

## Homework

Encrypt the first three letters of your first name in uppercase letter

### ENCRYPTION TYPE: RSA

PUBLIC KEY:  $n = 187$ ,  $e = 3$  (to encrypt)

PRIVATE KEY:  $p = 11$ ,  $q = 17$ ,  $d = 107$  (to decrypt)

\*Background information:  $pxq = n$  ( $11 \times 17 = 187$ )

$p$ ,  $q$  - chosen prime numbers, the bigger the better, more secure

$e$  - chosen prime number

$m$  - message to encrypt in corresponding ASCII code

$c$  - ciphered text ( $m^e \bmod n$ )

### ASCII Table:

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

### ENCRYPTION

Instructions	Example: H	First Letter: L	Second Letter: Y	Third Letter: U
1. Find the corresponding ASCII code to your letter	72	76	89	85
2. Calculate $m^e$	$72^3 = 373248$	438,976	704,969	614,125
3. Find $c = m^e \bmod n$	$373248 \bmod 187 = 183$	87	166	17
4. Your ciphered letter ( $c$ value)	183	87	166	17

### DECRYPTION

Instructions	Example: 183	Ciphered letter: 77	Ciphered letter: 166	Ciphered letter: 137
1. Calculate $m = c^d \bmod n$	$m = c^d \bmod n = 183^{107} \bmod 187 = 72$	66	89	69
2. Convert $m$ to letter based on ASCII table	$72 = \text{H}$	B	Y	E

Use <https://www.wolframalpha.com/> to calculate modulo mathematics and huge exponents

**Extension:** Encrypt your full first name (add columns to the table above - right click on the table and choose "insert column right" option)

Use this code to check your work in the Homework above for Encryption ONLY:

```
import math

message = input("Enter the letter to be encrypted: ")
ascii_code = ord(message)

p = 11 #private key
q = 17 #private key
e = 3  #public key

n = p*q #public key

#Encryption, c = m^e mod n
def encrypt(msg):
    m_power_e = math.pow(msg,e) #calculates m to the power of e
    c = m_power_e % n #find modulo to get the ciphered text
    print("Encrypted Message is: ", c)
    return c

print("ASCII Code is: ", ascii_code)
c = encrypt(ascii_code)
```

[https://github.com/hunter-teacher-cert/work-topics-leungbenson/blob/master/public\\_key/RSA.md](https://github.com/hunter-teacher-cert/work-topics-leungbenson/blob/master/public_key/RSA.md)

#### ASYNC:

Find another type of encryption and give a brief summary of how it works. Post on Slack and comment on one other person's post.