	322	1991	Quadratic sieve
110	365	1992	Quadratic sieve
120	398	1993	Quadratic sieve
129	428	1994	Quadratic sieve
130	431	1996	GNFS
140	465	1999	GNFS
155	512	1999	GNFS
160	530	2003	Lattice sieve
174	576	2003	Lattice sieve
200	633	2005	Lattice sieve

Bảng liệt kê các mốc phá mã RSA

→ **kích thước của N phải khoảng 1024 bít (309 chữ số)**

CHƯƠNG 4: Hệ mã hoá khoá công khai

Độ an toàn của RSA

- 3) Đo thời gian: Đây là một phương pháp phá mã không dựa vào mặt toán học của thuật toán RSA, mà dựa vào một “hiệu ứng lè” sinh ra bởi quá trình giải mã RSA. Hiệu ứng lè đó là thời gian thực hiện giải mã. Giả sử người phá mã có thể đo được thời giải mã $M = C^d \text{ mod } N$ dùng thuật toán bình phương liên tiếp. Trong thuật toán bình phương liên tiếp, nếu một bít của d là 1 thì xảy ra hai phép modulo, nếu bít đó là 0 thì chỉ có một phép modulo, do đó thời gian thực hiện giải mã là khác nhau. Bằng một số phép thử chosen-plaintext, người phá mã có thể biết được các bít của d là 0 hay 1 và từ đó biết được d .

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.1. Hệ mã Rabin

Encryption

The keys for the Rabin cryptosystem are generated as follows:

1. Choose two large distinct prime numbers p and q such that $p \equiv 3 \pmod{4}$ and $q \equiv 3 \pmod{4}$.
2. Compute $n = pq$.

Then n is the public key and the pair (p, q) is the private key.

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.1. Hệ mã Rabin

Decryption

The message m can be recovered from the ciphertext c by taking its square root modulo n as follows.

1. Compute the square root of c modulo p and q using these formulas:

$$m_p = c^{\frac{1}{4}(p+1)} \pmod{p}$$

$$m_q = c^{\frac{1}{4}(q+1)} \pmod{q}$$

2. Use the [extended Euclidean algorithm](#) to find y_p and y_q such that $y_p \cdot p + y_q \cdot q = 1$.

3. Use the [Chinese remainder theorem](#) to find the four square roots of c modulo n :

$$r_1 = (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \pmod{n}$$

$$r_2 = n - r_1$$

$$r_3 = (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \pmod{n}$$

$$r_4 = n - r_3$$

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.1. Hệ mã Rabin

Example

As an example, take $p = 7$ and $q = 11$, then $n = 77$. Take $m = 20$ as our plaintext. The ciphertext is thus $c = m^2 \bmod n = 400 \bmod 77 = 15$.

Decryption proceeds as follows:

1. Compute $m_p = c^{\frac{1}{4}(p+1)} \bmod p = 15^2 \bmod 7 = 1$ and $m_q = c^{\frac{1}{4}(q+1)} \bmod q = 15^3 \bmod 11 = 9$.
2. Use the extended Euclidean algorithm to compute $y_p = -3$ and $y_q = 2$.
3. Compute the four plaintext candidates:

$$r_1 = (-3 \cdot 7 \cdot 9 + 2 \cdot 11 \cdot 1) \bmod 77 = 64$$

$$r_2 = 77 - 64 = 13$$

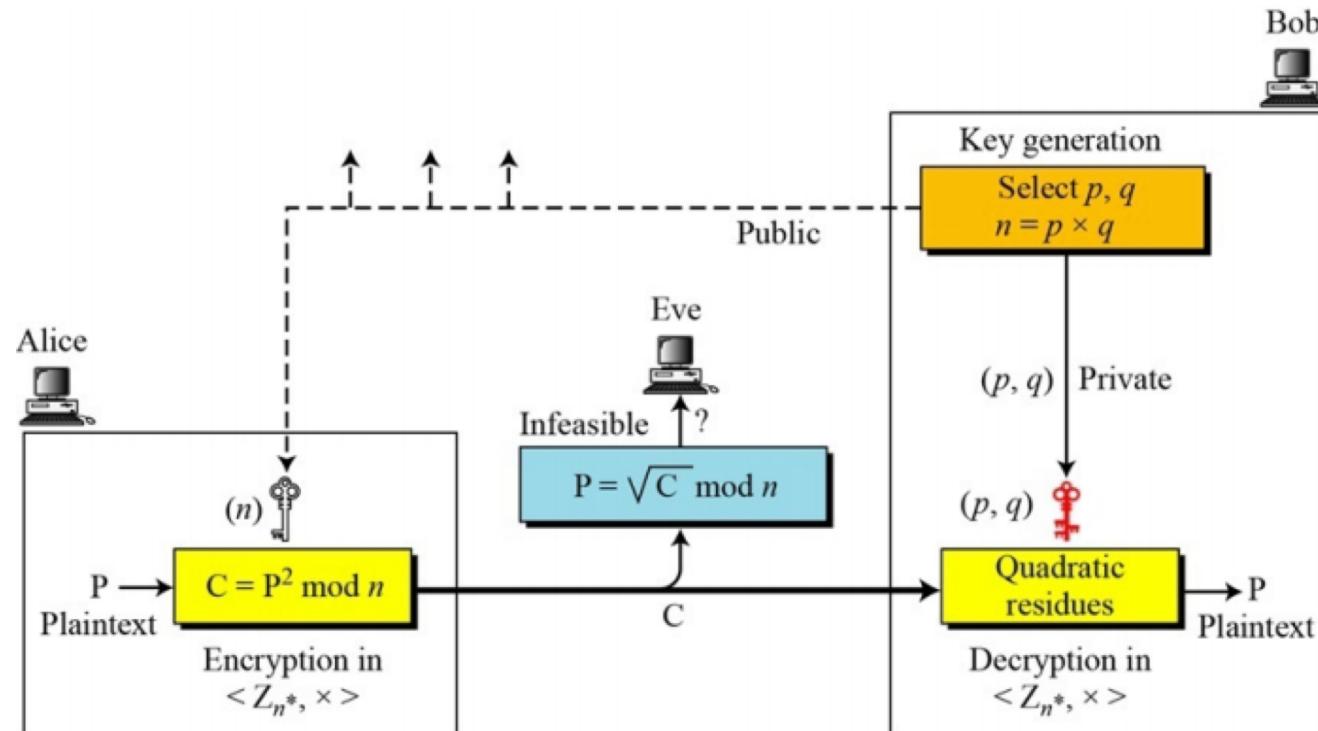
$$r_3 = (-3 \cdot 7 \cdot 9 - 2 \cdot 11 \cdot 1) \bmod 77 = 20$$

$$r_4 = 77 - 20 = 57$$

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.1. Hệ mã Rabin



Michael O. Rabin

(Nguồn: <http://www.rutherfordjournal.org>)

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.1. Hệ mã Rabin

Ưu điểm:

- + Tính an toàn được chứng minh hoàn toàn tương đương với bài toán PTTSNT, nói cách khác tính ATBM của Rabin là có thể chứng minh được (provable)
- + Ngoại trừ trường hợp RSA hoạt động với e nhỏ còn thuật toán sinh mã của Rabin nhanh hơn nhiều so với RSA là hệ đòi hỏi phải tính luỹ thừa. Thời gian giải mã thì tương đương nhau

Nhược điểm:

Vì phương trình giải mã cho 4 nghiệm nên làm khó dễ việc giải mã. Thông thường, bắn rõ trước khi được mã hoá cần được nối thêm vào đuôi một chuỗi số xác định để làm dấu vết nhận dạng

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.2. Hệ mã Elgamal

Tạo khoá: Alice chọn một số nguyên tố p và hai số nguyên ngẫu nhiên g và u , cả hai đều nhỏ hơn p . Sau đó tính

$$y = g^u \pmod{p}$$

Bây giờ khóa công khai của Alice được lấy là (p, g, y) , khoá mật là u .

Sinh mã:

1. Nếu Bob muốn mã hoá một tin X để truyền cho Alice thì trước hết anh ta chọn một số ngẫu nhiên k sao cho $(k, p-1) = 1$
2. Tính

$$a = g^k \pmod{p}$$

$$b = y^k X \pmod{p}$$

Mã là $Y = (a, b)$ và có độ dài gấp đôi bản rõ.

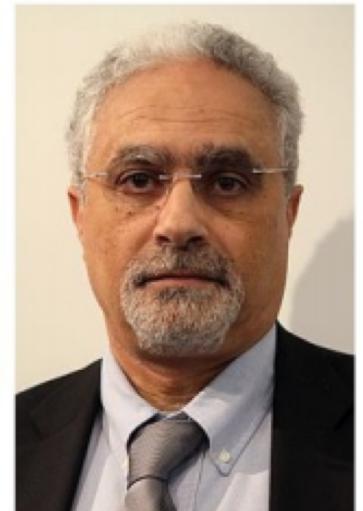
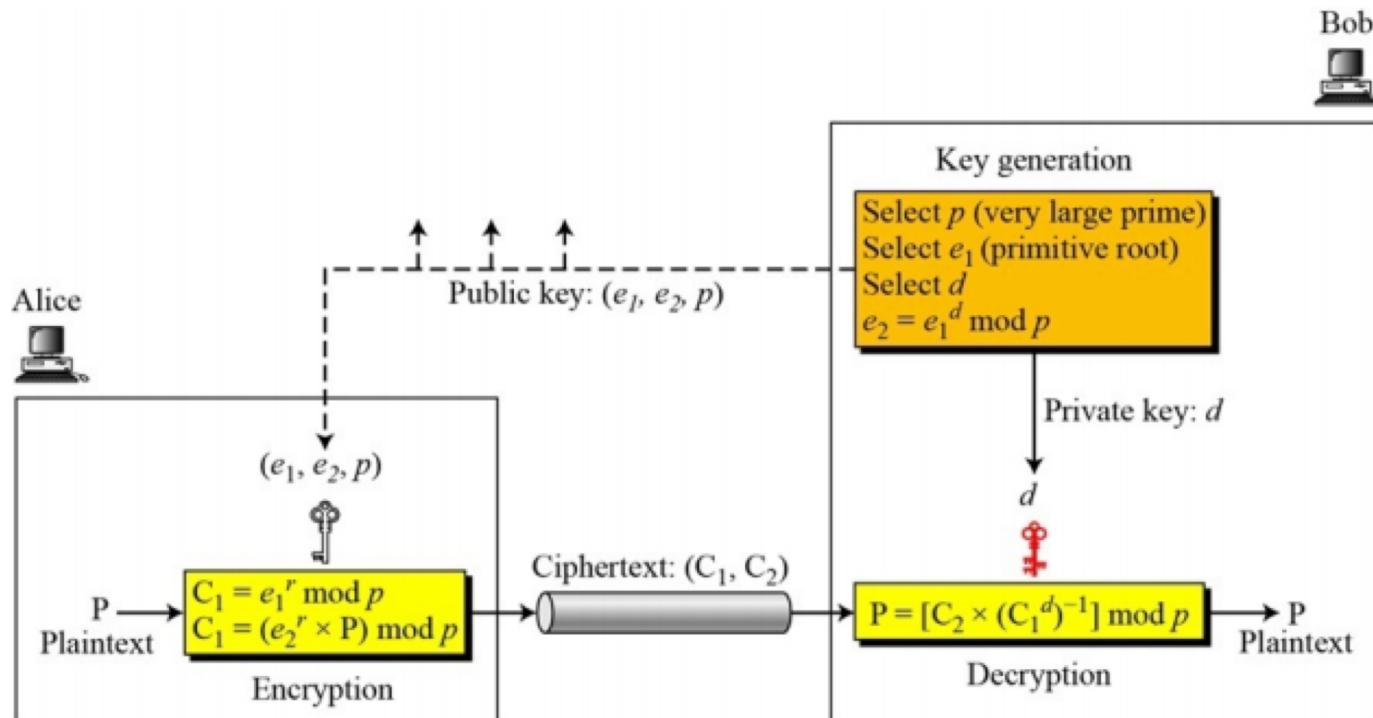
Giải mã: Alice nhận được $Y = (a, b)$ và giải ra X theo công thức sau:

$$X = \frac{b}{a^u} \pmod{p}$$

CHƯƠNG 4: Hệ mã hoá khoá công khai

4.3. Một số hệ mã hoá khoá công khai khác

4.3.2. Hệ mã Elgamal



Taher Elgamal

(Nguồn: <http://cs-exhibitions.uni-klu.ac>.