CptS 327 - Fundamentals of Cyber Security and Cryptography
Assignment 4

Washington State University
School of Electrical Engineering and Computer Science

Spring 2025
Due: TBD

Important Notice

Please carefully review the assignment instructions. To receive full credit, you must submit your encrypted text for Part 1, and your 2 decryption password (on the specified page) to Canvas. The use of AI is strictly prohibited!

1 Part 1: Caesar Cipher

The Caesar cipher is one of the simplest and oldest encryption techniques. It is a substitution cipher where each letter in the plaintext is shifted a fixed number of positions down the alphabet. For example, with a shift (or key) of 3, the letter `A` becomes `D`, `B` becomes `E`, and so on. Decryption simply shifts the letters back by the same number of positions.

1.1 Working Principle

Assuming the alphabet is `A-Z`, the encryption using a secret key k (suppose k is a number indicating how many letters to shift) is given by:

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

where P is the index of the plaintext letter (with $A = 0, B = 1, \ldots, Z = 25$) and C is the index of the ciphertext letter.

The decryption formula is:

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

Example:

- Plaintext: `HELLO`

- Key: $k = 3$

Encryption:

- `H` $\rightarrow$ `K`
- `E` $\rightarrow$ `H`
- `L` $\rightarrow$ `O`
- `L` $\rightarrow$ `O`
- `O` $\rightarrow$ `R`

The resulting ciphertext is `KHOOR`.

1.2 Python Implementation for the Caesar Cipher

1.2.1 Encryption Code

```
def caesar_encrypt(plaintext, key):
    """
    Encrypts the plaintext using the Caesar cipher.

    :param plaintext: The plaintext string containing only
        uppercase English letters.
    :param key: The shift key (integer).
    :return: The resulting ciphertext string.
    """
    ciphertext = ""
    for char in plaintext:
        if char.isalpha():
            # Assume all letters are uppercase
            shifted = (ord(char) - ord('A') + key) % 26
            ciphertext += chr(shifted + ord('A'))
        else:
            ciphertext += char  # Non-letter characters remain
                unchanged
    return ciphertext

# Example usage
plaintext = "HELLO"
key = 3
encrypted_text = caesar_encrypt(plaintext, key)
print("Encrypted Text:", encrypted_text)  # Output: KHOOR
```

Listing 1: Caesar Cipher Encryption

1.2.2 Decryption Code

```
def caesar_decrypt(ciphertext, key):
    """
    Decrypts the ciphertext using the Caesar cipher.

```

```
    :param ciphertext: The ciphertext string containing only
        uppercase English letters.
    :param key: The shift key (integer).
    :return: The decrypted plaintext string.
    """
    plaintext = ""
    for char in ciphertext:
        if char.isalpha():
            shifted = (ord(char) - ord('A') - key) % 26
            plaintext += chr(shifted + ord('A'))
        else:
            plaintext += char
    return plaintext

# Example usage
decrypted_text = caesar_decrypt(encrypted_text, key)
print("Decrypted Text:", decrypted_text)  # Output: HELLO
```

Listing 2: Caesar Cipher Decryption

1.3 Your Task

Please use the last digit of your WSU ID as the key to encrypt your plaintext using the Caesar Cipher. Your plaintext should be formatted as:

[First Name] [Last Name] cpts

(For example, if your name is David Jones, your plaintext would be: "david jones cpts", excluding the quotation marks). We don't include middle name here.

Encrypt the required string using the Caesar Cipher.

Submission. You then submit your encrypted text on the specified Quiz page.

2 Vigenère Cipher

The Vigenère cipher is a more complex polyalphabetic cipher compared to the Caesar cipher. It uses a keyword to determine the shift for each letter in the plaintext. Each letter in the keyword corresponds to a shift value (with `A` = 0, `B` = 1, ..., `Z` = 25). The keyword is repeated or truncated to match the length of the plaintext.

2.1 Working Principle

For a plaintext P (with length an arbitrary number) and a secret key K (with length m), the encryption for the i-th character is given by:

$$C_i = (P_i + K_{i \mod m}) \mod 26$$

where P_i is the index of the i-th plaintext letter and $K_{i \mod m}$ is the shift corresponding to the i-th letter (with wrap around) of the repeated key.

The decryption formula is:

$$P_i = (C_i - K_{i \mod m}) \mod 26$$

Example:

- Plaintext: `ATTACKATDAWN`
- Secret Key: `LEMON`

For encryption, the secret key is expanded to `LEMONLEMONLE` and each letter of the plaintext is shifted accordingly.

2.2 Python Implementation for the Vigenère Cipher

2.2.1 Encryption Code

```
def vigenere_encrypt(plaintext, key):
    """
    Encrypts the plaintext using the Vigen cipher.

    :param plaintext: The plaintext string containing only
        uppercase English letters.
    :param key: The keyword (string).
    :return: The resulting ciphertext string.
    """
    ciphertext = ""
    key = key.upper()
    key_length = len(key)
    plaintext = plaintext.upper()
    for i, char in enumerate(plaintext):
        if char.isalpha():
            # Determine shift from the key character
            shift = ord(key[i % key_length]) - ord('A')
            shifted = (ord(char) - ord('A') + shift) % 26
            ciphertext += chr(shifted + ord('A'))
        else:
            ciphertext += char  # Non-letter characters remain
                unchanged
    return ciphertext

# Example usage
plaintext = "ATTACKATDAWN"
key = "LEMON"
encrypted_text = vigenere_encrypt(plaintext, key)
print("Encrypted Text:", encrypted_text)
```

Listing 3: Vigenère Cipher Encryption

2.2.2 Decryption Code

```python
def vigenere_decrypt(ciphertext, key):
    """
    Decrypts the ciphertext using the Vigen cipher.

    :param ciphertext: The ciphertext string containing only
        uppercase English letters.
    :param key: The keyword (string).
    :return: The decrypted plaintext string.
    """
    plaintext = ""
    key = key.upper()
    key_length = len(key)
    ciphertext = ciphertext.upper()
    for i, char in enumerate(ciphertext):
        if char.isalpha():
            shift = ord(key[i % key_length]) - ord('A')
            shifted = (ord(char) - ord('A') - shift) % 26
            plaintext += chr(shifted + ord('A'))
        else:
            plaintext += char
    return plaintext

# Example usage
decrypted_text = vigenere_decrypt(encrypted_text, key)
print("Decrypted Text:", decrypted_text)
```

Listing 4: Vigenère Cipher Decryption

2.3 Your Task

Your task is to crack two given ciphertexts using the Vigen cipher – i.e., given the ciphertexts, find out the keys and the plaintexts.

We provide 4 sample ciphertext-password pairs to help you test your cracking method. You are highly recommended to write a script (code) to do it in an automated way, as our next assignment will use AWS to challenge your program.

Please note that all words in the ciphertext are in lowercase letters and the plaintexts is readable in the "English common sense". You may use an English word dictionary available for Python from online sources.

The password length **will not exceed 10 characters**. If your attempt reaches 11 characters without successfully decrypting the text, you have likely missed the correct answer.

Submission. Please submit the two cracked passwords in the designated Quiz page.

Important note. Additionally, you should consider the execution speed of your cracking program. Although there is no time limit for decryption in this assignment, next assignment does. Therefore, please optimize your program as much as possible to ensure its efficiency for future use.

Late Policy

- Submissions up to 24 hours late will incur a penalty 10%.
- Submissions received more than 24 hours late **will not be accepted** unless there is a valid reason.