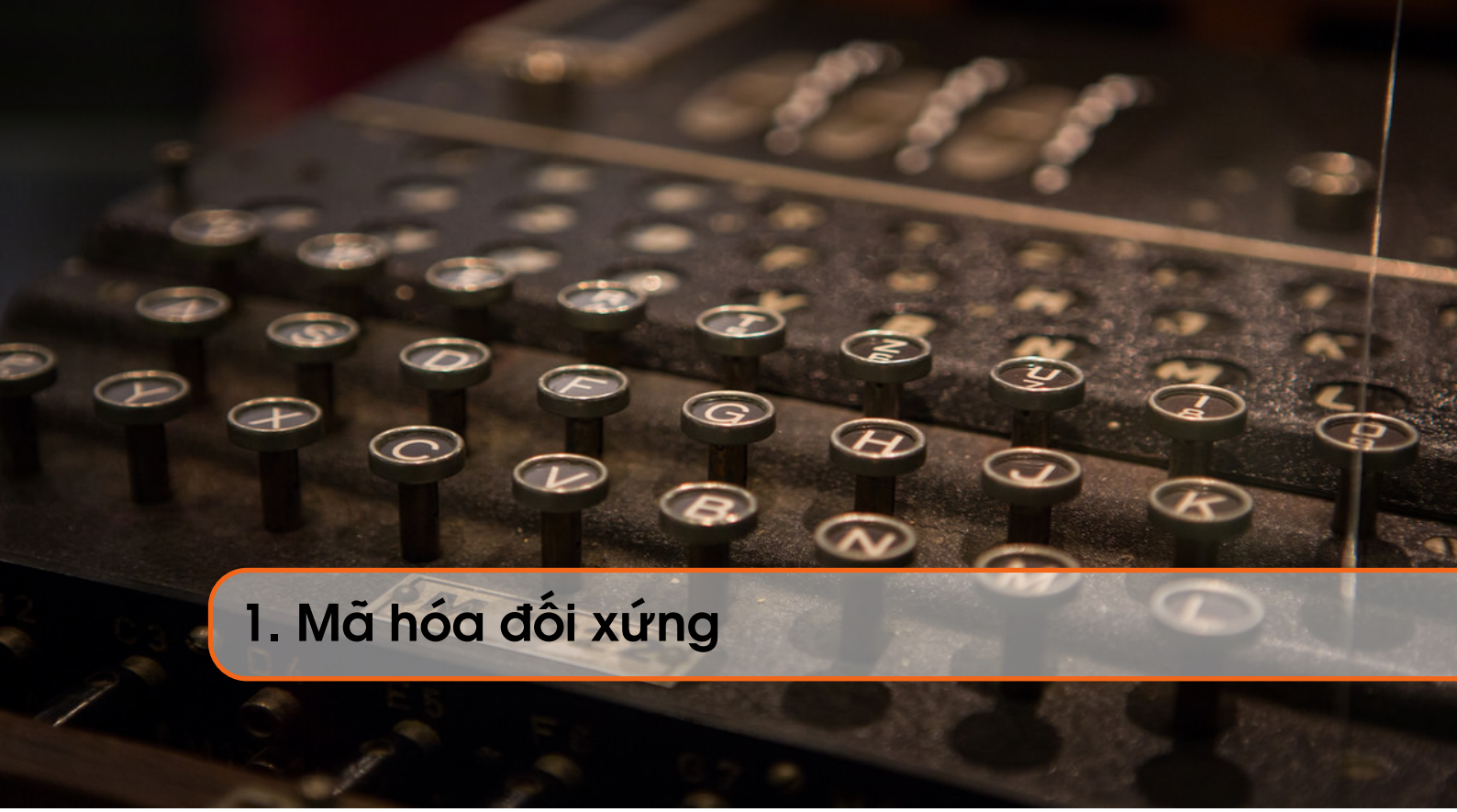


1. Mã hóa đối xứng

1.1 NỘI DUNG THỰC HÀNH

1.1.1 Mở đầu

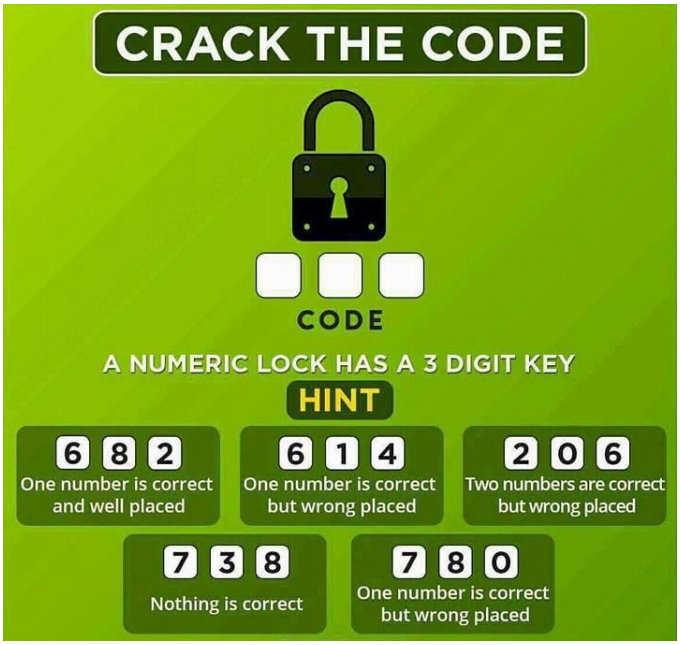
Task 1.1 Để mở đầu về mật mã học, sinh viên hãy đưa ra lời giải cho 2 bài tập với việc mã hóa thông tin cơ bản, không sử dụng thuật toán mã hóa sau đây:

1. Hãy tìm 3 chữ số (code) theo các gợi ý tại hình 1.1
2. Hãy tìm mã hóa tương ứng của các số từ 1 đến 9 theo dữ kiện từ bảng 1.1.1
   - Mỗi biểu tượng trong số 9 biểu tượng xuất hiện trong bảng dưới đây (△◁▷◯♡♠♢♣●) mã hóa duy nhất cho một trong các chữ số 1 đến 9.
   - Cột ngoài cùng bên phải là tổng của các số trong mỗi dòng.
   - Dòng dưới cùng là tổng của các số trong mỗi cột.
   - Một dấu **?** có thể đại diện cho một hoặc hai chữ số bất kỳ.

■

△	△	◁	◯	?
♡	♡	♠	♡	♢♢
?	?	◁	♣	●●
?	♡	♠	♡	●▷
●♡	♢♢	●●	●♢	

Bảng 1.1: Tìm mã hóa của các số từ 1 đến 9



Hình 1.1: Hãy tìm ra 3 chữ số với các dữ kiện đã cho sẵn.

1.1.2 Mã hóa Caesar

Task 1.2 Hãy viết 1 ứng dụng có thể mã hóa và giải mã sử dụng mã hóa Ceasar. Ngôn ngữ lập trình do sinh viên tự chọn *(khuyến khích sử dụng Python)*. Ứng dụng có các chức năng chính như sau:

- Nhập *plaintext* để mã hóa hoặc *ciphertext* để giải mã dựa vào khóa k tương ứng.
- Hỗ trợ *brute-force* (vét cạn) tất cả trường hợp của k để tìm *plaintext* khi chỉ cung cấp *ciphertext*.

Kiểm tra mã hóa, giải mã Caesar với một đoạn plaintext khoảng 100 ký tự bằng ứng dụng vừa xây dựng. Sau đó kiểm tra việc mã hóa, giải mã Caesar với cùng plaintext trên CrypTool. ■

Gợi ý 1.1. *Xét mỗi ký tự tương ứng với 1 số p từ 0 đến 25. Sử dụng công thức mã hóa:*

$$C = E(k,p) = (p+k) \ mod \ 26 \tag{1.1}$$

Công thức giải mã:

$$p = D(k,C) = (C-k) \ mod \ 26 \tag{1.2}$$

với C = Ciphertext, p = plaintext, k là mã dịch chuyển với $0 < k < 26$.

Task 1.3 Cho đoạn ciphertext sau, bằng ứng dụng đã xây dựng hãy tìm plaintext ban đầu và trình bày cách thực hiện. Loại mã hóa này có gì đặc biệt?
Crbcyr jub fhpprrq unir zbzraghz. Gur zber gurl fhpprrq, gur zber gurl jnag gb fhpprrq naq gur zber gurl svaq n jnl gb fhpprrq. Fvzvyneyl, jura fbzrbar vf

snvyvat, gur graqrapl vf gb trg ba n qbjajneq fcveny gung pna rira orpbzr n frys-shysvyyvat cebcurpl. ■

1.1.3 Mã hóa Monoalphabetic

Task 1.4 Sử dụng công cụ phù hợp để tìm *plaintext* của *ciphertext* đã được mã hóa thay thế sau. *Trình bày và giải thích cách thực hiện.*
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZVUEP HZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSXEPYEPOPDZSZUFP OMBZWPFUPZHMDJUDTMOHMQ ■

Gợi ý 1.2. *Dựa vào tần suất xuất hiện của các ký tự trong ciphertext và tần suất xuất hiện phổ biến của các chữ cái trong bảng chữ cái Tiếng Anh. (Hình 1).*
Trong CrypTool 2, để dò tìm plaintext được mã hóa bằng cách thay thế các ký tự trong bảng chữ cái, có thể sử dụng chức năng ***Monoalphabetic Substitution Analyzer***

Task 1.5 Sử dụng CrypTool hoặc công cụ tương đương để xác định plaintext của ciphertext được mã hóa thay thế sau:
GSVFMREVIHRGBLURMULINZGRLMGVXSMLOLTBRHZNVNYVILUE RVGMZNMZGRLMZOFMREVIHRGBSLXSRNRMSXRGBEMFSXNZMW RHGSVLMOBFMREVIHRGBLUERVGMZNGSZGFMWVIGZPVHRMULI NZGRLMZMWXLNNFMRXZGRLMGVXSMLOLTBIVHVZIXSZMWULX FHVWRMWVKGSGIZRMRMTGSVFMREVIHRGBSZHGSVBLFMTVHG NZMZTVNVMGIVHVZIXSZMWGVZXSRMTHGZUULUZMBEMFSXNN VNYVIYIRMTRMTTIVZGVMGSFHRZHNZOLMTDRGSWBMZNRXZM WXIVZGREVZWEZMGZTVH
Trình bày cách thực hiện. Loại mã hóa này là gì và có gì đặc biệt? ■

1.1.4 Mã hóa Polyalphabetic - Vigenère

Task 1.6 Xây dựng ứng dụng hỗ trợ mã hóa và giải mã Vigenère *(sinh viên tự chọn ngôn ngữ lập trình)*
Kiểm tra mã hóa, giải mã **Vigenère** với một đoạn *plaintext* khoảng 50 ký tự và 1 khóa từ 10-20 ký tự, sau đó kiểm tra lại bằng CrypTool hoặc công cụ tương đương. Giải thích cách hoạt động của ứng dụng đã xây dựng để tìm *ciphertext*. ■

Gợi ý 1.3. *Sử dụng công thức mã hóa của Vigenère:*
Công thức mã hóa của ký tự thứ i:

$$C_i = (p_i + k_{i \bmod m}) \bmod 26 \tag{1.3}$$

Công thức giải mã của ký tự thứ i:

$$p_i = (C_i - k_{i \bmod m}) \bmod 26 \tag{1.4}$$

với C = Ciphertext, p = plaintext, k là khóa, m là số ký tự của khóa.

Mở rộng 1.1 Trong trường hợp mã hóa đa bảng Vigenere, làm thế nào để phá mã *(tìm plaintext từ ciphertext)* khi không có khóa? Tìm hiểu và trình bày cách thực hiện phá mã

Vigenere, áp dụng cho ví dụ với *ciphertext* sau:
cv vvobobxy uocmgjg, olgiaqsliioa dynxyu fi axjwdy qgmqdgimpqz uvw jqygcgpemnqhu vqwpgpsgya wltupmw mtag utajqgimpemf khueqjbl hpp u axa qr lcel-dmmmw jcxwcehvuivl jcxfmw hnsizbajym bh atmhayvty gmlzcsya bu ymsa mocf uzx ocdx bh kgocxalt. fbmll fqnmktkzcampe mfohykfbul htq oaxk hal kkfrfiokhrtck dla syvxycfcwg hpp xqzpvmf abnpuho tuf hyzbmkoubbvp fi xkvvqwb whvm jzbccos, exi ddielpps iv mog uhbxypqn igk eahnbkgznqts eagunukoubbvpe mcvo ce wzxkkf wikk vduvlhefcwgz czx mfhkx

1.1.5 Mã hóa Playfair

Task 1.7 Xây dựng ứng dụng mã hóa và giải mã Playfair (sinh viên tự chọn ngôn ngữ lập trình). Ứng dụng có thể đáp ứng các yêu cầu sau:
- Nhập từ khóa, xuất ma trận Playfair *(ma trận 5x5)* các chữ cái kèm từ khóa dùng để mã hóa tương ứng
- Kiểm tra mã hóa, giải mã Playfair với một đoạn *plaintext* khoảng từ 100 ký tự, sau đó kiểm tra lại bằng một công cụ khác. Giải thích cách hoạt động của ứng dụng đã xây dựng. ■

Task 1.8 Sử dụng ma trận Playfair tại bảng 1.2 bên dưới để mã hóa thông điệp sau *(sử dụng CrypTool hoặc công cụ tự xây dựng)*:
Must see you over Cadogan West. Coming at once
Thông điệp này trích từ *Sherlock Holmes, The Adventure of the Bruce-Partington Plans*
■

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

Bảng 1.2: Ma trận Playfair Task 1.7

Task 1.9 Khi tàu Mỹ PT-109 do đại úy hải quân John F. Kennedy bị đánh chìm bởi một tàu khu trục Nhật vào tháng 8/1943, tại một đài vô tuyến điện của người Úc đã nhận được một thông điệp bằng mã hóa Playfair như sau:
KXJEY UREBE ZWEHE WRYTU HEYFS
KREHE GOYFI WTTTU OLKSY CAJPO
BOTEI ZONTX BYBNT GONEY CUZWR
GDSON SXBOU YWRHE BAAHY USEDQ
Khóa sử dụng là ***royal new zealand navy***.
Hãy giải mã thông điệp và trình bày cách thực hiện. ■

1.1.6 Các loại mã hóa khác

Task 1.10 Cho 1 ảnh *crypto01.jpg* chứa *Flag* và đã được mã hóa XOR với một khóa có độ dài 6 ký tự.
Sinh viên download tại `https://goo.gl/NqSNNu`.
Hãy tìm Flag và trình bày cụ thể cách thực hiện. ■

Gợi ý 1.4. *Có thể sử dụng các phần mềm để phân tích và phá mã XOR để tìm ra file ảnh ban đầu.*

1.2 Thực hành trên bộ thực hành SeedLabs

Task 1.11 Sinh viên thực hiện theo các hướng dẫn và yêu cầu bài lab "Secret-Key Encryption Lab"trong bộ thực hành SeedLabs (Xem tại: `https://seedsecuritylabs.org/Labs_20.04/Crypto/Crypto_Encryption/`) ■