Security and Privacy

Lab Worksheet 1 - Classical Ciphers in Python

João Vilela and Manuel E. Correia

Classical Ciphers and Attacks

1 - Implement encryption/decryption functions that take a key (as an integer in 0, 1,2, . . . , 25) and a string to operate as a Caesar cipher. The function should only operate on the characters 'a', 'b', . . . 'z' (both upper and lower case), and it should leave any other characters, unchanged.

2 - Implement a function that performs a brute force attack on a ciphertext, it should print a list of the keys and associated decryptions. It should also take an optional parameter/keyword that takes a substring and only prints out potential plaintexts that contain that keyword.

3 - Check the output of your encrypt function (point 1) on the following (key, plaintext) pairs:

- k = 6 plaintext = "Get me a vanilla ice cream, make it a double."
- k = 15 plaintext = "I don't much care for Leonard Cohen."
- k = 16 plaintext = "I like root beer floats."

4 - Check the output of your decrypt function (point 1) on the following (key, ciphertext) pairs:

- k = 12 ciphertext = 'nduzs ftq buzq oazqe.'
- k = 3 ciphertext = "fdhvdu qhhgv wr orvh zhljkw."
- k = 20 ciphertext = "ufgihxm uly numnys."

5 - Check the output of your attack function (point 2) on the following ciphertexts, if an optional keyword is specified, pass that to your attack function:

- ciphertext = 'gryy gurz gb tb gb nzoebfr puncry.' keyword = 'chapel'
- ciphertext = 'wziv kyv jyfk nyve kyv tpdsrcj tirjy.' keyword = 'cymbal'
- ciphertext = 'baeeq klwosjl osk s esf ozg cfwo lgg emuz.' no keyword

6 - Implement a function that performs frequency attacks on a mono-alphabetic substitution ciphers. This function should take a ciphertext string compute a histogram of the incidence each letter (ignoring all non alphabet characters), and return a list of pairs (letter, incidence percentage) sorted by incidence percentage.

7 - Implement a function that takes a partial mono-alphabetic substitution (python dictionary) and a ciphertext and returns a potential plaintext. The partial mono-alphabetic substitution should is a dictionary with the mapping between ciphertext letters and plaintext letters (if known).

The potential plaintext should be the ciphertext with values specified by the mono-alphabetic substitution replaced by the lower-case plaintext. If the corresponding character is unknown (i.e. does not exist in the dictionary) print the cipher text as an uppercase character.

Example:

```
> substitution = {'y':'o','v':'l','k':'a','w':'m','e':'u'}

> monoalphabetic_substitution("yvk wexny", substitution)

'ola muXNo'
```

8 - Use your functions from points (6) and (7) to decrypt the following cipher text that is in English:

“ztmn pxtne cfa peqef kecnp cjt tmn zcwsenp ontmjsw ztnws tf wsvp xtfwvfefw, c feb fcwvtf, xtfxevqea vf gvoenwk, cfa aeavxcwea wt wse rntrtpvwvtf wscw cgg lef cne xnecwea eymcg.”

Hint: Use the **letter frequency in english** as your guideline.