Assignment
Cryptanalysis of Ciphertexts

Casper Kristiansson

February 16, 2024

Introduction

The goal of the assignment is to solve and find the solution to an encrypted text. The assignment does provide basic information regarding the encryption like the specific characters it will contain and that an underscore encodes to a space and a hash character encodes a newline. This assignment was completed using cryptoanalysis to understand the cipher type and then decode it.

Cipher 1

When solving cipher 1 for this assignment I tried two different strategies to solve it; Caesar cipher and Substitution cipher [1]. I started to try to solve it as a Caesar cipher by shifting the characters but for all the different solutions, the text didn't produce any result. This pointed towards that the cipher was a Substitution cipher. Given the distribution of English characters in an English text we can see that the cipher text doesn't directly match it but still generally speaking it has the same distribution.

```
english_letter_distribution = {
    "A": 0.072, "B": 0.013, "C": 0.024, "D": 0.037, "E": 0.112, "F"
    : 0.020, "G": 0.018, "H": 0.054, "I": 0.061, "J": 0.001, "K":
    0.007, "L": 0.035, "M": 0.021, "N": 0.059, "O": 0.066, "P":
    0.017, "Q": 0.001, "R": 0.053, "S": 0.056, "T": 0.080, "U":
    0.024, "V": 0.009, "W": 0.021, "X": 0.001, "Y": 0.017, "Z":
    0.001, "_": 0.120,
}

text_distribution = {
```

```
    '_': 0.192, '8': 0.0960, 'F': 0.0675, 'Y': 0.0661, 'O': 0.0661,
     'Z': 0.0541, '9': 0.0519, 'C': 0.0506, 'K': 0.0496, 'H':
    0.0494, 'T': 0.0343, '5': 0.0323, 'O': 0.0257, 'B': 0.0236, 'I'
    : 0.0218, 'W': 0.0211, 'E': 0.0179, '2': 0.0154, 'U': 0.0142, '
    #': 0.0118, '3': 0.0101, 'Q': 0.0098, 'L': 0.0087, '1': 0.0068,
     'R': 0.0008, 'N': 0.0004, '6': 0.0002, 'X': 0.0001
}
```

I started by noticing that the "_" character had the most distribution so I let it be a space. I then started with the single character words *"I"* and *"A"* and assigned them to the text where they fit the best [2]. I then moved to the most frequent symbols such as *"E"*, *"T"*, *"A"*, *"O"*, and tried to guess the best mapping for them. It was during this time a couple of words started to formulate such as "*The*". After this, I then tried to start mapping two-letter words such as "an", "in" and "it". This consisted of a lot of guessing. Figuring out what character mapped to a new line was the hardest part.

After a couple of rounds of doing this, I was able to figure out a good mapping of the characters which resulted in:

```
mapping = {
    '_': ' ', '8': 'E', 'F': 'T', 'Y': 'A', 'O': 'O', 'Z': 'N', '9'
    : 'R', 'C': 'S', 'K': 'I', 'H': 'H', 'T': 'L', '5': 'D', 'O': '
    W', 'B': 'F', 'I': 'U', 'W': 'M', 'E': 'G', '2': 'C', 'U': 'Y',
     '#': 'B', '3': 'P', 'Q': 'K', 'L': 'V', '1': '\n', 'R': 'X', '
    N': 'J', '6': 'Q', 'X': 'Z'
}
```

After decoding the text I figure out that the original text mapped to:

"E HE SINKS FROM SENSE TO CONCEIT NOW IS ANSWERED WHAT YOU ASK OF THE RUNES GRAVEN BY THE GODS MADE BY THE ALL FATHER SENT BY THE POWERFUL SAGE LT IS BEST FOR MAN TO REMAIN SILENT FOR THESE THINGS GIVE THANKS AT NIGHTFALL THE DAY GONE A GUTTERED TORCH A SWORD TESTED THE TROTH OF A MAID ICE CROSSED ALE DRUNK HEW WOOD IN WINDTIME IN FINE WEATHER SAIL TELL IN THE NIGHTTIME TALES TO HOUSEGIRLS FOR TOO MANY..."

I want to note that apart I got stuck on is the few parts where the characters don't match. The symbol *"8"* I mapped to the character *"E"* which in pretty much all situations worked but not the first character. As you can see in the text it starts with the character *"E"* which doesn't make a lot of sense. Second, we have a random word of *"LT"* in the middle that also doesn't make a lot of sense. But because *"L"* and *"T"* work in all other situations it points out that *"LT"* is just a weird part of the text.

Cipher 2

When solving this assignment I came up that it is most likely that the ciphertext was encrypted using a Vigenere cipher or other polyalphabetic substation ciphers [1].

I found this out after calculating and estimating the key length based on the Index of Coincidence (IC) [3]. Because the Vigenere cipher encrypts information using a repeating key it will end up creating repeating patterns in the encrypted text. Using IC will help to measure the probability of repeated characters. IC can be calculated using this formula (N is the number of characters and f_i is the frequency of the i-th character):

$$IC = \frac{\sum_{i=1}^{N} f_i(f_i - 1)}{N(N-1)}$$

We then can by testing different key lengths calculate the probability of it. This can be completed via:

```
def index_of_coincidence(sequence):
  N = len(sequence)
  frequencies = Counter(sequence)
  return sum(f * (f - 1) for _, f in frequencies.items()) / (N * (N
     - 1))

def calculate_frequencies_for_key_length(cipher, key_length):
  chunks = []
  for i in range(key_length):
    chunk = ''
  for j in range(i, len(cipher), key_length):
    chunk += cipher[j]
  chunks.append(chunk)

  ics = []
  for chunk in chunks:
    ics.append(index_of_coincidence(chunk))

  return np.mean(ics)
```

Using this approach we will then get an estimate of the key length. For this specific text, we saw that at the key length 12, we got a spike the the probability. As the next step, we want to perform a frequency analysis on the cipher text. As stated earlier certain characters have a higher probability of appearing than other characters [3]. Therefore we can use chi-squared to calculate how well observed data can fit expected data. This can be useful because then we can estimate the best fit of characters. Chi-squared can be calculated by the following formula (O_i is the observed frequency and E_i is the expected frequency).

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

This can then be converted into code by:

```
def chi_squared_statistic(text, expected_freq):
  observed_freq = Counter(text)
  for char in characters:
    observed_freq[char] = observed_freq.get(char, 0) / len(text)

  chi_squared = 0
  for char in characters:
    expected = expected_freq.get(char, 0)
  if expected != 0:
    chi_squared += ((observed_freq[char] - expected) * * 2) /
    expected

  return chi_squared
```

The next step is to apply Caesar shift [4] to the cipher and the goal is to try to align and shift it until we get the highest probability of the character frequency distribution. This means that we first split the cipher text into different columns based on the key length. We then for each column perform Caesar shift between 1 to 38 (number of possible characters) and on each shifted text we calculate the chi-squared probability. We then know that the Caesar shift with the lowest chi-squared value has the best shift. For each of the twelve columns, we get the best Caesar shift for each of them which in this case results in [24, 28, 32, 36, 2, 6, 10, 14, 18, 22, 8, 6].

```
def caesar_shift(sequence, shift):
  shifted_sequence = ''
  for char in sequence:
    shifted_sequence += characters[(characters.index(char) - shift)
     % len(characters)]
  return shifted_sequence

def find_best_shift_for_column(column):
  chi_squared_by_shift = {}

  for shift in range(len(characters)):
    chi_squared_by_shift[shift] = chi_squared_statistic(
    caesar_shift(column, shift), english_letter_freq)

  return min(chi_squared_by_shift, key = chi_squared_by_shift.get)
```

We then can simply apply the best shifts to the cipher for each column and then combine the text to get its decrypted text. By doing this we can see that the original text is:

"HARD AS I CAN PERCEIVE FROM MANY CIRCUMSTANCES AND IN SHORT POSSESSES A LARGE STOCK OF INFORMATION WHEN HE

HEARD THAT I AM DRAWING A GOOD DEAL AND THAT I KNOW GREEK TWO WONDERFUL THINGS FOR THIS PART OF THE COUNTRY HE CAME TO SEE ME AND DISPLAYED HIS WHOLE STORE OF LEARNING FROM BATTEAUX TO WOOD FROM DE PILES TO WINKELMANN HE ASSURED ME HE HAD READ THROUGH THE FIRST PART OF..."

References

[1] D. Wikström, *Lecture 2: Applied cryptography (dd2520)*, KTH Royal Institute of Technology, Jan. 2022.

[2] *Cryptography 101: Basic solving techniques for substitution ciphers - dummies*, https://www.dummies.com/article/home-auto-hobbies/games/puzzles/cryptograms/cryptography-101-basic-solving-techniques-for-substitution-ciphers-195424/, (Accessed on 02/16/2024).

[3] C. Christensen, "Cryptanalysis of the vigenère cipher: The friedman test," *a a*, vol. 1, no. 1, p. 1, 2015.

[4] *Azrizonda state univerity - shift (caesar) ciphers*, https://math.asu.edu/sites/default/files/shift.pdf, (Accessed on 01/31/2024).