

Báo cáo Đồ án 01

Hệ thống lưu trữ file an toàn (mức độ sơ khai)

Nhóm: navi no1

Ngày 27 tháng 3 năm 2024

Mục lục

1	Thông tin nhóm	3
2	Thông tin kỹ thuật	3
3	Giới thiệu về Protocol	3
3.1	Đăng ký	3
3.2	Đăng nhập	3
3.3	Upload, Download & Chia sẻ ảnh	4
3.3.1	Upload ảnh	4
3.3.2	Download ảnh	5
3.3.3	Chia sẻ ảnh	6
3.4	API	6
4	Vận dụng	7
4.1	Các thuật toán sử dụng	7
4.1.1	RSA	7
4.1.2	Thuật toán Euclide mở rộng	7
4.1.3	Tìm modulo nghịch đảo	7
4.1.4	Thuật toán sinh khóa RSA ngẫu nhiên	8
4.1.5	AES - mã hóa và giải mã file ảnh	8
4.2	Cài đặt & chạy thử	9
4.2.1	Server	9
4.2.2	Client	10
4.2.3	Hướng dẫn sử dụng	10
4.3	Demo	12

5	Kết luận	13
5.1	Ưu điểm:	13
5.2	Nhược điểm	13
5.3	Cải tiến	13

1 Thông tin nhóm

MSSV	Họ và tên
19120064	Nguyễn Hồ Hoàng Duy
19120179	Võ Trương Trung Chánh
19120266	Nguyễn Hoàng Anh Kiệt
19120338	Trần Hoàng Quân

2 Thông tin kỹ thuật

Đồ án được thực hiện trên ngôn ngữ Python, sử dụng một số thư viện hỗ trợ sau:

- Flask Framework và Flask RESTful: xây dựng RESTful API với Flask webframework.
- Thư viện AES mã nguồn mở của PyCryptodome.
- Thư viện secrets: tạo ngẫu nhiên các chuỗi với độ dài xác định.
- Thư viện hashlib : hỗ trợ các thuật toán SHA256, MD5, ..
- Thư viện PIL, OpenCV: chuyển đổi định dạng ảnh và thao tác với ảnh.

Các yêu cầu kỹ thuật để cài đặt và chạy thử sẽ được miêu tả rõ hơn ở phần [4.2](#)

3 Giới thiệu về Protocol

3.1 Đăng ký

Quá trình đăng ký được miêu tả như sau:

1. User nhập tên của mình vào client.
2. Client tiến hành tạo ngẫu nhiên cặp khóa RSA e, d .
3. User chọn cặp khóa tùy ý, rồi submit lên server để đăng ký.

Ưu điểm của cách làm này là đảm bảo được user là người sở hữu khóa e . Sau đó user tiến hành ghi nhớ ID và khóa d , đây sẽ lần lượt là ID đăng nhập và mật khẩu đăng nhập cho tài khoản.

3.2 Đăng nhập

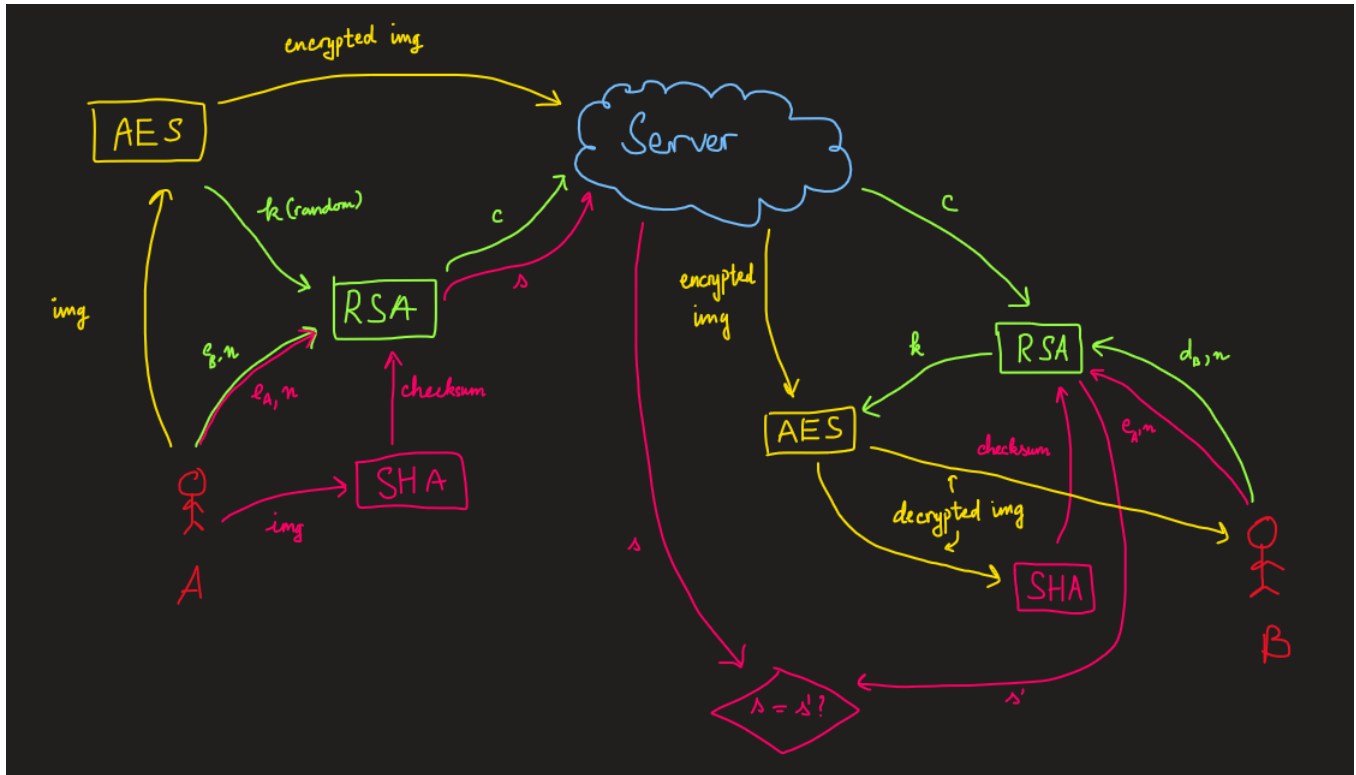
Quá trình đăng nhập được miêu tả như sau:

1. User tiến hành nhập ID và mật khẩu d .
2. Client sẽ gửi yêu cầu đăng nhập tới server, kèm theo ID của user.

3. Server sẽ tạo một token ngẫu nhiên¹ và lưu 1 bản sao trong database, sau đó encrypt token bằng public key e của user và gửi trả token đã được encrypt về client.
4. Client sẽ nhận token và dùng khóa d để giải mã token bằng thuật toán RSA, sau đó gửi token ngược lại server.
5. Nếu token gửi lên giống với token trong database, server sẽ cấp cho user một API token sử dụng trong suốt session của mình.

Cần lưu ý là khóa bí mật d không bao giờ được gửi đi khỏi client. Như vậy, giả sử người dùng có bị lộ khóa công khai e thì cũng không ảnh hưởng đến quá trình đăng nhập; ngược lại, giả sử khóa e bị kẻ tấn công thay thế từ pha đăng ký, khi đó tài khoản sẽ không thể đăng nhập từ lần đầu tiên - hạn chế khả năng bị đánh tráo khóa công khai.

3.3 Upload, Download & Chia sẻ ảnh



Hình 1: Ảnh minh họa các giao thức upload ảnh và download ảnh

3.3.1 Upload ảnh

Giao thức upload ảnh được miêu tả như sau: Giả sử A là người upload ảnh $a.jpg$, A có khóa công khai e_A

¹bằng `secrets.token_hex`

1. Client sinh khóa ngẫu nhiên k .
2. Client dùng khóa k mã hóa ảnh $a.jpg$ bằng thuật toán AES.
3. Client dùng khóa công khai e_A mã hóa k thành c .
4. Client gửi ảnh đã mã hóa và c lên server.

Trong giao thức này có một bước nữa: tạo checksum để kiểm tra tính toàn vẹn của ảnh:

1. Client hash ảnh bằng thuật toán SHA256 thành một chuỗi checksum h .
2. Client tiến hành mã hóa h bằng thuật toán RSA với khóa công khai e_A , tạo chữ ký s .
3. Client gửi chữ ký s lên server.

3.3.2 Download ảnh

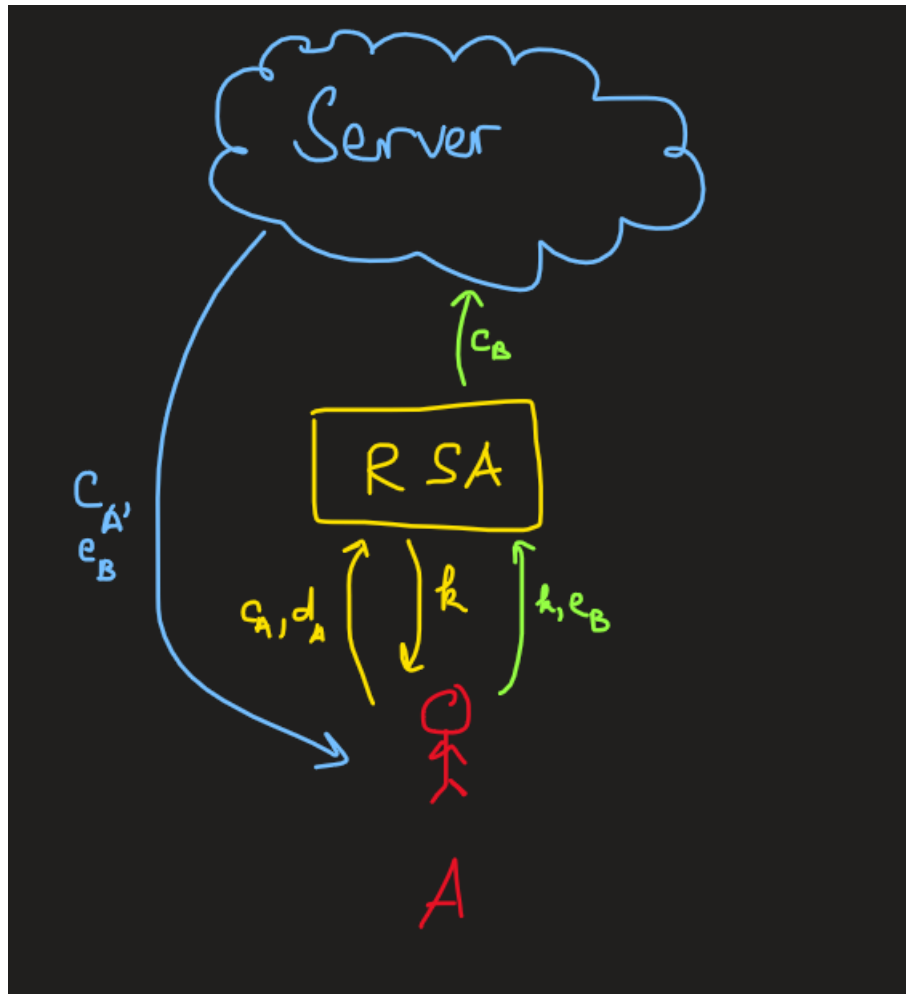
Giao thức download ảnh được miêu tả như sau: giả sử A có quyền download ảnh (được chia sẻ ảnh) và có khóa bí mật d_A ; B là người upload ảnh có khóa công khai là e_B ; Ảnh được upload lên server có chữ ký s .

1. Client download ảnh đã được mã hóa từ server.
2. Client yêu cầu lấy khóa c từ server.
3. Client dùng khóa bí mật d_A để giải mã c trở thành k bằng thuật toán RSA.
4. Client dùng k để giải mã ảnh đã được mã hóa.

Giao thức này còn một bước, là kiểm tra checksum của ảnh:

1. Client băm ảnh vừa được giải mã bên trên bằng thuật toán SHA256, tạo chuỗi checksum h .
2. Client dùng khóa công khai e_B của B mã hóa h thành chữ ký s' , sử dụng thuật toán RSA.
3. Client sau đó so sánh s' với s , nếu 2 chuỗi này trùng khớp thì ảnh được download nguyên vẹn, giống với ảnh gốc mà B đã upload.

3.3.3 Chia sẻ ảnh



Hình 2: Ảnh minh họa giao thức chia sẻ ảnh

Giao thức chia sẻ ảnh được miêu tả như sau: giả sử A là **chủ sở hữu của ảnh** muốn chia sẻ ảnh $x.jpg$ cho B

- A yêu cầu c_A, e_B lần lượt là khóa mã ảnh $x.jpg$ được mã hóa bởi A và khóa công khai của B .
- A tiến hành giải mã khóa c_A thành k bằng khóa bí mật d_A mình đang giữ, sử dụng thuật toán RSA.
- A mã hóa k thành c_B bằng khóa công khai e_B của B , sau đó gửi c_B trở lại server.

Khi đó B chỉ cần yêu cầu download ảnh từ server và đã có khóa c_B để giải mã.

3.4 API

Tài liệu API xem ở link này: <https://documenter.getpostman.com/view/18981203/UVRHiiVn>

4 Vận dụng

4.1 Các thuật toán sử dụng

4.1.1 RSA

Nhóm cài đặt lại RSA protocol theo nội dung được giảng dạy trên lớp.:

1. Chọn p, q là 2 số nguyên tố lớn.
2. Tính $n = pq, \phi = (p - 1)(q - 1)$
3. Chọn d, e sao cho $e \equiv d^{-1} \pmod{\phi}$
4. Bản mã $c \equiv m^e \pmod{n}$
5. Bản rõ $m \equiv c^d \pmod{n}$

Các phép nhân, phép mũ sử dụng thuật toán nhân nhanh và mũ nhanh theo modulo.

4.1.2 Thuật toán Euclide mở rộng

Dựa trên bổ đề Bézout: với hai số nguyên không âm a và b , $g = (a, b)$ thì:

- Tồn tại hai số nguyên x, y sao cho $ax + by = g$
- g là số nguyên nhỏ nhất có thể viết dưới dạng $ax + by$
- Mỗi số e có dạng $ax + by$ đều là bội của d .

Thuật toán nhận vào a, b , đưa ra $x, y; g = (a, b)$ sao cho $ax + by = g$

Algorithm 1 Thuật toán Euclide mở rộng (Extended Euclidean Algorithm)

```

function XEUCLIDEAN( $a, b$ )
  if  $a = 0$  then return ( $b, 0, 1$ )
  else
     $(g, y, x) \leftarrow XEuclidean(b \% a, a)$ 
     $x \leftarrow x - \left\lfloor \frac{b}{a} \right\rfloor \times y$ 
  return ( $g, x, y$ )
  end if
end function

```

4.1.3 Tìm modulo nghịch đảo

Vẫn dựa trên bổ đề Bézout bên trên: giả sử ta có $ax + by = g \iff ax \equiv g \pmod{y}$. Khi đó x là modulo nghịch đảo của a khi $g = 1$.

Algorithm 2 Thuật toán tìm modulo nghịch đảo

```

function INVERSE_MODULO(a, n)
     $(g, x) \leftarrow XEuclidean(a, n)$ 
    if  $g \neq 1$  then return 'DnE'
    else
        return  $x \% n$ 
    end if
end function
    
```

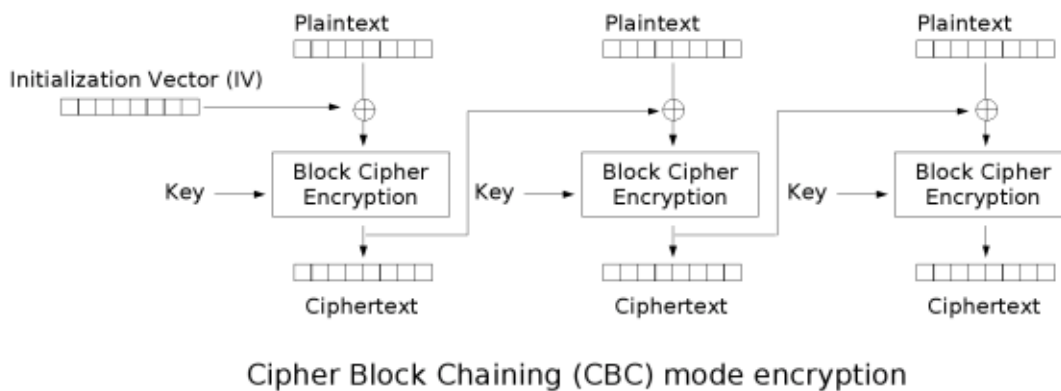
4.1.4 Thuật toán sinh khóa RSA ngẫu nhiên

Sinh khóa RSA sử dụng 2 thuật toán chính:

- Sinh số nguyên tố lớn (sử dụng để tạo p, q)
- Kiểm tra số nguyên tố

Để sinh d, e ta chỉ cần sinh một số ngẫu nhiên trong $(1, \phi]$ và tìm khả nghịch của số đó trong ϕ .

4.1.5 AES - mã hóa và giải mã file ảnh



Hình 3: Tóm tắt mode CBC.[2] Thuật toán block encryption sử dụng ở đây là AES.

Nhóm sử dụng thuật toán AES trong thư viện PyCryptodome với mode CBC. Với cách làm này ảnh vẫn sẽ giữ nguyên format nhưng nội dung được làm nhiễu. Quá trình mã hóa:

- Vì ta muốn mã hóa nhiều format ảnh khác nhau, nhóm ưu tiên chuyển các ảnh về định dạng PNG bằng thư viện PIL. Việc chuyển format về dạng PNG giúp đảm bảo chất lượng của ảnh.
- Khởi tạo khóa k là một chuỗi ngẫu nhiên, ở đây nhóm chọn độ dài là 16 bytes. Khởi tạo block $iv = 0000000000000000$ cũng có cùng độ dài.

- Tiến hành thêm padding cho file để file có kích thước là bội của 16. Byte cuối cùng của phần padding sẽ lưu số lượng dòng padding đã thêm. Chú ý là nếu file đã có kích thước là bội của 16, ta vẫn tiến hành thêm 16 bytes. Lúc này byte cuối cùng sẽ lưu số 16.²
- Tiến hành mã hóa với AES và lưu vào file có định dạng PNG.

Với quá trình giải mã:

- Từ file và key ban đầu, dùng thuật toán AES để giải mã và cho vào một ma trận.
- Phần tử cuối của ma trận lưu số dòng padding. Xóa các dòng padding đi.
- Lưu ma trận trên vào file với định dạng gốc, sử dụng OpenCV.

Cách làm này được tham khảo từ câu trả lời trên StackOverflow của tác giả Rotem^[1]

4.2 Cài đặt & chạy thử

4.2.1 Server

Sử dụng Terminal (Linux) hoặc Command Line (Windows)

1. Tạo virtual environment cho server:

Browse đến thư mục **server**, sau đó:

- Đối với Linux:
`python3 -m venv venv`
- Đối với Windows:
`python -m venv venv`

2. Kích hoạt virtual environment

- Đối với Linux:
`. venv/bin/activate`
- Đối với Windows:
`venv\Scripts\activate`

3. Cài các package cần thiết:

```
pip install -r requirements.txt
```

²Nếu không thì trong quá trình giải mã, decryptor sẽ nhận "nhầm" byte cuối lưu 1 điểm ảnh RGB là số dòng cần xóa!

4. Tạo mới database

- Đối với Linux:

```
export FLASK_APP=serverside
export FLASK_ENV=development
flask init -db
```

- Đối với Windows:

```
set FLASK_APP=serverside
set FLASK_ENV=development
flask init -db
```

5. Chạy server

```
flask run
```

4.2.2 Client

1. Trong folder `client`, chạy câu lệnh sau để cài các package cần thiết:

```
pip install -r requirements.txt
```

2. Chạy file `main.py` để khởi động client.

```
python main.py
```

4.2.3 Hướng dẫn sử dụng

Sau khi khởi động cả server và client, nhập IP của server và connection port để kết nối với server. Vì đây là một console application, chủ yếu người dùng sẽ nhập từ bàn phím để chọn các menu item / nhập liệu.

Đăng kí

- Người dùng chọn menu Register new account.
- Người dùng nhập tên.
- Hệ thống sẽ random ra cặp khóa e và d ngẫu nhiên. Nếu đồng ý thì bấm y , ngược lại bấm n sẽ tạo khóa mới.
- Nếu chọn khóa xong, người dùng sẽ thấy hiển thị thông báo đăng kí thành công.

Đăng nhập

- Người dùng chọn menu Login.
- Người dùng nhập ID.
- Người dùng nhập khóa bí mật và nhấn Enter.
- Thông báo đăng nhập thành công, người dùng sẽ được chuyển hướng đến menu chính.



```

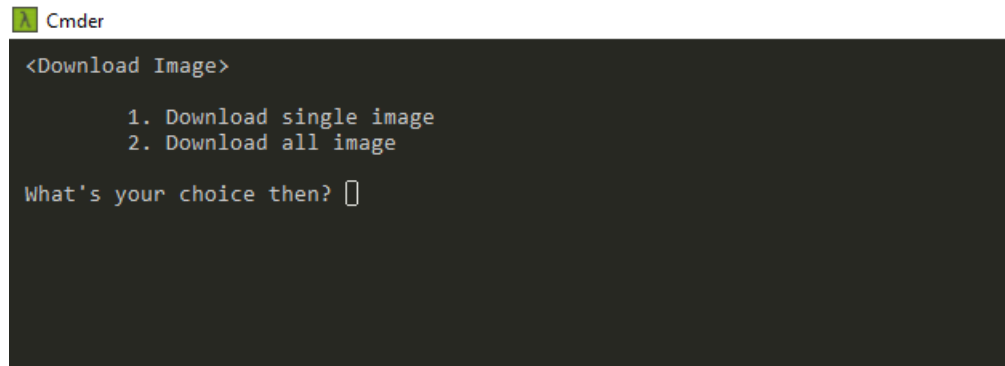
Cmder
Hello Tran Hoang Quan
Your current choice is:
  1. Log out
  2. Upload a new image to server
  3. Get image list from server
  4. Download image from server
  5. Share image for another user
Your choice: 
    
```

Hình 4: Menu chính của client

Đăng xuất Trong menu chính, người dùng chọn Logout.

Upload ảnh

- Trong menu chính, người dùng chọn Upload a new image to server.
- Người dùng nhập đường dẫn đến file ảnh. Chú ý: Hệ thống chỉ hỗ trợ các extension .png, .jpg/jpeg, .bmp.
- Nhấn enter, người dùng đã upload ảnh thành công.



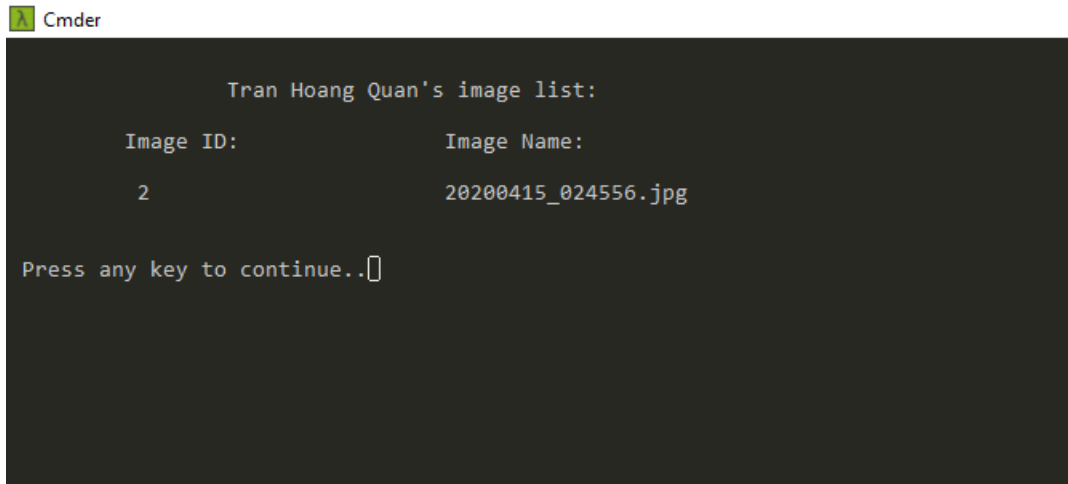
```

Cmder
<Download Image>
  1. Download single image
  2. Download all image
What's your choice then? 
    
```

Hình 5: Menu download

Download ảnh

- Trong menu chính, người dùng chọn Download image from server.
- Có 2 lựa chọn: download tất cả hoặc download 1 ảnh.
- Nếu download tất cả: hệ thống sẽ download tất cả những ảnh bạn đã upload / được chia sẻ.
- Nếu download 1 ảnh: nhập ID ảnh và download.



Hình 6: Danh sách ảnh của một User

Xem danh sách ảnh

- Trong menu chính, người dùng chọn Get image list from server.
- Tất cả các ảnh của user (upload, được share) sẽ được liệt kê ở đây (ID - tên ảnh)

Chia sẻ ảnh

- Trong menu chính, người dùng chọn Share image for another user.
- Người dùng nhập ID người cần share, ảnh cần share. Chú ý: người dùng phải là chủ (người upload) ảnh mới có thể share ảnh.

Chi tiết hướng dẫn sử dụng, mời bạn xem Demo.

4.3 Demo

Video demo quá trình cài đặt & chạy thử ở link này. <https://youtu.be/vQ0MhBgJ23A>

5 Kết luận

5.1 Ưu điểm:

- Hệ thống có thể chia sẻ hình ảnh một cách bảo mật, không làm lộ thông tin cho bên thứ 3.
- File được lưu trữ nguyên vẹn, không bị tổn hại.
- Hệ thống đảm bảo chỉ những user có quyền (được chia sẻ) mới có thể xem ảnh; user có quyền (chủ sở hữu) mới có thể chia sẻ ảnh.

5.2 Nhược điểm

Vì đây là hệ thống ở mức độ sơ khai và không có hệ thống bảo mật nào là an toàn tuyệt đối, nên còn nhiều khuyết điểm cần chỉ rõ:

- Hệ thống vẫn có thể làm lộ access_token qua quá trình truyền tải nếu không có SSL encryption.
- Vì hệ thống chỉ hỗ trợ đăng nhập bằng khóa bí mật d , nên user chưa thể thay đổi password.
- Hệ thống chưa hỗ trợ tính năng xóa ảnh.
- Nếu nhiều user sử dụng cùng 1 máy thì rất dễ lộ folder Download. Hiện tại hệ thống chỉ hỗ trợ 1 user / máy.

5.3 Cải tiến

- Có thể thay thế CBC mode bằng một số mode khác phức tạp hơn để đảm bảo độ an toàn cho ảnh đã encrypt.
- Có thể tăng độ dài key RSA và AES để đảm bảo an toàn.

References

- [1] Rotem (<https://stackoverflow.com/users/4926757/rotem>). *Create jpg/png from encrypted image*. Stack Overflow. URL:<https://stackoverflow.com/a/68060392> (version: 2022-01-10). URL: <https://stackoverflow.com/a/68060392>.
- [2] Wikimedia Commons, the free media repository. *File:Cbc encryption.png*. [Online; accessed January 10, 2022]. 2006. URL: https://commons.wikimedia.org/wiki/File:Cbc_encryption.png.