The Python script utilized for this HW assignment is shown on the final page of the assignment.

You intercept the following ciphertext generated using the following RSA public-key: pk={e,n}={23,20413}

Determine the prime numbers p and q:

I determined the following values for p and q: $p = 137$, and $q = 149$.

Determine Euler's totient function $\phi(n)$:

I calculated $\phi(n)$ as follows:

$$\begin{aligned}\phi(n) &= (p-1)\cdot(q-1)\\ \phi(n) &= (137-1)\cdot(149-1)\\ \phi(n) &= 20,128\end{aligned}$$

Determine the private-key={d,n}:

To compute d, the following relationship must hold:

$$d \cdot e \equiv 1 \mod \phi(n) \tag{1}$$

Knowing that $e = 23$, and $\phi(n) = 20,128$, I calculated $d = 13127$:[1]

$$\begin{aligned}e \cdot d &= 23 \cdot 13127\\ e \cdot d &= 301921\\ &= 301921 \mod 20128 = 1\end{aligned}$$

[1]Website used for calculation: https://www.cs.drexel.edu/ jpopyack/IntroCS/HW/RSAWorksheet.html

Compute the plaintext for EACH of the following ciphertext :

{236, 2743, 7983, 5919, 20213, 5520, 19563, 17083, 17083, 19326, 5919, 17258, 5919, 17215, 19563, 20213, 4940, 496}

The plaintext is shown below:

```
Plaintext: [65, 110, 100, 32, 115, 116, 105, 108, 108, 44, 32, 73, 32, 114, 105, 115, 101, 46]
```

Determine the ENGLISH plaintext:

The text output is shown below:

```
English plaintext:  And still, I rise.
```

Python Script:

```python
import math

def find_factors(a):
  factors = []
  for p in range(2,a-1):
    if (a % p) == 0:
      factors.append(p)
      factors.append( int( a/p ) ) # Append q
      break

  if len(factors) == 0:
    print("{} is prime".format(a))
  else:
    print("p = {0}, q = {1}".format(factors[0], factors[1]))
  return factors


def rsa_decode(a, p, q):
  n = p * q
  d = 13127
  # plaintext = ciphertext ^ d mod n

  output = (a ** d) % n
  return output


def main():
  inputs = [236, 2743, 7983, 5919, 20213, 5520, 19563, 17083, 17083, 19326, 5919,
    17258, 5919, 17215, 19563, 20213, 4940, 496]
  n = 20413
  e = 23
  p,q = find_factors(20413)
  outputs = ""
  plaintext = []
  for input in inputs:
    pt = rsa_decode(input, p, q)
    plaintext.append( pt )
    outputs +=  str( chr( pt ) )

  print("Plaintext: " + str(plaintext) )
  print("Output: " + outputs)


if __name__ == "__main__":
  main()
```

rsa_decode.py