a4q1

November 11, 2024

[1]:
```
import base64
import datetime
import json
import math
import random
from cryptography.hazmat.primitives import padding
from cryptography.hazmat.primitives.ciphers import Cipher, algorithms, modes
from sympy import mod_inverse
from sympy.ntheory import isprime, nextprime
```

[ ]:
```
# Extract RSA keys
with open('encrypted_assignment.json.txt', 'r') as file:
    data = json.load(file)

n = data["n"]
e = data["e"]
c_1 = data["c_1"]
c_2 = data["c_2"]
```

[ ]:
```
from sympy import factorint, mod_inverse



factors = factorint(n) # Factorized n and compute p, q
p, q = list(factors.keys())

phi_n = (p - 1) * (q - 1)

d = mod_inverse(e, phi_n)
```

[ ]:
```
# Decrypt AES key
aes_key_int = pow(c_1, d, n)
aes_key = aes_key_int.to_bytes(16, byteorder='big')

cipher = Cipher(algorithms.AES(aes_key), modes.ECB()).decryptor()
padded_plaintext = cipher.update(base64.urlsafe_b64decode(c_2)) + cipher.
 ↪finalize()
```

[ ]:
```
# Reomove the padding in the file
unpadder = padding.PKCS7(128).unpadder()
plaintext = unpadder.update(padded_plaintext) + unpadder.finalize()

with open("decrypted_assignment.pdf", 'wb') as file:
    file.write(plaintext)
```