

**Name: Ahmad Diab**

**Student ID: 1213295**

**Subject: Encryption**

**Course ID: COMP438**

**Title: Assignment 2**

**Encryption Methods Report**

**Question One: One-Time Pad Encryption**

**Theory:**

The One-Time Pad is a symmetric encryption method that utilizes a randomly generated key. The key must be as long as the plaintext message to ensure security. Each character or bit in the plaintext is XORed with the corresponding character or bit in the key. This method provides perfect secrecy when the key is truly random, used only once, and kept secret.

**Key features of the One-Time Pad include:**

1. Unbreakable Encryption - As long as the key is random and never reused, it is theoretically impossible to decrypt the ciphertext without the key.
2. Bitwise XOR Operation - Encryption and decryption are achieved using XOR, which is reversible.
3. Key Management Challenges - The security depends on securely generating, distributing, and storing the key.

**Implementation:**

I implemented the One-Time Pad encryption and decryption using Java. Below is a summary of the process:

1. **Key Generation:**
   * Generate a random key that matches the length of the plaintext.
   * Ensure the key is hashed and extended if necessary.
2. **Encryption Process:**
   * Convert the plaintext into bytes.
   * XOR each byte of the plaintext with the corresponding byte of the key.
   * Encode the result using Base64 for transmission.
3. **Decryption Process:**
   * Decode the ciphertext from Base64.
   * XOR the decoded bytes with the key.
   * Retrieve the original plaintext.

**Challenges Faced:**

* Handling Base64 encoding errors when input contained invalid characters.
* Ensuring that the key length matched the plaintext length.
* Implementing error handling for invalid Base64 strings.

**Example:**

Plaintext: HELLO Key: XMCKL Ciphertext: \x19\x06\x0F\x0F\x0E (encoded using Base64)

Decryption reverses this process by applying XOR with the same key.

**Question Two: Rail Fence Cipher**

**Theory:**

The Rail Fence Cipher is a form of transposition cipher that rearranges the characters of the plaintext in a zigzag pattern across multiple rails. It works as follows:

1. Write the plaintext diagonally in a zigzag pattern across the specified number of rails.
2. Read each rail row by row to form the ciphertext.
3. Decryption reconstructs the zigzag pattern and maps the ciphertext back to plaintext.

The Rail Fence Cipher relies on the number of rails as a key. It is simple to implement but vulnerable to frequency analysis and brute force attacks due to its limited key space.

**Implementation:**

I implemented the Rail Fence Cipher using Python. Below is a summary of the process:

1. **Encryption Process:**
   * Determine the zigzag pattern based on the number of rails.
   * Place the plaintext characters into the zigzag pattern.
   * Read each row sequentially to create the ciphertext.
2. **Decryption Process:**
   * Reconstruct the zigzag pattern using placeholders.
   * Fill in the placeholders with ciphertext characters.
   * Read the characters row by row to retrieve the plaintext.

**Challenges Faced:**

* Handling edge cases where plaintext lengths were not divisible by the number of rails.
* Implementing the zigzag pattern indexing to accurately read and write characters.
* Optimizing the algorithm to handle longer plaintexts.

**Example:**

Plaintext: HELLOWORLD Rails: 3 Zigzag Pattern:

H O L

E L W R D

L O

Ciphertext: HOL ELWRD LO

Decryption reconstructs the zigzag pattern and reads it diagonally.

**Comparison and Observations**

1. **Security:**
   * One-Time Pad offers perfect secrecy but faces challenges in key management.
   * Rail Fence Cipher is easy to break due to its small key space.
2. **Complexity:**
   * One-Time Pad requires more computational resources for random key generation.
   * Rail Fence Cipher is computationally simpler and faster.
3. **Practical Use Cases:**
   * One-Time Pad is ideal for scenarios requiring high-level security, such as diplomatic communications.
   * Rail Fence Cipher can be used in low-security applications or as a teaching tool for encryption basics.

**Conclusion**

In this report, I implemented and tested two encryption techniques: the One-Time Pad and the Rail Fence Cipher. The One-Time Pad provides theoretical security but demands robust key management, while the Rail Fence Cipher offers simplicity but is susceptible to attacks. Both implementations highlighted different aspects of cryptography, from perfect secrecy to transposition-based encryption. These examples illustrate the trade-offs between security, complexity, and usability in encryption systems.