Lab Sheet 2

Security

In our second lab, we will work on classical encryption techniques in Kali Linux. Boot your Kali Linux installation and use a terminal to invoke shell commands for encryption and decryption. The starting point of each task is an open terminal. When finished post a screenshot of your answers to slack.

1 Task 1: Caesar Cipher

For this task, you will need to install the Python libray pycipher by typing `pip install pycipher` into your terminal.

1.1 Encryption with a known key

Type `python` to invoke the Python interpreter and use `pycipher` to encrypt the message m = *'hello world'* with key $k = 3$. HINT: use `help(pyciper)` to find the right function.

1.2 Brute Force Attack on Caesar Cipher

Use a brute force attack with pycipher to decrypt the ciphertext c = *'FRUUR-JVBCANNC'*.

2 Task 2: Monoalphabetic Ciphers

Download the following script (link also available on Brightspace as crypto) and save it as a shell script file on Kali. HINT: don't forget to make it executable with `chmod +x`.

2.1 Frequency Analysis

Use the script to analyse the following ciphertext:

```
z i o l e g x k l t q o d l z g o f z k g r x e t n g x z g z i t h k o f e o h s t l q f r z t e i f o j
x t l g y l t e x k o f u e g d h x z t k l q f r e g d h x z t k f t z v g k a l v o z i y g e x l g f o
f z t k f t z l t e x k o z n z i t e g x k l t o l t y y t e z o c t s n l h s o z o f z g z v g h q k z
l y o k l z o f z k g r x e o f u z i t z i t g k n g y e k n h z g u k q h i n o f e s x r o f u i g v d
```

```
qfnesqlloeqsqfrhghxsqkqsugkozidlvgkaturtlklqrouozqslo
ufqzxktlqfrltegfrhkgcorofurtzqoslgyktqsofztkftzltexko
znhkgzgegslqsugkozidlqfrziktqzltuohltecokxltlyoktvqss
litfetngxvossstqkfwgzizitgktzoeqsqlhtezlgyegdhxztkqfr
ftzvgkaltexkoznqlvtssqligvziqzzitgknolqhhsotrofzitofz
tkftzziolafgvstrutvossitshngxofrtloufofuqfrrtctsghofu
ltexktqhhsoeqzogflqfrftzvgkahkgzgegslqlvtssqlwxosrofu
ltexktftzvgkal
```

Save it to a file and make sure it can be printed with `cat text.txt`. Use `crypto count letters text.txt percentsort` to see frequencies of letters. You can also use `crypto count digrams text.txt percentsort` to check the frequency of bigrams (double letters). Use the output to uncover most and least frequent letters and bigrams so you can replace them step by step and find the original text. HINT: Have a look at English letter frequency counts, e.g. Mayzner Revisited to help you find the connections.

3 Task 3: Vigenère Cipher

Use `pycipher` again to encrypt the plaintext p = *'centralqueensland'* with the following keys:

- *key = 'cat'*
- *key = 'dog'*
- *key = 'a'*
- *key = 'giraffe'*

4 Task 4: Transposition

Use `pycipher` again to decrypt the following text, using a Columnar Transposition Cipher:

lEEEIGTITGHDBONINSSRI

with the keyword k = *'doctor'*.