Nguyễn Thị Phương Anh -22174600085
Bài TH2

```python
import hashlib

# Mật khẩu gốc (đã băm trước đó) - ví dụ: "mypassword"
stored_password = hashlib.sha256(b"mypassword").hexdigest()

# Nhập mật khẩu từ người dùng
password = input("Nhập mật khẩu: ")

# Băm mật khẩu người dùng nhập
hashed_password = hashlib.sha256(password.encode()).hexdigest()

# So sánh mật khẩu băm với mật khẩu đã lưu
if hashed_password == stored_password:
    print(" Xác thực thành công!")
else:
    print(" Xác thực thất bại!")
```

```
Nhập mật khẩu: 12345678
 Xác thực thất bại!
```

```python
import pyotp
import time

# Tạo khóa bí mật và mã OTP
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)

print("Khóa bí mật (dùng để cài ứng dụng Google Authenticator):", secret)
print("Mã OTP của bạn là:", totp.now())

# Yêu cầu người dùng nhập mã OTP
otp_input = input("Nhập mã OTP: ")

# Xác thực mã OTP
if totp.verify(otp_input):
    print("Xác thực thành công!")
else:
    print(" Xác thực thất bại!")
```

```
Khóa bí mật (dùng để cài ứng dụng Google Authenticator): 2D5QT656DU7ISEUHUNNIKMZNVISSXNAT
Mã OTP của bạn là: 407181
Nhập mã OTP: 407181
 Xác thực thất bại!
```

```python
import hashlib
import pyotp
import time

# Bước 1: Xác thực bằng mật khẩu
stored_password = hashlib.sha256(b"mypassword").hexdigest()  # Mật khẩu lưu dưới dạng SHA-256 hash

password = input("Nhập mật khẩu: ")
hashed_password = hashlib.sha256(password.encode()).hexdigest()

if hashed_password == stored_password:
    print(" Xác thực mật khẩu thành công! Chuyển sang bước xác thực bằng mã OTP.")
else:
    print(" Xác thực mật khẩu thất bại!")
    exit()  # Thoát nếu sai mật khẩu

# Bước 2: Xác thực bằng mã OTP
# Tạo khóa bí mật
secret = pyotp.random_base32()
totp = pyotp.TOTP(secret)

# In mã OTP (trong thực tế sẽ gửi qua SMS hoặc email)
print("Mã OTP của bạn là:", totp.now())

# Yêu cầu người dùng nhập mã OTP
otp_input = input("Nhập mã OTP: ")

# Xác thực OTP
if totp.verify(otp_input):
    print(" Xác thực hai yếu tố thành công!")
else:
    print(" Xác thực bước 2 (OTP) thất bại!")
```

```
Nhập mật khẩu: 12345678
 Xác thực mật khẩu thất bại!
Mã OTP của bạn là: 556757
Nhập mã OTP: 556757
 Xác thực bước 2 (OTP) thất bại!
```

1. Tại sao xác thực hai yếu tố (2FA) lại an toàn hơn so với xác thực chỉ bằng mật khẩu?

Vì sao 2FA an toàn hơn?

Mật khẩu có thể bị đánh cắp

Do bị lộ, đoán ra, keylogger, phishing...

Nếu chỉ dùng mật khẩu → hacker đăng nhập được ngay.

2FA thêm một lớp bảo vệ khác

Ví dụ: mã OTP, xác nhận từ app, vân tay, v.v.

→ Kẻ tấn công phải có cả mật khẩu và yếu tố thứ hai mới truy cập được.

2. Có thể cải tiến thêm tính năng bảo mật nào cho chương trình này không?

Mã hóa khóa riêng RSA: Không lưu dưới dạng rõ ràng, dùng passphrase hoặc thiết bị bảo mật (HSM/TPM).

Xác minh khóa AES sau khi giải mã: Dùng HMAC hoặc checksum để kiểm tra tính toàn vẹn.

Tăng độ dài khóa RSA: Dùng 3072 hoặc 4096 bit thay vì 2048 nếu cần bảo mật cao hơn.

Dùng padding an toàn: Tiếp tục dùng PKCS1_OAEP (đã tốt), không dùng kiểu cũ.

Không in khóa AES ra console: Tránh rò rỉ thông tin nhạy cảm.

Mã hóa dữ liệu bằng AES: Không chỉ mã hóa khóa AES, mà dùng nó để mã hóa dữ liệu thực tế.

Dùng AES-GCM thay cho AES thường: Bảo vệ cả dữ liệu và tính toàn vẹn.

Anh/Chị rút ra được bài học gì về tính bảo mật của mật khẩu và mã OTP?

1. Mật khẩu không đủ mạnh và dễ bị đoán

Mật khẩu "12345678" quá đơn giản, dễ đoán → không an toàn.

Nếu chỉ dùng mật khẩu, hệ thống rất dễ bị tấn công (brute-force, phishing...).

2. OTP chỉ có hiệu lực trong thời gian ngắn

Mã OTP có hạn sử dụng ngắn (thường 30 giây).

Nếu nhập sai thời điểm hoặc sai mã → xác thực thất bại.

Điều này giúp tăng tính bảo mật, nhưng cũng yêu cầu người dùng thao tác nhanh và chính xác.

3. Cả hai yếu tố phải đúng để truy cập
Dù nhập đúng OTP, nếu sai mật khẩu → thất bại.

Dù đúng mật khẩu, nhưng sai OTP (hoặc đã hết hạn) → cũng thất bại.

Điều này cho thấy: 2FA buộc phải qua cả 2 bước bảo vệ, giúp giảm nguy cơ bị xâm nhập.