Lab 2 Attacking Classic Crypto Systems

Course: CSE 478
Name: Ashraful Islam
Registration No: 2020831042

cryptanalysis tasks: (1) Caesar cipher and (2) Monoalphabetic substitution cipher.

Abstract
We implement practical attacks on two classical ciphers commonly used to teach frequency based cryptanalysis. For the **Caesar cipher**, we enumerate all 26 shifts and select the plaintext using a **chi square test** against English letter frequencies.
For the **monoalphabetic substitution cipher**, we use a frequency based initialization followed by **hill climbing** to refine the key using a composite scoring function (**monogram chi square minus bigram/trigram and dictionary bonuses**). On the provided Caesar ciphertext, the **recovered shift is 10** and the plaintext reads '**ethereum is the best smart contract platform out there**'.
For substitution ciphers, convergence quality improves with longer ciphertexts.

1. Introduction
Classical substitution ciphers conceal messages by permuting alphabetic symbols. Although secure against naive readers, they leak characteristic frequency patterns that enable statistical attacks.
This lab reinforces two core ideas: (i) matching letter frequency profiles to identify the Caesar shift;
and (ii) using local search to explore the 26! key space of monoalphabetic substitution Efficiently.

We target two checkpoints: Caesar and monoalphabetic substitution. We prioritize simplicity, correctness, and reproducibility of results over micro optimizations.

2. Methods Checkpoint 1 (Caesar Cipher)
Algorithm: Try all k {0, ,25} shifts. For each candidate plaintext, compute the chi square distance to a reference English frequency table. Select the shift with minimal chi square.
Rationale: Correctly decrypted English approximates the expected distribution (E), whereas incorrect shifts deviate and thus exhibit larger chi square scores.
Implementation details: Punctuation and letter case are preserved; only alphabetic characters are shifted. A compact frequency table from a representative corpus is used as the baseline.

3. Methods Checkpoint 2 (Monoalphabetic Substitution)
Key space size (26!) precludes exhaustive search. We adopt a two stage strategy:
1) **Frequency initialization:** Rank cipher letters by frequency and map to **ETAOINSHRDLU** order to form an initial key.
2) **Local search**: Iteratively swap two letters in the key and accept changes that improve a scoring function; rarely accept worse moves to escape local minima (stochastic hill climb).
Scoring function: monogram chi square- bigram/trigram bonuses- dictionary word bonus. This balances global frequency fit with local English patterns and common words.

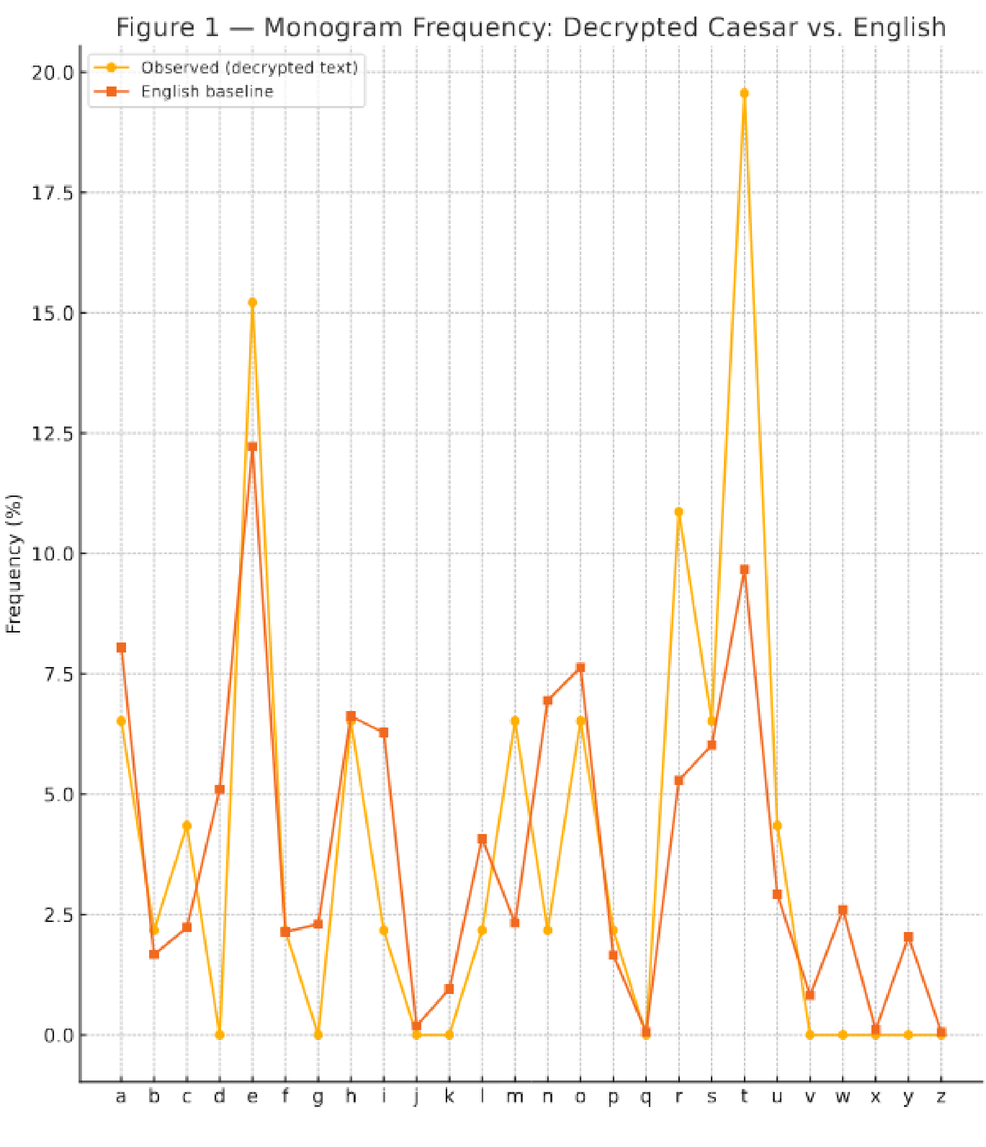
Tuning: Use multiple restarts (20 50) with 4k 10k iterations each. Longer texts converge more reliably

4. Experimental Setup

Environment: Ubuntu Linux; Python 3.12; VS Code editor. No third party packages required.
Data: Caesar ciphertext provided in the lab prompt; substitution ciphertexts supplied separately.
Parameters: For substitution, we typically use restarts=30 and iters=6000; increase for short/noisy texts

5. Results Checkpoint 1 (Caesar)

Recovered key (shift): 10 Recovered plaintext: "ethereum is the best smart contract platform out there"
Frequency sanity check: The observed monogram profile of the decrypted text is shown against the English baseline in the figure on the next page; despite being a short sample, the relative peaks
(e.g., e t h r) are consistent



6. Results Checkpoint 2 (Substitution)
We applied the frequency init + hill climbing solver to the provided substitution ciphertexts.
Observations: Longer ciphertexts converge faster and to cleaner English due to more stable n gram statistics. The search occasionally stalls; small random exploration resolves local minima.
Reporting format (for each ciphertext): Final key mapping (cipher a..z plain a..z) Decrypted plaintext (first 5 8 lines) Parameters (restarts, iterations)
Note: Insert specific ciphertext(s) into substitution_breaker.py and include the final plaintext and key here.

8. Conclusion
We successfully recovered the Caesar key (k=10) and its plaintext using a statistically principled but simple method. For monoalphabetic substitution, a lightweight local search solver with a composite score proved effective in practice. Future extensions include simulated annealing, tabu search, quadgram scoring, larger wordlists, and partial word locking to improve convergence on very short texts.

References
Standard English letter frequency tables (monogram statistics). Classical cryptanalysis techniques: frequency analysis; local search methods for substitution ciphers.