TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÀI KIỂM TRA GIỮA KÌ MÔN XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN GIỮA KÌ MÔN XÁC SUẤT VÀ THÔNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN

Người hướng dẫn: **TS. TRẦN LƯƠNG QUỐC ĐẠI**
Người thực hiện: **PHẠM PHƯỚC TẤN – 520H0418**
Lớp : **20H50204**
Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN

BÀI KIỂM TRA GIỮA KÌ MÔN XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN

TIỂU LUẬN GIỮA KÌ MÔN XÁC SUẤT VÀ THÔNG KÊ ỨNG DỤNG CHO CÔNG NGHỆ THÔNG TIN

Người hướng dẫn: **TS. TRẦN LƯƠNG QUỐC ĐẠI**
Người thực hiện: **PHẠM PHƯỚC TẤN**
Lớp **: 20H50204**
Khoá **: 24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022

LỜI CẢM ƠN

Em thật sự cảm ơn thầy Trần Lương Quốc Đại vì trong suốt thời gian em làm bài và trước khi làm bài thầy hướng dẫn em và nhờ sự hướng dẫn của thầy nên em có thể hoàn thành bài báo cáo này một cách hiệu quả và tốt hơn.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi và được sự hướng dẫn của TS Nguyễn Văn A. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 16 tháng 04 năm 2022

Tác giả

(ký tên và ghi rõ họ tên)

Phạm Phước Tấn

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

_____

_____

_____

_____

_____

_____

_____

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Trong bài tiểu luận này thì sẽ có một số vấn đề cần nghiên cứu đó là lý thuyết về encryption, decryption, symmetric và asymmetric cryptosystem. Nhưng hai vấn đề quan trọng nhất đó là phân tích mã hóa sử dụng Monoalphabet Substitution Cipher và giải mã sử dụng Frequency Analysis.

Hướng tiếp cận và cách giải quyết vấn đề ở bài này là tự học, tìm hiểu ở các nguồn trên internet và một số nguồn khác.

Kết quả đạt được sau khi em tự học và hoàn thành bài này là em sẽ hiểu được bản chất của việc bảo mật, mã hóa đoạn văn nào đó và cách giải mã đoạn văn đã mã hóa đó.

MỤC LỤC

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

CHƯƠNG 1 – INTRODUCTION

Giới thiệu: Các khái niệm về mã hóa và giải mã; đối xứng và hệ thống mật mã không đối xứng.

1.1 Định Nghĩa Mã Hóa

Mã hóa là một tiến trình thực hiện việc trao đổi hoặc kết hợp giữa các dữ liệu bao gồm chữ, số và kí tự, phổ biến nhất là thường áp dụng đối với mật khẩu, việc mã hóa nhằm tăng tính bảo mật cho người sử dụng, chỉ những người tạo ra dữ liệu đó hoặc một ai đó khác sử dụng công cụ hoặc ứng dụng nào đó có thể khôi phục những dòng mã hóa thành trạng thái ban đầu. Tính năng trên cực kì quan trọng đối với trên mạng xã hội internet, tin nhắn riêng và hầu như đều quan trọng với bất kì thứ gì liên quan đến bảo mật và thông tin, tài khoản cá nhân hoặc nhạy cảm. Và nó cung cấp cho người dùng sự tin cậy khi trong việc có cuộc giao tiếp với ai đó về công việc và sử dụng internet một cách an toàn và đảm bảo dữ liệu được lưu trữ trên hệ thống (được gọi là server) và chúng được vận chuyển hoặc truyền qua các phương thức của mạng máy tính. Mã hóa có một số tính năng cực kì quan trọng để đảm bảo dữ liệu có được truyền đi một cách toàn vẹn hay không, được gửi hay chưa và nhận hay là chưa nhận.

1.2 Định Nghĩa Giải Mã

Giải mã là một công cụ hoặc phương tiện để chuyển đổi dữ liệu từ dạng đã mã hóa không thể đọc được về thành đúng dạng ban đầu của nó được gửi đi và để giải mã được thì cần phải có yêu cầu khóa bảo mật hoặc mật khẩu từ dạng mã hóa gửi đến. Việc giải mã cũng rất quan trọng không riêng gì như mã hóa, vì khi thông tin được truyền qua internet thì cần phải kiểm trả một cách kĩ lưỡng việc truy cập từ các tổ chức hoặc cá nhân không được quyền hoặc không cho phép. Do đó, dữ liệu hạn chế việc kẻ xấu xâm nhập vào và giảm mất mát, đánh cấp dữ liệu.

Một vài dữ liệu phổ biến được mã hóa gồm tệp văn bản, hình ảnh, tin nhắn e-mail, dữ liệu người dùng và thư mục.

Người được yêu cầu giải mã thì người đó sẽ nhận được lời nhắn hoặc một cửa sổ ứng dụng nào đó mà trong đó cho người giải mã có thể nhập mật khẩu truy cập thì mới được phép truy cập vào dữ liệu đã mã hóa để giải mã nó.

Việc giải mã cũng có thể được giải mã tự động hoặc thủ công do một người nào đó được phép truy cập và nó được thực hiện với một bộ khóa hoặc mật khẩu.

Có nhiều loại giải mã nhưng cách giải mã chính xác, khả năng bảo mật và quan trọng nhất là giải mã HILL, tạo ra ma trận bảo mật.

1.3 Định Nghĩa Hệ Thống Mật Mã Đối Xứng

Hệ thống mật mã đối xứng là hệ thống sử dụng cùng mật khẩu hoặc khóa(key) cho cã hai quá trình mã hóa và giải mã để cho người gửi và người nhận có thể mã hóa và giải mã thì hai người trên phải cần trao đổi với nhau để đưa ra key cho quá trình mã hóa-giải mã được diễn ra hoặc trao đổi thông điệp.

Để cho quá trình trao đổi giữa người nhận và người gửi đồng ý trước khi trao đổi dữ liệu thì hệ thống phải sử dụng một loại giao thức để tăng độ bảo mật sự giao tiếp giữa hai người được tính riêng tư và không có ai có thể xâm nhập vào được đó là giao thức Diffie-Hellman.

Đối với hệ thống mật mã đối xứng thì có hạn chế đó là: cần có sự trao đổi key hoặc mật khẩu trước khi diễn ra quá trình trao đổi dữ liệu giữa người nhận và người gửi; Nếu như key hoặc mật khẩu không được thay đổi một cách đều đặn và thường xuyên thì kẻ gian có thể xâm nhập vào hệ thống vì kẻ gian có thể sử dụng khóa bị rò rỉ để làm cho đoạn liên lạc bị gián đoạn. Những đã có cách khắc phục ở bên hệ thống đó là hệ thống sử dụng MACs để đảm bảo tính toàn vẹn và xác thực cao nhất.

1.4 Định Nghĩa Hệ Thống Mật Mã Không Đối Xứng

Hệ thống mật mã không đối xứng là hệ thống cải tiến của hệ thống mật mã đối xứng bằng cách loại bỏ quá trình trao đổi mật khẩu và khóa trước khi trao đổi dữ liệu.

Hệ thống mật mã không đối xứng sử dụng hai khóa(key) khác nhau gồm 2 khóa đó là khóa công khai và khóa riêng tư, giữa hai khóa có sự liên kết về mặt toán học.

Điểm mạnh là người gửi và người nhận sẽ có cặp khóa riêng tư trong hệ thống này với mỗi quá trình trao đổi dữ liệu.

Hệ thống mật mã không đối xứng sử dụng lược đồ mã hóa El Gamal hoặc Mã hóa RSA. Đem lại rất an toàn và hơn hẳn hệ thống mật mã đối xứng cho quá trình trao đổi dữ liệu vì hệ thống sử dụng cặp khóa riêng tư và công khai, cặp đó ít bị rò rỉ và tỉ lệ dự đoán thấp. Hệ thống mật mã không đối xứng sử dụng chữ ký RSA để cung cấp tính toàn vẹn và xác thực.

CHƯƠNG 2 – MONOALPHABETIC SUBSTITUTION CIPHER

Những nghiên cứu thực nghiệm hoặc lý thuyết: Nêu vấn đề, các ràng buộc/điều kiện(nếu có), phương pháp/thuật toán; Ví dụ; Nhận xét,phân tích,đánh giá của bạn.

2.1 Định nghĩa, vấn đề, ràng buộc và phương pháp

Mật mã thay thế đơn pha là một loại mật mã phổ biến nhất, chúng sử dụng phương pháp thay thế mỗi chữ cái của plaintext (cũng có thể thay thế dấu chấm hoặc dấu cách hay kể cả kí tự đặc biệt, ngẫu nhiên) dựa trên phương pháp đó họ sẽ thay thế bằng một kí hiệu của ciphertext dựa trên KEY để hình thành ra Ciphertext và sau khi thay thế hết chữ cái của plaintext thì sẽ ra một đoạn text với kích thước tương tự như plaintext đó được gọi là ciphertext.

Vấn đề ở thuật toán này là mã hóa một đoạn plaintext và biến nó thành một dạng đã mã hóa được gọi là Ciphertext dựa trên một phương thức thay thế cố định hoặc

nói một cách dể hiểu hơn là trong plaintext thì mỗi lần xuất hiện của một ký hiệu plaintext được thay thế bằng một ký hiệu Ciphertext tương ứng để tạo Ciphertext.

Ràng buộc ở thuật toán này là việc hình thành từ plaintext của alphabet sang ciphertext của alphabet thì phải dựa vào key ở đề bài cung cấp thì mới có được ciphertext của tất cã chữ cái của plaintext. Cụ thể là tạo bảng chữ cái của ciphertext và mã hóa bằng cách so sánh bảng này với bảng chữ cái của plaintext dự vào KEY nhưng trong ciphertext không được trùng chữ cái alphabet, chữ cái nào đến trước thì dùng trước, khi thấy nó được lặp lại ở phía sau thì loại bỏ rồi tiếp tục chữ cái tiếp theo cho đến khi hết bảng chữ cái của plaintext.

Phương pháp là thay thế key vào từng chữ cái theo thứ tự của bảng chữ cái alphabet (ciphertext không được có chữ cái trùng). Đầu tiên là sẽ tạo một bảng chữ cái alphabet ở dưới dạng plain-text từ A->Z, ta sẽ được đề bài cung cấp một đoạn chữ cái nào đó, không tính dấu khoãng cách và sau đó sẽ lấy KEY gán vào ở đầu của ciphertext theo đúng thứ tự A-B-C-..-Z của plain-text và gán cho đến khi hết chữ cái trong KEY, Tiếp theo ta sẽ gán các chữ cái còn lại theo bảng chữ cái alphabet và chữ cái nào được lặp lại thì bỏ qua chữ đó và tiếp tục gán theo thứ tự tương ứng của plaintext cho đến khi hết chữ cái của plaintext. Nhưng đó chỉ là đoạn plaintext của bảng chữ cái để ta dựa vào đó để mã hóa theo KEY. Cuối cùng, yêu cầu của đề bài cần ta mã hóa một đoạn khác với nhiều hoặc ít chữ cái hơn so với plaintext của alphabet, đó được gọi là yêu cầu mã hóa đoạn plaintext của đề bài, và để mã hóa đoạn đó thì ta dựa vào đoạn bảng chữ cái của plaintext để đối chiếu tương ứng với ciphertext của alphabet rồi ta sẽ lấy chữ cái ở ciphertext đó để thay thế cho chữ cái ở đoạn plaintext của đề bài.

2.2 Ví dụ

Ví dụ: KEY = {LEBRON JAMES}, plaintext cần mã hóa đó là {I want to go DALAT city}

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

Bảng 2.2.1 Plaintext sang Ciphertext của Alphabet

P-A nghĩa là Plaintext Alphabet, C-A là Ciphertext Alphabet. Mới chỉ tạo ra cipher text của alphabet để mã hóa đoạn text yêu cầu của đề bài.

Đoạn plaintext yêu cầu đề bài cần mã hóa đó là “i want to go another trip to relax”, ta sẽ dựa vào bảng 1.2.1 để mã hóa đoạn plaintext trên. Chữ đầu tiên là “i”, ta sẽ nhìn vào bảng 1.2.1 và kiếm chữ “i” ở P-A và rồi nhìn xuống tương ứng cột của chữ cái “i” ta thấy nó được thay thế thành “M”, gặp khoãng cách thì ta không mã hóa nó, tới lúc thay thế xong thì sẽ thêm dấu khoãng cách vào. Tương tự với các chữ còn lại thì “w” thành “w”, “a” thành “l”, “n” thành “g”, “t” thành “t”, “t” thành “t”, “o” thành “h”, “g” thành “j”,“o” thành “h”, “a” thành “l”, “n” thành “g”, “o thành “h”, “t” thành “t”, “h” thành “a”, “e” thành “o”, “r” thành “p”, “t” thành “t”, “r” thành “p”, “i” thành “m”, “p” thành “i”, “t” thành “t”, “o” thành “h”, “r” thành “p”, “e” thành “o”, “l” thành “d”, “a” thành “l” và “x” thành “x”.

P-T	i	w	a	n	t	t	o	g	o	a	n	o	t	h	e	r	t	r	i	p	t	o	r	e	l	a	x
C-T	m	w	l	g	t	t	h	j	h	l	g	h	t	a	o	p	t	p	m	i	t	h	p	o	d	l	x

Bảng 2.2.2 Mã hóa “I want to go another trip to relax” sang Ciphertext

P-T là plaintext của yêu cầu đề bài, C-T là cipher text là đoạn mã hóa của plaintext.

Kết quả của đoạn plaintext ở đề bài “I want to go another trip to relax” thành “m wlgt th jh lghtaop tpmi th podlx”.

2.3 Nhận xét,phân tích và đánh giá

Phân tích: Trên thực tế có rất nhiều mật mã thay thế khác nhau vì một chữ cái có thể thay thế bất ký chữ cái hoặc kí hiệu nào đó khác với chữ cái cần thay thế. Thuật toán này cần tới bước để thực hiện thành công mã hóa vì ban đầu chúng ta phải tạo ra cipher text của alphabet dựa vào KEY, sau đó mới dựa vào đoạn cipher đó và tiến hành thay thế theo từng cột từng chữ đúng thứ tự.

Nhận xét: Phương pháp thay thế của thuật toán này cực kì đơn giản và dể hiểu, nhưng kẻ xâm nhập ngày nay có thể dò ra được KEY và phá vỡ đoạn mã hóa đó và điều đó cực kì nguy hiểm cho hệ thống và người sử dụng. Nhưng giải thuật này là một bước đà để phát triển cho giải thuật mã hóa và bảo mật cao hơn.

Đánh giá: Thuật toán mã hóa mật mã thay thế đơn pha ở hiện nay là cực kì yếu và dể bị kẻ xâm nhập phá vỡ cho nên hiện nay không còn sử dụng nữa và nó đã được cải tiến sang thứ mới mạnh mẽ bảo mật hơn.

CHƯƠNG 3 – FREQUENCY ANALYSIS

Những nghiên cứu thực nghiệm hoặc lý thuyết: Nêu vấn đề, các ràng buộc/điều kiện (nếu có), phương pháp/thuật toán; Ví dụ; Nhận xét, phân tích và đánh giá.

3.1 Định nghĩa, vấn đề, ràng buộc và phương pháp

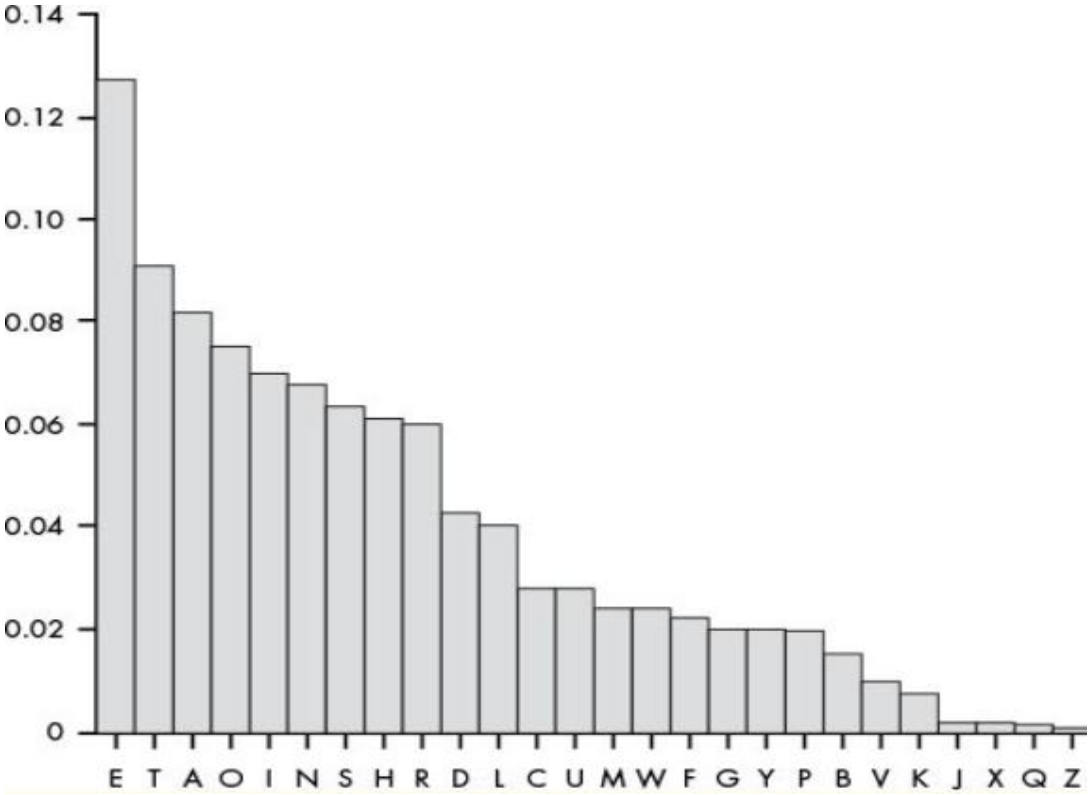
Định nghĩa: Phân tích tần số là một loại phân tích tần suất, khả năng và tần số xuất hiện của các chữ cái trong ciphertext, ciphertext là một chuỗi các kí tự đã được mã hóa từ đoạn văn nào đó.

Vấn đề ở thuật toán này là ta phải phá vỡ đoạn mã hóa đó bằng cách tính tần số xuất hiện của các chữ cái để giúp xác định các chữ cái lặp lại nhiều nhất trong văn bản mật mã. Tính tần số dựa trên đoạn đã được mã hóa, qua các mật mã sau khi mã hóa thì họ ít dùng các chữ cái như là Z, Q, J và X nhưng họ sử dụng nhiều những từ như là E,

T, A và O đối với ngôn ngữ tiếng anh, còn một số ngôn ngữ khác thì họ sử dụng đa dạng từ ngữ và chữ cái kí tự.

Ràng buộc là đối với thuật toán này thì ta nên dùng văn bản tiếng anh vì không phải tất cả các chữ cái đều xảy ra với tần suất giống nhau và số lượng từ phải nhiều thì mới phân tích một cách đúng đắng được.

Phương pháp: Giải thuật này dùng để dịch và phá vỡ các mật mã thay thế như là mật mã thay thế đơn pha. Tính tần số để giúp xác định các chữ cái lặp lại nhiều nhất trong đoạn văn bản đã mã hóa. Sau đó, ta có thể dễ dàng nhận biết được cách hoạt động của cấu trúc đoạn văn bản đó và cách thay thế của nó. Sự phân tích tần số cần sự cẩn thận về số lần xuất hiện của mỗi kí tự trong đoạn văn bản đã mã hóa. Sau khi đã tính số lần xuất hiện của các chữ cái trong đoạn đã mã hóa thì sau đó ta sẽ dựa vào hình bên dưới:



Hình 3.1 Tần số thay thế của các chữ cái để thay thế cho đoạn văn đã mã hóa

(Nguồn: inventwithpython.com)

Ta sẽ sắp xếp các chữ cái sau khi tính tần suất lần xuất hiện của các chữ cái trong đoạn văn đã mã hóa theo xu hướng giảm dần, ta sẽ lấy các chữ cái được lặp lại nhiều nhất từ nhiều lần xuất hiện đến ít lần xuất hiện và ta sẽ lấy từ đó thử thay thế theo thứ tự ưu tiên như hình trên như là E, T, A, O, I, N, …., Z. Và trong lúc thử ta nên xem xét kĩ xem liệu chữ mà ta muốn thay thế nó có phù hợp hay là không, nếu nó phù hợp theo ngữ nghĩa từ của tiếng anh thì sự thay thế là đúng và tiếp tục cứ lấy chữ cái từ xuất hiện nhiều lần cho đến xuất hiện ít lần hơn trong câu rồi thay thế cho đến hết đoạn sắp xếp tần suất xuất hiện đó.

3.2 Ví dụ

Ví dụ là em có đoạn văn đã mã hóa là “m wlgt th jh lghtaop tpmi th podlx” và em muốn giải mã nó để trở thành lại đoạn văn nguyên mẫu của nó.

Độ ưu tiên thay thế của các chữ cái sẽ thay thế cho đoạn mã hóa là như sau theo thứ tự từ cao đến thấp: E T A O I N S H R D L C U M W F G Y P B V K J X Q Z.

Sau đó, em sẽ tính tần suất xuất hiện trong đoạn mã hóa: {'m': 2, ' ': 7, 'w': 1, 'l': 3, 'g': 2, 't': 5, 'h': 4, 'j': 1, 'a': 1, 'o': 2, 'p': 3, 'i': 1, 'd': 1, 'x': 1} và em sẽ sắp xếp nó lại theo thứ tự giảm dần, không tính khoãng trắng: {'t': 5, 'h': 4, 'l': 3, 'p': 3, 'o': 2, 'm': 2, 'g': 2, 'a': 1,'d': 1, 'w': 1, 'j': 1, 'i': 1, 'x': 1}.

Sau khi sắp xếp thì ta thấy chữ cái ‘t’ xuất hiện nhiều nhất cho nên em sẽ thử thay thế chữ ‘t’ thành ‘e’ vì độ ưu tiên thay thế đầu tiên là chữ ‘e’, đoạn mã cần giải mã từ “m wlgt th jh lghtaop tpmi th podlx” thành “m wlge eh jh lgheaop epmi eh podlx”, sau khi đã thử thay thế thì em thấy một điều bất thường là trong tiếng anh không có từ nào hai chữ cái mà bắt đầu bằng chữ ‘e’ như là “eh” cho nên em nghĩ chữ ‘e’ sẽ không phù hợp cho sự thay thế chữ ‘t’. Tiếp tục em sẽ thử thay thế chữ ‘t’ thành ‘t’ dựa vào thứ tự ưu tiên, đoạn mã sau khi thay thế sẽ như lúc ban đầu vì ‘t’ thành ‘t’ ra là “m wlg**t** **t**h jh lgh**t**aop **t**pmi **t**h podlx” xem xét qua thì em thấy “th” sẽ có một số từ tiếng anh có thể được như là “to” nó có thể xãy ra cho nên sẽ quyết định thay thế chữ ‘t’.

Sau khi đã thay thế chữ ’t’, số lần xuất hiện nhiều nhất đứng sau ‘t’ là chữ ‘h’, chữ ‘h’ em sẽ thử thay thế chữ ‘e’ và đoạn mã sẽ thành là “m wlgt te je lgetaop tpmi te podlx” em thấy rằng là “te” trong tiếng anh không phải là từ tiếng anh nên là thay bằng chữ ‘e’ sẽ không phù hợp cho nên em sẽ tiếp tục chữ ưu tiên tiếp theo ngoại trừ ‘e, t’ vì ‘t’ đã thay thế rồi, cho nên em sẽ thử thay thế chữ ‘a’ theo độ ưu tiên thay thế thì đoạn mã sẽ thành “m wlgt ta ja lgataop tpmi ta podlx”, từ “ta” cũng không có nghĩa nên chữ ‘a’ cũng không phù hợp, tiếp tục em sẽ thử thay thế thành chữ ‘o’, đoạn mã sẽ thành là “m wlgt to jo lgotaop tpmi to podlx”, từ “to” có nghĩa trong tiếng anh nên sự thay thế này là phù hợp. Đoạn mã sau khi thay thế chữ ‘h’ thành ‘o’ là “m wlg**t to** j**o** lg**ot**aop **t**pmi **to** podlx”.

Sau khi đã thay thế chữ ’h’, số lần xuất hiện nhiều nhất đứng sau ‘h’ là chữ ‘l’, và em sẽ thử thay thế chữ ‘l’ thành ‘e’ thì đoạn mã sẽ thành là “m wegt to jo egotaop tpmi to podex”, em nhận thấy là từ “podex” là không khả thi vì trước từ đó là từ “to” đã thay thế xong, thì đứng sau “to” phải là động từ nguyên mẫu, em nghĩ là từ đó có thể là “index” chỉ có mỗi từ đó nên từ đó là danh từ nên là không phù hợp cho “to”, cho nên thay thế đó không phù hợp. Tiếp tục sẽ thử thay thế chữ ‘a’ cho chữ ‘l’ thì đoạn mã sẽ thành là “m wagt to jo agotaop tpmi to podax” là phù hợp vì “podax” có thể là “relax” là động từ nên phù hợp ở đây. Sau khi thay thế thì đoạn mã thành: “m w**a**g**t to** j**o a**g**ot**aop **t**pmi **to** pod**a**x”.

Sau khi đã thay thế chữ ’l’, số lần xuất hiện nhiều nhất đứng sau ‘l’ là chữ ‘p’, và em sẽ thử thay thế chữ ‘l’ thành ‘r’ vì em thấy nó là hợp lí ở đây, vì em muốn nó thành từ “relax” cho nên em sẽ thay thế và đoạn mã sẽ thành: “m w**a**g**t to** j**o a**g**ot**ao**r** **tr**mi **to r**od**a**x”.

Sau khi đã thay thế chữ ’p’, số lần xuất hiện nhiều nhất đứng sau ‘p’ là chữ ‘o’, và em sẽ thử thay thế nó thành chữ ‘e’ và đoạn mã sẽ thành là “m w**a**g**t to** j**o a**g**ot**a**er** **tr**mi **to re**d**a**x” vì em thấy từ “**re**d**a**x” là phù hợp vì em đang hướng tới từ “relax” .

Sau khi đã thay thế chữ ’o’, số lần xuất hiện nhiều nhất đứng sau ‘o’ là chữ ‘m’, và em sẽ thử thay thế nó thành chữ ‘i’ theo độ ưu tiên, không lặp lại chữ và đoạn mã sẽ thành là “**i** w**a**g**t to** j**o a**g**o**t**a**e**r tri**i **to re**d**a**x” vì em thấy từ “i” và từ “trii” sẽ thành “trip” nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ ’m’, số lần xuất hiện nhiều nhất đứng sau ‘m’ là chữ ‘g’, và em sẽ thử thay thế nó thành chữ ‘n’ theo độ ưu tiên, không lặp lại chữ và đoạn mã sẽ thành là “**i** w**ant to** j**o anot**a**e**r **tri**i **to re**d**a**x” vì em thấy từ “want” có nghĩa trong tiếng anh nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ ’g’, số lần xuất hiện nhiều nhất đứng sau ‘g’ là chữ ‘a’, và em sẽ thử thay thế nó thành chữ ‘s’ theo độ ưu tiên, không lặp lại chữ và đoạn mã sẽ thành là “**i** w**ant to** j**o anotser tri**i **to re**d**a**x” vì em thấy từ “**anotser** ” không có nghĩa trong tiến anh nên em nghĩ sự thay thế này không phù hợp cho nên em sẽ thay thế chữ ưu tiên tiếp theo đó là chữ ‘h’ cho chữ ‘a’ và đoạn mã sẽ thành “**i** w**ant to** j**o another tri**i **to re**d**a**x” vì em thấy từ “**another** ” có nghĩa trong tiếng anh nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ ’a’, số lần xuất hiện nhiều nhất đứng sau ‘a’ là chữ ‘d’, và em sẽ thử thay thế nó thành chữ ‘s’ theo độ ưu tiên, không lặp lại chữ và đoạn mã sẽ thành là “**i** w**ant to** j**o another tri**i **to resa**x” vì em thấy từ “**resa**x” không có nghĩa trong tiến anh nên em nghĩ sự thay thế này không phù hợp và chữ ưu tiên tiếp theo là ‘d’ thì từ “**reda**x” không có nghĩa trong tiến anh nên em nghĩ sự thay thế này không phù hợp luôn, cho nên em sẽ thay thế chữ ưu tiên tiếp theo đó là chữ ‘l’ cho chữ ‘d’ và đoạn mã sẽ thành “**i** w**ant to** j**o another tri**i **to rela**x” vì em thấy từ “**another**” có nghĩa trong tiếng anh nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ ’d’, số lần xuất hiện nhiều nhất đứng sau ‘d’ là chữ ‘w’, và em sẽ thử thay thế nó thành chữ ‘w’ luôn vì ‘w’ rất khó xuất hiện nên em nghĩ là nó

sẽ giữ nguyên nên sẽ thành là "**i want to** j**o another tri**i **to rela**x" vì em thấy từ "want" có nghĩa trong tiếng anh nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ 'w', số lần xuất hiện nhiều nhất đứng sau 'w' là chữ 'j', và em sẽ thử thay thế nó thành chữ 'g' đoạn mã sẽ thành là "**i want to go another tri**i **to rela**x" vì em thấy từ "**go**" có nghĩa trong tiếng anh và phù hợp với "**i want to**" phía sau là một động từ nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế chữ 'j', số lần xuất hiện nhiều nhất đứng sau 'j' là chữ 'i', và em sẽ thử thay thế nó thành chữ 'p' đoạn mã sẽ thành là "**i want to go another trip to rela**x" vì em thấy từ "**trip**" có nghĩa trong tiếng anh và em không thấy từ nào là phù hợp hơn "trip" nên em nghĩ sự thay thế sẽ phù hợp.

Cuối cùng, sau khi đã thay thế chữ 'i', số lần xuất hiện nhiều nhất đứng sau 'i' là chữ 'x', và em sẽ thử thay thế nó thành chữ 'x' vì "relax" giữ nguyên cho chữ cái 'x' là phù hợp và đoạn mã sẽ thành là "**i want to go another trip to relax**" là đúng từ vựng, có nghĩa trong tiềng anh và cho nên em nghĩ sự thay thế sẽ phù hợp.

Sau khi đã thay thế hết chữ cái theo thứ tự giảm dần thì em thấy đoạn văn nguyên bản của đoạn mã hóa là "i want to go another trip to relax" và các chữ cái đã sử dụng để thay thế theo thứ tự thay thế "T,O, A, R, E, I, N, H, L, W, G, P".

3.3 Nhận xét,phân tích và đánh giá

Phân tích: Trên thực tế thì phương pháp giải mã phân tích tần số như trên thì khả năng chính xác về giải mã là khá cao đối với đoạn văn bảng bằng tiếng anh với đoạn văn bản đủ dài và đoạn mã hóa có sử dụng nhiều từ như E, T, A và O. Nếu chúng ta vận dụng sự thay thế đó nếu chữ cái thay thế cảm thấy không phù hợp thì ta tiếp tục thử các chữ cái tiếp theo và nó sẽ giải mã ra đúng với ý của chúng ta.

Nhận xét: Phương pháp phân tích tấn số này có thể được áp dụng với lúc trước nhưng hiện tại thì ít sử dụng vì tỉ lệ chính xác của nó trong code thì ít chính xác hơn khi giải tay. Vì trên thực tế thì đoạn văn bản muốn giải mã nó rất là dài và việc giải tay

nó tốn rất là nhiều thời gian nên là nó sẽ áp dụng giải mã dưới dạng áp dụng các đoạn code cho nên tỉ lệ chính xác đúng với đoạn văn bản nguyên bản thì cũng không khá cao và các kẻ gian cũng sẽ dễ dàng lấy được đoạn mã hóa đó và phân tích ra được. Vì thế, phương pháp này ít được sử dụng ở hiện nay và thực tế.

Đánh giá: Thuật toán phân tích tần số ở hiện nay là ít phổ biến, ít được sử dụng và dể bị kẻ xâm nhập lấy được đoạn mã hóa và phân tích ra được như người giải mã cho nên hiện nay ít sử dụng nữa và hiện nay đã có nhiều cách giải mã đa dạng hơn ở thực tế.

CHƯƠNG 4 – EXPERRIMENTS

Các thư viện và biến mà em dùng:

```
import random
from collections import Counter
alpha ="abcdefghijklmnopqrstuvwxyz"
l = list(alpha)
random.shuffle(l)
key = ''.join(l)
```

Mã hóa:

```python
#encryption
def encryption(plaintext,key):
    alphabet ="abcdefghijklmnopqrstuvwxyz " #bao gồm cả khoảng cách
    NewPlaintext = "" # để chứa các kí tự của plantext nếu nó thuộc kí tự alphabet
    newKey = "" #dùng để chứa các kí tự của key cộng với các kí tự còn lại trong alphabet(không trùng)
    for c in plaintext:
        if c in alphabet:
            NewPlaintext += c
    for c in key:
        if c in alphabet:
            if c != " ":
                if c not in newKey:
                    newKey = newKey + c #key
    for c in alphabet:
        if c not in newKey:
            newKey = newKey + c #thêm key trước rồi tới các kí tự còn lại của alphabet(không trùng))

    indexOfPlainText = []  #thứ tự cua plaintext trong alphabet bao gồm khoảng cách
    for c in NewPlaintext:
        indexOfPlainText.append(alphabet.index(c)) #lấy các số của chữ cái alphabet trong đoạn plaintext

    Ciphertext = ""
    characters= "" #dùng để chứa kí tự của số trong indexOfPlainText
    for idx in indexOfPlainText:
        characters += newKey[idx]  #thêm kí tự của số thứ tự idx trong indexOfPlainText

        Ciphertext += characters  #sau khi thêm thì truyền characters đó vào ciphertext

        characters= "" #reset số của kí tự về rỗng
    return Ciphertext #trả về ciphertext cho hàm giải mã
```

Giải mã:

```
from collections import Counter
def decryption(ciphertext):
    storedLetter = {} #dictionary
    for c in ciphertext:
        if c not in storedLetter:
            storedLetter[c] = 1
        else:
            storedLetter[c] += 1

    #sort letter turn into array
    cipherSorted = Counter(storedLetter).most_common()

    j =0
    for i in cipherSorted:
        if i[0] == ' ':
            continue
        else:
            ciphertext = ciphertext.replace(i[0],charactersUutien[j])
            j = j+1

    decrypted_plaintext=ciphertext

    return(decrypted_plaintext)
```

Thử nghiệm:

Với 50 từ tiếng anh là “”e can only offer plain text views through our Plain Text client in Checklist. This will provide a screenshot of the plain text version of your email. If a plain text version is not present you may see a message saying that a plain text version was not present now":

```
#test

charactersUutien = ['e','t','a','o','i','n','s','h','r','d','l','c','u', #kí tự ưu tiên được thay thế
                    'm','w','f','g','y','p','b','v','k','j','x','q','z']


plaintext = "We can only offer plain text views through our Plain Text client in Checklist. This will provi
plaintext = plaintext.lower()
print(key)
print(plaintext)


print("\nsau khi mã hóa:")
ciphertext = encryption(plaintext, key)
print(ciphertext)

print("\nsau khi giải mã:")
print(decryption(ciphertext))
```

```
PS C:\AcademicTDT\XSTK\midterm> py .\520H0418.py
tbnsjcfpowmuahedqlikxvzyrg
we can only offer plain text views through our plain text client in checklist. this will provide a screenshot of the plain text version of your email. if a plain text version is not present yo
u may see a message saying that a plain text version was not present now

sau khi mã hóa:
zj nth ehur eccjl dutoh kjyk vojzi kplexfp exl dutoh kjyk nuojhk oh npjnmuoik kpoi zouu dlevosj t inljjhipek ec kpj dutoh kjyk vjlioeh ec rexl jatou oc t dutoh kjyk vjlioeh oi hek dljijhk rex
atr ijj t ajiitfj itrohf kptk t dutoh kjyk vjlioeh zti hek dljijhk hez

sau khi giải mã:
wu ukb ubhv uwwuv dhkob kupk gouwk klvuypl uyv dhkob kupk uhoubk ob uluuvhokk klok wohh dvugoku k kuvuubkluk uw klu dhkob kupk guvkoub uw vuyv ubkoh ow k dhkob kupk guvkoub ok buk dvukubk vuy
bkv kuu k bukkkpu kkvobp klkk k dhkob kupk guvkoub wkk buk dvukubk buw
PS C:\AcademicTDT\XSTK\midterm>
```

Với 100 từ tiếng anh là “A long time ago, in a galaxy far, far away... It is a dark time for the Rebellion. Although the Death Star has been destroyed, Imperial troops have driven the Rebel forces from their hidden base and pursued them across the galaxy. Evading the dreaded Imperial Starfleet, a group of freedom fighters led by Luke Skywalker has established a new secret base on the remote ice world of Hoth. The evil lord Darth Vader, obsessed with finding young Skywalker, has dispatched thousands of remote probes into the far reaches of space":

```python
charactersUutien = ['e','t','a','o','i','n','s','h','r','d','l','c','u', #kí tự ưu tiên được thay thế
                    'm','w','f','g','y','p','b','v','k','j','x','q','z']


plaintext ="A long time ago, in a galaxy far, far away... It is a dark time for the Rebellion. Although
plaintext = plaintext.lower()
print(key)
print(plaintext)


print("\nsau khi mã hóa:")
ciphertext = encryption(plaintext, key)
print(ciphertext)

print("\nsau khi giải mã:")
print(decryption(ciphertext))
```

```
PS C:\AcademicTDT\XSTK\midterm> py .\520H0418.py
usldaqwrxgznkvfcjotipeymnb
a long time ago, in a galaxy far, far away... it is a dark time for the rebellion. although the death star has been destroyed, imperial troops have driven the rebel forces from their hidden ba
se and pursued them across the galaxy. evading the dreaded imperial starfleet, a group of freedom fighters led by luke skywalker has established a new secret base on the remote ice world of ho
th. the evil lord darth vader, obsessed with finding young skywalker, has dispatched thousands of remote probes into the far reaches of space

sau khi mã hóa:
u nfvw ixka uwf xv u wunumh quo quo uyuh xi xt u duoz ixka qfo ira oasannxfv unirfpwr ira dauir tiuo rut saav datiofhad xkcaoxun ioffct ruea doxeav ira oasan qfolat qofk iraxo rxddav suta uvd
cpotpad irak ulofft ira wunumh aeudxvw ira doaudad xkcaoxun tiuoqnaai u wofpc fq qoaadfk qxwriaot nad sh npza tzhyunzao rut atiusnxtrad u vay taloai suta fv ira oakfia xla yfond fq rfir ira ae
xn nfod duoir eudao fstattad yxir qxvdxvw hfpvw tzhyunzao rut dxtcuilrad irfptuvdt fq oakfia cofsat xvif ira quo oaulrat fq tcula

sau khi giải mã:
i ddpu orjk iud rp i uidijf gia gia ibif ro ri i fiav orjk gda owk akwkddrdp idowdbuw owk fkiow ioia wii wkkp fkioadfkf rjgkarid oaddgi wikk farkkp owk akwkd gdapki gadj owkra wrffkp wiik ipf
gbaibkf oakj ipadii owk uidijf kkifrpu owk fakifkf rjgkarid ioiagdkko i uadbg dg gakkfdj gruwokai dkf wf dbvk ivfbidvka wii kioiwdriwkf i pkb ikpako wiik dp owk akjdok rpk bdadf dg wdow owk kk
rd ddaf fiaow kifka dwikiikf brow grpfrpu fdbpu ivfbidvka wii frigiopwkf owdbiipfi dg akjdok gadwki rpod owk gia akipwki dg igipk
PS C:\AcademicTDT\XSTK\midterm>
```

Dòng chạy đầu tiên là đoạn Key.

Dòng thứ hai là đoạn plaintext được gắn vào lúc ở đề bài.

Dòng thứ ba là đoạn mã hóa.

Dòng thứ ba là đoạn giải mã.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. https://www.internetsociety.org/issues/encryption/what-is/.
2. https://economictimes.indiatimes.com/definition/decryption.
3. https://www.educative.io/edpresso/what-are-symmetric-and-asymmetric-cryptosystems.
4. https://www.101computing.net/frequency-analysis/#:~:text=When%20trying%20to%20decrypt%20a,the%20letters%20in%20the%20text.