

TRƯỜNG ĐẠI HỌC SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO AN TOÀN BẢO MẬT
HỆ THỐNG THÔNG TIN

Tên đề tài:
MÃ HÓA AES TRONG PHẦN MỀM QUẢN LÝ
THƯ VIỆN

Họ tên thành viên trong nhóm:

Nguyễn Tấn Hòa	3119410135
Nguyễn Thanh Hải	3119410105
Phạm Văn Lợi	3118410256
Nguyễn Thanh Châu	3118410038

GIÁO VIÊN HƯỚNG DẪN: Huỳnh Nguyễn Khắc Huy
TP. HỒ CHÍ MINH, năm 2022

MỤC LỤC

LỜI MỞ ĐẦU	2
I. THUẬT TOÁN MÃ HÓA KHÓA ĐỐI XỨNG AES	3
1.1. Tổng quan về AES.....	3
1.2. Xây dựng thuật toán.....	3
1.2.1. Xây dựng bảng S-box.....	3
1.2.2. Giải thuật sinh khóa phụ.....	4
1.2.3. Quá trình mã hóa	5
1.3. Quá trình giải mã	8
1.3.1. Tổng quan	8
1.3.2. Thuật toán giải mã	8
1.4. Các dạng tấn công vào AES và phương pháp phòng chống	9
1.4.1. Side-channel attack.....	9
1.4.2. Known attacks.....	9
1.4.3. Các phương pháp phòng chống	9
II. ÁP DỤNG AES VÀO PHẦN MỀM QUẢN LÝ THƯ VIỆN.	11
2.1. Giao diện mô tả mã hóa	11
2.2. Coding function	14
III. TỔNG KẾT.....	16
3.1. Kết quả đạt được	16
3.2. Đánh giá thuật toán.....	16
TÀI LIỆU THAM KHẢO.....	17

LỜI MỞ ĐẦU

Vấn đề đảm bảo an ninh, an toàn thông tin dữ liệu là nội dung nghiên cứu thiết thực, là chủ đề luôn được các cấp, các ngành quan tâm trong lĩnh vực công nghệ thông tin. Nhu cầu đảm bảo an ninh thông tin dữ liệu trên mạng máy tính là cấp thiết trong các hoạt động kinh tế xã hội, đặc biệt là đối với các mạng máy tính chuyên dùng phục vụ công tác an ninh, quốc phòng, đối ngoại của các cơ quan Đảng, Nhà nước... Thực tế ứng dụng công nghệ thông tin trong các lĩnh vực liên quan đến an ninh chính trị, quốc phòng luôn gặp phải những rủi ro đột nhập trái phép, tấn công, lấy cắp thông tin,...

Với việc sử dụng kiến thức về mã hóa của mình. Chúng em đã thực hiện mã hóa phần mềm: “*Quản lý thư viện*”, một phần để tổng hợp kiến thức, phần là tích lũy kinh nghiệm cho bản thân. Với sự giúp đỡ tận tình của giảng viên chúng em đã hoàn thành được đề tài. Mặc dù vậy do sự phức tạp của đề tài nên chúng em vẫn còn những sai sót. Mong bạn đọc và giảng viên góp ý kiến để nhóm em chỉnh sửa bổ sung.

Chúng em xin chân thành cảm ơn.

I. THUẬT TOÁN MÃ HÓA KHÓA ĐỐI XỨNG AES

1.1. Tổng quan về AES

AES (viết tắt của từ tiếng anh: Advanced Encryption Standard, hay Tiêu chuẩn mã hóa nâng cao) là một thuật toán mã hóa khối được chính phủ Hoa Kỳ áp dụng làm tiêu chuẩn mã hóa.

Thuật toán được xây dựng dựa trên Rijndael Cipher phát triển bởi 2 nhà mật mã học người Bỉ: Joan Daemen và Vincent Rijmen.

AES làm việc với các khối dữ liệu 128bit và độ dài khóa 128bit, 192bit hoặc 256bit. Các khóa mở rộng sử dụng trong chu trình được tạo ra bởi thủ tục sinh khóa Rijndael.

Hầu hết các phép toán trong thuật toán AES đều thực hiện trong một trường hữu hạn của các byte. Mỗi khối dữ liệu đầu vào 128bit được chia thành 16byte, có thể xếp thành 4 cột, mỗi cột 4 phần tử hay một ma trận 4x4 của các byte, nó gọi là ma trận trạng thái.

Tùy thuộc vào độ dài của khóa khi sử dụng 128bit, 192bit hay 256bit mà thuật toán được thực hiện với số lần lặp khác nhau.

1.2. Xây dựng thuật toán

1.2.1. Xây dựng bảng S-box

a. Bảng S-box thuận

Bảng S-box thuận được sinh ra bằng việc xác định nghịch đảo cho một giá trị nhất định trên $GF(28) = GF(2)[x] / (x^8+x^4+x^3+x+1)$ (trường hữu hạn Rijndael). Giá trị 0 không có nghịch đảo thì được ánh xạ với 0. Những nghịch đảo được chuyển đổi thông qua phép biến đổi affine.

Công thức tính các giá trị bảng S-box và bảng S- box tương ứng:

$$\begin{bmatrix}
 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1
 \end{bmatrix}
 \begin{bmatrix}
 x_0 \\
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7
 \end{bmatrix}
 +
 \begin{bmatrix}
 1 \\
 1 \\
 0 \\
 0 \\
 0 \\
 1 \\
 1 \\
 0
 \end{bmatrix}$$

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	fe	0e	61	35	57	b9	86	e1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

b. Bảng S-box nghịch đảo

S-box nghịch đảo chỉ đơn giản là S-box chạy ngược. Nó được tính bằng phép biến đổi affine nghịch đảo các giá trị đầu vào. Phép biến đổi affine nghịch đảo được biểu diễn như sau:

$$\begin{bmatrix}
 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0
 \end{bmatrix}
 \begin{bmatrix}
 x_0 \\
 x_1 \\
 x_2 \\
 x_3 \\
 x_4 \\
 x_5 \\
 x_6 \\
 x_7
 \end{bmatrix}
 +
 \begin{bmatrix}
 1 \\
 0 \\
 1 \\
 0 \\
 0 \\
 0 \\
 0 \\
 0
 \end{bmatrix}$$

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1x	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2x	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3x	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4x	72	f8	fe	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5x	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6x	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7x	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8x	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9x	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
ax	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
bx	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
cx	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
dx	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
ex	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
fx	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

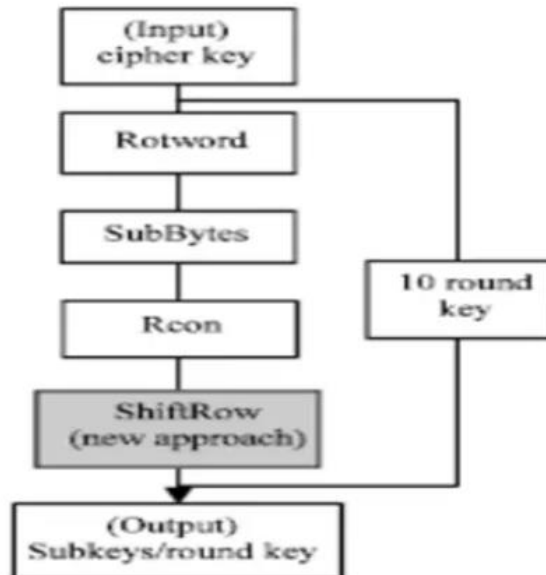
1.2.2. Giải thuật sinh khóa phụ

Quá trình sinh khóa gồm 4 bước:

- Rotword: quay trái 8 bit

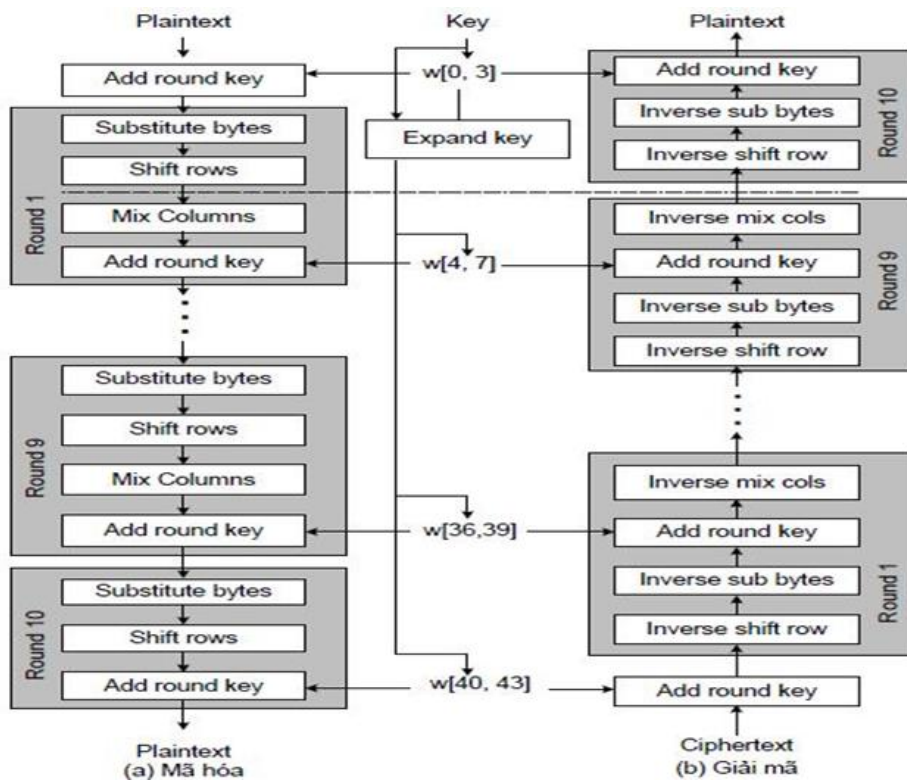
- SubBytes
- Rcon: tính giá trị Rcon(i) Trong đó :

$$Rcon(i) = x(i-1) \bmod (x^8 + x^4 + x^3 + x + 1).$$
- ShiftRow



1.2.3. Quá trình mã hóa

a. Sơ đồ tổng quát



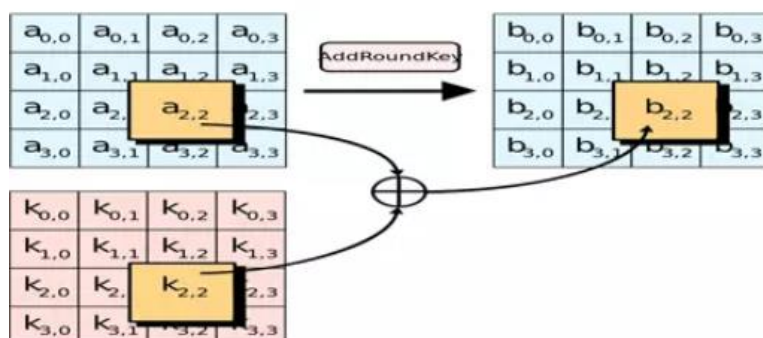
b. Hàm AddRoundKey

Được áp dụng từ vòng lặp thứ 1 tới vòng lặp Nr

Trong biến đổi Addroundkey(), một khóa vòng được cộng với state bằng một phép XOR theo từng bit đơn giản.

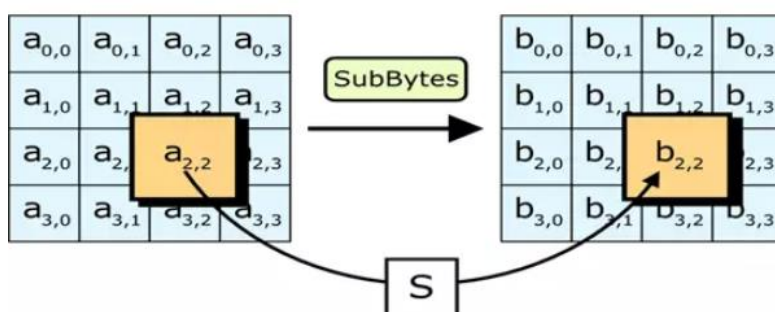
Mỗi khóa vòng gồm có 4 từ (128 bit) được lấy từ lịch trình khóa. 4 từ đó được cộng vào mỗi cột của state, sao cho:

$$[S'_{0,c}, S'_{1,c}, S'_{2,c}, S'_{3,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c}] \oplus [W(4*i + c)] \text{ với } 0 \leq c < 4.$$



c. Hàm SubBytes

Biến đổi SubBytes() thay thế mỗi byte riêng rẽ của state $S_{r,c}$ bằng một giá trị mới $S'_{r,c}$ sử dụng bảng thay thế (S - box) được xây dựng ở trên.



d. Hàm ShiftRow

Trong biến đổi ShiftRows(), các byte trong ba hàng cuối cùng của trạng thái được dịch vòng đi các số byte khác nhau (độ lệch). Cụ thể :

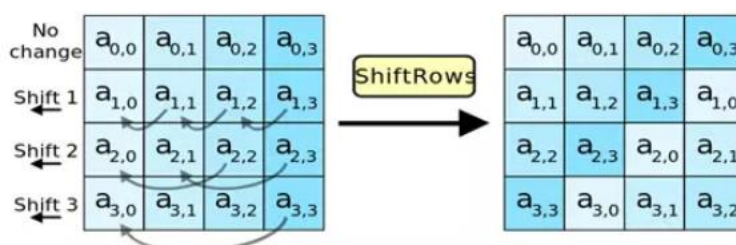
$$S'r,c = Sr,(c + \text{shift}(r, Nb)) \bmod Nb \quad (Nb = 4)$$

Trong đó giá trị dịch shift (r, Nb) phụ thuộc vào số hàng r như sau:

$$\text{Shift}(1,4) = 1, \text{shift}(2,4) = 2, \text{shift}(3,4) = 3.$$

Hàng đầu tiên không bị dịch, ba hàng còn lại bị dịch tương ứng:

- Hàng thứ 1 giữ nguyên.
- Hàng thứ 2 dịch vòng trái 1 lần.
- Hàng thứ 3 dịch vòng trái 2 lần.
- Hàng thứ 4 dịch vòng trái 3 lần.



e. Hàm MixColumns

Biến đổi MixColumns() tính toán trên từng cột của state. Các cột được coi như là đa thức trong trường GF(28) và nhân với một đa thức $a(x)$ với:

$$a(x) = (03)x^3 + (01)x^2 + (01)x + (02)$$

Biến đổi này có thể được trình bày như phép nhân một ma trận, mà mỗi byte được hiểu như là một phần tử trong trường GF(28): $s'(x) = a(x) \square s(x)$:

Mô tả bằng ma trận như sau :

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix}$$

1.3. Quá trình giải mã

1.3.1. Tổng quan

Thuật toán giải mã khá giống với thuật toán mã hóa về mặt cấu trúc nhưng 4 hàm sử dụng là 4 hàm ngược của quá trình mã hóa.

Mã Hóa	Giải Mã
AddRoundKey()	InvAddRoundKey()
SubBytes()	InvSubBytes()
ShiftRows()	InvShiftRows()
MixColumns()	InvMixColumns()

1.3.2. Thuật toán giải mã

```

InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin

byte state[4,Nb]
state = in
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 downto 1
    InvShiftRows(state)
    InvSubBytes(state)
    AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    InvMixColumns(state)
end for

InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
out = state
end

```

Trong đó :

- In[] : Mảng dữ liệu đầu vào Input.
- Out[] : Mảng dữ liệu đầu ra Output.
- Nr : Số vòng lặp.(Nr = 10).
- Nb : Số cột(Nb = 4).
- W[] : Mảng các w[i] có độ dài 4 bytes.

1.4. Các dạng tấn công vào AES và phương pháp phòng chống

1.4.1. Side-channel attack

Side Channels (Kênh kề) được định nghĩa là các kênh đầu ra không mong muốn từ một hệ thống.

Tấn công kênh bên hay còn gọi là Tấn công kênh kề là loại tấn công dễ thực hiện trong các loại tấn công mạnh chống lại quá trình triển khai mã hóa, và mục tiêu của loại tấn công này là phân tích các nguyên tố, các giao thức, modul, và các thiết bị trong mỗi hệ thống.

Phân loại :

- Tấn công thời gian.
- Tấn công dựa vào lỗi.
- Tấn công phân tích năng lượng.
- Tấn công phân tích điện từ.

1.4.2. Known attacks

Vào năm 2002, Nicolas Courtois và Josef Pieprzyk phát hiện một tấn công trên lý thuyết gọi là tấn công XSL và chỉ ra điểm yếu tiềm tàng của AES.

Tuy nhiên, một vài chuyên gia về mật mã học khác cũng chỉ ra một số vấn đề trong cơ sở toán học của tấn công này và cho rằng các tác giả đã có sai lầm trong tính toán. Việc tấn công dạng này có thực sự trở thành hiện thực hay không vẫn còn để ngỏ và cho tới nay thì tấn công XSL vẫn chỉ là suy đoán.

1.4.3. Các phương pháp phòng chống

Phương pháp 1: Mã hóa cực mạnh

Sử dụng các biện pháp để tăng tính bảo mật của các thuật toán mã hóa.

Phương pháp 2: Bảo vệ dữ liệu theo phương pháp vật lý

Nếu một kẻ tấn công không thể tiếp cận vật lý với dữ liệu, dĩ nhiên khả năng đánh cắp khóa mã hóa sẽ khó khăn hơn. Vì vậy, trước những cuộc tấn công qua âm thanh tiềm tàng, bạn có thể sử dụng các giải pháp bảo vệ vật lý như đặt laptop vào các hộp cách ly âm thanh, không để ai lại gần máy tính khi đang giải mã dữ liệu hoặc sử dụng các nguồn âm thanh băng rộng tần số đủ cao để gây nhiễu.


Phương pháp 3: Kết hợp cả 2 cách trên.

II. ÁP DỤNG AES VÀO PHẦN MỀM QUẢN LÝ THƯ VIỆN

2.1. Giao diện mô tả mã hóa

Staff Management

Staff Management



Open image

[Go back](#)

ID:

Name:

Gender: ☐ M... ☒ Fe...

Address:

Role: Staff

DOB:

Phone:

Salary:

Search by ☒ Username ☐ ID

ID	Name	DOB	Gender	Address	Phone	roleID	Salary
5	Lap	2000-12-01	Male	Ha Noi city	0915253626	Staff	5000000
6	Loi14	1998-06-18	Male	Ho Chi Minh city	0918564127	Staff	2780000
7	abcde	2020-06-01	Female	1	111	Staff	1112
1	abc	2000-01-02	Female	Ho Chi Minh city	909090909	Admin	10000000
4	Quan	2000-12-01	Male	Ho Chi Minh City	808080808	Admin	5500000
8	Van	2000-07-19	Female	273 An Duong Vuong	914452336	Admin	3500000

Staff sau khi được mã hóa AES:

staffID	staffName	staffdob	staffAddr	staffGender	staffPhone	staff_roleID	staffSalar
1	XJ8hLlEsXQNSOIg2dOMQQ==	2000-01-02	adHo25MeHUqAqO4ZItu0U3hGg9vwuOZQSSZ1...	Female	YObWDIFQnhHBeWg1JFPQ6w==	2	10000000
4	bIBXwt0XOQJcXIPnekJbg==	2000-12-01	cV/m3QJC3rYK80dkX+NMYnhGg9vwuOZQSSZ1...	Male	HINzRJ0v9dJNN53cy+dbYZA==	2	5500000
5	0fo6S9K1OEStz23zuXLOfg==	2000-12-01	QISA+EM4IDxfRwre2U3rog==	Male	hRvdELQsXTZzYZd6/Fg97A==	1	5000000
6	VNjrbcvk3B+u4Y4BXPgEdw==	1998-06-18	adHo25MeHUqAqO4ZItu0U3hGg9vwuOZQSSZ1...	Male	vSRU4eFNoqmEQKywJA1IWA==	1	2780000
7	yjSkDutAcw6t8lg+BHbug==	2020-06-01	PSK+827gv3FO7Z9HEXvHJA==	Female	V80edzEwdAL5nkqu+CsnlQ==	1	1112
8	APufhkaYQRfStsMI0av1ug==	2000-07-19	G4jyd0tZnKrpTWqBosRD8cZt3rHo9KbnZbeBBt0...	Female	FoHhk5WsqITYOWoiwfsqg==	2	3500000
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Book

Menu

Book

ID

1

Author

Ernest Hemingwa

Add

Name

The Old Man and The Fish

Categ...

Literature

Update

Publis...

1952

Price

300000

Refresh

Availa...

269

Remove

Open image

Search ...

☒ Na...
 ☐ ID

Search

ID	Book Name	Author	Category	Publisher	Available	Price
1	The Old Man and The Fish	Ernest Hemin...	Literature	1952	269	300000
2	New Book 2	Alex	Literature	1985	12	800000
4	New Book 7	Ernest Hemin...	Literature	2016	35	175000
5	New Book 3	Ernest Hemin...	Literature	1965	11	73000
6	New Book 4	Ernest Hemin...	Literature	1969	44	126000
7	New Book 4	Patricia High...	Literature	2000	200	99000
8	New Book 77	Patricia High...	Math	2016	7	846000
9	New Book 300	Patricia High...	Physics	1965	47	400000
10	New Book 31	Patricia High...	Math	1965	44	40000
11	New Book 6	Patricia High...	Math	2019	350	400000
12	New Book 3	Alex	Physics	2020	47	11000
21	New Book 312	Ernest Hemin...	Math	2	3	4
13	New Book 311	Patricia High...	Math	1965	23	40000

Book sau khi mã hóa AES:

	bookID	bookName	bookAuthorID	bookCategoryID	bookPublisher	bookPrice	available	bookImg
▶	1	wQHLoO2OO3YBx0U8ohyKexfRn9XQz+5hpsXII...	1	1	1952	300000	269	BLOB
	2	Bbzi4eMHhG4lZCbtxKOdcA==	3	1	1985	800000	12	BLOB
	4	dgQUg9Mc7smc4jd5qx2ZeQ==	1	1	2016	175000	35	BLOB
	5	WVNC+XuNOxf4b1wSIAQxMQ==	1	1	1965	73000	11	BLOB
	6	GE/YK3HENIf4Ks5Ra8INQ==	1	1	1969	126000	44	BLOB
	7	GE/YK3HENIf4Ks5Ra8INQ==	2	1	2000	99000	200	BLOB
	8	3Z35nCMmHeFDy5TJAfgc0A==	2	2	2016	846000	7	BLOB
	9	Z5t428Si5jN5sysSw9d13A==	2	4	1965	400000	47	BLOB

Account

Menu
Account

ID

Start Day

Username

Out of Day

Password

Staff ID

Search... ☒ Na... ☐ ID

ADD

REMOVE

UPDATE

CLEAR

Search

ID	Name	Password	Start Day	Out of Day	Staff ID
1	lap123	e10adc3949...	2020-06-11	2031-08-01	Lap
2	quan123	e10adc3949...	2020-06-11	2031-08-01	Quan
9	vanloi	e10adc3949...	2022-05-11	2026-05-29	Lap
10	vanloi2	e10adc3949...	2022-05-11	2027-05-20	Lap

User Account sau khi được mã hóa AES:

	userID	userName	userPassword	startDay	outofDay	staffID
▶	1	lap123	e10adc3949ba59abbe56e057f20f883e	2020-06-11	2031-08-01	5
	2	quan123	e10adc3949ba59abbe56e057f20f883e	2020-06-11	2031-08-01	4
	9	vanloi	e10adc3949ba59abbe56e057f20f883e	2022-05-11	2026-05-29	5
	10	vanloi2	e10adc3949ba59abbe56e057f20f883e	2022-05-11	2027-05-20	5
✱	NULL	NULL	NULL	NULL	NULL	NULL

2.2. Coding function

GeneratorKey function

```
public static void generatorKey(String mykey) {
    MessageDigest sha=null;
    try {
        key = mykey.getBytes("UTF-8");
        sha = MessageDigest.getInstance("SHA1");
        key = sha.digest(key);
        key = Arrays.copyOf(key, 16);
        secretKeySpec = new SecretKeySpec(key, "AES");
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    catch (UnsupportedEncodingException ex) {
        ex.printStackTrace();
    }
}
```

Encrypt function

```
public static String encrypt(String strToEncrypt)
{
    try
    {
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec);
        return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF-8")));
    }
    catch (Exception e)
    {
        System.out.println("Error while encrypting: " + e.toString());
    }
    return null;
}

public static String decrypt(String strToDecrypt)
{
    try
    {
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
        cipher.init(Cipher.DECRYPT_MODE, secretKeySpec);
        return new String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));
    }
    catch (Exception e)
    {
        System.out.println("Error while decrypting: " + e.toString());
    }
    return null;
}
```

GetMd5


```

public static String getMd5(String input)
{
    try {

        // Static getInstance method is called with hashing MD5
        MessageDigest md = MessageDigest.getInstance("MD5");

        // digest() method is called to calculate message digest
        // of an input digest() return array of byte
        byte[] messageDigest = md.digest(input.getBytes());

        // Convert byte array into signum representation
        BigInteger no = new BigInteger(1, messageDigest);

        // Convert message digest into hex value
        String hashtext = no.toString(16);
        while (hashtext.length() < 32) {
            hashtext = "0" + hashtext;
        }
        return hashtext;
    }

    // For specifying wrong message digest algorithms
    catch (NoSuchAlgorithmException e) {
        throw new RuntimeException(e);
    }
}

```


III. TỔNG KẾT

3.1. Kết quả đạt được

- Khái quát tổng quan về thuật toán mã hóa khóa đối xứng AES và áp dụng nó vào bảo mật dữ liệu phần mềm.
- Hoàn thành mã hóa dữ liệu một số dữ liệu trong hệ thống phần mềm quản lý thư viện.

3.2. Đánh giá thuật toán

- Thiết kế và độ dài khóa của thuật toán AES (128, 192 và 256 bit) là đủ an toàn để bảo vệ các thông tin được xếp vào loại tối mật nhưng về an ninh của AES thì các nhà khoa học đánh giá là chưa cao. Nếu các kỹ thuật tấn công được cải thiện thì AES có thể bị phá vỡ.
- Một vấn đề khác nữa là cấu trúc toán học của AES khá đơn giản.

TÀI LIỆU THAM KHẢO

Bibliography

- [1] H. N. K. Huy. [Online]. Available:
<https://classroom.google.com/u/0/c/NDY2OTY2ODM4MjM0>.
- [2] B. H. Hoat, "Viblo," [Online]. Available: <https://viblo.asia/p/tim-hieu-thuat-toan-ma-hoa-khoa-doi-xung-aes-gAm5yxOqlDb>.
- [3] L. Phúc, "Bài giảng," in *Bảo mật hệ thống thông tin*.