CAN304 Lab 1

Programing And Hacking Classical Ciphers

In this lab, you will learn the principle of and how to program and hack three classical ciphers: the Caesar cipher, the transposition cipher, and the Vigenère cipher. Please note theses ciphers are all broken and don't provide true security. Don't use any of the encryption programs in this lab to secure your actual files.

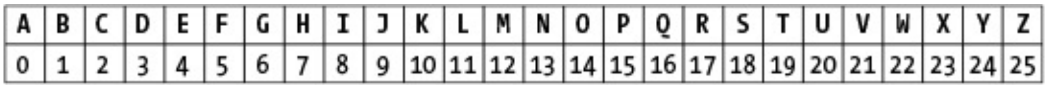
Dependencies

- Python 3

1. The Caesar cipher

The Caesar cipher (also known as the shift cipher) is named after Julius Caesar who used it 2000 years ago. It works by substituting each letter of a message with a new letter after shifting the alphabet over.

Considerring encrypting English text, you can associate A with 0; B with 1; ...; Z with 25 (Figure 1). The secret key $k \in \{0,\cdots,25\}$. To encrypt using key k, shift every letter of the plaintext by k positions to the right (with wraparound). Decryption just reverses the process.

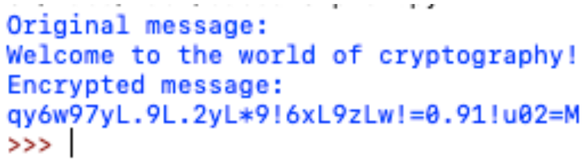


A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Figure 1 Numbering the alphabet from 0 to 25

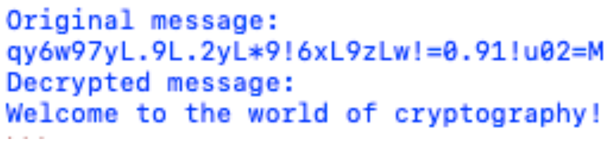
Practices:

(a) Run caesarCipher.py to encrypt the message. You should see the following:



```
Original message:
Welcome to the world of cryptography!
Encrypted message:
qy6w97yL.9L.2yL*9!6xL9zLw!=0.91!u02=M
>>> |
```

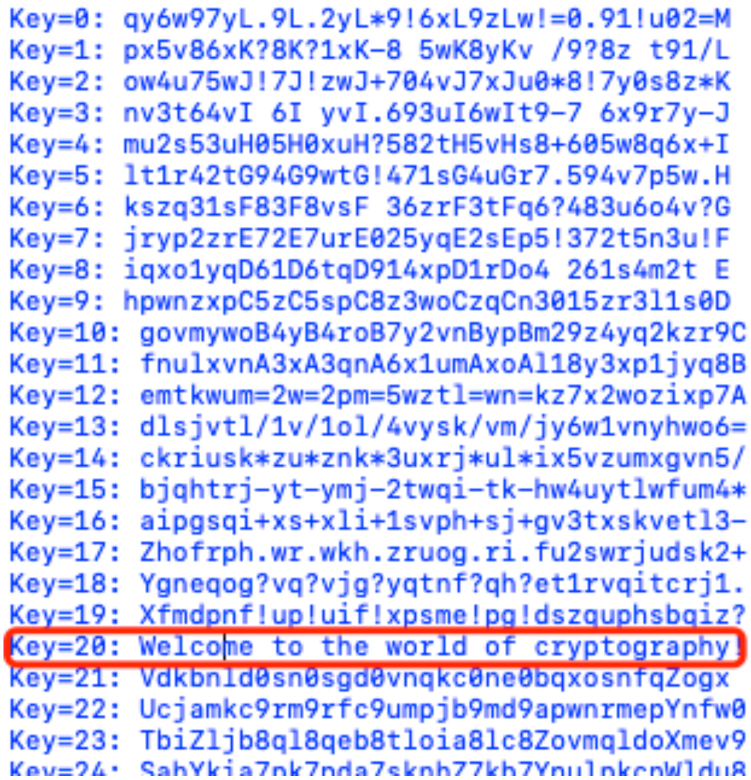
(b) Change the mode to "decrypt" by commenting line 16, 25 and uncommenting line 17, 26. Then run the code. You should see:



```
Original message:
qy6w97yL.9L.2yL*9!6xL9zLw!=0.91!u02=M
Decrypted message:
Welcome to the world of cryptography!
```

(c) Exercise the encrypting and decrypting modes with your own messages.

(d) Run the caesarHacker.py to break a ciphertext with brute-force. You should see:



```
Key=0: qy6w97yL.9L.2yL*9!6xL9zLw!=0.91!u02=M
Key=1: px5v86xK?8K?1xK-8 5wK8yKv /9?8z t91/L
Key=2: ow4u75wJ!7J!zwJ+704vJ7xJu0*8!7y0s8z*K
Key=3: nv3t64vI 6I yvI.693uI6wIt9-7 6x9r7y-J
Key=4: mu2s53uH05H0xuH?582tH5vHs8+605w8q6x+I
Key=5: lt1r42tG94G9wtG!471sG4uGr7.594v7p5w.H
Key=6: kszq31sF83F8vsF 36zrF3tFq6?483u6o4v?G
Key=7: jryp2zrE72E7urE025yqE2sEp5!372t5n3u!F
Key=8: iqxo1yqD61D6tqD914xpD1rDo4 261s4m2t E
Key=9: hpwnzxpC5zC5spC8z3woCzqCn3015zr3l1s0D
Key=10: govmywoB4yB4roB7y2vnBypBm29z4yq2kzr9C
Key=11: fnulxvnA3xA3qnA6x1umAxoAl18y3xp1jyq8B
Key=12: emtkwum=2w=2pm=5wztl=wn=kz7x2wozixp7A
Key=13: dlsjvtl/1v/1ol/4vysk/vm/jy6w1vnyhwo6=
Key=14: ckriusk*zu*znk*3uxrj*ul*ix5vzumxgvn5/
Key=15: bjqhtrj-yt-ymj-2twqi-tk-hw4uytlwfum4*
Key=16: aipgsqi+xs+xli+1svph+sj+gv3txskvetl3-
Key=17: Zhofrph.wr.wkh.zruog.ri.fu2swrjudsk2+
Key=18: Ygneqog?vq?vjg?yqtnf?qh?et1rvqitcrj1.
Key=19: Xfmdpnf!up!uif!xpsme!pg!dszquphsbqiz?
Key=20: Welcome to the world of cryptography!
Key=21: Vdkbnld0sn0sgd0vnqkc0ne0bqxosnfqZogx
Key=22: Ucjamkc9rm9rfc9umpjb9md9apwnrmepYnfw0
Key=23: TbiZljb8ql8qeb8tloia8lc8ZovmqldoXmev9
Key=24: SahYkia7pk7pda7sknhZ7kb7YnulpkcnWldu8
```

The program tries with every possible key. Only with key = 20 the result is readable English text. Therefore, the secret key is 20.

2. The transposition cipher

The Caesar cipher isn't secure; it doesn't take much for a computer to brute-force through all possible keys. The transposition cipher, on the other hand, is more difficult to brute-force because the number of possible keys depends on the message's length. There are many different types of transposition ciphers, including the rail fence cipher, route cipher, Myszkowski transposition cipher, and disrupted transposition cipher. This lab covers a simple transposition cipher called the columnar transposition cipher.

Instead of substituting characters with other characters, the transposition cipher rearranges the message's symbols into an order that makes the original message unreadable. Because each key creates a different ordering, or permutation, of the characters, a cryptanalyst doesn't know how to rearrange the ciphertext back into the original message.

The steps for encrypting with the transposition cipher are as follows:

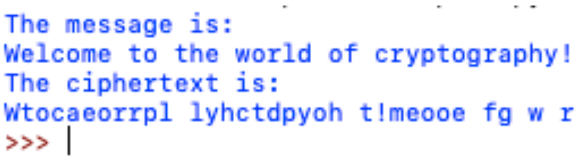
1. Count the number of characters in the message and the key.
2. Draw a row of a number of boxes equal to the key (for example, 8 boxes for a key of 8).
3. Start filling in the boxes from left to right, entering one character per box.
4. When you run out of boxes but still have more characters, add another row of boxes.
5. When you reach the last character, shade in the unused boxes in the last row.

6. Starting from the top left and going down each column, write out the characters. When you get to the bottom of a column, move to the next column to the right. Skip any shaded boxes. This will be the ciphertext.

Practices:

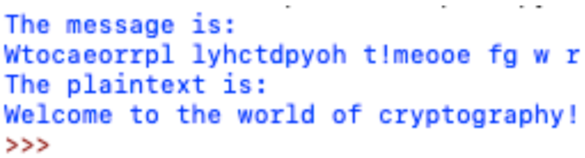
Run the codes to understand the transposition cipher:

(a) Run transpositionCipher.py to encrypt the message. You should see the following:



```
The message is:
Welcome to the world of cryptography!
The ciphertext is:
Wtocaeorrpl lyhctdpyoh t!meooe fg w r
>>> |
```

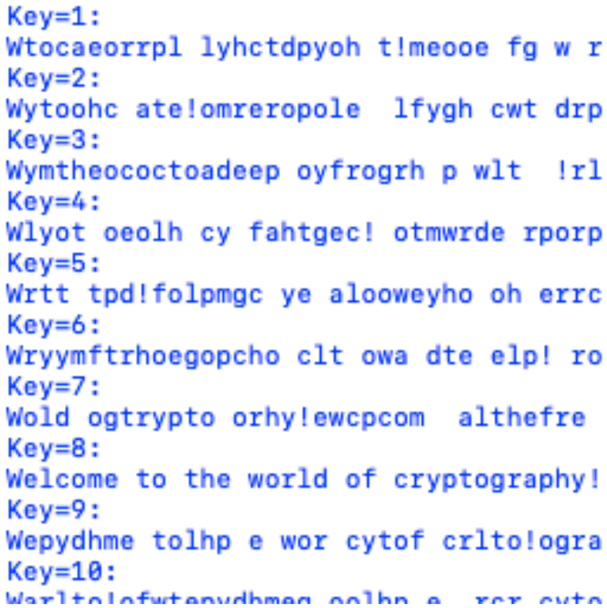
(b) Change the mode to “decrypt” by commenting line 14, 19, 22, 26 and uncommenting line 15, 20, 23, 27. Then run the code. You should see:



```
The message is:
Wtocaeorrpl lyhctdpyoh t!meooe fg w r
The plaintext is:
Welcome to the world of cryptography!
>>>
```

(c) Exercise the encrypting and decrypting modes with your own messages.

(d) Run the transpositionHacker.py to break a ciphertext with brute-force. You should see:



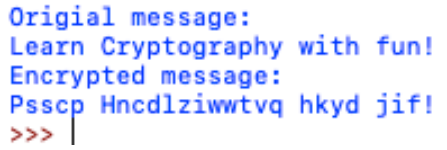
```
Key=1:
Wtocaeorrpl lyhctdpyoh t!meooe fg w r
Key=2:
Wytoohc ate!omreropole  lfygh cwt drp
Key=3:
Wymtheococtoadeep oyfrogrh p wlt  !rl
Key=4:
Wlyot oeolh cy fahtgec! otmwrde rporp
Key=5:
Wrtt tpd!folpmgc ye alooweyho oh errc
Key=6:
Wryymftrhoegopcho clt owa dte elp! ro
Key=7:
Wold ogtrypto orhy!ewcpcom  althefre
Key=8:
Welcome to the world of cryptography!
Key=9:
Wepydhme tolhp e wor cytof crlto!ogra
Key=10:
Warlto!ofwtepydhmeg oolhp e  rcr cyto
```

The program tries with every possible key. Only with key = 8 the result is readable English text. Therefore, the secret key is 20.

3. The Vigenère cipher

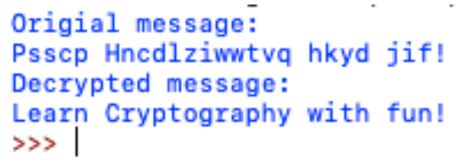
Practices:

(a) Run vigenereCipher.py to encrypt the message. You should see the following:



```
Origial message:
Learn Cryptography with fun!
Encrypted message:
Psscp Hncdlziwwtvq hkyd jif!
>>>
```

(b) Change the mode to “decrypt” by commenting line 14, 18 and uncommenting line 15, 19. Then run the code. You should see:



```
Origial message:
Psscp Hncdlziwwtvq hkyd jif!
Decrypted message:
Learn Cryptography with fun!
>>>
```

(c) Exercise the encrypting and decrypting modes with your own messages.

4. Homework:

(a) Using the transpositionCipher as module, write a (or pair of) programme to encrypt and decrypt a file (.txt)

(b) Write a detectEnglish function to improve the transpositionHacker.py using the dictionary.txt. Then use your improved code to break the encrypted file.

(c) Program to hack the vigenereCipher using non-brute-force methods. Using your code to break the following ciphertext:
Hwxqkj, S., ebv Sgqhqof, X. Pjs hwjpeyesbk tp hncdlziwwtvq.