Lab 2 Report: Attacking Classic Crypto Systems

Checkpoint 1: Caesar Cipher Attack

Problem Description

The ciphertext provided was:

```
odroboewscdrolocdcwkbdmyxdbkmdzvkdpybwyeddrobo
```

This cipher was created using the Caesar cipher, which is a monoalphabetic substitution cipher where each letter in the plaintext is shifted by a fixed number of positions in the alphabet.

Approach

The Caesar cipher has a fundamental weakness: there are only 26 possible keys (shifts from 0 to 25). This makes it vulnerable to a **brute force attack**, where we can simply try all possible shifts and identify the correct decryption.

Implementation Strategy

1. **Brute Force Method**: Since there are only 26 possible shifts (one for each letter in the alphabet), the most straightforward approach is to try all possible shifts and examine the results.
2. **Decryption Function**: The decryption function works by:
   - Taking each character in the ciphertext
   - If it's a letter, shifting it backward by the specified number of positions
   - Wrapping around the alphabet (using modulo 26)
   - Preserving non-alphabetic characters as-is
3. **Key Identification**: After trying all 26 shifts, the correct plaintext can be identified by:
   - Looking for readable English text
   - Checking for common words and patterns

    - Verifying that the decrypted text makes sense

Code Implementation

The solution implemented a simple Python function that:

- Iterates through all possible shifts (0-25)
- Decrypts the ciphertext for each shift
- Displays all results for manual inspection

Results

After trying all 26 possible shifts, the correct decryption was found at **shift 10**:

Decrypted Message:

```
ethereumisthebestsmartcontractplatformoutthere
```

This translates to: “ethereum is the best smart contract platform out there”

Analysis

The Caesar cipher was easily broken because:

1. **Small Key Space**: Only 26 possible keys make brute force attacks trivial
2. **No Key Management**: The key is simply a number from 0-25
3. **Deterministic**: The same letter always maps to the same ciphertext letter
4. **No Diffusion**: The structure of the plaintext is preserved in the ciphertext

---

Checkpoint 2: Substitution Cipher Attack

Problem Description

Two ciphertexts were provided, both encrypted using substitution ciphers with different keys:

Cipher-1 (Length: 495 characters):

```
af p xpkcaqvnpk pfg, af ipqe qpri, gauuikifc tpw, ceiri udvk tiki afgarxifrphni
cd eao--wvmd popkwn, hiqpvri du ear jvaql vfgikrcpfgafm du cei xkafqaxni
r du xrwqedearcdkw pfg du ear aopmafpcasi xkdhafmr afcd fit pkipr. ac tpr
qdoudkcafm cd lfdt cepc au pfwceafm epxxifig cd ringdf eaorinu hiudki cei
```

```
opceiopcaqr du cei uaing qdvng hi qdoxnicinw tdklig dvc--pfg edt rndtnw
ac xkdqiigig, pfg edt odvfcpafdvr cei dhrcpqnir--ceiki tdvng pc niprc kiopaf
dfi mddg oafg cepc tdvng qdfcafvi cei kiripkqe
```

Cipher-2 (Length: 1948 characters):

```
aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlc
vu shv zcbkg guovz, upuv zcmdu lcz vuwovroaeu jczoyyuovomdu omj qmu
byudkuj vukqvm. klu vcdluz lu loj avhqnlk aodr svhw lcz kvopuez loj mht au
dhwu o ehdoe eunumj, omj ck toz yhyqeoveg auecupuj, tlokupuv klu hej sh
er wcnlk zog, klok klu lcee ok aon umj toz sqee hs kqmmuez zkqssuj tckl k
vuozqvu. omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lcz yvhehm
nuj pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu us
sudk hm wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok m
cmukg-mcmu klug aunom kh doee lcw tuee-yvuzuvpuj; aqk qmdlomnuj thq
ej lopu auum muovuv klu wovr. kluvu tuvu zhwu klok zlhhr klucv luojz omj k
lhqnlk klcz toz khh wqdl hs o nhhj klcmn; ck zuuwuj qmsocv klok omghmu
zlhqej yhzzuzz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg) cm
ubloqzkcaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqv
oe, omj kvhqaeu tcee dhwu hs ck! aqk zh sov kvhqaeu loj mhk dhwu; omj o
z wv. aonncmz toz numuvhqz tckl lcz whmug, whzk yuhyeu tuvu tceecmn
kh shvncpu lcw lcz hjjckcuz omj lcz nhhj shvkqmu. lu vuwocmuj hm pczckc
mn kuvwz tckl lcz vueokcpuz (ubduyk, hs dhqvzu, klu zodrpceeu- aonncm
zuz), omj lu loj wamg juphkuj ojwcvuvz owhmn klu lhaackz hs yhhv omj qm
cwyhvkomk sowcecuz. aqk lu loj mh dehzu svcumjz, qmkce zhwu hs lcz g
hqmnuv dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z soph
qvcku, toz ghqmn svhjh aonncmz. tlum aceah toz mcmukg-mcmu lu ojhyku
j svhjh oz lcz lucv, omj avhqnlk lcw kh ecpu ok aon umj; omj klu lhyuz hs kl
u zodrpceeu- aonncmzuz tuvu scmoeeg jozluj. aceah omj svhjh loyyumuj k
h lopu klu zowu acvkljog, zuykuwauv 22mj. ghq loj aukkuv dhwu omj ecpu l
uvu, svhjh wg eoj, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv ac
vkljog-yovkcuz dhwshvkoaeg khnukluv. ok klok kcwu svhjh toz zkcee cm l
cz ktuumz, oz klu lhaackz doeeuj klu cvvuzyhmzcaeu ktumkcuz auktuum dl
cejlhhj omj dhwcmn hs onu ok klcvkg-klvuu
```

Approach

Substitution ciphers are more complex than Caesar ciphers because they have a much larger key space (26! ≈ 4 × 10^{26} possible keys). However, they can still be broken using **frequency analysis** combined with **pattern recognition**.

Step 1: Frequency Analysis

The first step was to calculate the frequency distribution of characters in each ciphertext and compare them side-by-side with the known frequency distribution of English characters. This comparison helps identify potential character mappings based on statistical similarity.

Frequency Analysis for Cipher-1:

The frequency distribution was calculated and compared with English letter frequencies:

Cipher-1 Character	Frequency	English Character	Expected Frequency
`i`	11.33%	`e`	12.22%
`d`	8.87%	`t`	9.67%
`c`	8.13%	`a`	8.05%
`p`	7.88%	`o`	7.63%
`a`	7.64%	`i`	6.28%
`f`	7.39%	`n`	6.95%

Frequency Analysis for Cipher-2:

Similarly, for Cipher-2, the frequency distribution showed:

Cipher-2 Character	Frequency	English Character	Expected Frequency
`u`	12.81%	`e`	12.22%
`k`	8.54%	`t`	9.67%
`o`	8.34%	`a`	8.05%
`h`	7.37%	`o`	7.63%
`c`	6.60%	`i`	6.28%
`z`	6.14%	`n`	6.95%

This side-by-side comparison provided the foundation for creating initial character mappings.

Step 2: Initial Mapping

Based on frequency analysis, an initial character mapping was created by matching:

- The most frequent ciphertext character → most frequent English character (`e`)
- Second most frequent → second most frequent (`t`)
- And so on...

However, this initial mapping was often incomplete or incorrect, requiring refinement.

Step 3: Pattern Recognition

After frequency analysis, the next step was to identify common patterns (bigrams, trigrams, and repeated words) in the ciphertext. This pattern recognition is crucial because it provides concrete evidence for character mappings.

For Cipher-1:

Common patterns identified and their likely plaintext equivalents:

- **`cei`** → **`the`**: This trigram appeared frequently and is the most common trigram in English. This established: `c` → `t`, `e` → `h`, `i` → `e`
- **`pfg`** → **`and`**: A very common trigram. This established: `p` → `a`, `f` → `n`, `g` → `d`
- **`af`** → **`in`**: A common bigram. Combined with previous mappings, this confirmed: `a` → `i`, `f` → `n`
- **`du`** → **`of`**: Another common bigram. This established: `d` → `o`, `u` → `f`
- **`ac`** → **`it`**: Common bigram, consistent with `a` → `i` and `c` → `t`
- **`cd`** → **`to`**: Common bigram, consistent with `c` → `t` and `d` → `o`

For Cipher-2:

Common patterns identified and their likely plaintext equivalents:

- **`klu`** → **`the`**: This trigram appeared very frequently throughout the text. This established: `k` → `t`, `l` → `h`, `u` → `e`
- **`toz`** → **`was`**: A common word. This established: `t` → `w`, `o` → `a`, `z` → `s`
- **`omj`** → **`and`**: A very common trigram. This established: `o` → `a`, `m` → `n`, `j` → `d`

- `puvg` → `very`: Common word. This established: `p` → `v`, `g` → `y`
- `vcdl` → `rich`: Common word. This established: `v` → `r`, `d` → `c`
- `upuv` → `ever`: Common word, consistent with `u` → `e`, `p` → `v`, `v` → `r`

Step 4: Iterative Refinement Process

The initial frequency-based mapping was refined through an iterative process. This involved multiple rounds of:

1. **Applying partial mappings**: Using the mappings established from pattern recognition to partially decrypt the text
2. **Analyzing partial decryptions**: Examining the partially decrypted text to identify new patterns
3. **Refining mappings**: Adding new character mappings based on the partially decrypted text
4. **Repeating**: Continuing this process until the full mapping is complete

Example from Cipher-1 Round 1:
After initial mappings (`cei` → `the`, `pfg` → `and`, `af` → `in`), the partial decryption revealed:

- `gauuikifc` → `_iffe_ent`, suggesting `g` → `d` and `k` → `r`
- `ipqe` → `ea_h`, suggesting `q` → `c`
- `pfwceafm` → `an_thin_`, suggesting `w` → `y` and `m` → `g`

Example from Cipher-2 Round 1:
After initial mappings (`klu` → `the`, `toz` → `was`, `omj` → `and`), the partial decryption revealed:

- `puvg` → `_e__`, suggesting `p` → `v` and `g` → `y`
- `vcdl` → `___h`, suggesting `v` → `r`, `c` → `i`, `d` → `c`
- `yudqecov` → `_e____a_`, suggesting `y` → `p`, `q` → `u`, `e` → `l`

This iterative approach allowed for systematic refinement of the character mappings.

Implementation Details

The solution involved:

1. Calculating character frequencies for both ciphertexts

2. Creating initial mappings based on frequency analysis
3. Applying the mappings to generate partially decrypted text
4. Identifying common patterns (bigrams, trigrams, common words)
5. Refining the mappings based on pattern recognition
6. Manually correcting any remaining errors

Results

Cipher-1 Final Mapping:

Complete character substitution mapping:

- `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:

```
in a particular and, in each case, different way, these four were indispensa
ble to him--yugo amaryl, because of his quick understanding of the principl
es of psychohistory and of his imaginatije probings into new areas. it was c
omforting to know that if anything happened to seldon himself before the m
athematics of the field could be completely worked out--and how slowly it
proceeded, and how mountainous the obstacles--there would at least rema
in one good mind that would continue the research
```

Cipher-2 Final Mapping:

Complete character substitution mapping:

- `u` → `e`, `k` → `t`, `l` → `h` (from `klu` → `the`)
- `t` → `w`, `o` → `a`, `z` → `s` (from `toz` → `was`)
- `m` → `n`, `j` → `d` (from `omj` → `and`)
- `v` → `r`, `d` → `c`, `p` → `v`, `g` → `y`, `y` → `p`, `q` → `u`
- `w` → `m`, `r` → `k`, `s` → `f`, `n` → `g`, `i` → `j`, `b` → `x`

Decrypted Cipher-2:

```
bilbo was very rich and very peculiar, and had been the wonder of the shire
for sixty years, ever since his remarkable disappearance and unexpected r
eturn. the riches he had brought back from his travels had onw become a l
ocal legend, and it was popularly 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 blso 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-pr
eserved; but unchanged would have been nearer the mark. there were som
e that shook their heads and thought this was too much of a good thing; it s
eemed unfair that anyone should possess (apparently) perpetual youth as
well as (reputedly) inexhaustible wealth. it will have to be paid for, they sai
d. it isn't natural, and trouble will come of it! but so far trouble had oot com
e; and as mr. baggins was generous with jis money, most people were willi
ng to forgive him jis oddities and jis good fortune. he remained on visiting t
erms with jis relatives (except, of course, the sackville- bagginses), and he
had many devoted admirers bmong the hobbits of poor and unimportant fa
milies. but he had no close friends, until some of jis younger cousins began
to grow up. the eldest of these, and bilbo's favourite, was yong frodo baggi
ns. when bilbo was ninety-nine he adopted frodo as jis heir, and brought ji
m to live at bag end; and the hopes of the sackville- bagginses were finally
dashed. bilbo and frodo happened to have the same birthday, september 2
2nd. you had better come and live here, frodo my lad, said bilbo one day; a
nd then we can celebrate our birthday-parties comfortably together. at that
time frodo was still in jis tweens, as the hobbits called the irresponsible twe
nties between childhood and coming of age at thirty-three
```

Note: The decrypted text contains some minor errors (e.g., “onw” instead of “now”, “blso” instead of “also”, “jis” instead of “his”, “bmong” instead of “among”, “yong” instead of “young”, “jim” instead of “him”), which are likely due to incomplete mapping or typos in the original ciphertext. However, the overall meaning is clear, and the text is recognizable as an excerpt from “The Hobbit” by J.R.R. Tolkien.

Which Cipher Was Easier to Break?

Cipher-2 was significantly easier to break for the following reasons:

1. **Length**: Cipher-2 is much longer (1948 characters) compared to Cipher-1 (495 characters). Longer texts provide:
   - More statistical data for frequency analysis, leading to more accurate frequency distributions
   - More context for pattern recognition, with more instances of common words and phrases
   - More instances of repeated words (like "the", "and", "was") that can be used to verify mappings
   - Better validation of mappings through cross-referencing multiple occurrences
2. **Pattern Recognition**: The longer text in Cipher-2 allows for:
   - Better identification of common bigrams and trigrams with higher confidence
   - More instances of repeated words (like "the", "and", "was") for verification
   - Clearer frequency distributions that are closer to expected English frequencies due to larger sample size
3. **Error Correction**: With more text, errors in initial frequency-based mapping can be more easily identified and corrected through:
   - Cross-referencing multiple occurrences of words to verify mappings
   - Pattern matching with known text structures and literary context
   - Statistical validation across the entire text, allowing for more reliable frequency analysis

---

Conclusion

Both cryptographic systems were successfully broken, demonstrating their fundamental weaknesses:

1. **Caesar Cipher**: Vulnerable to brute force attacks due to its small key space (only 26 possible keys).
2. **Substitution Cipher**: Vulnerable to frequency analysis and pattern recognition, especially when:

- The ciphertext is long enough to provide reliable statistical data
- The plaintext follows standard English letter frequency distributions
- Common words and patterns can be identified

The key takeaway is that **classical cryptographic systems are fundamentally insecure** because they preserve patterns from the plaintext in the ciphertext. Modern cryptography addresses these weaknesses through:

- Larger key spaces
- Diffusion (spreading plaintext influence across ciphertext)
- Confusion (making the relationship between key and ciphertext complex)
- Secure key management