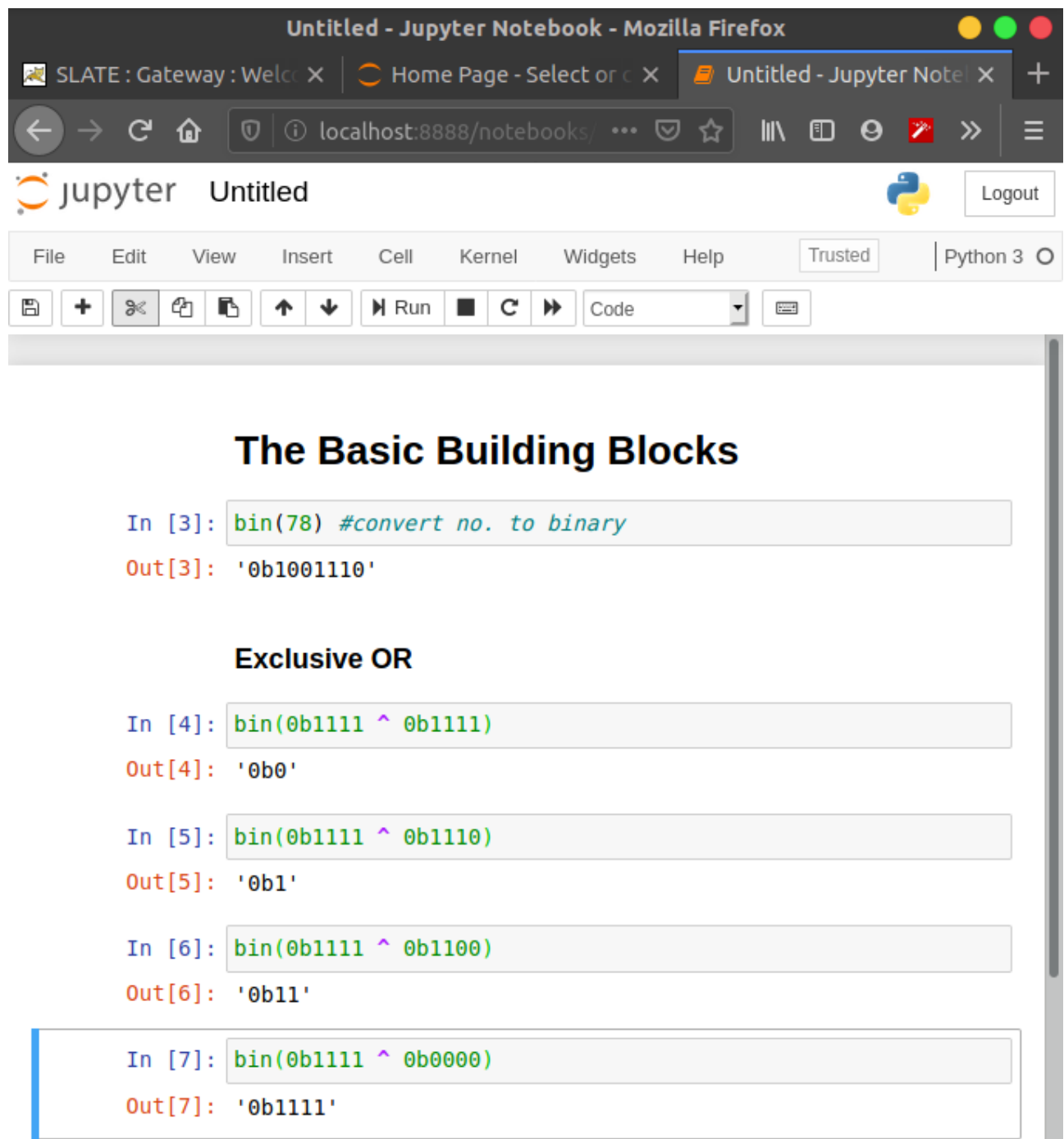


ASAD ZAMAN

p18-0034

(B)



```
Untitled - Jupyter Notebook - Mozilla Firefox
SLATE : Gateway : Welc X | Home Page - Select or c X | Untitled - Jupyter Note X +
localhost:8888/notebooks/
jupyter Untitled Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code

The Basic Building Blocks

In [3]: bin(78) #convert no. to binary
Out[3]: '0b1001110'

Exclusive OR

In [4]: bin(0b1111 ^ 0b1111)
Out[4]: '0b0'

In [5]: bin(0b1111 ^ 0b1110)
Out[5]: '0b1'

In [6]: bin(0b1111 ^ 0b1100)
Out[6]: '0b11'

In [7]: bin(0b1111 ^ 0b0000)
Out[7]: '0b1111'
```

Untitled - Jupyter Notebook - Mozilla Firefox

SLATE : Gateway : Welc X | Home Page - Select or c X | Untitled - Jupyter Note X

localhost:8888/n 90%

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Getting Unicode Representation

In [14]: 1 `ord('A')`

Out[14]: 65

In [16]: 1 `ord('À')`

Out[16]: 226

In [17]: 1 *# don't actually need to convert to binary to do XOR*
2
3 `print(0b1111 ^ 0b0011)`
4 `print(15 ^ 3)`

12
12

In [18]: 1 *# if you XOR the no. with itself, you get the original no. back*
2 `print(bin(0b1111 ^ 0b1111))`

0b0

In [20]: 1 *# if you XOR with the same thing twice,*
2 *# you get the original back*
3
4 `r1 = 0b1010 ^ 0b0110`
5 `print(bin(r1))`
6
7 `r2 = r1 ^ 0b0110`
8 `print(bin(r2))`

0b1100
0b1010

Untitled - Jupyter Notebook - Mozilla Firefox

SLATE : Gatewa X Home Page - Se X Untitled - Jupyter X Visualize Python X

localhost:8888/n 90%

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Building XOR-Based CRYPTO

```
In [27]: 1 message = "Hello, I am hiding in the bunker, Come get me, Please"
```

```
In [33]: 1 for c in message[:5]:
2         print(ord(c))
```

72
101
108
108
111

```
In [34]: 1 chr(72)
```

Out[34]: 'H'

```
In [35]: 1 key = 0b10101010
```

```
In [36]: 1 int(key)
```

Out[36]: 170

```
In [ ]: 1 |
```

The screenshot displays the Jupyter Notebook interface within a Mozilla Firefox browser window. The browser's title bar reads "Untitled - Jupyter Notebook - Mozilla Firefox". The address bar shows the URL "localhost:8888/n". The Jupyter Notebook toolbar is visible, featuring buttons for File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The "Trusted" status is indicated, and the "Python 3" kernel is selected. The notebook content area shows a code cell with the text "print('Hello, World!')".

Lets do the Encryption

```
In [37]: 1 message = "Hello, I am hiding in the bunker, Come get me, Please"
```

```
In [39]: 1 def encrypt(msg , key):
2         encrypted_message = ""
3
4         for c in msg:      # each character gets Encrypted Separately
5
6             c_bin = ord(c)
7             c_encrypted = c_bin ^ key
8             c_encrypted = chr(c_encrypted)
9
10            encrypted_message += c_encrypted
11
12        return encrypted_message
```

```
In [41]: 1 key = 170
          2 cypher_text = encrypt(message, key)
```

```
In [42]: 1 print(cypher_text)
```

àïæåãēçâñîäíãäþâtëβääĩøéǻçïíïρçĩúæïèùĩ

```
In [44]: 1 print( ord('H') ^ 170 )
          2 print( chr(ord('H') ^ 170) )

226
à
```

Untitled - Jupyter Notebook - Mozilla Firefox

SLATE : Gatewa X Home Page - Se X Untitled - Jupyter X Visualize Python X +

localhost:8888/n 90%

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Decryption

```
In [46]: 1 def decrypt(msg , key):
          2
          3     decrypted_message = ""
          4
          5     for c in msg:
          6
          7         c_bin = ord(c)
          8         c_decrypted = c_bin ^ key
          9         c_decrypted = chr(c_decrypted)
         10
         11         decrypted_message += c_decrypted
         12
         13     return decrypted_message
```

```
In [47]: 1 key = 170
          2 decrypted = decrypt(cypher_text, key)
```

```
In [48]: 1 print(decrypted)
```

Hello, I am hiding in the bunker, Come get me, Please

```
In [ ]: 1 |
```

The screenshot shows the Jupyter Notebook interface in a Mozilla Firefox browser window. The browser title is "Untitled - Jupyter Notebook - Mozilla Firefox". The address bar displays "localhost:8888/n". The Jupyter Notebook toolbar is visible, featuring buttons for File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The "Cell" menu is open, showing options like Run, Stop, and Restart. The interface is running on Python 3, as indicated by the "Python 3" label in the bottom right corner.

Breaking The Key

```
In [50]: 1 # lets try brute forcing it !
          2
          3 key = 4
          4 decrypted = decrypt(cypher_text, key)
          5 print(decrypted)
```

æĒĂĀĄçİǺȚÊÇÀÉÇÁÚÆËÎŮÀĖÛıÁǺĖĖÛǺĖŗǺĖİŸĖ

```
In [51]: 1 for key in range(160, 172):
          2     print(decrypt(cypher_text, key))
```

Boffe&*C*kg*bncndm*cd~bo*hdaox&*Iego*mo~*go&*Zfokyo
Cnggd'+B+jf+cbobel+be+cn+i~e'ny'+Hdfn+ln+fn'+[gnjxn
@mddg\$(A(ie('alafo(af(|'m(j)fcmz\$(Kgem(om|(em\$(Xdmi{m
Aleef%)(h)d)a'm'gn')g)al)k|gbl{%)Jfdl)nl)d%)Yelhlz
Fkbb~.G.oc.fggj.i.g.zfk.l'ek'|.Mack.ikz.c%'.~bko)k
Gjcc##/F/nb/gfkfah/fa/[gj/mzad]##/L'b/j/hj/[bj#/c_jn|j
Di``c ,E,ma,dehebk,eb,xdi,nybgi~ ,Ocai,kix,ai ,\`imi
Ehaab!-D-l'-edidcj-dc-yeh-oxcfh!-Nb'h-jhy-'h!-lahl~h
Jgnnm."K"co"jkfkle"kl"vjg"~wligp."Amog"egv"og."Rngcqg
Kfool/#J#bn#k#jgmd#jm#wkf#avmhfq/#@lnf#dfw#nf/#Sofbpf
Hello, I am hiding in the bunker. Come get me, Please
Idmmn-!H!`l!ihehof!ho!uid!ctojs-!Bnld!fdu!ld-!Qmd`rd

In []: 1

Public Key Crypto

```
In [65]: 1 p = 2  
2 q = 7
```

```
In [66]: 1 n = p * q
```

```
In [67]: 1 phi = (p-1)*(q-1)
```

```
In [68]: 1 print(n, phi)
```

14 6

```
In [69]: 1 def gcd(a, b):  
2     while b != 0:  
3         a, b = b, a % b  
4     return a
```

```
In [70]: 1 def get_e(phi):  
2     e = 2  
3  
4     while True:  
5         if gcd(e, phi) == 1:  
6             break  
7         e += 1  
8  
9     return e
```

```
In [71]: 1 e = get_e(phi)  
2 print(e)
```

5

```
In [72]: 1 def get_d(intial_val = 1):  
2  
3     d = intial_val  
4  
5     while True:  
6         if (e * d % phi) == 1:  
7             break  
8         d += 1  
9  
10    return d
```

```
In [73]: 1 d = get_d(10)  
2 print(d)
```

11

msg = 2

```
In [74]: 1 enc = msg**e % n
```

```
In [75]: 1 print(enc)
```

12

```
In [76]: 1 # To Decrypt
```

```
In [77]: 1 dec = enc**d % n
```

```
In [78]: 1 print(dec)
```

10

Let's Say I wanted to tell you the amount I owe you but in a way that provides a guarantee that only I could have written that number --sort of like a digital signature!

```
In [79]: 1 amount = 1000
```

```
In [80]: 1 # sign it -- meaning encrypt it with 'd' this time
2 # of e
3
```

```
In [81]: 1 p = 199
2 q = 131
3 n = p * q
4 phi = (p - 1)*(q - 1)
5 e = get_e(phi)
6 d = get_d()
7 print('n : ', n)
8 print('e : ', e)
9 print('d : ', d)
10 print('phi : ', phi)

n : 26069
e : 7
d : 22063
phi : 25740
```

```
In [82]: 1 sign = amount**d % n
2 print(sign)

17403
```

```
In [83]: 1 dec = sign**e % n
2 print(dec)

1000
```

Final Piece

You don't actually take the full message, You take the hash of the message and sign it.

```
In [93]: 1 p = 983
2 q = 719
3 n = p * q
4 phi = (p - 1)*(q - 1)
5 e = get_e(phi)
6 d = get_d()
7 print('n : ', n)
8 print('e : ', e)
9 print('d : ', d)
10 print('phi : ', phi)

n : 706777
e : 3
d : 470051
phi : 705076
```

```
In [94]: 1 def bad_hash(msg):
2     s = 0
3     for c in msg:
4         s += ord(c)
5     return int(s % 1e10)
```

```
In [95]: 1 message = 'I owe you a total of PKR 1000. Collect on April 28, 2028'
```

```
In [96]: 1 digest = bad_hash(message)
```

```
In [97]: 1 print(digest)

4260
```



```
In [98]: 1 sign = digest**d % n  
2 print(sign)
```

277917

```
In [99]: 1 (message, sign)
```

Out[99]: ('I owe you a total of PKR 1000. Collect on April 28, 2028', 277917)

```
In [100]: 1 digest = bad_hash(message)  
2 print(digest)
```

4260

```
In [101]: 1 dec = sign**e % n  
2 print(dec)
```

4260

```
In [102]: 1 dec == digest
```

Out[102]: True