Bài 2.
Các hệ mật mã

ONE LOVE. ONE FUTURE.

1

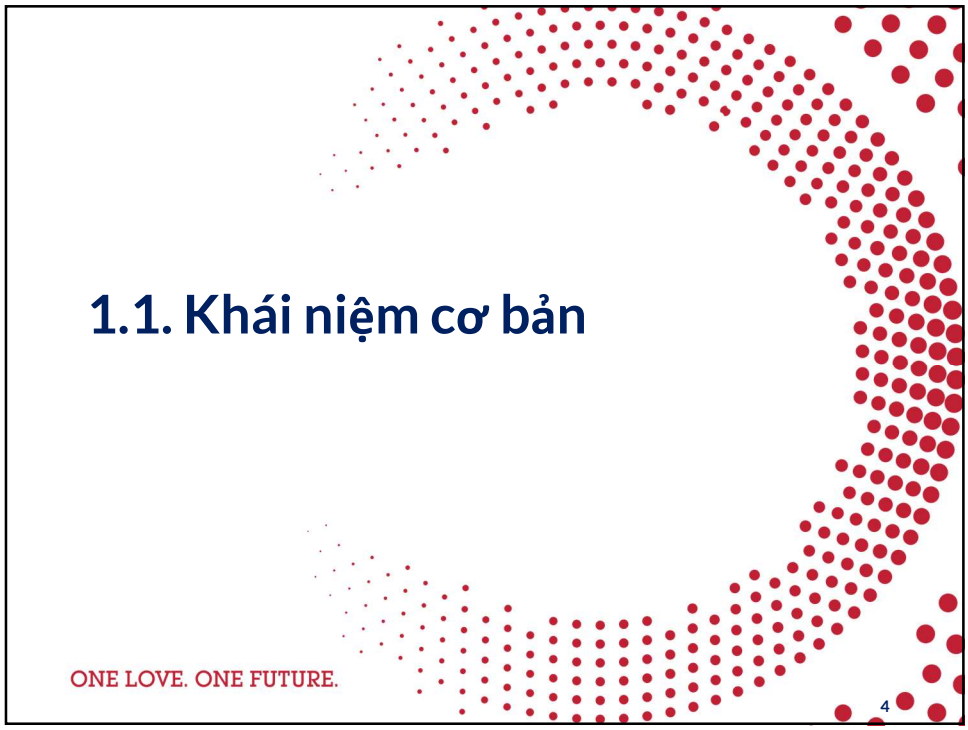
Nội dung

- Mật mã (cipher) là gì?
- Nguyên tắc chung của các hệ mật mã
- Hệ mật mã khóa đối xứng
- Hệ mật mã khóa bất đối xứng

2


1. Mật mã là gì?
ONE LOVE. ONE FUTURE.
3

3


1.1. Khái niệm cơ bản
ONE LOVE. ONE FUTURE.
4

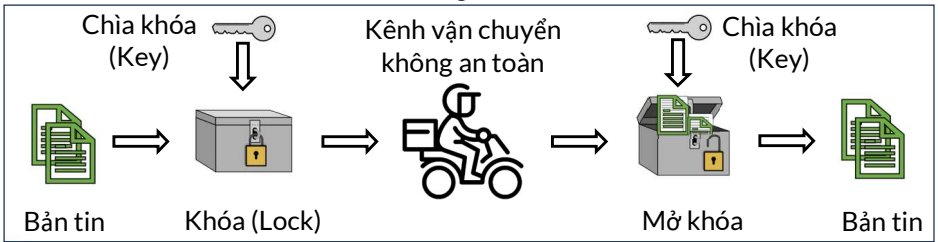
4

Khái niệm cơ bản

- Mã hóa (code): biến đổi cách thức biểu diễn thông tin
- Mật mã (cipher):
  - Khái niệm cũ: Mã hóa để che giấu, giữ mật thông tin
  - Khái niệm mới: đảm bảo bí mật + toàn vẹn + xác thực
- Mật mã học (cryptography): ngành khoa học nghiên cứu các phương pháp toán học để mã hóa giữ mật thông tin
- Thám mã (cryptoanalysis): nghiên cứu các phương pháp toán học để phá vỡ hệ mật mã
- Trong học phần này, chỉ đề cập đến khái niệm cơ bản và cách thức sử dụng các phương pháp mật mã

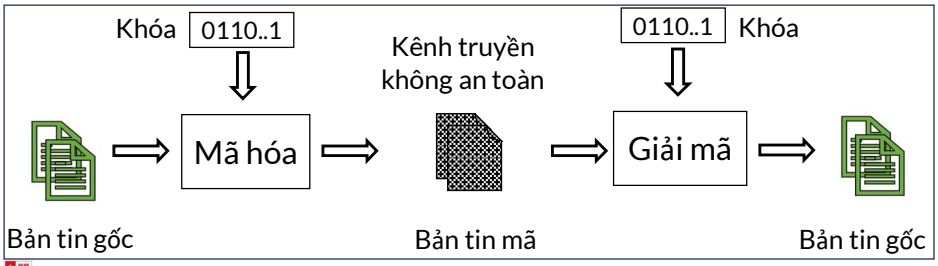
Mục tiêu của mật mã: Tính bí mật

- Tính bí mật(Confidentiality): Đối phương không đọc được bản tin
- Giải pháp 1: Không sử dụng công cụ tính toán
  - Alice cất bản tin vào hòm và khóa lại
  - Alice chuyển bản tin (đã được cất trong hòm) quá kênh vận chuyển không bí mật
  - Eve có thể đánh cắp hòm nhưng không thể mở được nếu không có chìa khóa
  - Bob nhận được hòm và sử dụng chìa khóa để mở



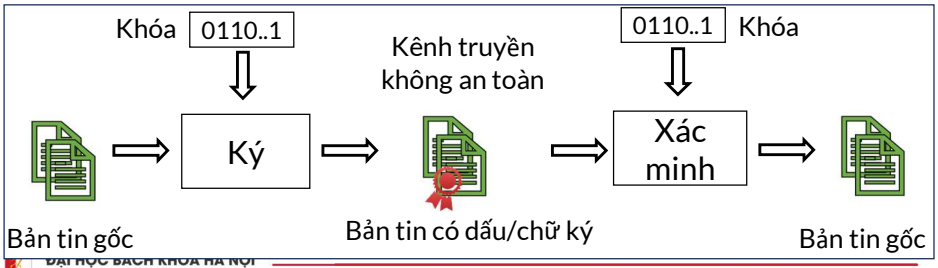
Mục tiêu của mật mã: Tính bí mật

- Tính bí mật(Confidentiality): Đối phương không đọc được bản tin
- Giải pháp 2: Sử dụng mật mã học
  - Alice mã hóa bản tin gốc (plaintext) với khóa mã hóa
  - Alice gửi bản tin mã (ciphertext) qua kênh truyền không bí mật
  - Eve có thể đánh cắp bản tin mã nhưng không thể đọc được nếu không có khóa để giải mã
  - Bob giải mã bằng khóa giải mã



Mục tiêu của mật mã: Tính toàn vẹn và xác thực

- Tính toàn vẹn(Integrity): Đối phương không thay đổi được bản tin mà không bị phát hiện
- Tính xác thực(Authenticity): Đối phương không giả mạo được hai bên mà không bị phát hiện
- Cách thực hiện
  - Alice tạo dấu/chữ ký xác thực cho bản tin với khóa ký
  - Alice gửi bản tin có dấu/chữ ký qua kênh truyền không an toàn
  - Mallory có thể sửa đổi/giả mạo bản tin nhưng không thể tạo dấu/chữ ký hợp lệ
  - Bob kiểm tra dấu/chữ ký bằng khóa xác minh

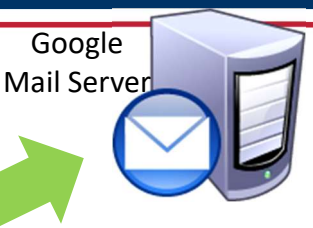


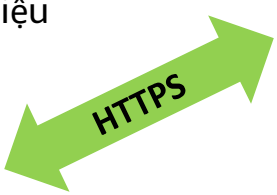
Ứng dụng của mật mã

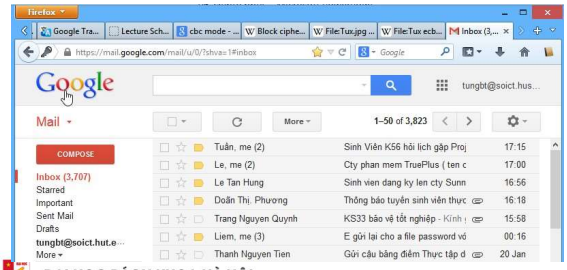
- Giữ bí mật cho thông tin,
- …và không chỉ vậy…
- Chữ ký số(Digital Signature)
- Liên lạc ẩn danh (Anonymous Communication)
- Tiền ẩn danh (Anonymous digital cash)
- Bầu cử điện tử (E-voting)

Ứng dụng của mật mã – Ví dụ

- Bước 1: Trao đổi khóa
- Bước 2: Mã hóa dữ liệu







Web Browser

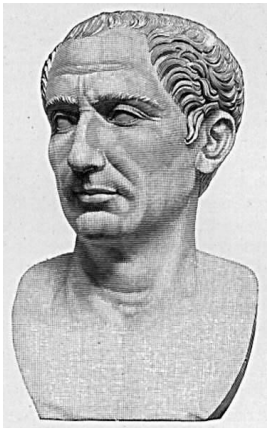
Ứng dụng của mật mã – Ví dụ

Thiết bị lưu trữ

Alice "hôm nay" truyền tin bí mật cho Alice "ngày mai"

Một ví dụ - Mật mã Caesar

- Julius Caesar đưa ra vào thế kỷ thứ 1 trước CN, sử dụng trong quân sự
- Ý tưởng: thay thế một ký tự (bản rõ) trong bảng chữ cái bằng ký tự (bản mật) đứng sau nó 3 (khóa) vị trí.
  - Sử dụng bảng chữ cái vòng
  - A → D, B → E, C → F,..., X → A, Y → B, Z → C
- Mô hình hóa bằng toán học(Mã dịch vòng)
  - Khóa $1 \leq k \leq 25$
  - Mã hóa: $c = (m + k) \bmod 26$
  - Giải mã: $m = (c - k) \bmod 26$
- Dễ dàng bị phá ngay cả khi k thay đổi



Gaius Julius Caesar

Mật mã Caesar – Ví dụ

- Bảng thay thế

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- Bản tin gốc (Plaintext – Bản rõ): PARIS
- Bản mật (Ciphertext): SDULV
- Bản tin gốc: NEWYORK

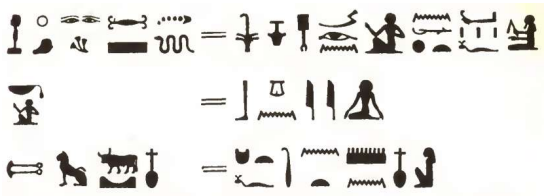
13

1.2. Lược sử mật mã (Đọc thêm)

14

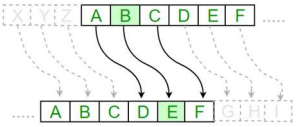
Kỹ thuật mật mã cổ điển (1)

- Các công trình khảo cổ cho thấy kỹ thuật mật mã xuất hiện trong hầu hết các nền văn minh từ rất sớm
  - Người Ai Cập cổ đại sử dụng các chữ tượng hình không theo tiêu chuẩn
  - Người Hy Lạp cổ đại sử dụng gậy scytale tạo ra kỹ thuật mật mã dựa trên hoán vị
  - Hoàng đế Julius Ceasar sử dụng kỹ thuật mật mã dựa trên thay thế ký tự



Mật mã tại khu mộ của pharaoh Amenemhet II
(bên trái là bản mã, bên phải là bản rõ)

Gậy scytale

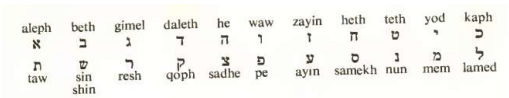


Mật mã Caesar

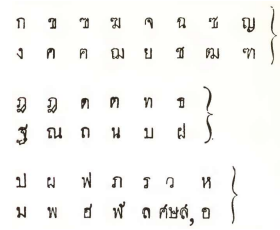
15

Kỹ thuật mật mã cổ điển (2)

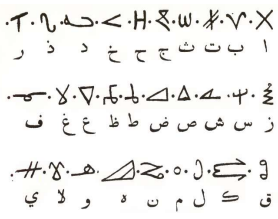
- Các kỹ thuật mã trở nên phức tạp hơn, nhưng chủ yếu vẫn sử dụng kỹ thuật thay thế
  - Quy tắc thay thế ít biến đổi, thường soạn thảo thành từ điển mã (codebook)
  - Sớm bị đánh bại bởi các kỹ thuật thống kê của nhà phá mã (code breaker)



Một từ điển mã tiếng Do Thái



Một từ điển mã tiếng Thái



Một từ điển mã tiếng Ả rập

16

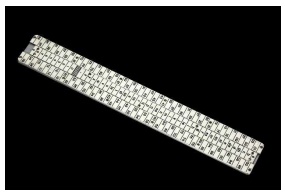
Mật mã Vigenère

- Phát minh bởi Giovan Battista Bellaso vào năm 1553
- Blaise de Vigenère đưa ra cải tiến quan trọng vào năm 1586
- Sử dụng kỹ thuật thay thế trên nhiều bảng chữ cái thay vì sử dụng đơn bảng chữ cái như từ điển mã
  - Trên cùng một văn bản, một mẫu ký tự gốc xuất hiện ở vị trí khác nhau có thể thay thế bằng những mẫu ký tự mã khác nhau.

  →tạo ra sự hỗn loạn về thống kê
  - Bền vững trong khoảng 300 năm trước khi bị phá

Đĩa mật mã sử dụng trong nội chiến Hoa Kỳ

Thước mật mã sử dụng trong nội chiến Hoa Kỳ

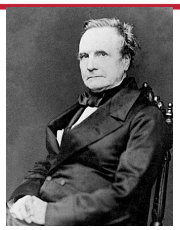


Thước mật mã sử dụng trong quân đội Thụy Sĩ (1914-1940)

17

Sự dịch chuyển thành khoa học mật mã (cryptology)

- Các kỹ thuật mật mã được sử dụng âm thầm, nhưng vẫn chưa phát triển một cách có hệ thống cho đến thế kỷ 19
- Charles Babbage công bố phương pháp phá mã Vigenère dựa trên phân tích thống kê vào năm 1846
- Friedrich Kasiski hoàn thiện phương pháp của C. Babbage và trình bày như một công trình đầy đủ vào năm 1863
  - "he had wrought a revolution in cryptology" – David Kahn, *The Codebreakers (1967)*
- Cryptology = Cryptography + Cryptanalysis



Charles Babbage (1791 – 1871)

Friedrich Kasiski (1805 – 1881)

18

Nguyên lý Kerckhoffs

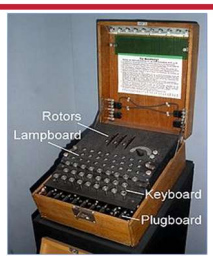
- Giới thiệu 6 nguyên lý thiết kế hệ mật mã vào năm 1883:
  1. Tính thực tế
  2. Thiết kế mở: Không giữ bí mật thiết kế
  3. Khóa có thể dễ dàng lưu giữ và thay đổi
  4. Có thể sử dụng cho điện tín
  5. Thiết bị gọn nhẹ
  6. Dễ hiểu và dễ cài đặt
- Trở thành tôn chỉ cho cả mật mã hiện đại
- Nguyên lý thứ 2 trở thành tiêu chuẩn để đánh giá cho hệ thống an toàn an ninh thông tin bất kỳ

Auguste Kerckhoffs (1835 – 1903)

Sự ra đời của máy mật mã

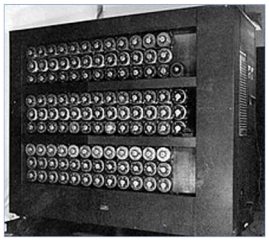
- Các kỹ thuật mật mã cổ điển tiếp tục được sử dụng trong suốt CTTG lần 1
  - Tuy nhiên, công cụ được sử dụng chủ yếu vẫn là giấy, bút, từ điển mã (codebook) và một số công cụ cơ khí đơn giản
  - Các phương pháp thám mã dựa trên phân tích thống kê vẫn hiệu quả
- Máy mật mã (rotor machine) được chế tạo và sử dụng trong CTTG lần 2:
  - Mã hóa và giải mã trở nên đơn giản
  - Các phương pháp thám mã dựa trên phân tích thống kê dần kém hiệu quả hơn
  - Tuy nhiên, vẫn có những thành công quan trọng.
    - Ví dụ: Turing đã thành công trong việc phá mã Enigma

Kỹ thuật mật mã cổ điển trong CTTG



Máy mật mã Enigma

Nhà thám mã đầu tiên của Hoa Kỳ Elizebeth Friedman (1892 – 1980)



Máy phá mã Enigma

Alan Turing (1912 – 1954)

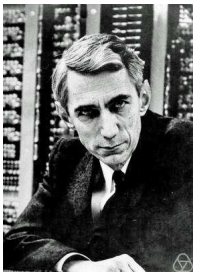
Sự dịch chuyển sang mật mã hiện đại

- Năm 1917, Gilbert Vernam phát minh phương pháp mật mã dựa trên mật mã Vigenère
- Năm 1919, Vernam đăng ký bằng sáng chế cho thiết kế sử dụng rơ le cơ điện
  - Sơ đồ mạch tương ứng với toán tử XOR
  - Đáp ứng nguyên lý của Kerckhoffs
  - "perhaps one of the most important in the history of cryptography", *NSA(Cơ quan an ninh quốc gia Hoa Kỳ)*
- Trên lý thuyết, hệ mật mã Vernam là hoàn hảo (không thể bị phá vỡ trong mọi điều kiện) nếu khóa hoàn toàn ngẫu nhiên và không lặp lại
  - Rất khó đáp ứng yêu cầu về khóa
  - Ngược lại, nếu khóa bị lặp thì rất dễ bị phá. Ví dụ: dự án VENONA của Hoa Kỹ đã giải mã thành công hơn 3000 bức điện tín của tình báo Liên Xô từ 1943-1980

Gilbert Vernam (1890 – 1960)

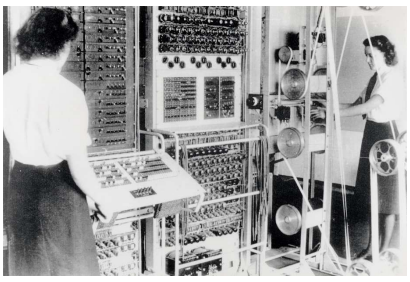
Lý thuyết Shannon

- Claude E. Shannon: "Cha đẻ của lý thuyết thông tin"
- Năm 1945, công bố "A Mathematical Theory of Cryptography"
- Năm 1948 công bố "A Mathematical Theory of Communication" đặt nền móng cho lý thuyết thông tin
- Năm 1949 công bố "Communication Theory of Secrecy Systems" đặt nền móng cho mật mã học hiện đại
- Đưa ra khái niệm về "hệ mật mã hoàn hảo" (perfect secrecy) và cách chứng minh



Claude E. Shannon (1916 – 2001)

Máy tính trở thành công cụ mật mã



Máy tính Colossus (1943) được quân đội Anh sử dụng để phá mã SZ42 của Đức quốc xã

Hệ thống máy mật mã điện tử KW-26 (1957) được Hoa Kỳ đưa vào sử dụng

- Sự ra đời và tiến hóa của máy tính ảnh hưởng tới sự phát triển của tất cả các lĩnh vực, trong đó có an toàn an ninh thông tin

Mật mã DES

- Đến thập niên 1970s, máy tính điện tử đã xuất hiện và dần hoàn thiện
- Chính phủ Hoa Kỳ chưa có một phương pháp mật mã thống nhất để sử dụng → thúc đẩy nhu cầu thiết kế một tiêu chuẩn mật mã có thể cài đặt trên máy tính điện tử
- Hệ mật mã DES (Data Encryption Standard) được thiết kế bởi IBM vào năm 1975 và chuẩn hóa vào năm 1977
  - Kích thước khóa: 56 bit
- Ngay sau khi ra đời, rất nhiều tranh cãi về sự an toàn của DES
  - Khóa 56 bit được cho là có nguy cơ bị vét cạn (brute-force attack)
  - Các cải tiến được thiết kế: 2DES, 3DES
- Hết hạn vào năm 2005 do không còn an toàn để sử dụng

Mật mã AES

- DES và các biến thể ngày càng trở nên kém an toàn do máy tính điện tử ngày một "nhanh" hơn
  - Nguy cơ tấn công vét cạn khóa ngày càng rõ
- Năm 1981, công nghệ máy tính lượng tử được đề xuất bởi Richard Feynman kéo theo nhu cầu về một hệ mật mã an toàn có thể sử dụng trong tương lai xa hơn
- Chính phủ Hoa Kỳ giao cho NIST khởi động dự án thiết kế hệ mật mã tiên tiến từ năm 1997
- Hệ mật mã Rijmen được lựa chọn từ 15 thiết kế
  - Thiết kế và ciới thiệu bởi V. Rijmen và cộng sự vào năm 1998
  - Chuẩn hóa thành AES (Advanced Encryption Standard) vào năm 2002
  - Kích thước khóa: 128, 192, 256 bit
  - Là phương pháp mật mã an toàn nhất cho mục đích dân sự. Khóa có kích thước 256 bit chống được tấn công vét cạn trên máy tính lượng tử

Sự ra đời của mật mã khóa công khai (KCK)

- Trong thập niên 1970s, mạng máy tính đã bắt đầu hình thành
- Bài toán chia sẻ khóa bí mật của hệ mật mã KĐX trở nên cấp thiết do nhu cầu trao đổi dữ liệu trên mạng máy tính tăng cao
- Năm 1976, Whitfield Diffie và Martin Hellman công bố phương pháp trao đổi khóa mới trong công trình "New Directions in Cryptography". Khóa bí mật được tính toán dựa trên khóa công khai(public key) và khóa riêng(private key)
  - Trở thành phương pháp chủ đạo để trao đổi khóa
- Mật mã KCK: sử dụng khóa mã hóa và khóa giải mã khác nhau
  - Khóa công khai(Public key): Công bố cho mọi người biết
  - Khóa riêng(Private key): Chỉ người sở hữu biết

Whitfield Diffie

Martin Hellman

1.3. Một số nguyên lý chung

ONE LOVE. ONE FUTURE.

Nguyên lý Kerckhoffs

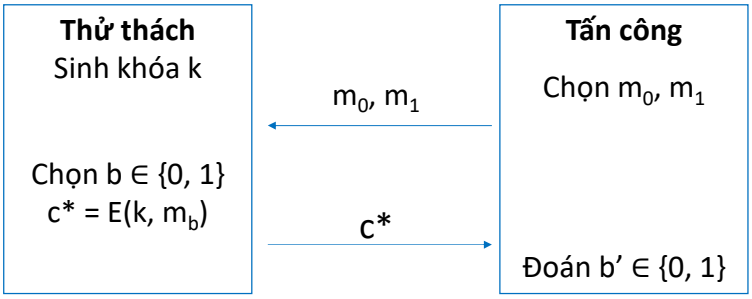
- Hệ mật mã gồm {KeyGen(), E(), D()}
- Làm cách nào để ngăn cản kẻ khác giải mã?
- Nguyên lý Kerckhoffs: "Một hệ mật mã cần an toàn ngay cả đối phương biết mọi thông tin về hệ, trừ khóa bí mật"
- Tại sao?

Hệ mật hoàn hảo

- **Định nghĩa:** Hệ mật là hoàn hảo khi và chỉ khi ∀m và ∀c mà Pr(C = c) > 0: Pr(M = m | C = c) = Pr(M = m)
- **Bổ đề:** ∀ cặp m_0, m_1 có độ dài như nhau, ∀c
  $$Pr(C = c \mid M = m_0) = Pr(C = c \mid M = m_1)$$
- Bản mật hoàn toàn không chứa thông tin về bản rõ
- **Định lý:** Một hệ mật mã là hoàn hảo thì $||K|| \geq ||M||$

Hệ mật hoàn hảo

- Thử thách tấn công biết trước bản rõ(Known plaintext attack)



- Kẻ tấn công thắng nếu đoán đúng b' = b
- Hệ mật là hoàn hảo nếu với mọi thuật toán, xác suất kẻ tấn công đoán đúng là P = ½ → không thể phân biệt được bản rõ nào đã được mã hóa

Lý thuyết Shannon

- Định lý: Một hệ mật có ||M|| = ||K|| = ||C|| là hoàn hảo khi và chỉ khi:
  1. Xác suất xuất hiện của mọi giá trị khóa k là như nhau
  2. Tồn tại duy nhất giá trị khóa k sao cho

     $c = E(k, m) \quad \forall m, \forall c$
- Có thể chứng minh được rằng định lý trên đưa ra 2 yêu cầu cần cho một hệ mật hoàn hảo:
  - Kích thước khóa k bằng kích thước bản tin m
  - Khóa k chỉ được dùng 1 lần

An toàn theo tính toán

- Hệ mật hoàn hảo: Không có bất cứ thông tin về bản rõ (plaintext) nào bị lộ ngay cả khi kẻ tấn công có vô hạn tài nguyên tính toán.
- Chi phí sử dụng hệ mật hoàn hảo là quá lớn hoặc không khả thi.
- Thực tế, chỉ cần hệ mật mã yếu hơn, nhưng đủ mạnh để thỏa mãn đồng thời 2 điều kiện:
  - Chống lại được các phương pháp tấn công trong khoảng thời gian nào đó
  - Kẻ tấn công chỉ có thể thành công với xác suất không đáng kể

→ Hệ mật an toàn theo tính toán

An toàn theo tính toán

- Định nghĩa 1: Hệ mật được gọi là an toàn theo tính toán với độ an toàn (t, ε) nếu kẻ tấn công thực hiện phá mã trong thời gian tối đa là t thì chỉ đạt được xác suất thành công tối đa là ε
- Ví dụ: Khóa có kích thước n, kẻ tấn công cần phải giải mã thử với 2^n giá trị khóa (Tấn công vét cạn). Giả sử rằng mỗi lần thử mất 1 chu kỳ CPU. Nếu t = 100 năm, $\varepsilon = 2^{-60}$
  - CPU = 16 GHz → n = ?
  - CPU = 10^6 x 16 GHz → n = ?
- Tuy nhiên, ở góc độ lý thuyết, định nghĩa này không dùng cho chứng minh độ an toàn.

An toàn theo tính toán

- Định nghĩa 2: Một hệ mật được gọi là an toàn theo tính toán nếu với mọi thuật toán tấn công hiệu quả (độ phức tạp tính toán đa thức) thì xác suất thành công là ε không đáng kể
  - Thời gian tấn công: t = poly(n)
  - Xác suất tấn công: ε = f(n) sao cho ε nhỏ tùy ý ∀n ≥ N.
  - Thực tế, xác suất không đáng kể: $\varepsilon \leq 2^{-80}$
  - Xác suất đáng kể: $\varepsilon \geq 2^{-30}$

Lý thuyết Shannon (tiếp)

- Độ dư thừa của ngôn ngữ: Sự xuất hiện của n ký tự cho phép đoán nhận đúng ký tự xuất hiện tiếp theo với xác xuất *p* nào đó.
- Đối với thám mã: sử dụng phương pháp vét cạn, cần phải thu được tối thiểu *u* ký tự mật mã để tìm được chính xác khóa.

u: khoảng cách unicity (unicity distance)

→ *u* càng lớn độ an toàn của hệ càng cao

Lý thuyết Shannon (tiếp)

- Tính toán khoảng cách unicity

$$u = \frac{l_K H(k)}{H(c) - H(m)}$$

l_K: Kích thước khóa

$H(k), H(m), H(c)$: entropy của ký tự. Ví dụ

$H(m) = -\sum p(m_i) \times log_2(p(m_i))$: entropy của ký tự bản rõ

$p(m_i)$: xác suất xuất hiện của ký tự trong không gian bản rõ

- Nếu khóa và bản mật xuất hiện hoàn toàn ngẫu nhiên, và chung bảng chữ cái:

$$u = \frac{l_K log_2(N)}{log_2(N) - H(m)}$$

N: số ký tự của bảng chữ cái

- Làm thế nào để tăng độ an toàn khi sử dụng mật mã?

Thông tin tham khảo – Kích thước khóa

- Khóa có kích thước bao nhiêu?
  - Mật mã được coi là an toàn khi phương pháp vét cạn (brute-force) là cách nhanh nhất để bẻ khóa
  - Mục tiêu: giảm thiểu nguy cơ bị tấn công vét cạn (đạt độ an toàn theo tính toán)
- Bạn nghe ở đâu đó, "dễ dàng" bẻ khóa mật mã DES có kích thước khóa 56 bit?
  - Năm 1998, hệ thống phá mã EFF DES (trị giá 250K$) bẻ khóa DES trong khoảng 1 ngày
  - Năm 2006, hệ thống phá mã COPACOBANA (trị giá 10K$) bẻ khóa DES trong 6,4 ngày

Sử dụng định luật Moore để tính thời gian bẻ khóa trong năm 2020 với chi phí 10K$?

Thông tin tham khảo – Kích thước khóa

- Chi phí để bẻ khóa DES (năm 2006)
  - 56 bit: $10.000
  - 87 bit: $100.000.000.000 (thời gian bẻ khóa không đổi)
- Cần giữ thông tin mật trong bao lâu khi hệ thống phá mã là COPACOBANA? (năm 2006)
  - 56 bit: 6.4 ngày
  - 128 bit: ?
- Tham khảo kích thước khóa nên sử dụng trong tương lai tại địa chỉ
  http://csrc.nist.gov/groups/ST/toolkit/key_management.html

Thông tin tham khảo – Kích thước khóa

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1** SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256 SHA-384 SHA-512

http://www.keylength.com

2. Hệ mật mã khóa đối xứng

ONE LOVE. ONE FUTURE.

41

41

Hệ mật mã khóa đối xứng

- Symmetric cryptography, Secret-key cryptography: sử dụng cùng một khóa khi mã hóa và giải mã.
- Được phát triển từ rất sớm
- Thuật toán mã hóa: phối hợp các toán tử
  - Thay thế
  - Đổi chỗ (hoán vị)
  - XOR
- Tốc độ thực hiện các thuật toán nhanh, có thể thực hiện bằng dễ dàng bằng phần cứng
- Một số hệ mật mã khóa đối xứng hiện đại: DES, 2DES, 3DES, AES, RC4, RC5

42

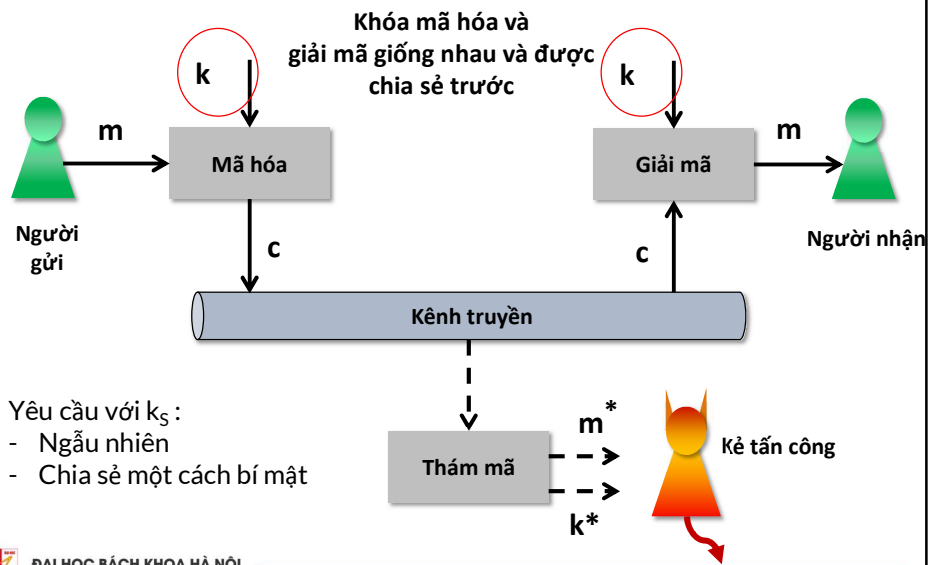
2.1. Sơ đồ nguyên lý

Hệ mật mã gồm:

- Bản rõ (plaintext-m): thông tin không được che dấu
- Bản mật (ciphertext-c): thông tin được che dấu
- Khóa (key- k): giá trị đã được chia sẻ bí mật
- Sinh khóa KeyGen()
  - Là hàm ngẫu nhiên
- Mã hóa (encrypt-E): c = E(k, m)
  - E là hàm ngẫu nhiên
- Giải mã (decrypt): m = D(k, c)
  - D là hàm xác định
- Tính đúng đắn D(k, E(k, m)) = m

43

Sơ đồ chung



44

Yêu cầu của lược đồ mã hóa KĐX

- Tính đúng đắn:

$$\text{Dec}(k, \text{Enc}(k, m)) = m \quad \forall m, k$$

- Tính hiệu quả: thời gian mã hóa/giải mã ngắn
- Tính bí mật
  - Lý thuyết Shannon(nhắc lại): bản mã không cung cấp bất cứ thông tin nào về bản gốc

Thám mã

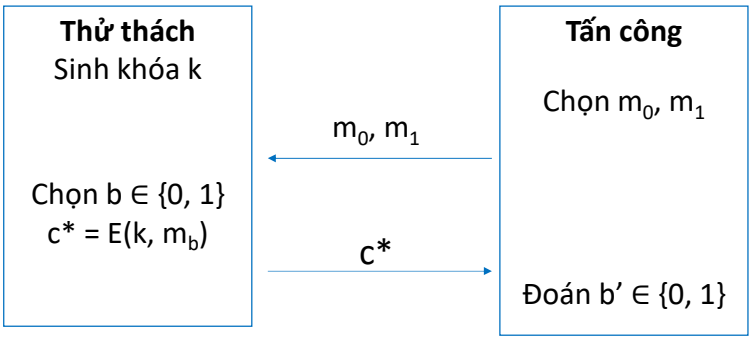
- Nhắc lại nguyên lý Kerckhoffs "Một hệ mật mã cần an toàn ngay cả đối phương biết mọi thông tin về hệ, trừ khóa bí mật"
  - Kẻ thám mã đã biết giải thuật sinh khóa, mã hóa, giải mã nhưng không biết khóa bí mật
- Tấn công chỉ biết bản mật:
  - Kẻ thám mã có các bản mật (COA: ciphertext-only attack)
  - Phương pháp phá mã: thử tất cả các tổ hợp khóa có thể để tìm ra tổ hợp khóa thích hợp. Trong trường hợp không gian khóa lớn thì phương pháp này không thực hiện được.
  - Đối phương cần phải phân tích văn bản mật, thực hiện các kiểm nghiệm thống kê để giảm số lượng trường hợp cần thử.

Tấn công biết trước bản rõ

- Known-plaintext attack - KPA
- Kẻ tấn công đã có các cặp bản tin (m_i, c_i) mã hóa với cùng khóa k ***(kẻ tấn công không biết k)***
- Mục đích: xác định được thông tin bí mật trong các bản mã $c \neq c_i$ được tạo ra khi sử dụng cùng khóa k ở trên
- Phương thức tấn công:
  - Vét cạn
  - Phân tích để đoán giá trị khóa

Thám mã (tiếp)

- Kiểm chứng tính an toàn trước tấn công KPA



- Hệ mật chống lại được tấn công KPA (độ an toàn IND-KPA) nếu với mọi thuật toán tấn công hiệu quả thì $P(b' = b) \leq \frac{1}{2} + \varepsilon$

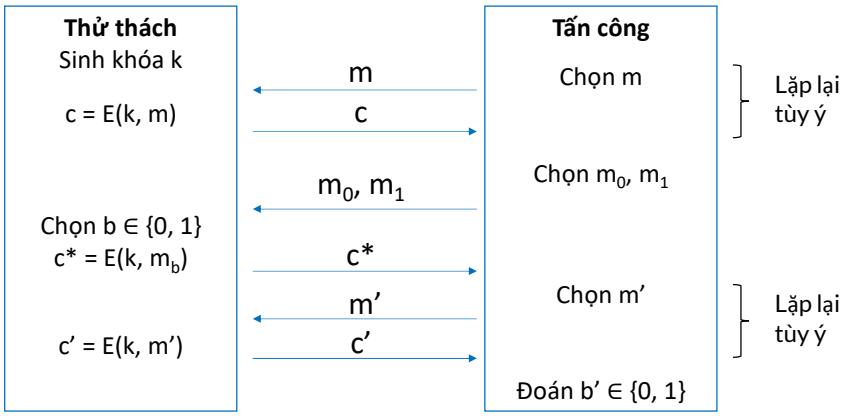
ε: không đáng kể (Hiện tại: $\leq 2^{-80}$)

Tấn công chọn trước bản rõ

- Chosen-plaintext attack - CPA
- Kẻ tấn công có quyền truy cập không hạn chế vào thành phần mã hóa, ***nhưng hắn không biết giá trị khóa k***
- Kẻ tấn công lựa chọn một số bản rõ (plaintext) theo ý muốn để mã hóa → nhận được các bản mã tương ứng
- Dựa vào các bản mã nhận được thì kẻ tấn công đoán nhận bản tin gốc mà các bên truyền đi / hoặc đoán giá trị khóa

Thám mã (tiếp)

- Kiểm chứng tính an toàn trước tấn công CPA



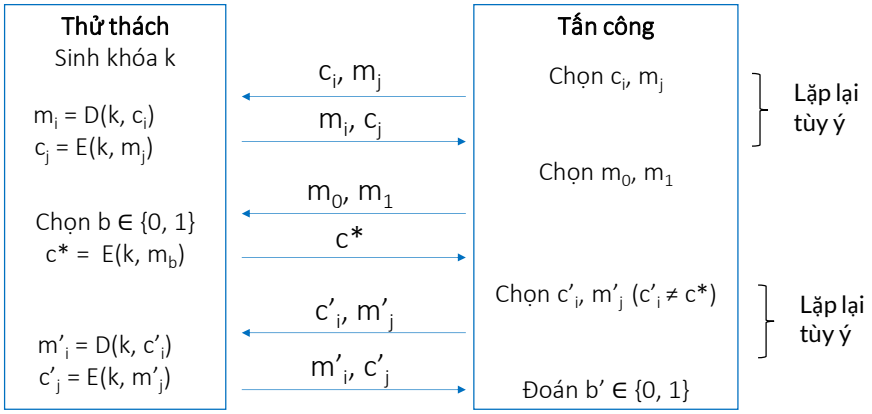
- Hệ mật chống lại được tấn công CPA (độ an toàn IND-CPA) nếu với mọi thuật toán tấn công hiệu quả thì $P(b' = b) \leq \frac{1}{2} + \varepsilon$

Tấn công chọn trước bản mật

- Chosen-ciphertext attack - CCA
- Tương tự tấn công tấn công CPA, nhưng kẻ tấn công có nhiều quyền hơn
- Kẻ tấn có thêm quyền truy cập tùy ý vào thành phần giải mã
- Kẻ tấn công có thể lựa chọn không giới hạn bản mã c và nhận được bản rõ tương ứng

Thám mã (tiếp)

- Kiểm chứng tính an toàn trước tấn công CCA



- Hệ mật chống lại được tấn công CCA (độ an toàn IND-CCA) nếu với mọi thuật toán tấn công hiệu quả thì $P(b' = b) \leq \frac{1}{2} + \varepsilon$

Tổng kết - Các phương pháp thám mã

- COA < KPA < CPA < CCA

53

2.2. Mật mã cổ điển

54

Mật mã thay thế(Substitution cipher)

- Một/một mẫu ký tự được thay thế bằng một/một mẫu ký tự khác.
- Mật mã Ceasar
- Mật mã dịch vòng (Shift Cipher): mã từng ký tự
  - Mật mã đơn bảng thế
  - Khóa: $1 \le k \le 25$
  - Mã hóa: $c = (m + k) \bmod 26$
  - Giải mã: $m = (c - k) \bmod 26$

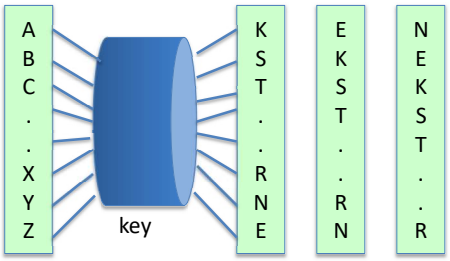
Mật mã thay thế(Substitution cipher)

- Mật mã Vigener: mã 1 khối ký tự
  - Mật mã đa bảng thế: sử dụng nhiều bảng thay thế

k	=	C	R	Y	P	T	O	C	R	Y	P	T	O	C	R	Y	P	T
																		(+ mod 26)
m	=	W	H	A	T	A	N	I	C	E	D	A	Y	T	O	D	A	Y
c	=	Z	Z	Z	J	U	C	L	U	D	T	U	N	W	G	C	Q	S

Mật mã thay thế(Substitution cipher)

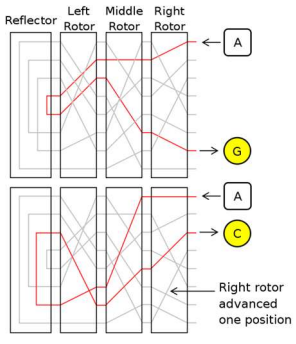
- Máy rotor (Rotor machine)



Hebern machine

Mật mã thay thế(Substitution cipher)

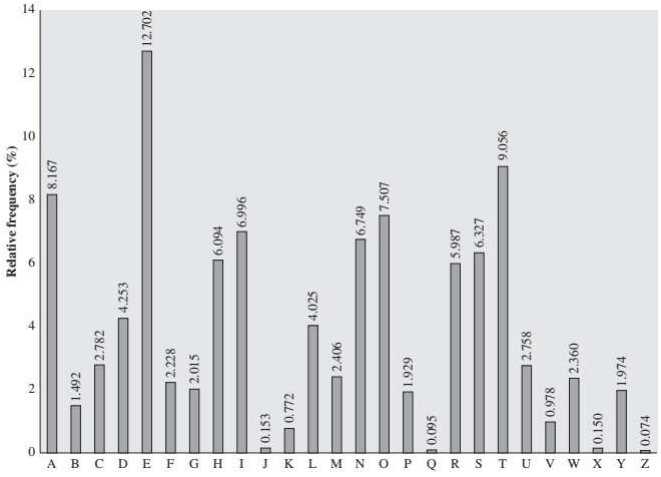
- Máy rotor (Rotor machine)



Enigma

Phá mã hệ mật mã thay thế(Đọc thêm)

- Chỉ có bản mã: Dựa trên phương pháp thống kê
- Ví dụ: tiếng Anh



Thuộc tính thống kê của tiếng Anh

- Phân nhóm ký tự theo tần suất

I	e
II	t,a,o,i,n,s,h,r
III	d,l
IV	c,u,m,w,f,g,y,p,b
V	v,k,j,x,q,z

- Một vài mẫu ký tự có tần suất xuất hiện cao
  - Bigrams: th, he, in, an, re, ed, on, es, st, en at, to
  - Trigrams: the, ing, and, hex, ent, tha, nth, was, eth, for, dth

Ví dụ: Phá mã dịch vòng

```
YKHLBA JCZ SVIJ JZB TZVHI JCZ VHJ DR IZXKHLBA VSS
RDHEI DR YVJV LBXSKYLBA YLALJVS IFZZXC CVI
LEFHDNZY EVBLRDSY JCZ FHLEVHT HZVIDB RDH JCLI CVI
WZZB JCZ VYNZBJ DR ELXHDZSZXJHDBLXI JCZ XDEFSZQLJT
DR JCZ RKBXJLDBI JCVJ XVB BDP WZ FZHRDHEZY WT JCZ
EVXCLBZ CVI HLIZB YHVEVJLXVSST VI V HXXIKSJ DR
JCLI HZXZBJ YZNZXDFEZBJ LB JZXCBDSDAT EVBT DR JCZ
XLFCZH ITIJZEIJCVJ PZHZ DBXZ XDBILYXHZYIZKHZ
VHZBDP WHZVMVWSZ
```


61

Ví dụ: Phá mã dịch vòng

Ký tự:	*A*	*B*	*C*	*D*	*E*	*F*	*G*
Tần suất:	*5*	*24*	*19*	*23*	*12*	*7*	*0*
Ký tự:	*H*	*I*	*J*	*K*	*L*	*M*	*N*
Tần suất:	*24*	*21*	*29*	*6*	*21*	*1*	*3*
Ký tự:	*O*	*P*	*Q*	*R*	*S*	*T*	*U*
Tần suất:	*0*	*3*	*1*	*11*	*14*	*8*	*0*
Ký tự:	*V*	*W*	*X*	*Y*	*Z*		
Tần suất:	*27*	*5*	*17*	*12*	*45*		

Z → e
f_J=29, f_V = 27
f_{JCZ} = 8 → ' the'
⇒J → t, C → h
V đứng riêng: V → a

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}
t a h
JZB → te? {teo, tei, ten, tes, ter}: B → n

62

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHLnA the SaIt ten TeaHI the aHt DR IeXKHLnA aSS
RDHEI DR Yata LnXSKYLnA YLALtaS IFeeXh haI
LEFHDNeY EanLRDSY the FHLEaHT HeaIDn RDH thLI haI
Ween the aYNent DR ELXHDeSeXtHDnLXI the XDEFSeQLtT
DR the RKnXtLDnI that Xan nDP We FeHRDHEeY WT the
EaXhLne haI HLIen YHaEatLXaSST aI a HXXIKSt DR
thLI HeXent YeNeXDFEent Ln teXhnDSDAT EanT DR the
XLFheH ITIteEIthat PeHe DnXe XDnILYXHeYIeKHe
aHenDP WHeaMaWSe
```

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}

t a n h

aI → a? {ao, ai, as, ar}: I → s

63

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHLnA the Sast ten TeaHs the aHt DR seXKHLnA aSS
RDHEs DR Yata LnXSKYLnA YLALtaS sFeeXh has
LEFHDNeY EanLRDSY the FHLEaHT HeasDn RDH thLs has
Ween the aYNent DR ELXHDeSeXtHDnLXs the XDEFSeQLtT
DR the RKnXtLDns that Xan nDP We FeHRDHEeY WT the
EaXhLne has HLsen YHaEatLXaSST as a HXXsKSt DR
thLs HeXent YeNeXDFEent Ln teXhnDSDAT EanT DR the
XLFheH sTsteEsthat PeHe DnXe XDnsLYXHeYseKHe
aHenDP WHeaMaWSe
```

Nhóm: {J, V, B, H, D, I, L, C} → {t, a, o, i, n, s, h, r}

t a n s h

Rút gọn: {H, D, L} → {o, i, r}

thLs = th?s {thos, this, thrs}: L → i

64

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKHinA the Sast ten TeaHs the aHt DR seXKHinA aSS
RDHEs DR Yata inXSKYinA YiAitaS sFeeXh has
iEFHDNeY EaniRDSY the FHiEaHT HeasDn RDH this has
Ween the aYNent DR EiXHDeSeXtHDniXs the XDEFSeQitT
DR the RKnXtiDns that Xan nDP We FeHRDHEeY WT the
EaXhine has Hisen YHaEatiXaSST as a HXXsKSt DR
this HeXent YeNeXDFEent in teXhnDSDAT EanT DR the
XiFheH sTsteEsthat PeHe DnXe XDnsiYXHeYseKHe
aHenDP WHeaMaWSe
```

Nhóm: {H, D} → {o, r}

aHt = a?t {aot, art}: H → r, D → o

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKrinA the Sast ten Tears the art oR seXKrinA aSS
RorEs oR Yata inXSKYinA YiAitaS sFeeXh has
iEFroNeY EaniRoSY the FriEarT reason Ror this has
Ween the aYNent oR EiXroeSeXtroniXs the XoEFSeQitT
oR the RKnXtions that Xan noP We FerRorEeY WT the
EaXhine has risen YraEatiXaSST as a rXXsKSt oR
this reXent YeNeXoFEent in teXhnoSoAT EanT oR the
XiFher sTsteEsthat Pere onXe XonsiYXreYseKre
arenoP WreaMaWSe
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	n	h	o				r	s	t		i										a				e

reason Ror this has Ween→ reason for this has been

this reXent → this recent

R → f, W → b, X → c

Ví dụ: Phá mã dịch vòng (tiếp)

```
YKrinA the Sast ten Tears the art of secKrinA aSS
forEs of Yata incSKYinA YiAitaS sFeech has
iEFroNeY EanifoSY the FriEarT reason for this has
been the aYNent of EicroeSectronics the coEFSeQitT
of the fKnctions that can noP be FerforEeY bT the
Eachine has risen YraEaticaSST as a rccsKSt of
this recent YeNecoFEent in technoSoAT EanT of the
ciFher sTsteEsthat Pere once consiYcreYseKre
arenoP breaMabSe
```

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	n	h	o				r	s	t		i						f				a	b	c		e

of the fKnctions → of the functions
of the ciFher → of the cipher
K → u, F → p

67

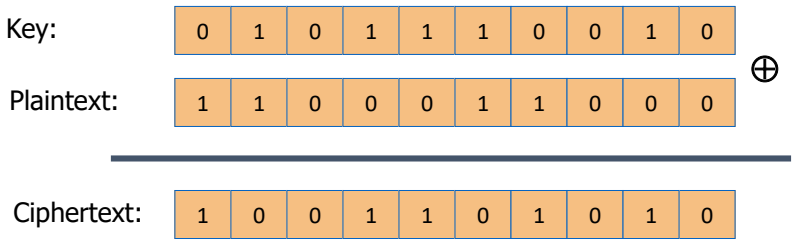
2.3. Mật mã hiện đại

ONE LOVE. ONE FUTURE.

68

Mật mã one-time-pad (OTP)

- Vernam (1917)



- KeyGen(): Sinh khóa ngẫu nhiên
  - Kích thước của khóa bằng kích thước của bản rõ
  - Khóa chỉ dùng để mã hóa cho 1 thông điệp
- Mã hóa: E(k, m) = m ⊕ k
- Giải mã: D(k, c) = c ⊕ k
- Shannon : mật mã OTP là hệ mật hoàn hảo

Mật mã OTP

- Khóa được dùng nhiều hơn 1 lần (two-time-pad): không còn an toàn

$$c_1 \leftarrow m_1 \oplus k$$

$$c_2 \leftarrow m_2 \oplus k$$

Nếu kẻ tấn công có được bản mã:

$$c_1 \oplus c_2 \rightarrow m_1 \oplus m_2$$

Nếu kích thước bản tin đủ dài

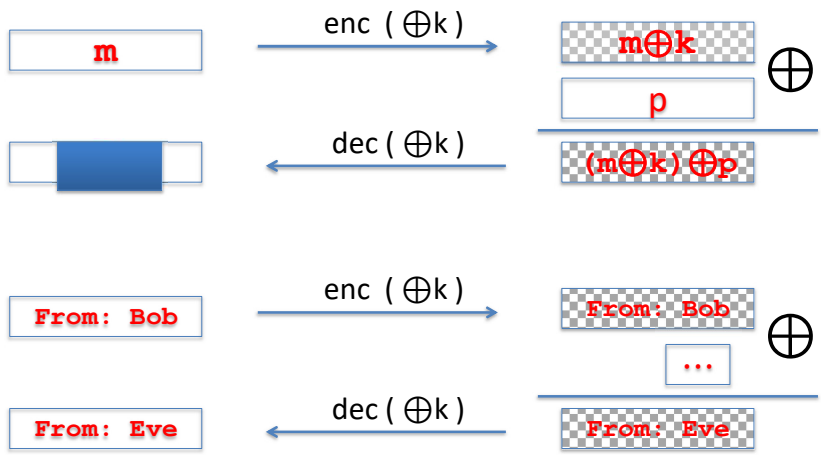
$$m_1 \oplus m_2 \rightarrow m_1, m_2$$

Hệ mật là an toàn trong một số điều kiện, ngược lại trở nên không an toàn

Two-time-pad trong lịch sử: Dự án VENONA

- Tình báo Liên Xô đã sử dụng two-time-pad trong suốt chiến tranh thế giới lần 2
- Dự án VENONA: thực hiện bởi tình báo Hoa Kỳ để bẻ khóa các thông điệp thu được trong thời gian trên:
  - Bao gồm xác định các gián điệp trong dự án Manhattan
  - Dự án kéo dài đến 1980
- Không được giải mật cho đến 1995
  - Bí mật đến mức tổng thống Truman không được thông báo về dự án
  - Xô Viết phát hiện vào 1949, mặc dù hệ thống mật mã của họ đã được sửa chữa từ 1948

Tấn công vào tính toàn vẹn của OTP



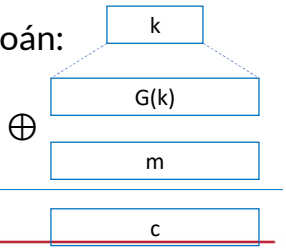
Mật mã dòng (Stream Cipher)

- Xử lý văn bản rõ theo dòng byte, thời gian thực
  - RC4 (900 Mbps), SEAL (2400 Mbps), RC5(450 Mbps)
- Phù hợp với các hệ thống truyền dữ liệu thời gian thực trên môi trường mạng máy tính
- An toàn nếu khóa chỉ dùng 1 lần (one-time-pad)
- Trên thực tế, sử dụng hàm sinh khóa giả ngẫu nhiên (PRG - Pseudo Random Number Generator)

  G: K → $\{0, 1\}^n$ (len(K) << n)

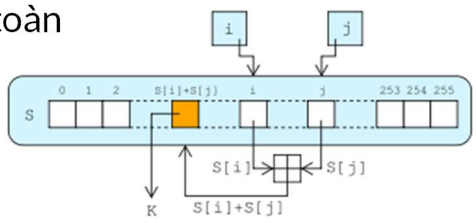
Hàm PRNG phải có tính không thể tiên đoán:

Với mọi thuật toán hiệu quả, nếu đã biết i bit đầu tiên thì xác suất đoán đúng bit thứ i + 1 là ≤ ½ + ε



Mã RC4 (Rivest Cipher 4)

- Rivest Cipher 4: ra đời năm 1987
- Kích thước khóa: 40 hoặc 128 bit
- Hoạt động: gồm 2 thuật toán chính
  - Key-scheduling algorithm (KSA): mở rộng khóa mã hóa thành 1 giá trị S có kích thước 256 byte
  - Pseudo-random generation algorithm (PRGA): lựa chọn 1 byte K từ S để XOR 1 byte thông điệp
- Hiện không còn an toàn



Mã eStream

- Phương pháp mật mã dòng mới nhất được thiết kế để thay thế cho các phương pháp mã dòng cũ
- Hiện đang được phát triển, chưa công bố thành tiêu chuẩn
- Hàm sinh khóa giả ngẫu nhiên:

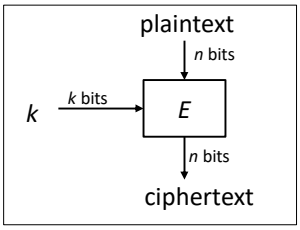
$$PRG: \quad \{0,1\}^s \times R \longrightarrow \{0,1\}^n$$

R: giá trị chỉ dùng 1 lần, không lặp lại

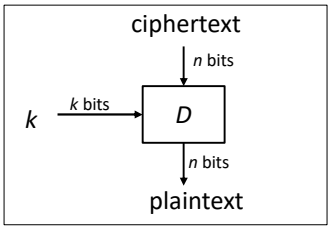
- Mã hóa: $E(k, m ; r) = m \oplus PRG(k ; r)$
- Ví dụ: Salsa20 có s = 128 hoặc 256 bit, R có kích thước 64 bit

Mật mã khối (Block Cipher)

- Xử lý khối dữ liệu có kích thước cố định
- Khóa có kích thước cố định



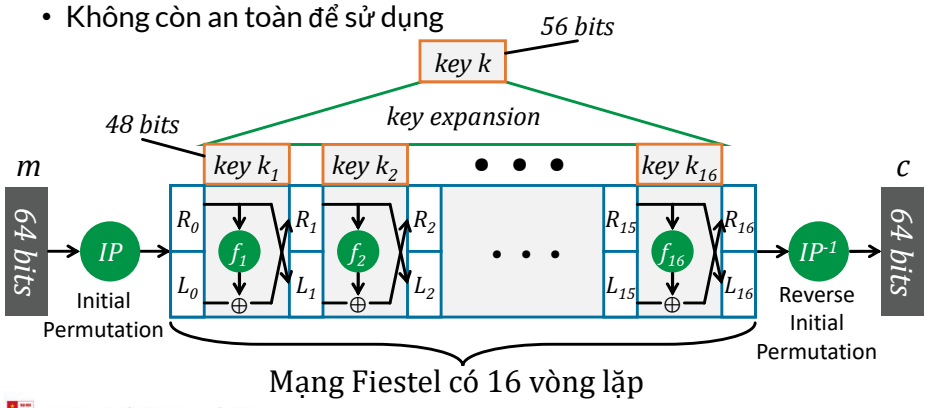
Mã hóa: c = E(k, m)
Đầu vào: khóa k bit, khối dữ liệu rõ n bit
Đầu ra: khối dữ liệu mã n bit



Giải mã: m = D(k, c)
Đầu vào: khóa k bit, khối dữ liệu mã n bit
Đầu ra: khối dữ liệu rõ n bit

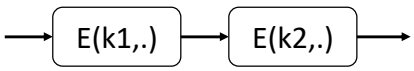
Mật mã DES - Data Encryption Standard

- Kích thước khóa: 56 bit
- Kích thước khối dữ liệu: 64 bit
- Giải mã giống mã hóa nhưng đảo ngược thứ tự dùng khóa
- Không còn an toàn để sử dụng

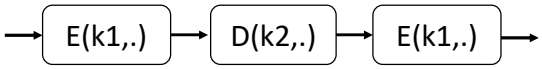


Cải tiến DES

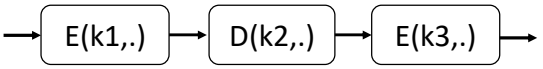
- DES trở nên không an toàn do kích thước khóa ngắn
- 2DES: Sử dụng 2 khóa DES (k1,k2) = 112 bit



  - Tuy nhiên, 2DES không an toàn hơn đáng kể so với DES vì có thể bị tấn công meet-in-the-middle
- 3DES:
  - Sử dụng 2 khóa DES:



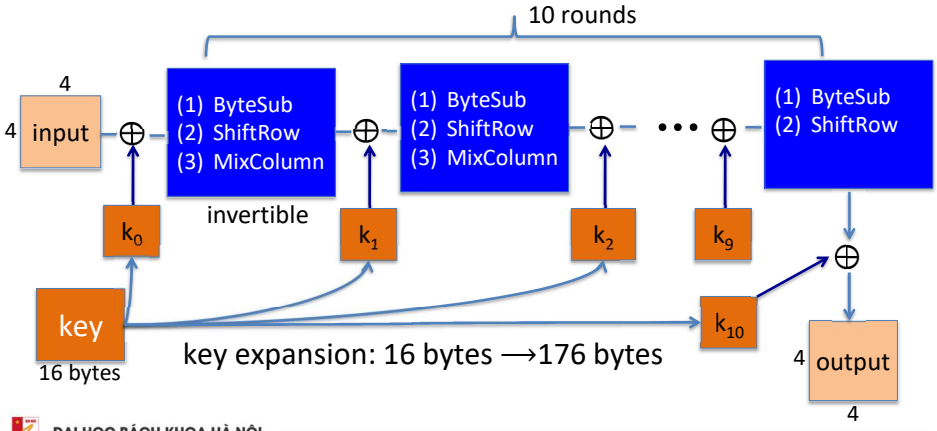
  - Sử dụng 3 khóa DES:



  - Sử dụng 3 khóa không an toàn hơn so với sử dụng 2 khóa

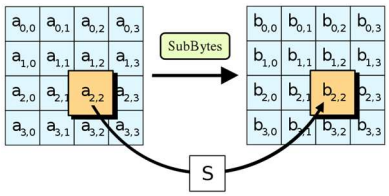
Mật mã AES – Advanced Encryption Standard

- Kích thước khóa: 128, 192, 256 bit
- Kích thước khối: 128 bit
- Số vòng lặp: 10, 12, 14 theo kích thước khóa

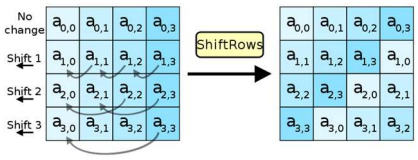


AES – Hàm lặp (Tham khảo)

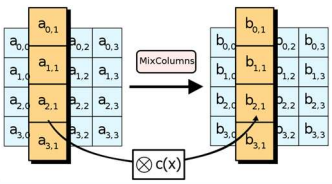
- ByteSub:



- ShiftRows:

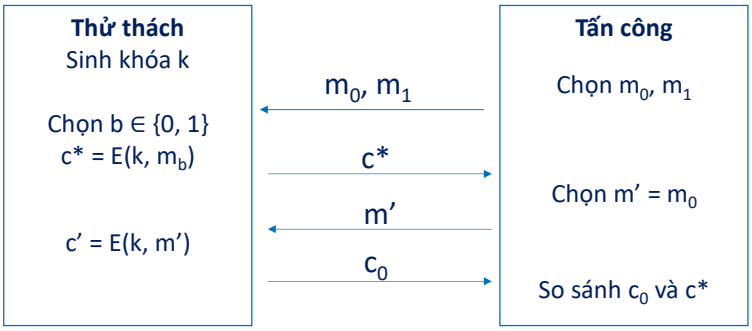


- MixColumns:



Các vấn đề của mã khối

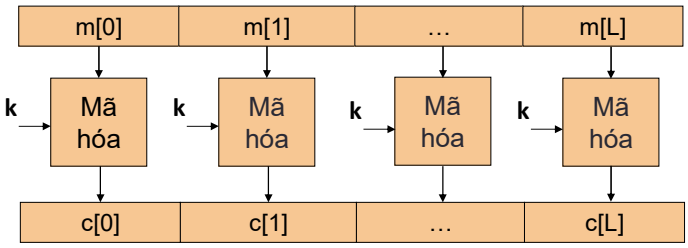
- Mã khối chỉ xử lý khối dữ liệu có kích thước cố định
  - Làm cách nào để mã hóa bản tin có kích thước lớn hơn?
- Mã khối không an toàn trước tấn công CPA



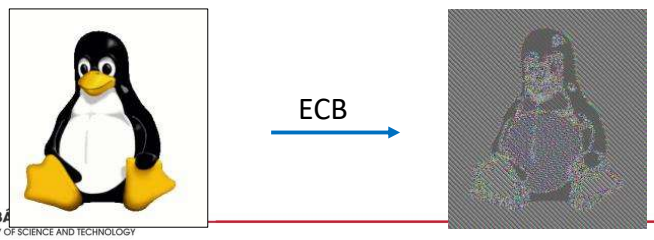
Mã hóa là hàm xác định thì không an toàn trước tấn công CPA

Các chế độ mã khối

- Electronic Code Book (ECB): Mã từ điển

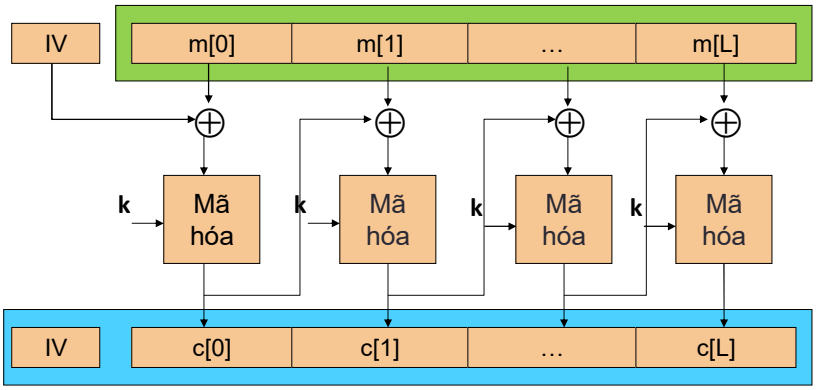


- Hạn chế: ECB không chống lại được tấn công KPA



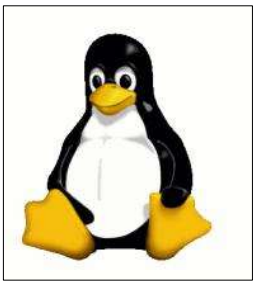
Chế độ CBC - Cipher Block Chaining

- Chế độ mã móc xích

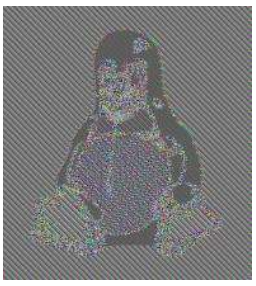


CBC chống lại được tấn công CPA nếu IV (Initial Vector) ngẫu nhiên

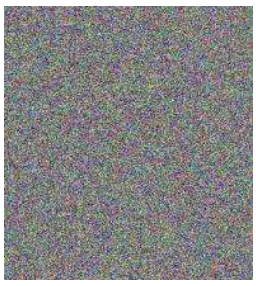
CBC – So sánh với ECB



Ảnh gốc



Mã hóa ở chế độ ECB



Mã hóa ở chế độ CBC

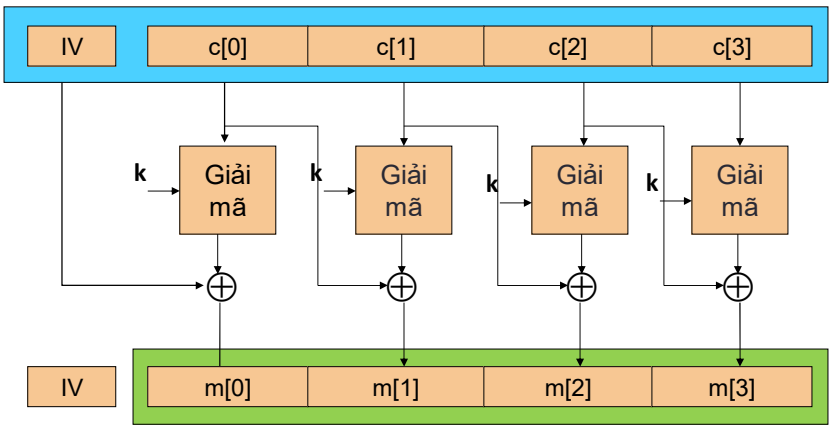
Tấn công CPA khi đoán được IV

- Giả sử kẻ tấn công đoán được giá trị IV*

Thử thách		Tấn công
Sinh khóa k		Sinh m = 0s
	m = 0s ←	
c = [IV, E(k, IV)]	c →	**c = IV || E(k,IV)**
		Sinh m_0 = IV* ⊕ IV
		$m_1 \neq m_0$
	m_0, m_1 ←	
Chọn b ∈ {0, 1}		
b = 0 → c* = [IV*, E(k, IV)]		
b = 1 → c* = [IV*, E(k, m_1 ⊕ IV*)]		
	c* →	**c* = IV* || [...]**
		Nếu c* = c đoán b' = 0
		Ngược lại b' = 1

85

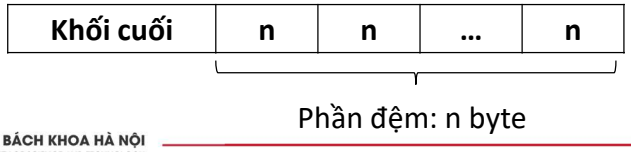
CBC – Giải mã



86

Phần đệm cho ECB và CBC

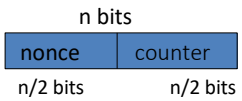
- Khi kích thước bản tin gốc không chia hết cho một khối:
  - r = Len(message) mod Len(block)
  - Phần đệm có kích thước Len(block) – r
- Khi kích thước bản tin gốc chia hết cho 1 khối: thêm phần đệm có kích thước là 1 khối
- Giá trị phần đệm khác nhau với mỗi chuẩn
  - Không dùng chuỗi bit 0 để làm phần đệm
- Chuẩn PKCS#7: Nếu cần đệm n byte thì dùng phần đệm là chuỗi byte có giá trị mỗi byte là n



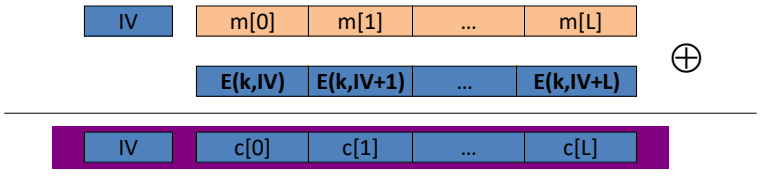
87

Chế độ CTR – Counter Mode

- Initial Vector:
  - nonce: giá trị ngẫu nhiên
  - counter: khởi tạo bằng 0



- Mã hóa: không cần padding



- Nếu IV lặp lại, chế độ CTR không an toàn

88

Độ an toàn của các chế độ mã

- Khóa được dùng nhiều lần → giảm độ an toàn
- Nếu gọi:
  - q: số bản tin được mã hóa cùng với khóa không đổi
  - L: số khối dữ liệu có trong bản tin dài nhất
  - X: Số lượng giá trị có thể của 1 khối dữ liệu
- Chế độ CBC an toàn trước tấn công CPA khi $q^{2*}L^2 << X$
- Chế độ CTR an toàn trước tấn công CPA khi $q^{2*}L << X$
- Để xác suất tấn công là không đáng kể ($\leq 2^{-80}$) thì sau bao nhiêu khối phải đổi khóa?
- Tất cả các chế độ mã đã đề cập không an toàn trước tấn công CCA

Độ an toàn của các chế độ mã

- CBC: $q^{2*}L^2 << X$, X: Số giá trị có thể có của 1 khối
- Kích thước 1 khối là n bit → $X = 2^n$

→ $q^{2*}L^2 << 2^n$

→ $q^{2*}L^2/2^n << 1$

→ $q^{2*}L^2/2^n <= 2^{-80}$

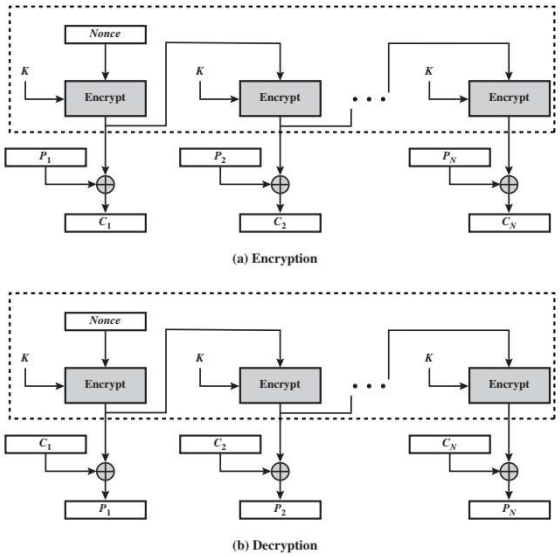
- Ví dụ: mật mã AES có kích thước khối là 128 bit

AES-CBC còn an toàn nếu $q^{2*}L^2/2^{128} <= 2^{-80}$

$q^{2*}L^2 <= 2^{48}$ → $q*L <= 2^{24}$

→Mã tối đa 2^{24} * 16 byte = 2^{28} byte = 256 MB, sau đó cần đổi khóa mới.

Output Feedback - OFB



Tấn công vào mật mã khối

- Tấn công vét cạn (Exhaustive Search): Kẻ tấn công thử mọi giá trị khóa k khi có được một vài cặp (m_i, c_i)
  - DES: Với 2 cặp, xác suất tìm được đúng khóa k là ~ 1 – 2^{-71} với thời gian vét cạn 2^{56} giá trị
  - AES-128: Với 2 cặp, xác suất tìm được đúng khóa k là ~ 1 – 2^{-128} với thời gian vét cạn 2^{128} giá trị
  - Sử dụng tính toán lượng tử: thời gian vét cạn còn $T^{1/2}$ → sử dụng AES-256

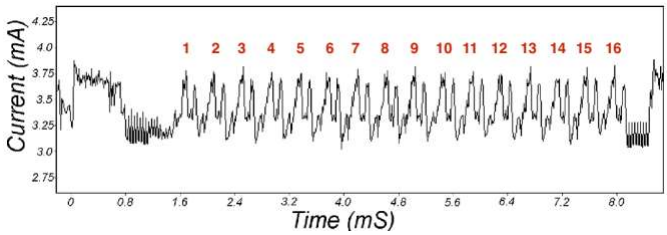
1976	DES adopted as federal standard		
1997	Distributed search	3 months	
1998	EFF deep crack	3 days	$250,000
1999	Distributed search	22 hours	
2006	COPACOBANA (120 FPGAs)	7 days	$10,000

Tấn công vào mật mã khối

- Tấn công vét cạn (Exhaustive Search): Kẻ tấn công thử mọi giá trị khóa k khi có được một vài cặp (m_i, c_i)
  - DES: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{71}$ với thời gian vét cạn 2^{56} giá trị
  - AES-128: Với 2 cặp, xác suất tìm được đúng khóa k là ~ $1 - 1/2^{128}$ với thời gian vét cạn 2^{128} giá trị
  - Sử dụng tính toán lượng tử: thời gian vét cạn còn $T^{1/2}$ → sử dụng AES-256
- Tấn công tuyến tính (Linear Attack): Kẻ tấn công tính toán khóa k khi có rất nhiều cặp (m_i, c_i)
  - DES: Với 2^{42} cặp có thể tìm thấy khóa K trong thời gian 2^{43}
  - AES-256: Với 2^{99} cặp có thể tìm thấy khóa K trong thời gian 2^{99}

Tấn công vào mật mã khối

- Tấn công kênh bên (side-channel attack): phán đoán giá trị các bit khóa bằng cách ước lượng thời gian, lượng điện năng tiêu thụ, bức xạ điện từ… khi mã hóa, giải mã
  - Ví dụ: phương pháp tấn công DES của Kocher và Jaffe năm 1998



- Tấn công dựa vào lỗi (Fault attacks): lỗi xảy ra ở vòng lặp cuối cùng trong DES sẽ làm lộ thông tin về khóa

2.3. Sinh số ngẫu nhiên

Entropy: độ đo tính bất định (tính khó đoán trước) của thông tin. Đơn vị tính: bit

ONE LOVE. ONE FUTURE.

Vấn đề sinh khóa mã hóa

- Cách thức 1: Sinh khóa dựa trên mật khẩu
  - Vấn đề: Entropy của mật khẩu rất thấp
  - Giải quyết: Sử dụng hàm dẫn xuất khóa từ mật khẩu PBKDF2() theo chuẩn PKCS#5
    - Hàm lõi: HMAC (Hashed MAC)
    - Thêm **giá trị ngẫu nhiên**(seed) có entropy lớn
    - Thực hiện lặp N vòng (N nên lớn 1000)

    →Khóa trông giống như chuỗi bit ngẫu nhiên
- Cách thức 2: Sử dụng nguồn **ngẫu nhiên** thực sự
- Cách thức 3: Sử dụng nguồn giả **ngẫu nhiên**

Tại sao sinh số ngẫu nhiên là quan trọng?

- Sinh khóa mã hóa giải mã
- Sinh giá trị IV/nonce
- Sinh các giá trị ngẫu nhiên trong các kịch bản ứng dụng khác
- Nếu đối phương có thể đoán trước giá trị ngẫu nhiên được sinh ra, hệ thống không còn an toàn
- Nguồn sinh số ngẫu nhiên có entropy càng cao thì càng ngẫu nhiên
  - Phân bố xác suất đều có entropy cao nhất
  - Nguồn có entropy n bit thì tương đương với phân bố xác suất 2^{-n}

Bộ sinh số ngẫu nhiên thực sự

- Để sinh số ngẫu nhiên thực sự, cần có nguồn vật lý. Ví dụ:
  - Sự biến động của các đại lượng trên mạch điện tử
  - Hoạt động của người dùng trong một khoảng thời gian
  - ...
- Hạn chế:
  - Không cân bằng giữa số lượng bit 0 và bit 1
  - Tốc độ chậm
  - Chi phí cao

Nguồn sinh số ngẫu nhiên thực sự tại tập đoàn Clouflare: bức tường bóng đèn đối lưu

Bộ sinh số giả ngẫu nhiên

- PRNG: Pseudo Random Number Generator
- Thuật toán biến đổi 1 chuỗi bit thực sự ngẫu nhiên (seed- số mầm) ngắn thành 1 chuỗi bit dài hơn và "trông giống như" ngẫu nhiên
- Hàm PNRG có tính xác định: chuỗi bit đầu ra được sinh theo thuật toán
  - Nhưng đối với kẻ tấn công, nếu không biết được seed thì không thể phân biệt sự khác nhau giữa đầu ra của hàm PRNG với chuỗi bit ngẫu nhiên thực sự
  - Tên khác: DRBG(Deterministic Random Bit Generator)
- Xây dựng hàm PRNG:
  - Từ hàm mã hóa ở chế độ CTR
  - Từ hàm HMAC

Tính an toàn của PRNG

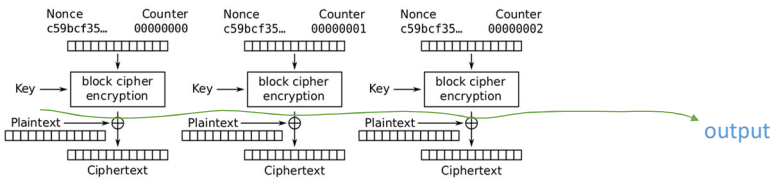
- PRNG không thể tạo ra giá trị thực sự ngẫu nhiên:
  - Giá trị đầu ra xác định theo giá trị seed khởi tạo
  - Nếu seed có kích thước s bit thì chỉ có thể tạo ra 2^s đầu ra có thể.
- An toàn của PRNG:
  - Thử thách: kẻ tấn công cần phân biệt một chuỗi bit ngẫu nhiên và một chuỗi bit là đầu ra của PRNG
  - PRNG là an toàn nếu xác suất phân biệt đúng không lớn hơn $\frac{1}{2} + \varepsilon$
- Định nghĩa tương đương: kẻ tấn công không thể đoán được chuỗi bit đầu ra của PRNG

PRNG: Chống hồi phục (rollback resistant)

- Dù kẻ tấn công biết được kết quả của một bước trung gian, hắn cũng không học được gì về kết của bước tính toán trước đó hoặc giá trị đầu ra.
- Thử thách: kẻ tấn công biết được kết quả tính toán trung gian và cần phân biệt một chuỗi bit ngẫu nhiên và một chuỗi bit đầu ra trước đó của PRNG
  - Xác suất phân biệt đúng không lớn hơn 1/2 + ε
- PRNG không cần đáp ứng yêu cầu chống hồi phục nhưng là một tính năng hữu dụng
  - Ví dụ: sử dụng PRNG để tạo sinh khóa bí mật và giá trị IV
  - Kẻ tấn công biết được kết quả tính toán trung gian
  - Nếu PRNG không chống hồi phục, kẻ tấn công có thể đoán được giá trị khóa bí mật

CTR-DRBG

- Sử dụng chế độ mã khối CTR để tạo chuỗi bit giả ngẫu nhiên
- Hạn chế: không có tính năng kháng hồi phục



Counter (CTR) mode encryption

- Có thể sử dụng HMAC-DRBG để thay thế
  - Bài học sau sẽ giới thiệu về HMAC

3. Hệ mật mã khóa bất đối xứng

Khóa mã hóa ≠ Khóa giải mã

ONE LOVE. ONE FUTURE.

Những hạn chế của mật mã khóa đối xứng

- Cần kênh mật để chia sẻ khóa bí mật giữa các bên
  - Làm sao để chia sẻ một cách an toàn cho lần đầu tiên
- Quá trình trao đổi khóa và trao đổi dữ liệu đòi hỏi cả 2 bên đều online
  - Giải pháp sử dụng bên thứ 3 tin cậy (trusted 3rd party) không giải quyết được vấn đề?
- Số lượng khóa lớn: n(n-1)/2
- Không dễ dàng để xác thực thông tin quảng bá (Chúng ta sẽ quay trở lại vấn đề này trong những bài sau)

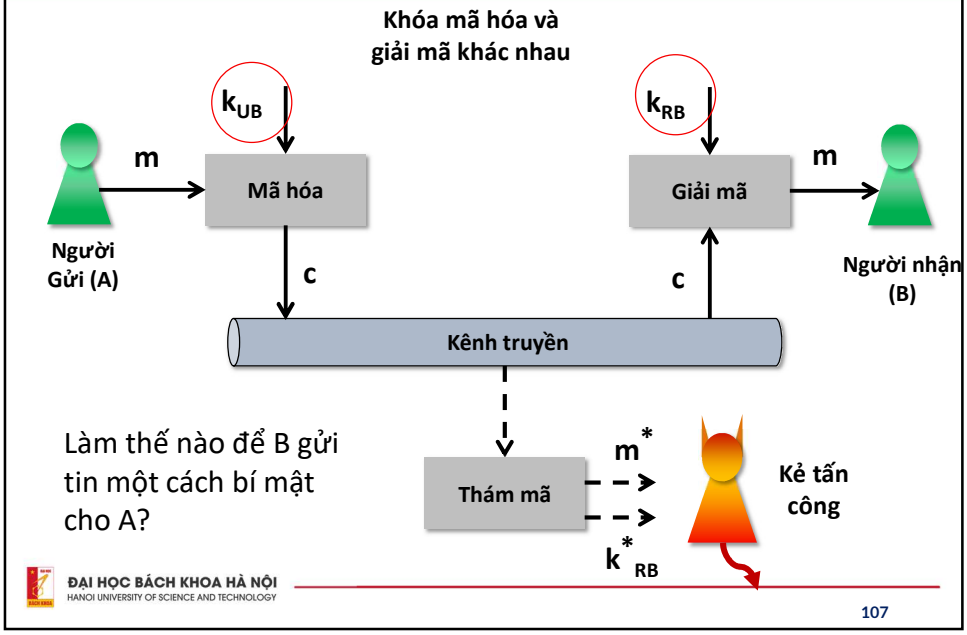
Hệ mật mã khóa bất đối xứng

- Asymmetric key cryptography, Public key cryptography
- Sử dụng một cặp khóa:
  - Khóa công khai k_U: Công bố cho tất cả cùng biết
  - Khóa cá nhân k_R: Chỉ chủ sở hữu biết, giữ bí mật
  - Mã hóa bằng khóa này thì giải mã bằng khóa còn lại.
- Cơ sở an toàn: Dựa trên một số bài toán không có lời giải trong thời gian đa thức
  - Ví dụ: Phân tích một số thành thừa số nguyên tố
- Các thuật toán dựa trên các hàm toán học
- Một số hệ mật mã khóa công khai: RSA, El-Gamal, Eliptic Curve Cipher (ECC)

Sơ đồ nguyên lý

- Hệ mật mã gồm:
  - Bản rõ (plaintext-m): thông tin không được che dấu
  - Bản mật (ciphertext-c): thông tin được che dấu
- Khóa: Bên nhận có **1 cặp** khóa (k_{UB}, k_{RB})
- Mã hóa (encrypt-E): $c = E(k_{UB}, m)$
  - Là hàm ngẫu nhiên
- Giải mã (decrypt): $m = D(k_{RB}, c)$
  - Là hàm xác định
- Tính đúng đắn: $D(k_{RB}, E(k_{UB}, m)) = m$
- Nếu hệ mật mã KCK an toàn trước tấn công KPA thì cũng an toàn trước tấn công CPA

Sơ đồ nguyên lý (tiếp)



Hệ mật RSA

- Sinh khóa:
  - Chọn p,q là hai số nguyên tố ngẫu nhiên
  - Tính $n = p \times q$, $\Phi(n) = (p-1)\times(q-1)$
  - Chọn e sao cho UCLN($\Phi(n)$, e) = 1 ;1< e < $\Phi(n)$
  - Tính d sao cho $(e \times d) \bmod \Phi(n) = 1$; $1 < d < \Phi(n)$
  - Khóa công khai : $k_U = (e, n)$
  - Khóa riêng : $k_R = (d, n)$
- Mã hóa : $c = m^e \bmod n$ (điều kiện: m < n)
- Giải mã: $m = c^d \bmod n$ (điều kiện: c < n)

Hệ mật RSA

- Sinh khóa:
  - Chọn p = 5, q = 11
  - Tính n = p × q = 55, $\Phi(n) = (p-1)\times(q-1) = 40$
  - Chọn e sao cho UCLN($\Phi(n)$, e) = 1 và $1 < e < \Phi(n)$

  VD: e = 7
  - Tính d sao cho $(e \times d) \bmod \Phi(n) = 1$, $1 < d < \Phi(n)$

  d = 23

  Cặp khóa : $k_U = (7,55)$, $k_R = (23,55)$
- Mã hóa: m = 6 → $c = m^e \bmod n = 6^7 \bmod 55 = 41$
- Giải mã: c = 41 → $m = c^d \bmod n = 41^{23} \bmod 55 = 6$

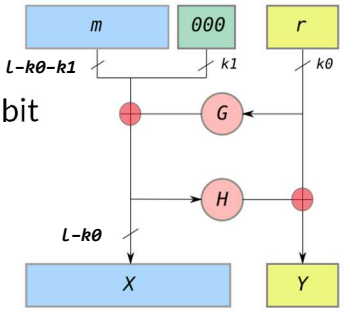
Nếu kẻ tấn công có k_U, làm thế nào để tính k_R?

Những vấn đề của mật mã RSA

- Bản tin gốc m có kích thước nhỏ → kẻ tấn công có thể thực hiện kiểm tra vét cạn để xác định bản tin gốc.
  - $m \leq n^{1/e}$ với e đủ nhỏ
- Giá trị e nhỏ cho phép kẻ tấn công xác định được các bản tin gốc nếu chúng có liên quan với nhau
- Giá trị e nhỏ cho phép kẻ tấn công đoán nhận được bản tin gốc nếu bản tin đó được mã hóa và gửi tới nhiều đích
- Nếu biết c và số lượng bản tin m có thể là đủ nhỏ → thử mã hóa tất cả bản tin m và so sánh với c

RSA-OEAP (Chuẩn PKCS#1 v2.0)

- Nếu bản tin m được mã 2 lần với cùng khóa k thì nội dung bản mã không thay đổi → không chống được tấn công KPA → không an toàn
- RSA-OEAP: sử dụng thêm khối đệm(padding) và giá trị ngẫu nhiên trong quá trình mã hóa
- Chống lại được tấn công CCA
- Xử lý bản m trước khi mã hóa:
  - r: giá trị ngẫu nhiên kích thước k0 bit
  - G, H: hàm băm
- Mã hóa:
  - X = (m || padding) XOR G(r)
  - Y = H(X) XOR r
  - Mã hóa RSA (X||Y)



111

Độ an toàn của RSA

Date	Minimum of Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1** SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256 SHA-384 SHA-512

http://www.keylength.com

112

Tấn công vào RSA

- Tấn công kênh bên: quan sát quá trình giải mã
  - Phân tích thời gian [Kocher et al. 1997]: quá trình giải mã có thể lộ thông tin về khóa riêng
  - Phân tích mức độ tiêu thụ năng lượng [Kocher et al. 1999]
  - Phân tích tiếng ồn phát ra từ CPU [Daniel Genkin et al. 2013]
- Tấn công dựa vào lỗi tính toán
- Tấn công do sinh khóa không ngẫu nhiên:
  - Giả sử quá trình sinh khóa sử dụng p1 = p2 nhưng q1 ≠ q2 → UCLN(N_1, N_2) = p
  - Thực tế: 0.4% số lần sinh khóa ra trong giao thức HTTPS gặp lỗi trên

```
x = C
for j = 1 to n
  x = mod(x^2, N)
  if dj == 1 then
    x = mod(x^C, N)
  end if
return x
```

Kết hợp mật mã KCK và mật mã KĐX

- Ưu điểm của mật mã khóa công khai:
  - Không cần chia sẻ khóa mã hóa k_{UB} một cách bí mật
  - Khóa giải mã k_{RB} chỉ có B biết:
    - An toàn hơn
    - Có thể sử dụng k_{RB} để xác thực nguồn gốc thông tin (Chúng ta sẽ quay lại vấn đề này trong bài sau)
  - Số lượng khóa để mã mật tỉ lệ tuyến tính với số phần tử (n phần tử → n cặp khóa)
- Nhưng...

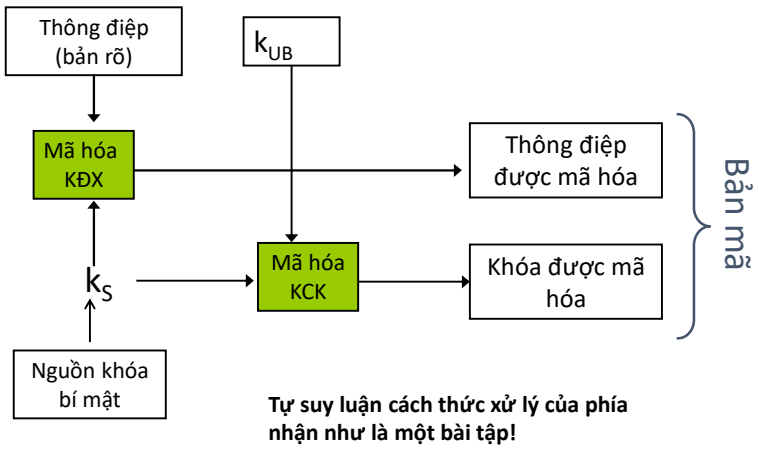
Kết hợp mật mã KCK và mật mã KĐX

- Hạn chế của mật mã khóa công khai so với mật mã khóa đối xứng:
  - Kém hiệu quả hơn: khóa có kích thước lớn hơn, chi phí tính toán cao hơn
  - Có thể bị tấn công toán học

→ Kết hợp 2 hệ mật mã

Sơ đồ "lai"

- Phía gửi



Tự suy luận cách thức xử lý của phía nhận như là một bài tập!

Những sai lầm khi sử dụng mật mã

- Không thay đổi giá trị IV(Initial Vector)
- Sử dụng chế độ mã từ điển (ECB)
- Case study: Lỗi sử dụng mật mã trong các ứng dụng Android (2013)
  - Phân tích 11.748 ứng dụng

	# apps	violated rule
48%	5,656	Uses ECB (BouncyCastle default) (R1)
31%	3,644	Uses constant symmetric key (R3)
17%	2,000	Uses ECB (Explicit use) (R1)
16%	1,932	Uses constant IV (R2)
	1,636	Used iteration count < 1,000 for PBE(R5)
14%	1,629	Seeds SecureRandom with static (R6)
	1,574	Uses static salt for PBE (R4)
12%	1,421	No violation

Những sai lầm khi sử dụng mật mã

- Lỗ hổng trên HĐH Android được phát hiện vào năm 2013 cho thấy quá trình sinh khóa không đủ ngẫu nhiên
  - Các ứng dụng sử dụng cơ chế mã hóa bị ảnh hưởng, trong đó có các ứng dụng sử dụng Bitcoin để thanh toán
- Lỗ hổng trên Chromebooks: sinh giá trị ngẫu nhiên chỉ có 32 bit thay vì 256 bit
- Sửa đổi/Thêm một vài yếu tố bí mật vào giải thuật, hệ mật mã sẽ an toàn hơn
- Sử dụng các hàm ngẫu nhiên không an toàn
  - Ví dụ: các hàm ngẫu nhiên tiêu chuẩn của ngôn ngữ lập trình

Một số lưu ý khác

- Chỉ sử dụng thuật toán chuẩn và các thư viện lập trình được phê chuẩn: OpenSSL, Bouncy Castle, Libgcrypt, RSA BSAFE, wolfCrypt
- Nếu có thể, sử dụng các thuật toán mạnh nhất
- Nếu phải sinh khóa từ một giá trị cho trước, sử dụng hàm PBKDF2()
- Sử dụng mật mã theo tiêu chuẩn. Ví dụ: PKCS, FIPS
- Cẩn trọng khi không gian bản gốc là hẹp và chúng có sự khác biệt về kích thước

Case study: Bảo mật trên iPhone

ONE LOVE. ONE FUTURE.

Bảo mật trên iPhone

- Triết lý bảo mật của Apple:
  - Nếu điện thoại trong tay bạn, bạn có thể truy cập mọi thứ
  - Nếu điện thoại trong tay người khác, nó chỉ là một "cục gạch"
- Apple triển khai tất cả các cơ chế bảo mật trên chip Secure Enclave Processor (SEP)
  - Secure Enclave Processor là TCB của thiết bị
- Mọi phần còn lại của thiết bị được coi là không tin cậy
  - Bộ nhớ là không tin cậy, vì vậy mọi dữ liệu được mã hóa
  - CPU phải gửi yêu cầu tới Secure Enclave để giải mã
  - Một vài dữ liệu bí mật chỉ có thể đọc bởi Secure Enclave

Bảo mật trên iPhone

- Khóa chủ K_{phone} được sử dụng để mã hóa các khóa mật mã khác
- K_{phone} được mã hóa bởi K_{user}, sinh từ mật khẩu của người dùng
  - K_{user} = PBKDF(password, hardcode)
  - hardcode có kích thước 256 bit được sinh và lưu trữ trên chip SEP
- Làm cách nào kẻ tấn công có thể đánh cắp được dữ liệu mà không có mật khẩu người dùng?