

CS-GY6803- ISSEM Information Systems Security Engineering and Management Lab 1

For labs in this class, we will be using Jupyter notebooks which is a useful application for writing and compiling Python code. It contains all the necessary libraries and packages which can be used for labs and projects for this class.

Basic python questions:

- 1) Install the Jupyter notebook. You can install in one of the following ways:
 - Refer to link: <https://jupyter.org/install>
 - Alternatively, you can install Anaconda using below link in your system. Based on the OS you have, please select appropriate version.
<https://www.anaconda.com/products/individual-d>
- 2) Post installation, register into the anaconda to access Jupyter notebook.
- 3) After installing and registering on Anaconda, create a new Python 3 notebook.
- 4) In the first cell, put the heading as Lab1 using Heading option in cell type.
- 5) Also, in the first cell, change the cell type to mark down and write the name of all group members and their net ids under the heading.
- 6) Create a new cell (type- code) and write the following code snippet and run the cell.

```
Group = "<Your Group Number>"
Course = "ISSEM"
Lab = "Lab1"
print("Group:"+Group, Lab, "Course:"+Course)
```

- 7) This is the question from the Python assessment. What operation following piece of code is doing. Use comments to explain the relevant parts of the code.

```
str="The quick brown fox jumps over the lazy dog"
arr = str.split(" ")
temp=[]
temp.append(arr[0])
temp.append(arr[1])
arr[0]= arr[len(arr)-2]
arr[1]= arr[len(arr)-1]
arr[len(arr)-2] = temp[0]
arr[len(arr)-1] = temp[1]
str=' '.join(arr)
```

- 8) Write the following function in the next cell.

```
def dummy(var):
    x=str(var)
    y=len(x)-1
    z=""
    while y >= 0:
```

```
z = z + x[y]
y -=1
return int(z)
```

Using an example, explain what this function is doing in comments.

- 9) In the next cell, use the above dummy function to check if below numbers are palindromes or not.

121, 123, 414, 866. 988, 101

- 10) Following is the code for Calculate_Discount function which you implement in the Python Assessment.

```
def Calculate_Discount(price):
    if price >= 500:
        final_price = price * (70/100)
    elif price >= 350 and price < 500:
        final_price = price * (80/100)
    elif price >= 150 and price < 350:
        final_price = price * (90/100)
    elif price >= 10 and price < 150:
        final_price = price * (95/100)
    elif price > 0 and price < 10:
        final_price = price * (50/100)
    else:
        final_price = "Price cannot be negative. Please enter a positive value."
    return final_price
```

Write this code in next Cell and Calculate the discount for the following prices (Hint: Use for loop):

- 440
- 265
- 144
- -10
- 8

Basic cybersecurity questions:

11) Please explain the following terms:

- Cryptography
- Encryption
- Plain Text
- Cipher Text
- Decryption
- Double Strength Encryption
- Hybrid Encryption

Please read the following text carefully:

Shift Ciphers

If you have a message you want to transmit securely, you can encrypt it (translate it into a secret code). One of the simplest ways to do this is with a shift cipher. Famously, Julius Caesar used this type of cipher when sending messages to his military commanders.

We'll call this number the encryption key. It is just the length of the shift we are using. For example, upon encrypting the message “**cookie**” using a shift cipher with encryption key **3**, we obtain the encoded message (or ciphertext): **FRRNLH**.

Let's consider the following conversion table for English alphabets:

A	B	C	D	E	F	G	H	I	J	K	L	M
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
13	14	15	16	17	18	19	20	21	22	23	24	25

Table 2: Encoding English capital letters using integers from \mathbb{Z}_{26} .

- 1) Using the table, we can represent the letters in our message “cookie” with their corresponding numbers: 2 14 14 10 8 4.
- 2) Now add 3 (the encryption key) to each number to get: 5 17 17 13 11 7
- 3) Now use the table to replace these numbers with their corresponding letters: FRRNLH

Now, let's try doing the same thing for the word "zero". What letter would we use to replace the letter "z" when we encrypt? We use the letter "c", which can be viewed as being 3 places further along than "z" if, after we reach "z", we cycle the alphabet around to the beginning again.

After performing shift cipher encryption key 3, the message "zero" becomes CHUR. In terms of mathematical representation of our letters, the encryption of the message "zero" looks this way,

$$25\ 4\ 17\ 14 \rightarrow 28\ 7\ 20\ 17$$

What have we done mathematically? There is a handy mathematical concept that describes this very nicely. Define the following notation for integers a and b and integer $m > 1$:

$$a \equiv b \pmod{m} \text{ means } m \text{ is a divisor of } a - b.$$

In summary, our encryption of the message "zero" using a shift cipher with encryption key 3 looks like this

$$\begin{aligned} z &\rightarrow 25 \rightarrow 25 + 3 \equiv 28 \pmod{26} \rightarrow C \\ e &\rightarrow 4 \rightarrow 4 + 3 \equiv 7 \pmod{26} \rightarrow H \\ r &\rightarrow 17 \rightarrow 17 + 3 \equiv 20 \pmod{26} \rightarrow U \\ o &\rightarrow 14 \rightarrow 14 + 3 \equiv 17 \pmod{26} \rightarrow R \end{aligned}$$

How is the original (plaintext) message recovered from the ciphertext if the encryption key is known?

The following ciphertext was produced using a shift cipher with encryption key 3: CHUR. To decrypt it (i.e., to recover the plaintext message), we need to subtract 3 (. . . or add 23 . . . why is that the same?) to each of the numbers representing the ciphertext letters.

$$\begin{aligned} C &\rightarrow 2 \rightarrow 2 - 3 \equiv -1 \pmod{26} \rightarrow Z \\ H &\rightarrow 7 \rightarrow 7 - 3 \equiv 4 \pmod{26} \rightarrow E \\ U &\rightarrow 20 \rightarrow 20 - 3 \equiv 17 \pmod{26} \rightarrow R \\ R &\rightarrow 17 \rightarrow 17 - 3 \equiv 14 \pmod{26} \rightarrow O \end{aligned}$$

12) Implement the above shift cipher in Python. You are expected to write two functions:

Encryption: An encryption function that takes the plain text as input and returns the ciphertext as output. A good way to check if your function works correctly is to make sure that the output text of your function after encryption is NOT the same as your input text and manually check if each alphabet on in input has been replaced by 3 alphabets to the right

Decryption: This function will take the output of the above encryption function and decode the message to give out the original plain text.

Note: You are expected to handle only alphabetic inputs for both your encryption and decryption functions.

Below is a sample set of inputs that your submission will be tested against and 10 more random inputs on top of these to make sure the functionality works for any alphabetic inputs.

- USA
- ISSEM
- CHICAGO
- CAPPUCCINO
- ZUPPA
- XAVIERS
- YELLOW
- SECURITY
- CYBERHUB
- HURRICANE

13) What is hash function in cryptography?

14) Write a hash function in Python to create the hash value of below message:

Message = "Information Systems Security Engineering and Management"

To hash the above function use Python library- hashlib (SHA256)

15) Hash verification: Data can be compared to a hash value to confirm its integrity. The data is hashed at a certain time and the hash value is protected in some way (you can note it down, for this lab). Later, the data can be hashed again and compared to the protected value. If the hash values match, the data has not been altered. If the values do not match, the data has been corrupted.

Write a Python function which implements Hash verification by above method. Use that function to verify the data integrity of the message in part B of the question.

Submission:

This is a group submission. Please submit only one python notebook per group. Make sure to run all the functions and then download the notebook (format – ipynb). Title of the notebook should be lab1_group_<your group number>.ipynb. Good luck!