Homework 01: Decrypting a Ciphertext Using Frequency Analysis

Nicolas Leone
Student ID: 1986354
Cybersecurity

October 7, 2025

Contents

1 Introduction

Monoalphabetic substitution ciphers represent one of the classical encryption techniques where each letter of the plaintext is replaced by another letter throughout the entire message. While these ciphers provide a basic level of security, they are vulnerable to frequency analysis attacks due to the statistical properties of the used language's words and letters.

This report documents the approach used to decrypt a given ciphertext using:

- **Frequency Analysis**: Statistical analysis of letter occurrences
- **Pattern Recognition**: Identification of common English word structures

1.1 Methodology

In order to use correctly the approaches described above, the decryption process involved several steps:

1. **Analyzing the Ciphertext**: Calculate the frequency of each character in the ciphertext.
2. **Frequency Analysis Implementation**: Obtain the frequency of each letter in the standard English language.
3. **Initial Mapping**: Create a substitution map based on the frequencies.
4. **Iterative Refinement**: Adjust the mapping manually by making logical assumptions about common words and letter patterns.
5. **Plaintext Formulation**: Generate the plaintext using the current mapping.

2 Problem Statement

Given a ciphertext encrypted using a monoalphabetic substitution cipher, the objective was to recover the original plaintext message. The ciphertext contained the marker "TT" which appeared to serve as line breaks in the original message structure.

2.1 Encrypted Message

The initial encrypted ciphertext was:

```
"OUETOJIDECEJTCE
   OTUWSTDWCCILETIOTBIFATFWOUTITSYICCTNILTHATUWSTSUHRCBEJTT
IABTUET HCCHFEBTOUETAIJJHFTJHIBTOUIOTCEBTIQJHSSTOUET
   WECBSTFUEJETOUETLJISSTT
FISTSOWCCTFEOTFWOUTBEFTIABTOUETIWJTFISTQHHCTFUWCETOUETSHRABTH TNWJBSTT
EQUHEBT JHYTOUETOJEESTOUIOTSOHHBTIOTOUETEBLETH TOUET HJESOTIABTOUETSXGTT
LJEFTNJWLUOEJTFWOUTEDEJGTSOEPTISTOUETSRATJHSETNEUWABTOUETBWSOIAOTUWCCSTT
IABTOUETSUIBHFSTSUHJOEAEBTFUWCETOUETQHCHJSTH TOUETEIJOUTQUIALEBTSCHFCGTT
 JHYTLJEGTOHTLHCBTISTW TOUETCIABTWOSEC TFEJETIFIXEAWALTOHTLJEEOTUWYTHATT
UWSTKHRJAEGTOUIOTSEEYEBTOHTSOJEOQUTEABCESSCGTNE HJETUWYTCWXETITJWDEJTT
FUHSETQHRJSETFISTRAXAHFATGEOTPJHYWSEBTOHTCEIBTSHYEFUEJETNEGHABTOUETUHJWMHATT
```

```
FUEJETOUETQWOGTUETBJEIYEBTH TSOHHBTFWOUTFICCSTH TSOHAETIABTOHFEJSTOUIOTT
LCWOOEJEBTWATOUETCWLUOTH TOUETYHJAWALTSRATOUIOTSUHAETFWOUTITNJWCCWIAQETT
OUIOTYIBETUWYTOUWAXTH TSOHJWESTUETUIBTUEIJBTISTITQUWCBTFUEATUETSIOTNGTT
OUET WJETIABTCWSOEAEBTOHTUWSTYHOUEJTSPEIXTH TUEJHESTFUHTOJIDECEBT IJTT
IABT HRABTFHABEJSTOUIOTQUIALEBTOUEWJTCWDEST HJEDEJTFUWCETUETFHABEJEBTW
   TT
UWSTHFATSOEPSTFHRCBTHAETBIGTNJWALTUWYTOHTSRQUTPCIQESTHJTW TOUETJHIBTT
FHRCBTSWYPCGTQHAOWARETFWOUHROTEABTIQJHSSTDICCEGSTIABTHDEJTJWDEJSTIABTT
OUJHRLUT HJESOSTOUIOTLJEFTBIJXEJTISTOUETCWLUOTH TBIGT
   IBEBTIABTOUETYHHATT
IPPEIJEBTWATOUETSXGTOHTFIOQUTHDEJTUWYTFWOUTSWCEAOTEGESTOUIOTSEEYEBTOHTT
LRWBETUWST EEOTICHALTPIOUSTUETUIBTAEDEJTXAHFATEVWSOEBTRAOWCTOUETAWLUOTT
HPEAEBTOUEYTNE HJETUWYTIABTOUETSOIJSTINHDETYIJXEBTUWSTFIGTFWOUTOUEWJTT
BWSOIAOTCWLUOTOUIOTLCWOOEJEBTCWXETSWCDEJTBRSOTSQIOOEJEBTIQJHSSTITNCIQXTT
QCHOUTOUIOTQHDEJEBTOUETUEIDEASTIABTYIBETUWYT EECTNHOUTSYICCTIABTGEOTT
PIJOTH TSHYEOUWALTLJEIOEJTOUIATUWYSEC TISTW TOUETRAWDEJSETUIBTNEEATT
FIWOWALT HJTUWYTOHTFICXTOUWSTDEJGTJHIBTIABTOHTCEIJATOUIOTOUETKHRJAEGTT
FISTAHOTHACGTINHROTJEIQUWALTOUETQWOGTUETSHRLUOTNROTINHROTBWSQHDEJWALTT
OUETSOJEALOUTOUIOTCIGTFWOUWATUWYTIABTOUETQHRJILETOUIOTLJEFTFWOUTEIQUTT
SOEPTOUIOTUETOHHXTWAOHTOUETRAXAHFATFUEJETOUETPJHYWSETH TOHYHJJHFTT
FIWOEBTCWXETITCWLUOTOUIOTQHRCBTAEDEJTNETEVOWALRWSUEBTAHTYIOOEJTUHFTT
CHALTOUETAWLUOTHJTUHFT
   IJTOUETJHIBTOUIOTCEBTUWYTEDEJTHAFIJBTWAOHTOUETBWSOIAQETT"
```

3 Methodology

3.1 Phase 1: Analyzing the Ciphertext

The first step involved preparing the ciphertext for analysis by handling the "TT" markers and extracting the core encrypted content.

```python
from pathlib import Path

ciphertext_path = Path(__file__).with_name("ciphertext.txt")
ciphertext = ciphertext_path.read_text()

# ciphertext replacing 'tt' letters with '\n' char for final plaintext
ciphertext_in_rows = ciphertext.replace("TT", "\n")

# ciphertext without 'tt' letters to work on frequency analysis
ciphertext_no_tt = ciphertext.replace("TT", "")
```

Listing 1: Initial Data Processing Code

3.2 Phase 2: Frequency Analysis Implementation

The core of the cryptanalytic approach relied on frequency analysis, exploiting the fact that letter frequencies in the ciphertext should mirror those of English text.

```python
# count the occurrences of each letter in the TT-stripped ciphertext
ciphertext_letter_occurrences = {}
for char in ciphertext_no_tt:
    if char.isalpha() or char == " ":
        ciphertext_letter_occurrences[char] =
           ciphertext_letter_occurrences.get(char, 0) + 1
```

```
6
7  # sort by descending frequency so the most common letters come first
8  sorted_letter_occurrences = dict(
9      sorted(ciphertext_letter_occurrences.items(), key=lambda entry:
           entry[1], reverse=True)
10 )
```

Listing 2: Frequency Analysis Implementation

3.3 Phase 3: Initial Mapping Based on English Frequencies

Using standard English letter frequencies, an initial substitution mapping was created:

```
1  # expected frequency order (space first, then letters by common English
       usage - source: Wikipedia)
2  frequency_reference = [
3      " ", "E", "T", "A", "O", "I", "N", "S", "R", "H", "L", "D", "C", "U"
           , "M", "W", "F", "G", "Y", "P", "B", "V", "K", "J", "X", "Q", "Z"
           ,
4  ]
5
6  # first attempt of frequency analysis decryption via frequency matching
7  sorted_cipher_chars = list(sorted_letter_occurrences.keys())
8  decryption_map = {
9      cipher_char: frequency_reference[idx]
10     for idx, cipher_char in enumerate(sorted_cipher_chars)
11     if idx < len(frequency_reference)
12 }
13
14 first_plaintext = "".join(
15     decryption_map.get(char, char)
16     for char in ciphertext_in_rows
17 )
```

Listing 3: Initial Frequency-Based Mapping

Analysis: This initial mapping provided a first result that unfortunately contained many nonsensical words, indicating the need for refinement through pattern analysis.

3.4 Phase 4: Iterative Refinement Process

Each refinement step was motivated inside the code by specific observations about English language patterns:

```
1  # second attempt - swapping O with A since the O was frequently used as
       a single letter word, which is usually 'A' in English
2  second_plaintext = first_plaintext.translate(str.maketrans({"O": "A", "A
       ": "O"}))
```

Listing 4: First Manual Adjustment

```
1  # third attempt - swapping O with H since the word TOE is frequent in
       the text, and could be THE
2  third_plaintext = second_plaintext.translate(str.maketrans({"O": "H", "H
       ": "O"}))
```

Listing 5: THE Pattern Recognition

```
# fourth attempt - swapping O with N and L with D since the frequently
    used word AOL could be AND
fourth_plaintext = third_plaintext.translate(str.maketrans({"O": "N", "N
    ": "O", "L": "D", "D": "L"}))
```

Listing 6: AND Pattern Recognition

```
# fifth attempt - swapping U with G because EDUE could be EDGE
fifth_plaintext = fourth_plaintext.translate(str.maketrans({"U": "G", "G
    ": "U"}))
```

Listing 7: EDGE Pattern Recognition

```
# sixth attempt - swapping I with O since the word BEYIND could be
    BEYOND
sixth_plaintext = fifth_plaintext.translate(str.maketrans({"I": "O", "O"
    : "I"}))
```

Listing 8: BEYOND Pattern Recognition

```
# seventh attempt - swapping P with V since ABOPE could be ABOVE
seventh_plaintext = sixth_plaintext.translate(str.maketrans({"P": "V", "
    V": "P"}))
```

Listing 9: ABOVE Pattern Recognition

```
# eighth attempt - swapping S with I since VSLLAGE could be VILLAGE
eighth_plaintext = seventh_plaintext.translate(str.maketrans({"S": "I",
    "I": "S"}))
```

Listing 10: VILLAGE Pattern Recognition

```
# ninth attempt - swapping R with S since the word HIR is frequently
    used and could be HIS
ninth_plaintext = eighth_plaintext.translate(str.maketrans({"R": "S", "S
    ": "R"}))
```

Listing 11: HIS Pattern Recognition

```
# tenth attempt - swapping M with F since the word LEMT could be LEFT
tenth_plaintext = ninth_plaintext.translate(str.maketrans({"M": "F", "F"
    : "M"}))
```

Listing 12: LEFT Pattern Recognition

```
# eleventh attempt - swapping C with W since CITH could be WITH and
    FOLLOCED could be FOLLOWED
eleventh_plaintext = tenth_plaintext.translate(str.maketrans({"C": "W",
    "W": "C"}))
```

Listing 13: WITH Pattern Recognition

```
# twelfth attempt - swapping C with M since SCALL could be SMALL and
    AMROSS could be ACROSS
twelfth_plaintext = eleventh_plaintext.translate(str.maketrans({"C": "M"
    , "M": "C"}))
```

Listing 14: SMALL and ACROSS Pattern Recognition

```
# final attempt - swapping K with P since STEK could be STEP and SKEAP could be SPEAK
final_plaintext = twelfth_plaintext.translate(str.maketrans({"K": "P", "P": "K"}))
```

Listing 15: Final Adjustment - STEP and SPEAK

4 Plaintext Formulation

4.1 Final Cipher-to-Plaintext Mapping

After the complete refinement process, the final mapping was established:

Cipher	Plain	Cipher	Plain	Cipher	Plain
A	N	J	R	S	S
B	D	K	J	T	” ”
C	L	L	G	U	H
D	V	M	Q	V	X
E	E	N	B	W	I
F	W	O	T	X	K
G	Y	P	P	Y	M
H	O	Q	C	Z	Z
I	A	R	U	” ”	F

4.2 Decrypted Message

The final decrypted plaintext revealed a coherent narrative:

```
"THE TRAVELER LEFT HIS VILLAGE AT DAWN WITH A SMALL BAG ON HIS SHOULDER
AND HE FOLLOWED THE NARROW ROAD THAT LED ACROSS THE FIELDS WHERE THE GRASS
WAS STILL WET WITH DEW AND THE AIR WAS COOL WHILE THE SOUND OF BIRDS
ECHOED FROM THE TREES THAT STOOD AT THE EDGE OF THE FOREST AND THE SKY
GREW BRIGHTER WITH EVERY STEP AS THE SUN ROSE BEHIND THE DISTANT HILLS
AND THE SHADOWS SHORTENED WHILE THE COLORS OF THE EARTH CHANGED SLOWLY
FROM GREY TO GOLD AS IF THE LAND ITSELF WERE AWAKENING TO GREET HIM ON
HIS JOURNEY THAT SEEMED TO STRETCH ENDLESSLY BEFORE HIM LIKE A RIVER
WHOSE COURSE WAS UNKNOWN YET PROMISED TO LEAD SOMEWHERE BEYOND THE HORIQON
WHERE THE CITY HE DREAMED OF STOOD WITH WALLS OF STONE AND TOWERS THAT
GLITTERED IN THE LIGHT OF THE MORNING SUN THAT SHONE WITH A BRILLIANCE
THAT MADE HIM THINK OF STORIES HE HAD HEARD AS A CHILD WHEN HE SAT BY
THE FIRE AND LISTENED TO HIS MOTHER SPEAK OF HEROES WHO TRAVELED FAR
AND FOUND WONDERS THAT CHANGED THEIR LIVES FOREVER WHILE HE WONDERED IF
HIS OWN STEPS WOULD ONE DAY BRING HIM TO SUCH PLACES OR IF THE ROAD
WOULD SIMPLY CONTINUE WITHOUT END ACROSS VALLEYS AND OVER RIVERS AND
THROUGH FORESTS THAT GREW DARKER AS THE LIGHT OF DAY FADED AND THE MOON
APPEARED IN THE SKY TO WATCH OVER HIM WITH SILENT EYES THAT SEEMED TO
GUIDE HIS FEET ALONG PATHS HE HAD NEVER KNOWN EXISTED UNTIL THE NIGHT
OPENED THEM BEFORE HIM AND THE STARS ABOVE MARKED HIS WAY WITH THEIR
```

```
DISTANT LIGHT THAT GLITTERED LIKE SILVER DUST SCATTERED ACROSS A BLACK
CLOTH THAT COVERED THE HEAVENS AND MADE HIM FEEL BOTH SMALL AND YET
PART OF SOMETHING GREATER THAN HIMSELF AS IF THE UNIVERSE HAD BEEN
WAITING FOR HIM TO WALK THIS VERY ROAD AND TO LEARN THAT THE JOURNEY
WAS NOT ONLY ABOUT REACHING THE CITY HE SOUGHT BUT ABOUT DISCOVERING
THE STRENGTH THAT LAY WITHIN HIM AND THE COURAGE THAT GREW WITH EACH
STEP THAT HE TOOK INTO THE UNKNOWN WHERE THE PROMISE OF TOMORROW
WAITED LIKE A LIGHT THAT COULD NEVER BE EXTINGUISHED NO MATTER HOW
LONG THE NIGHT OR HOW FAR THE ROAD THAT LED HIM EVER ONWARD INTO THE
   DISTANCE"
```