Bài thực hành 1: Mã hóa đối xứng và bất đối xứng (AES và RSA)

1. Mục tiêu của bài thực hành:
   - Giúp sinh viên hiểu rõ cách hoạt động của mã hóa đối xứng (AES) và mã hóa bất đối xứng (RSA).
   - Áp dụng thuật toán AES để mã hóa và giải mã một đoạn văn bản.
   - Sử dụng thuật toán RSA để mã hóa và giải mã một khóa đối xứng.
2. Yêu cầu chuẩn bị:
   - Máy tính có cài đặt Python và thư viện pycryptodome.
   - Một trình soạn thảo mã nguồn như Visual Studio Code.
   - Cài đặt thư viện pycryptodome

Mở terminal và chạy lệnh: **pip** install pycryptodome

3. Nội dung thực hành:

Bước 1: Mã hóa và giải mã bằng AES (Mã hóa đối xứng)

Mô tả: Sinh viên sẽ 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 bằng AES.

Code mẫu:

```
1  from Crypto.Cipher import AES
2  from Crypto.Random import get_random_bytes
3  from Crypto.Util.Padding import pad, unpad
4  import time
5
6  # Tạo khóa mã hóa 128-bit và khởi tạo AES
7  key = get_random_bytes(16)
8  cipher = AES.new(key, AES.MODE_CBC)
9
10 plaintext = b"Hello, this is a test message for AES encryption!"
11
12 # Đo thời gian mã hóa AES
13 start_time = time.time()
14 ciphertext = cipher.encrypt(pad(plaintext, AES.block_size))
15 end_time = time.time()
16 aes_encryption_time = end_time - start_time
17
18 print("Văn bản mã hóa (AES):", ciphertext)
19 print("Thời gian mã hóa AES:", aes_encryption_time, "giây")
20
21 # 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")
```

Kết quả cần đạt được:

- In ra văn bản mã hóa và giải mã bằng AES.
- In ra thời gian thực thi của mã hóa và giải mã bằng AES (thường rất nhanh, dưới 0.01 giây).

Kết quả thực hiện code….

```
Văn bản mã hóa (AES):
b'\xb0\xca\xff\xba2L\x17\xed_\xcc(\x93\xba\xe9\xcb\xa5\x01\x01_\x8f\xda\x89\xac8\xb3\x98\x99\xa5\xda[4o\x13\xd4V\xbe\xccp\xc9\xcd\xfc\xd0L\x9e\x98\xc2B\xf8\x0bzj4\xd6\xa4\xd8\x99\x1d9\xdb\xa6\xd5\x0fF}'
Thời gian mã hóa AES: 0.0020008087158203125 giây
Văn bản giải mã (AES): Hello, this is a test message for AES encryption!
Thời gian giải mã AES: 0.0 giây
```

Bước 2: Mã hóa và giải mã bằng RSA (Mã hóa bất đối xứng)

Mô tả: Sinh viên sẽ tạo một cặp khóa RSA (công khai và bí mật) và sử dụng chúng để mã hóa một khóa AES. Đồng thời, đo thời gian thực thi của quá trình mã hóa và giải mã bằng RSA.

Code mẫu:

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP

# 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")
```

Kết quả cần đạt được:

- In ra khóa AES sau khi mã hóa và giải mã bằng RSA.
- In ra thời gian thực thi của mã hóa và giải mã bằng RSA (thường chậm hơn AES đáng kể).

Kết quả thực hiện code…..

```
Khóa AES sau khi mã hóa bằng RSA: b'\x1b\x8b\x88:\xf4\xbc\xdd\xaeX\xd0%\x80\xe1-
\xe6\xb5\x02\xec0\x0e\xbeWj\xfa\xbc\x1cR\xact\xd2\xf0<\x98\xfc\xc2;#|/}g\x07z\x11\x
c8\x97\xe6\xd3U\x18\x91\xe7\x17\xf4\xdf\xc2\x10`\xf1\x10\xa8U\x0f2\xcc\t\xe6G\x14y
]\xee\xcecO\x95\xe9\x14\x7f\x1c\xa3\xf7\x82\x94\x92\t3\x83\x9b\xb7\xf0f\x12\xe0\x8b\
xa0\xdfW\x9b\x05\xa6\xb0\x7fq\xd3e\xffNz"\xb6I\'\x8a\x9d#"\x88\xcf\x16\xe7\x1fv\x9
1Gi\x86^`\xe7\x96\xa0\xa2K\xbc\xa9\xf0c\x05\xbe\xd2\xd7h%\xda\xddL\x7f\xf4[o\xb3
ec\xf1\xc0\xdcnG\x89\x15J\x1bRE\xa0P\xff\xackwc$\xe1\x81$\x1e\xc4E\xcf\xfd\x08\x0
4\xa0P\x1c\\f$3\x16\x05c\x1e<\xa5\x95\x96\xf6G\xed\xa6\xa6\xe5\xd7\xc0y80*\x94\xf
1\xb1\xb0+\x83\xf6
\x1b\x93\xb4\xb5\xf4kv\xd9\xdbs\x06\xbak\x9br\xd1\x04\xe9"\x81f\xec\xb5\x80\x05\xf
2\xe3a\xbe\x1e\x0f\xb2\xa2\x92\xdeUbQ'
Thời gian mã hóa RSA: 0.0020003318786621094 giây
Khóa AES sau khi giải mã: b'\xe8/\xde[\xed\xa5\xaf\x94\xf4\x17^#\x15\xf8\xc0Y'
Thời gian giải mã RSA: 0.007999181747436523 giây
```

Bước 3: So sánh thời gian thực thi giữa AES và RSA

Mô tả: So sánh và nhận xét về thời gian mã hóa và giải mã giữa AES và RSA dựa trên kết quả thực hành.

Code mẫu:

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP

# 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()



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

Câu hỏi thảo luận:

1. Tại sao mã hóa AES có tốc độ nhanh hơn đáng kể so với RSA?
2. Trong thực tế, tại sao người ta thường kết hợp cả AES và RSA trong một hệ thống bảo mật?
3. Dựa trên kết quả đo thời gian, loại mã hóa nào phù hợp hơn cho việc mã hóa dữ liệu dung lượng lớn?