Quiz: Cryptography with Python

Due Feb 1 at 3:30pm **Points** 12 **Questions** 6 **Time Limit** None

Instructions

For this assignment, you will need to first follow the instructions in **this document (https://app.box.com/embed_widget/s/g8kn4sm7b75yeadn6er0xi3m0mpkmhuf?view=list&sort=name&direction=ASC&theme=dark)** . This assignment has no time limit, and it is an individual assignment. After you write your python code, answer the following questions. For many of you, this is your first time using python. Please review the **Intro to Python** assignment. If you struggle to complete the assignment, please ask for help.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	302 minutes	8 out of 12 *

* Some questions not yet graded

Score for this quiz: **8** out of 12 *
Submitted Jan 26 at 10:02pm
This attempt took 302 minutes.

Question 1

2 / 2 pts

In Part 1, what are the first 10 digits of your private key d?

- 9762946661
- 6886694454 — Correct!
- 7299900118
- 1003163509

Question 2

0 / 2 pts

In Part 2, what are the first ten dgits of the cypher-text of the sentence "Running late. Wait for me."

- 6772190022
- 6820092213
- You Answered — 1325060482
- Correct Answer — 5628730804

Question 3

2 / 2 pts

In Part 3, what is the decrypted message?

- Congrats! You just encrypted and decrypted the message!
- Congrats! You just decrypted the message!
- Correct! — Congrats! You just decrypted your first message!
- Congrats!

Question 4

2 / 2 pts

In Part 4, what are the first ten digits of the signature?

- Correct! — 3141272798
- 1982734091
- 6099814821

1762980002

Question 5

2 / 2 pts

In Part 5, is the signature valid?

Correct!

- (selected) Yes
- No

Question 6

Not yet graded / 2 pts

Copy and paste your code here.

Your Answer:

```
p=112481050639317229656723018120659623829736571015511322021617837187076258724819
q=89185111938335771293328323333111422985697062149139368049232365065924632677343

n=p*q
e= 65537

from sympy import mod_inverse
import hashlib

d=mod_inverse(e,(p-1)*(q-1))
d

message = "Running late. Wait for me."
print(message)
# this is the plaintext message we want to send.

# First, we transform m into a number.
m = int.from_bytes(message.encode('utf-8'), "big")  # message.encode converts the string into bytes using the utf-8
# encoding. int.from_bytes converts a number from bytes to integer. 'big' means that the most significant byte is at
```

```
# the beginning  (search MSByte for more info)
print(m)
c = pow(m, e, n)
print(c)

c=902972792334038486842651888055416788261194306570392783861752151
574206685997488594749464296388356840824043012559963682499657708011
602291905026901703377766 7
m3 = pow(c, d, n)

m3_bytes = m3.to_bytes((m3.bit_length() + 7) // 8, 'big')  # the function x.to_bytes needs to now how long the bit
# length should be, and where the MSByte is ("big").

# Now that the m3 is in bytes, we can decode it into string.
message3 = m3_bytes.decode()
print(message3)

m_hash=hashlib.sha256(b"Congrats! You just decrypted your first message!").hexdigest()
m_hash

M3 = int.from_bytes(m_hash.encode('utf-8'), "big")
M3

S = pow(M3, d, n)
print("The Signature message is %s. The signature is %s" % (M3,S))

M4 = pow(S, e, n)
print("The Verified Signature message is %s. The Original Signature message was %s" % (M4,M3))
```

Quiz Score: **8** out of 12