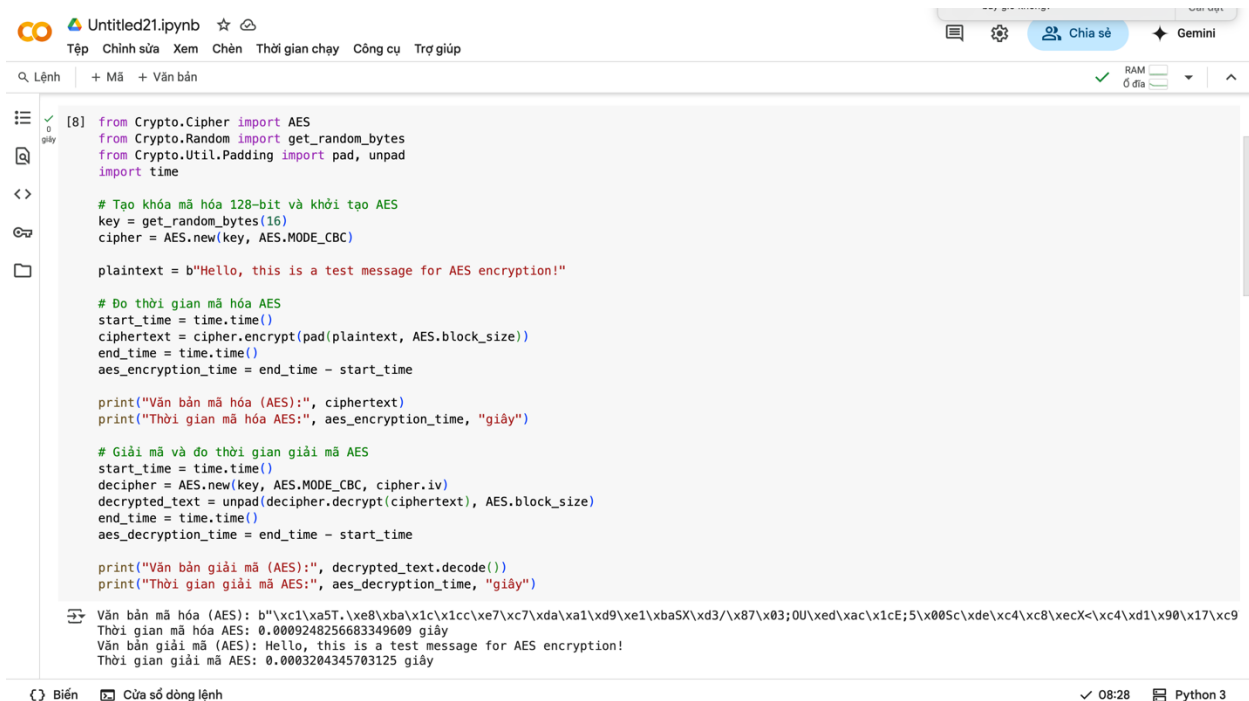


NGUYỄN THU TRANG - DHKL16A1HN

MSV: 22174600114

Bài thực hành số 1



The screenshot shows a Jupyter Notebook titled "Untitled21.ipynb". The code implements AES encryption and decryption using the Crypto module. It includes imports for AES, random bytes, padding, and time. The encryption process generates a 128-bit key, encrypts the plaintext "Hello, this is a test message for AES encryption!", and prints the ciphertext and encryption time. The decryption process uses the same key to decrypt the ciphertext and prints the plaintext and decryption time.

```
[8] 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!"

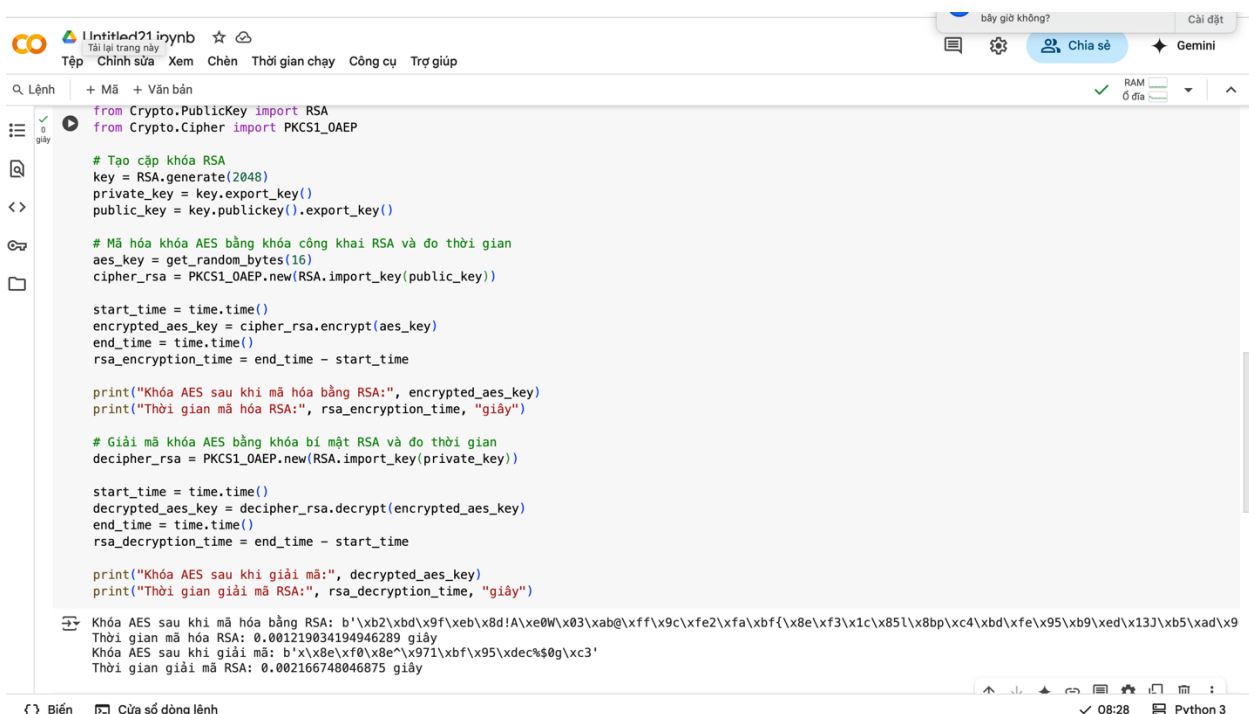
# Đo thời gian mã hóa AES
start_time = time.time()
ciphertext = cipher.encrypt(pad(plaintext, AES.block_size))
end_time = time.time()
aes_encryption_time = end_time - start_time

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")
```

Văn bản mã hóa (AES): b'\xc1\xa5T.\xe8\xba\x1c\x1c\xe7\xc7\xda\xa1\xd9\xe1\xba5\xda3/\x87\x03;0U\xed\xac\x1cE;5\x00Sc\xde\xc4\xc8\xecX<\xc4\xda1\x90\x17\xc9
Thời gian mã hóa AES: 0.0009248256683349609 giây
Văn bản giải mã (AES): Hello, this is a test message for AES encryption!
Thời gian giải mã AES: 0.0003204345703125 giây



The screenshot shows a Jupyter Notebook titled "Untitled21.ipynb". The code implements RSA key generation and AES encryption/decryption using the Crypto module. It includes imports for RSA, Cipher, and time. The encryption process generates an RSA key pair, uses the public key to encrypt an AES key, and prints the encrypted AES key and encryption time. The decryption process uses the private key to decrypt the encrypted AES key and prints the decrypted AES key and decryption time.

```
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")
```

Khóa AES sau khi mã hóa bằng RSA: b'\xb2\xbd\x9f\xeb\x8d!\A\xe0W\x03\xab\xf9\xc\xfe2\xfa\xbf\x8e\xf3\x1c\x85l\x8bp\xc4\xbd\xfe\x95\xb9\xed\x13J\xb5\xad\x9
Thời gian mã hóa RSA: 0.001219034194946289 giây
Khóa AES sau khi giải mã: b'\x8e\xf0\x8e\x97l\xbf\x95\xdec%\$0g\xc3'
Thời gian giải mã RSA: 0.002166748046875 giây

```
.ệnh | + Mã | + Văn bản
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")

Khóa AES sau khi mã hóa bằng RSA: b'\xb2\xbd\x9f\xeb\x8d!\A\xe0W\x03\xab@\xff\x9c\xfe2\xfa\xbf{\x8e\xf3\x1c\x85l\x8bp\xc4\xbd\xfe\x95\xb9\xed\x13J\xb5\xad\x9
Thời gian mã hóa RSA: 0.001219034194946289 giây
Khóa AES sau khi giải mã: b'\x8e\xf0\x8e"\x971\xbf\x95\xdec%$0g\xc3'
Thời gian giải mã RSA: 0.002166748046875 giây

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()
start_time = time.time()
# 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))
print("Khóa AES sau khi giải mã:", decrypted_aes_key)
print("Thời gian giải mã RSA:", rsa_decryption_time, "giây")

Khóa AES sau khi giải mã: b'\xf6m\x1ee\xdd\x5\x86\xe4\xee_\xde \x84.'
Thời gian giải mã RSA: 0.00189971923828125 giây
```

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

- AES (Advanced Encryption Standard) là một thuật toán mã hóa đối xứng, sử dụng cùng một khóa cho cả mã hóa và giải mã. Các thuật toán đối xứng như AES hoạt động theo các khối dữ liệu và được thiết kế để thực hiện nhanh, hiệu quả trên cả phần cứng lẫn phần mềm.
- RSA là thuật toán mã hóa bất đối xứng, sử dụng cặp khóa công khai và bí mật, và dựa vào các phép toán số học phức tạp như lũy thừa mô-đun trên các số nguyên rất lớn. Do đó, RSA tiêu tốn nhiều tài nguyên tính toán hơn.

Kết luận: Mã hóa AES nhanh hơn RSA do sự đơn giản và hiệu quả trong thiết kế thuật toán, trong khi RSA chậm hơn vì bản chất toán học phức tạp.

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

- RSA được dùng để bảo mật việc truyền khóa, còn AES được dùng để mã hóa dữ liệu.
- Cách kết hợp thường thấy là:
 - Khóa AES (tạo ngẫu nhiên) được mã hóa bằng RSA.
 - Dữ liệu thực tế được mã hóa bằng AES (vì nhanh và hiệu quả).
- Phương pháp này được gọi là mã hóa lai (hybrid encryption):
 - Kết hợp tính bảo mật cao của RSA cho quản lý khóa.
 - Kết hợp tốc độ xử lý nhanh của AES cho dữ liệu lớn.

Kết luận: Kết hợp AES và RSA tận dụng điểm mạnh của cả hai: bảo mật khóa tốt (RSA) và tốc độ mã hóa dữ liệu nhanh (AES).

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

- Từ kết quả trong đoạn code:
 - AES mã hóa và giải mã mất dưới 0.01 giây.
 - RSA mất thời gian dài hơn rõ rệt cho cùng thao tác.
- AES có thể xử lý hàng MB hay GB dữ liệu rất nhanh, còn RSA chỉ thích hợp để mã hóa những đoạn dữ liệu nhỏ (như khóa phiên).

Kết luận: AES là lựa chọn phù hợp hơn để mã hóa dữ liệu dung lượng lớn, trong khi RSA chỉ nên dùng để mã hóa khóa hoặc dữ liệu nhỏ.