

SOLUTION REPORT 2

MODERN CRYPTOLOGY (CS641)

COMPUTER SCIENCE AND ENGINEERING

Level 2

Team: **team58**

Ayush Bansal (160177)

Gunjan Jalori (170283)

Siddharth Nohria (160686)

Date: January 23, 2020

1 Chapter 2 (The Caveman)

There are 2 sub-levels in the chapter, first one doesn't have any cipher which needs to be decrypted.

The second sub-level is a **Vigenere Cipher**, the answer to - "how it was recognised and solved" is explained in the subsection after the following list of commands.

The detailed explanation on *Vigenere Cipher* can be found in section 3.2.

Below is the solution to each of the sub-levels:

1. read
2. the_cave_man_be_pleased

1.1 Vigenere Cipher

The ciphertext given was:

```
Lg ccud qh urg tgay ejbwdk, wmgf su bgud nkudnk lrd vjfbg. Yrhfm qvd vng sfuuxytj
"vkj_ecwo_ogp_ej_rnfkukf" wt iq urtuwjm. Ocz iqa jdag vio uzthsivi pqx vkj pgdy encpggt.
Uy hopg yjg fhkz arz hkscv ckoq pgfn vu wwygt nkioe ztft djkt.
```

For identifying what kind of cipher is applied in the above text, we will use the **Index of Coincidence**.

The detailed explanation on *Index of Coincidence* can be found in section 3.1.

The *Index of Coincidence* of the above ciphertext is about 0.042, which is closer to the uniform distribution of English text, this suggests that the cipher is *Poly-alphabetic* such as *Vigenere Cipher*, it may be some other Poly-alphabetic cipher as well but we still have to give it a shot.

For solving the *Vigenere Cipher*, the following steps were employed:

1. Remove all characters from the text which are not part of the English alphabets and capitalize all characters.
2. Partition the text according to different key lengths and sort them according to the *Index of Coincidences* achieved, since higher the IC, closer it is to valid English Text.
3. For each keylen, perform frequency analysis to get the best key possible with the given length.
4. Try out all the keys retrieved and see which one gives some valid English text.

The code used in this part is in the file - `break_vigenere.py`.

The plaintext revealed after using the above decryption key is:

```
Be wary of the next chamber, there is very little joy there. Speak out the password
"the_cave_man_be_pleased" to go through. May you have the strength for the next chamber.
To find the exit you first will need to utter magic words there.
```

From the above, the solution is revealed: `the_cave_man_be_pleased`.

2 Appendix

This section explains each of the things used in between the solutions without proper explanation.

2.1 Index of Coincidence

The **Index of Coincidence** is a measure of how similar a frequency distribution is to the uniform distribution.

$$I.C. = \frac{\sum_{i=A}^{i=Z} f_i(f_i - 1)}{N(N - 1)}$$

where f_i is the count of letter i (where $i = A, B, \dots, Z$) in the ciphertext, and N is the total number of letters in the ciphertext.

Important facts about the *Index of Coincidence*:

- The *Index of Coincidence* of valid English text is about 0.07.
- The *Index of Coincidence* for uniform distribution of English text is about 0.038.
- The *Index of Coincidence* remains the same for the ciphertext and plaintext if cipher is **Mono-alphabetic** (i.e. Substitution Cipher).
- The *Index of Coincidence* of ciphertext is closer to uniform distribution if cipher is **Poly-alphabetic** (such as Vigenere Cipher).

We can get an approximate idea of what kind of cipher is used to generate the ciphertext by using the *Index of Coincidence*.

2.2 Vigenere Cipher

The Vigenere Cipher is a polyalphabetic substitution cipher.

Suppose, the length of the encryption key is k , then the string formed by picking out each letter with a multiple of k letters in between them will be a *Caesar Cipher*.

Since each such string is a *Caesar Cipher*, the *Index of Coincidence* of this string will be closer to that of valid English text rather than closer to uniform distribution.

Using the above principle, we can crack the *Vigenere Cipher*.