Caesar Cipher Cryptanalysis Report

Training Exercise: CS101_BASIC_CIPHER_BROKEN

Prepared for Basic Cryptanalysis Training

October 21, 2025

1 Case Overview

Target: Two messages encrypted using the Caesar Cipher

Objective: Decrypt the encrypted messages and recover the original plaintext

Flag: CS101_BASIC_CIPHER_BROKEN

Context: Basic cryptanalysis training exercise

This report details the successful decryption of two Caesar Cipher-encrypted messages using brute-force and known-plaintext attack techniques. The exercise highlights the vulnerabilities of classical substitution ciphers and underscores the importance of modern cryptographic methods.

2 Background

The Caesar Cipher is a monoalphabetic substitution cipher that shifts each letter in the plaintext by a fixed number of positions in the alphabet (A=0, B=1, ..., Z=25). With only 25 possible shifts, the cipher's limited keyspace makes it susceptible to brute-force attacks and frequency analysis.

3 Original Plaintext

Message: MEET ME AT THE PARK TODAY
This known plaintext serves as a reference for verifying decryption results.

4 Encryption Process

4.1 Cipher 1: Caesar Cipher (Key = 3)

Encryption Rule: Shift each letter forward by 3 positions
Mapping:

- MEET → PHHW
- ME → PH
- AT → DW
- THE → WKH
- PARK → SDUN
- TODAY → WRGDB

Full Ciphertext 1: PHHW PH DW WKH SDUN WRGDB
Secret Key: 3

4.2 Cipher 2: Caesar Cipher (Key = 7)

Encryption Rule: Shift each letter forward by 7 positions
Mapping:

- MEET → TLLA
- ME → TL
- AT → HA
- THE → AOL
- PARK → WHYR
- TODAY → AVKHF

Full Ciphertext 2: TLLA TL HA AOL WHYR AVKHF
Secret Key: 7

5 Decryption Process

5.1 Cipher 1 Decryption (Key = 3)

Decryption Rule: Shift each letter backward by 3 positions
Mapping:

- PHHW → MEET
- PH → ME
- DW → AT
- WKH → THE
- SDUN → PARK
- WRGDB → TODAY

Recovered Plaintext: MEET ME AT THE PARK TODAY

5.2 Cipher 2 Decryption (Key = 7)

Decryption Rule: Shift each letter backward by 7 positions
Mapping:

- TLLA → MEET
- TL → ME
- HA → AT
- AOL → THE
- WHYR → PARK
- AVKHF → TODAY

Recovered Plaintext: MEET ME AT THE PARK TODAY

6 Cryptanalysis Methodology

6.1 Cipher 1: Brute-Force Attack

Ciphertext: PHHW PH DW WKH SDUN WRGDB
Approach: Sequentially tested all possible keys (1 to 25)

- Key 1: OGGV OG CV VJG RCTM VQFCA (incorrect)
- Key 2: NFFU NF BU UIF QBSL UPEBZ (incorrect)
- Key 3: MEET ME AT THE PARK TODAY (correct)

Result: Key 3 successfully recovered the plaintext.

6.2 Cipher 2: Known-Plaintext Attack

Ciphertext: TLLA TL HA AOL WHYR AVKHF
Approach: Hypothesized the message begins with "MEET" based on Cipher 1's plaintext.

- Compared T $\rightarrow$ M: 7 positions back
- Applied Key 7 to the entire ciphertext, yielding the correct plaintext: MEET ME AT THE PARK TODAY

Result: Key 7 confirmed via known-plaintext analysis.

7 Mathematical Foundation

The Caesar Cipher relies on modular arithmetic over the 26-letter alphabet.

7.1 Encryption Formula

$$C = (P + K) \mod 26$$

- C: Ciphertext letter (0 to 25)
- P: Plaintext letter (0 to 25)
- K: Key (shift value)

7.2 Decryption Formula

$$P = (C - K) \mod 26$$

- Ensures non-negative results before applying modulo.

7.3 Example

Encryption: Letter "M" ($P = 12$) with Key = 3

$$C = (12 + 3) \mod 26 = 15 \rightarrow \text{"P"}$$

Decryption: Letter "P" ($C = 15$) with Key = 3

$$P = (15 - 3) \mod 26 = 12 \rightarrow \text{"M"}$$

8 Vulnerabilities

1. **Limited Keyspace:** Only 25 possible keys, enabling rapid brute-force attacks.
2. **Predictable Structure:** Common words (e.g., "THE") and short messages facilitate guessing.
3. **Static Keys:** Lack of key management simplifies cryptanalysis.

9 Recovered Information

Item	Value
Cipher 1 Key	3
Cipher 2 Key	7
Plaintext	MEET ME AT THE PARK TODAY
Flag	CS101_BASIC_CIPHER_BROKEN

10 Conclusion

Both Caesar ciphers were successfully decrypted using brute-force and known-plaintext attacks. This exercise demonstrates the inherent weaknesses of the Caesar Cipher, including its small keyspace and susceptibility to basic cryptanalytic techniques.

11 Security Recommendations

The Caesar Cipher is insecure for practical use due to its vulnerabilities. For secure communication, employ modern cryptographic standards such as AES, RSA, or ECC, which offer robust protection against cryptanalysis.