TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Bộ môn: An toàn và phục hồi dữ liệu

Giảng viên: Thái Hùng Văn

Đặng Trần Minh Hậu

Nhóm thực hiện:

20127308: Phan Minh Sáng

20127318: Phan Trí Tài

20127569: Tô Đình Phương Nam

Hồ Chí Minh, ngày 31 tháng 12 năm 2023

Bảng phân công công việc

STT	Công việc	Phụ trách
1	Tổ chức file	Cả nhóm
2	Tổng hợp các tiêu chí	Cả nhóm
3	Xóa phần tử có khả năng phục hồi ngoại trừ tình huống đặc biệt cần phải xóa hẳn, các phần tử đã xóa quá lâu cũng không cần phải phục hồi).	
	Tổ chức minh họa thông tin cơ bản (Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD,…), trong đó số CCCD và số ĐT cần bảo mật.	
	Thông tin quản lý cần thiết, như ngày tạo lập, key mã hóa /giải mã, …	
	Cơ chế kiểm tra mật khẩu động hoặc passkey mỗi khi file được mở, có khống chế thời gian (nhập sai nhiều lần thì phải đợi một thời gian sau mới có thể nhập tiếp, vẫn sai nữa thì thời gian đợi bị tăng thêm)	
	Xây dựng chương trình demo minh họa với ít nhất các thao tác tạo /thêm /xóa /sửa phần tử trên cả 2 DS, liệt kê 1 đoạn trong DS, đổi mật khẩu /cơ chế mã hóa.	

Đánh giá mức độ: 100%

Tổ chức file

Bảng quản lý trạng thái	Các thông tin tạo lập	**Kích thước mảng GV**	**Dữ liệu mảng GV**	**Kích thước mảng dự trữ**	**Dữ liệu mảng dự trữ**	**Kích thước mảng SV**	**Dữ liệu mảng SV**

- Bảng quản lý trạng thái cảu các phần tử trong mảng: trạng thái của các phần tử (0 – tương ứng với phần tử chưa được sử dụng hoặc để xóa vĩnh viễn, 1- tương ứng với phần tử đang được sử dụng, 2 – tương ứng với phần tử đang bị xóa tạm thời)
- Các thông tin tạo lập: key mã hóa/giải mã và ngày tạo lập
- Kích thước mảng giáo viên (GV)

- Dữ liệu mảng giáo viên (GV)
- Kích thước mảng dự trữ
- Dữ liệu mảng dự trữ
- Kích thước mảng sinh viên (SV)
- Dữ liệu mảng sinh viên (SV)

Tổng hợp tiêu chí

1. Xóa phần tử có khả năng phục hồi ngoại trừ tình huống đặc biệt cần phải xóa hẳn, các phần tử đã xóa quá lâu cũng không cần phải phục hồi).
    - Xóa phần tử có khả năng phục hồi lại: Trạng thái của phần tử trong bảng quản lý trạng thái được chuyển từ 1 thành 2.
    - Xóa hẳn phần tử: Trạng thái của phần tử trong bảng quản lý trạng thái được chuyển từ 1 thành 0. Tạm thời ghi đè lại dữ liệu trong mảng.

2. Tổ chức minh họa thông tin cơ bản (Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD, …), trong đó số CCCD và số ĐT cần bảo mật.
    - Mỗi thành phần trong mảng sẽ có các thông tin ID, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD.
    - CCCD và số điện thoại được mã hóa bằng thuật toán AES.

3. Thông tin quản lý cần thiết, như ngày tạo lập, key mã hóa /giải mã, …
    - Ngày tạo lập: ngày tháng năm tạo ra file.
    - Key: được sinh ra cho thuật toán AES.

4. Cơ chế kiểm tra mật khẩu động hoặc passkey mỗi khi file được mở, có khống chế thời gian (nhập sai nhiều lần thì phải đợi một thời gian sau mới có thể nhập tiếp, vẫn sai nữa thì thời gian đợi bị tăng thêm)
    - Khi khởi động chương trình thì mật khẩu động được gửi tới mail, mỗi lần nhập sai thì thời gian đợi sẽ tăng thêm **timeWait ^ 2 + 30.**

5. Xây dựng chương trình demo minh họa với ít nhất các thao tác tạo /thêm /xóa /sửa phần tử trên cả 2 DS, liệt kê 1 đoạn trong DS, đổi mật khẩu /cơ chế mã hóa.
    - Menu cho phép lựa chọn các mục sau:
        - 1. Liệt Kê
        - 2. Thêm
        - 3. Xóa
        - 4. Phục Hồi Dữ Liệu
        - 5. Sửa

DEMO

Link demo: https://youtu.be/5vBQWk4EYI8

1. **Xóa phần tử có khả năng phục hồi ngoại trừ tình huống đặc biệt cần phải xóa hẳn, các phần tử đã xóa quá lâu cũng không cần phải phục hồi)**

```
# Hàm liệt kê danh sách người
def list_People(file_name):
    choose = input(
        "Bạn muốn liệt kê danh sách người (0 - Tất cả, 1 - Giáo viên, 2 - Sinh viên): "
    )

    (
        GV_read,
        SV_read,
        state_read,
        GV_size_read,
        reserve_size_read,
        SV_size_read,
        K_read,
    ) = readFile(file_name)
```

- Bước 1: Yêu cầu người dùng lưa chọn in ra tất cả giáo viên hoặc học sinh hoặc cả 2.
- Bước 2: Đọc các thông tin cần thiết từ file.
- Bước 3: Tùy vào lựa chọn của người dùng ở bước 1 mà in duyệt qua mảng tương ứng. In ra các phần tử có trạng thái là 1 (đang được sử dụng).

2. **Tổ chức minh họa thông tin cơ bản (Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD, …), trong đó số CCCD và số ĐT cần bảo mật**

```
# Class Person
class Person:
    # Mã, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD,…
    def __init__(self, id, name, birthday, join_date, phone_number, cccd):
        self.id = id
        self.name = name
        self.birthday = birthday
        self.join_date = join_date
        self.phone_number = phone_number
        self.cccd = cccd
```

```
# mã hóa số điện thoại và CCCD
phone_number_bytes = (
    encrypt_AES(K, People.phone_number.encode("utf-8")).hex().encode("utf-8")
    + b"\0"
)
file.write(phone_number_bytes)

cccd_bytes = (
    encrypt_AES(K, People.cccd.encode("utf-8")).hex().encode("utf-8") + b"\0"
)
file.write(cccd_bytes)
```

- Mỗi thành phần trong mảng sẽ có các thông tin ID, Họ Tên, Ngày sinh, Ngày tham gia, Số ĐT, Số CCCD.
- Chương trình sẽ tự sinh ra key mã hóa cho thuật toán AES, sau đó mã hóa số điện thoại và CCCD để lưu vào file.

3. Thông tin quản lý cần thiết, như ngày tạo lập, key mã hóa /giải mã, …

```
# Ghi Key
K = generate_AES_key()
file.write(K)

# Ghi ngày tạo lập
d = date.today()

year = str(d.year)
month = str(d.month)
day = str(d.day)
```

- Sử dụng hàm **os.urandom(32)** để sinh ra key cho thuật toán AES.
- Sử dụng thư viện **datetime** để lấy được ngày tháng năm.
- Ghi các thông tin trên vào file.

4. Cơ chế kiểm tra mật khẩu động hoặc passkey mỗi khi file được mở, có khống chế thời gian (nhập sai nhiều lần thì phải đợi một thời gian sau mới có thể nhập tiếp, vẫn sai nữa thì thời gian đợi bị tăng thêm)

```
    digits = "0123456789"
    OTP = ""
    for i in range(6):
        OTP += digits[math.floor(random.random() * 10)]
    return OTP
```

```python
# Kết nối đến máy chủ SMTP và gửi email
try:
    with smtplib.SMTP(smtp_server, smtp_port) as server:
        server.starttls()  # Kích hoạt chế độ TLS
        server.login(email_address, email_password)
        server.send_message(message)
    print("Email sent successfully!")
except Exception as e:
    print("Error:", str(e))
```

```python
# Người nhận và nội dung email
to_address = "phanminhsang147@gmail.com"
subject = "OTP"
```

- Lưu ý: cần thay đổi email người nhận để nhận được email OTP (Hình 3)
- Bước 1: Sinh ra 6 số ngẫu nhiên để làm OTP (Hình 1)
- Bước 2: Kết nối với SMTP server của Google để gửi mail tới người nhận. Mỗi lần nhập sai thì thời gian đợi sẽ tăng thêm **timeWait ^ 2 + 30.**

5. Xây dựng chương trình demo minh họa với ít nhất các thao tác tạo /thêm /xóa /sửa phần tử trên cả 2 DS, liệt kê 1 đoạn trong DS, đổi mật khẩu /cơ chế mã hóa

```python
writeBinary("test.dat", GV, SV, 3, 4, 5)
ans = True
while ans:
    os.system("cls")
    print("1. Liệt Kê")
    print("2. Thêm")
    print("3. Xóa")
    print("4. Phục Hồi Dữ Liệu")
    print("5. Sửa")
    print("0. Thoát")
    ans = input("Tác vụ muốn thực hiện ")
    os.system("cls")
```

```
giao_vien_1 = Person(
    "GV001", "Nguyen Van A", "01/01/1980", "01/01/2010", "123456789", "123456789012"
)
giao_vien_2 = Person(
    "GV002", "Tran Thi B", "05/05/1985", "01/01/2015", "987654321", "987654321012"
)
giao_vien_3 = Person(
    "GV003", "Nguyen Van C", "10/10/1990", "01/01/2018", "111222333", "111222333444"
)
# Tạo đối tượng Sinh viên
sinh_vien_1 = Person(
    "SV001", "Nguyen Van C", "10/10/1995", "01/01/2018", "111222333", "111222333444"
)
sinh_vien_2 = Person(
    "SV002", "Tran Thi D", "15/03/1998", "01/01/2019", "222333444", "222333444555"
)
```

- Demo sẽ được cung cấp những dữ liệu có sẵn (Hình 2).
- Người dùng sẽ được lựa chọn các mục sau từ Menu:
  - 1. Liệt Kê
  - 2. Thêm
  - 3. Xóa
  - 4. Phục Hồi Dữ Liệu
  - 5. Sửa

Tài liệu tham khảo

- Thư viện để lấy ngày tháng năm: https://docs.python.org/3/library/datetime.html
- Thư viện dùng để tạo thuật toán AES: https://pycryptodome.readthedocs.io/en/latest/src/cipher/aes.html