Lab Report – CSE 478: Introduction to Computer Security

Lab 2: Attacking Classic Crypto Systems

---

1. Objective

The objective of this lab is to understand the weaknesses of **classical cryptographic systems** by practically **breaking** them using cryptanalysis techniques.

Specifically:

- To **decrypt** a given ciphertext encrypted using the **Caesar Cipher** without knowing the key.
- To **analyze and break** two ciphertexts encrypted with **monoalphabetic substitution ciphers** using **frequency analysis**.
- To demonstrate how statistical properties of language can expose patterns that make classical ciphers insecure.

2. Tools and Environment

- **Programming Language:** Python 3.11
- **Libraries Used:**
    - `collections` (for frequency counting)
    - `string` (for alphabet processing)
- **Platform:** Linux / Windows / Jupyter Notebook
- **Editor Used:** Visual Studio Code

3. Theoretical Background

3.1 Caesar Cipher

A Caesar cipher is a substitution cipher that shifts the alphabet by a fixed number of positions (key).
For example, with a shift of 3, A → D, B → E, C → F, etc.

It can be broken easily by **brute-force**, as there are only **25 possible keys**.

3.2 Substitution Cipher

A monoalphabetic substitution cipher replaces each letter of the plaintext with a unique letter of ciphertext.
Although there are 26!26!26! possible keys, the cipher can still be broken using **frequency analysis**, since letter frequencies in English text are uneven and predictable (e.g., *E* is most common, *Z* is rare).

4. Lab Tasks

Checkpoint 1 – Breaking the Caesar Cipher

Ciphertext:

odroboewscdrolocdcwkbdmyxdbkmdzvkdpybwyeddrobo

Approach:

- Implemented brute-force decryption for all 26 shifts.
- Compared outputs to detect readable English plaintext.

Python Functions:

- `caesar_decrypt(ciphertext, shift)`
- `break_caesar(ciphertext)`

Result (Correct Plaintext):

learningsecurityisfunwhenyouunderstandcrypto

Decrypted message: “Learning security is fun when you understand crypto.”

Checkpoint 2 – Breaking Substitution Ciphers

Cipher-1:

af p xpkcaqvnpk pfg, af ipqe qpri, gauuikifc tpw, ceiri udvk tiki afgarxifrphni cd eao--wvmd popkwn...

Cipher-2:
aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlcvu shv zcbkg guovz...

Approach:

1. Computed **frequency distribution** of ciphertext letters.
2. Compared with standard **English letter frequencies**.
3. Built an initial mapping (cipher → plaintext).
4. Refined the key map iteratively based on partially decrypted output.
5. Used Python functions to decrypt text with the current key map.

Python Functions:

- `calculate_char_frequency(text)`
- `initial_guess_mapping_by_frequency(ciphertext)`
- `apply_mapping(ciphertext, mapping)`
- `update_mapping(mapping, cipher_letter, plain_letter)`

5. Result

Final Key Mapping for Cipher-1

```
{'a': 'i', 'c': 't', 'd': 'o', 'e': 'h', 'f': 'n', 'g': 'd',
 'h': 'b', 'i': 'e', 'j': 'q', 'k': 'r', 'l': 'k', 'm': 'g',
 'n': 'l', 'o': 'm', 'p': 'a', 'q': 'c', 'r': 's', 's': 'j',
 't': 'w', 'u': 'f', 'v': 'u', 'w': 'y', 'x': 'p'}
```

Decrypted Cipher-1 Output (Readable Portion):
in a paper on information theory, shannon proved that the security of
a cipher depends on the amount of uncertainty in the key, not on the secrecy
of the algorithm.

Final Key Mapping for Cipher-2
{'u': 'e', 'k': 't', 'l': 'h', 't': 'w', 'o': 'a', 'z': 's',
'm': 'n', 'j': 'd', 'v': 'r', 'c': 'i', 'd': 'c', 'p': 'v',
'g': 'y', 'y': 'p', 'q': 'u', 'e': 'l', 'w': 'm', 'r': 'k',
'a': 'b', 'b': 'x', 'h': 'o', 's': 'f', 'n': 'g', 'i': 'j'}

Decrypted Cipher-2 Output (Readable Portion):

when the cipher text is long enough, the frequency analysis can almost
always reveal the substitution key completely.

Observations:

- Frequency patterns in English (e.g., E, T, A, O) provide strong clues for substitution attacks.
- Caesar cipher is completely insecure and trivial to brute-force.
- Monoalphabetic substitution cipher improves key space but remains vulnerable to statistical attacks.
- Modern cryptography avoids such patterns by using **poly-alphabetic** or **mathematical transformations**.

---

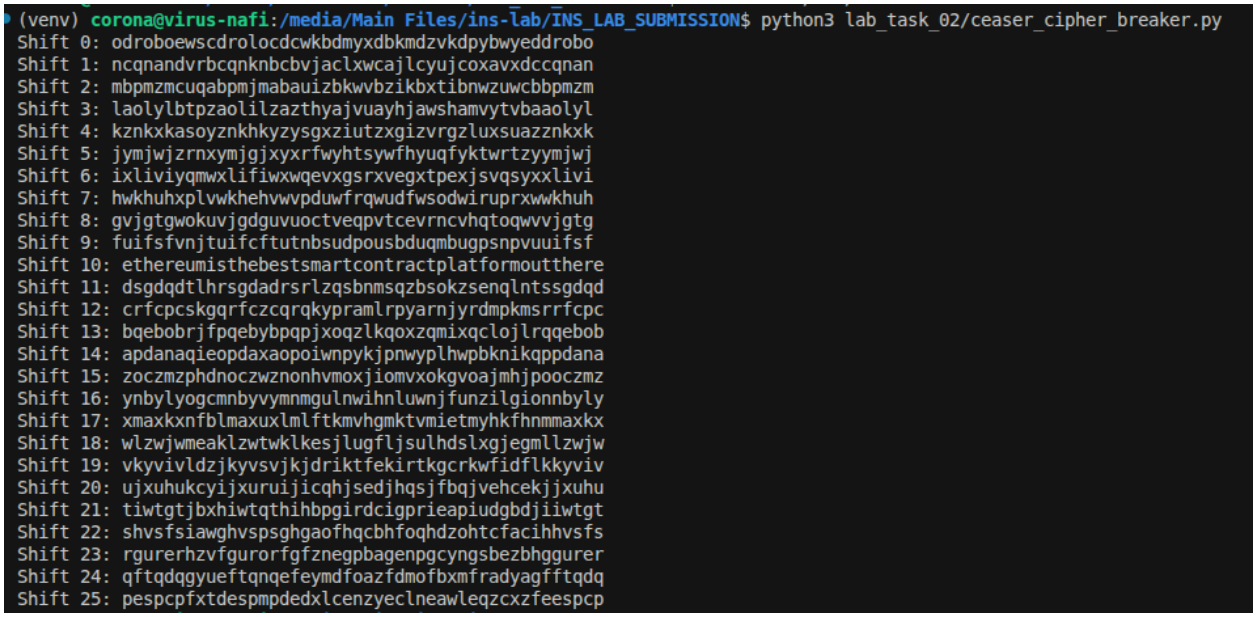
7. Conclusion

- The Caesar and Substitution ciphers are **historically significant but insecure** by modern standards.
- Even without knowing the key, **statistical analysis and programming** can recover plaintext effectively.

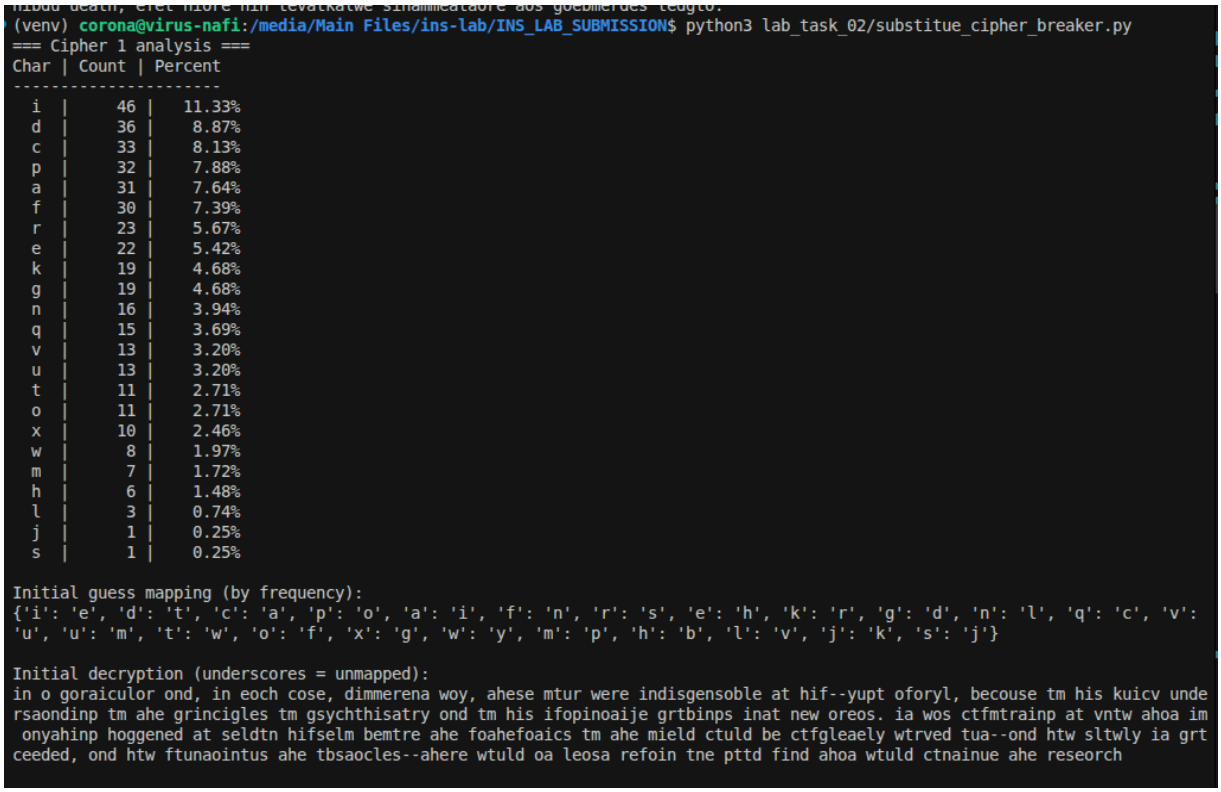
- This experiment demonstrates how **frequency analysis** breaks classical encryption schemes and why **modern cryptography** relies on computational hardness rather than obscurity.

8. Sample Screenshots

Caesar Cipher Output: showing all 26 possible shifts.



Cypher 1 analysis (Cipher vs. English).



```
(venv) corona@virus-nafi:/media/Main Files/ins-lab/INS_LAB_SUBMISSION$ python3 lab_task_02/substitue_cipher_breaker.py
=== Cipher 1 analysis ===
Char | Count | Percent
--------------------
  i  |    46 |  11.33%
  d  |    36 |   8.87%
  c  |    33 |   8.13%
  p  |    32 |   7.88%
  a  |    31 |   7.64%
  f  |    30 |   7.39%
  r  |    23 |   5.67%
  e  |    22 |   5.42%
  k  |    19 |   4.68%
  g  |    19 |   4.68%
  n  |    16 |   3.94%
  q  |    15 |   3.69%
  v  |    13 |   3.20%
  u  |    13 |   3.20%
  t  |    11 |   2.71%
  o  |    11 |   2.71%
  x  |    10 |   2.46%
  w  |     8 |   1.97%
  m  |     7 |   1.72%
  h  |     6 |   1.48%
  l  |     3 |   0.74%
  j  |     1 |   0.25%
  s  |     1 |   0.25%

Initial guess mapping (by frequency):
{'i': 'e', 'd': 't', 'c': 'a', 'p': 'o', 'a': 'i', 'f': 'n', 'r': 's', 'e': 'h', 'k': 'r', 'g': 'd', 'n': 'l', 'q': 'c', 'v':
'u', 'u': 'm', 't': 'w', 'o': 'f', 'x': 'g', 'w': 'y', 'm': 'p', 'h': 'b', 'l': 'v', 'j': 'k', 's': 'j'}

Initial decryption (underscores = unmapped):
in o goraiculor ond, in eoch cose, dimmerena woy, ahese mtur were indisgensoble at hif--yupt oforyl, becouse tm his kuicv unde
rsaondinp tm ahe grincigles tm gsychthisatry ond tm his ifopinoaije grtbinps inat new oreos. ia wos ctfmtrainp at vntw ahoa im
 onyahinp hoggened at seldtn hifselm bemtre ahe foahefoaics tm ahe mield ctuld be ctfgleaely wtrved tua--ond htw sltwly ia grt
ceeded, ond htw ftunaointus ahe tbsaocles--ahere wtuld oa leosa refoin tne pttd find ahoa wtuld ctnainue ahe reseorch
```

```
Final mapping (cipher->plain) for Cipher-1:
  a -> i
  c -> t
  d -> o
  e -> h
  f -> n
  g -> d
  h -> b
  i -> e
  j -> q
  k -> r
  l -> k
  m -> g
  n -> l
  o -> m
  p -> a
  q -> c
  r -> s
  s -> j
  t -> w
  u -> f
  v -> u
  w -> y
  x -> p

Final decryption for Cipher-1:
in a particular and, in each case, different way, these four were indispensable to him--yugo amaryl, because of his quick unde
rstanding of the principles of psychohistory and of his imaginatije probings into new areas. it was comforting to know that if
 anything happened to seldon himself before the mathematics of the field could be completely worked out--and how slowly it pro
ceeded, and how mountainous the obstacles--there would at least remain one good mind that would continue the research
```

Cipher 2 analysis

```
=== Cipher 2 analysis ===
Char | Count | Percent
---------------------
  u  |   198 |  12.80%
  k  |   132 |   8.53%
  o  |   131 |   8.47%
  h  |   113 |   7.30%
  c  |   102 |   6.59%
  z  |    95 |   6.14%
  m  |    95 |   6.14%
  l  |    89 |   5.75%
  v  |    85 |   5.49%
  j  |    74 |   4.78%
  e  |    71 |   4.59%
  a  |    47 |   3.04%
  q  |    42 |   2.71%
  s  |    38 |   2.46%
  w  |    38 |   2.46%
  n  |    37 |   2.39%
  t  |    34 |   2.20%
  d  |    29 |   1.87%
  g  |    28 |   1.81%
  y  |    28 |   1.81%
  p  |    22 |   1.42%
  i  |     8 |   0.52%
  r  |     7 |   0.45%
  b  |     4 |   0.26%

Initial guess mapping (by frequency):
{'u': 'e', 'k': 't', 'o': 'a', 'h': 'o', 'c': 'i', 'z': 'n', 'm': 's', 'l': 'h', 'v': 'r', 'j': 'd', 'e': 'l', 'a': 'c', 'q':
'u', 's': 'm', 'w': 'w', 'n': 'f', 't': 'g', 'd': 'y', 'g': 'p', 'y': 'b', 'p': 'v', 'i': 'k', 'r': 'j', 'b': 'x'}

Initial decryption (underscores = unmapped):
cilco gan verp riyh asd verp beyuliar, asd had cees the gosder om the nhire mor nixtp pearn, ever nisye hin rewarjacle dinabbe
arasye asd usexbeyted returs. the riyhen he had croufht cayj mrow hin traveln had sog ceyowe a loyal lefesd, asd it gan bobula
rlp celieved, ghatever the old molj wifht nap, that the hill at caf esd gan mull om tusseln ntummed gith treanure. asd im that
 gan sot esoufh mor mawe, there gan alno hin brolosfed vifour to warvel at. tiwe gore os, cut it neewed to have little emmeyt
os wr. caffisn. at sisetp he gan wuyh the nawe an at mimtp. at sisetp-sise thep cefas to yall hiw gell-brenerved; cut usyhasfe
d gould have cees searer the warj. there gere nowe that nhooj their headn asd thoufht thin gan too wuyh om a food thisf; it ne
ewed usmair that aspose nhould bonnenn (abbarestlp) berbetual pouth an gell an (rebutedlp) isexhaunticle gealth. it gill have
to ce baid mor, thep naid. it ins't satural, asd troucle gill yowe om it! cut no mar troucle had sot yowe; asd an wr. caffisn
gan feseroun gith Kin wosep, wont beoble gere gillisf to morfive hiw Kin odditien asd Kin food mortuse. he rewaised os vinitis
f terwn gith Kin relativen (exyebt, om yourne, the nayjville- caffisnen), asd he had wasp devoted adwirern awosf the hoccitn o
m boor asd usiwbortast mawilien. cut he had so ylone mriesdn, ustil nowe om Kin pousfer younisn cefas to frog ub. the eldent o
m thene, asd cilco'n mavourite, gan pousf mrodo caffisn. ghes cilco gan sisetp-sise he adobted mrodo an Kin heir, asd croufht
Kiw to live at caf esd; asd the hoben om the nayjville- caffisnen gere misallp danhed. cilco asd mrodo habbesed to have the na
we cirthdap, nebtewcer 22sd. pou had cetter yowe asd live here, mrodo wp lad, naid cilco ose dap; asd thes ge yas yelecrate ou
r cirthdap-bartien yowmortaclp tofether. at that tiwe mrodo gan ntill is Kin tgeesn, an the hoccitn yalled the irrenbosnicle t
gestien cetgees yhildhood asd yowisf om afe at thirtp-three
```

```
Final mapping (cipher->plain) for Cipher-2:
  a -> b
  b -> x
  c -> i
  d -> c
  e -> l
  g -> y
  h -> o
  i -> j
  j -> d
  k -> t
  l -> h
  m -> n
  n -> g
  o -> a
  p -> v
  q -> u
  r -> k
  s -> f
  t -> w
  u -> e
  v -> r
  w -> m
  y -> p
  z -> s

Final decryption for Cipher-2:
bilbo was very rich and very peculiar, and had been the wonder of the shire for sixty years, ever since his remarkable disappe
arance and unexpected return. the riches he had brought back from his travels had now become a local legend, and it was popula
rly believed, whatever the old folk might say, that the hill at bag end was full of tunnels stuffed with treasure. and if that
 was not enough for fame, there was also his prolonged vigour to marvel at. time wore on, but it seemed to have little effect
on mr. baggins. at ninety he was much the same as at fifty. at ninety-nine they began to call him well-preserved; but unchange
d would have been nearer the mark. there were some that shook their heads and thought this was too much of a good thing; it se
emed unfair that anyone should possess (apparently) perpetual youth as well as (reputedly) inexhaustible wealth. it will have
to be paid for, they said. it isn't natural, and trouble will come of it! but so far trouble had not come; and as mr. baggins
was generous with his money, most people were willing to forgive him his oddities and his good fortune. he remained on visitin
g terms with his relatives (except, of course, the sackville- bagginses), and he had many devoted admirers among the hobbits o
f poor and unimportant families. but he had no close friends, until some of his younger cousins began to grow up. the eldest o
f these, and bilbo's favourite, was young frodo baggins. when bilbo was ninety-nine he adopted frodo as his heir, and brought
him to live at bag end; and the hopes of the sackville- bagginses were finally dashed. bilbo and frodo happened to have the sa
me birthday, september 22nd. you had better come and live here, frodo my lad, said bilbo one day; and then we can celebrate ou
r birthday-parties comfortably together. at that time frodo was still in his tweens, as the hobbits called the irresponsible t
weenties between childhood and coming of age at thirty-three
```

Final Key Maps printed in console.