Họ và tên: Hoàng Sỹ Việt
Mã sinh viên: 22174600094
Lớp: DHKL16A1HN

B1: Mã hóa và giải mã bằng AES(Mã hóa đối xứng)

Viết chương trình mã hóa một đoạn văn bản bằng thuật toán AES với khóa 128-bit và giải mã để kiểm tra tính chính xác. Đồng thời, đo thời gian thực thi của quá trình mã hóa và giải mã AES

```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from Crypto.Util.Padding import pad, unpad
import time

# Tạo khóa mã hóa 128-bit và khởi tạo AES
key = get_random_bytes(16)
cipher = AES.new(key, AES.MODE_CBC)

plaintext = b"Hello, this is a test message for AES encryption!"

# Đo thời gian mã hóa AES
start_time = time.time()
ciphertext = cipher.encrypt(pad(plaintext, AES.block_size))
end_time = time.time()
aes_encryption_time = end_time - start_time

print("Văn bản mã hóa (AES):", ciphertext)
print("Thời gian mã hóa AES:", aes_encryption_time, "giây")

# Giải mã và đo thời gian giải mã AES
start_time = time.time()
decipher = AES.new(key, AES.MODE_CBC, cipher.iv)
decrypted_text = unpad(decipher.decrypt(ciphertext), AES.block_size)
end_time = time.time()
aes_decryption_time = end_time - start_time

print("Văn bản giải mã (AES):", decrypted_text.decode())
print("Thời gian giải mã AES:", aes_decryption_time, "giây")
```

[3] Python

```
Văn bản mã hóa (AES): b'\x15\xd3\x1a\x1d\xb9\xd3k?\xbd\xde\'\xf55\x08u,\x02\x19\x9c8\xbcJm\xa7\x8f\x02\xb5o\x00"^f\xde\xe3\x96\x92\xa8K\xa7\xe3\x9d\x90*r\xeef\x9b\xb2\xcd4\xdd\x9d\xdcC\xceBc2\xeb\xe1@\x
Thời gian mã hóa AES: 0.00027728080749511172 giây
Văn bản giải mã (AES): Hello, this is a test message for AES encryption!
Thời gian giải mã AES: 0.00037407875061035156 giây
```

B2: Mã hóa và giải mã RSA (Mã hóa bất đối xứng) để mã hóa khóa AES

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from Crypto.Random import get_random_bytes
import time

# Tạo cặp khóa RSA
key = RSA.generate(2048)
private_key = key.export_key()
public_key = key.publickey().export_key()

# Mã hóa khóa AES bằng khóa công khai RSA và đo thời gian
aes_key = get_random_bytes(16)
cipher_rsa = PKCS1_OAEP.new(RSA.import_key(public_key))

start_time = time.time()
encrypted_aes_key = cipher_rsa.encrypt(aes_key)
end_time = time.time()
rsa_encryption_time = end_time - start_time

print("Khóa AES sau khi mã hóa bằng RSA:", encrypted_aes_key)
print("Thời gian mã hóa RSA:", rsa_encryption_time, "giây")

# Giải mã khóa AES bằng khóa bí mật RSA và đo thời gian
decipher_rsa = PKCS1_OAEP.new(RSA.import_key(private_key))

start_time = time.time()
decrypted_aes_key = decipher_rsa.decrypt(encrypted_aes_key)
end_time = time.time()
rsa_decryption_time = end_time - start_time

print("Khóa AES sau khi giải mã:", decrypted_aes_key)
print("Thời gian giải mã RSA:", rsa_decryption_time, "giây")
```

[4] Python

```
Khóa AES sau khi mã hóa bằng RSA: b'@\xc8hA\x1b]\xdeB\xa7\xbc\xc8C\x0b\xbb\x07I\x9d<\x7fg\x95D\x1f9\xf6\x87\xbd\x93\xd1\xff5\x9fM\x19\xadH\xd7\xc8\xfe\xd7\xeb\x1f\x0eH\xc4;Q: \x9f\x9fW\|\xe5\xea\xec\x90
Thời gian mã hóa RSA: 0.000972747802734375 giây
Khóa AES sau khi giải mã: b'\x0b\x11>\xd8\xe1\xa2i\xf4:\xdb\x92\xf0\xa4e\xd0\x16'
Thời gian giải mã RSA: 0.0037391185760498047 giây
```

B3: So sánh thời gian thực thi giữa AES và RSA

```
----- So sánh thời gian mã hóa -----
Thời gian mã hóa AES: 0.000277 giây
Thời gian mã hóa RSA: 0.000973 giây

----- So sánh thời gian giải mã -----
Thời gian giải mã AES: 0.000374 giây
Thời gian giải mã RSA: 0.003739 giây

Mã hóa AES nhanh hơn mã hóa RSA
Giải mã AES nhanh hơn giải mã RSA
```

1. Tại sao mã hóa AES có tốc độ nhanh hơn đáng kể so với RSA?

AES là thuật toán mã hóa đối xứng, sử dụng cùng một khóa cho mã hóa và giải mã. Thuật toán này thiết kế để xử lý nhanh các khối dữ liệu lớn, rất phù hợp với mã hóa dữ liệu có dung lượng lớn. RSA là thuật toán mã hóa bất đối xứng, sử dụng cặp khóa công khai và khóa bí mật khác nhau. Thuật toán này phức tạp hơn nhiều, xử lý dựa trên các phép toán số học lớn (số nguyên tố lớn, luỹ thừa modulo...), nên rất chậm khi thực hiện trên dữ liệu lớn. Do đó, AES thường nhanh và hiệu quả hơn RSA rất nhiều.

2. Trong thực tế, tại sao người ta thường kết hợp cả AES và RSA trong một hệ thống bảo mật?

RSA thường được dùng để mã hóa khóa AES, chứ không để mã hóa trực tiếp dữ liệu lớn, vì RSA xử lý chậm và tốn tài nguyên. AES dùng để mã hóa dữ liệu chính với tốc độ nhanh. Việc kết hợp giúp tận dụng ưu điểm của cả hai: Tính bảo mật cao của RSA trong việc truyền khóa một cách an toàn. Tốc độ xử lý nhanh của AES cho dữ liệu lớn. Ngoài ra, việc dùng khóa AES ngẫu nhiên cho từng phiên làm tăng tính bảo mật, vì ngay cả khi một khóa AES bị lộ, các phiên khác vẫn an toàn.

3. Dựa trên kết quả đo thời gian, loại mã hóa nào phù hợp hơn cho việc mã hóa dữ liệu dung lượng lớn?

AES phù hợp hơn để mã hóa dữ liệu dung lượng lớn do tốc độ nhanh, hiệu quả, chiếm ít tài nguyên tính toán. RSA không thích hợp để mã hóa trực tiếp dữ liệu lớn do hiệu suất thấp, thường chỉ dùng để mã hóa khóa AES hoặc dữ liệu nhỏ. Vì vậy trong các ứng dụng thực tế, thường dùng song song: Mã hóa dữ liệu lớn bằng AES. Mã hóa khóa AES bằng RSA để truyền khóa an toàn.