Cryptography System Report

Aditya Yadav (2021374)
Himanshu Choudhary (2021391)

February 1, 2024

1 Hash Function

The system utilizes the SHA-256 hash function to ensure the integrity of plaintext data. This widely adopted hashing algorithm produces a 64-character hexadecimal string.

1.1 Custom Mapping

A custom mapping is applied to the resulting hash value, representing each hexadecimal character with a corresponding character from 'a' to 'p'. This mapping aims to enhance efficiency and readability. Each hexadecimal character is replaced with a corresponding character according to a predefined mapping.

2 Encryption Process

The encryption process combines the plaintext with its hash value. A fixed key of size 9 is used for transposition encryption. This process adds an extra layer of security by introducing complexity and variability to the encryption.

3 Transposition Encryption

Transposition encryption is implemented using the fixed key. The plaintext is padded if its length is not a multiple of the key size. The transposition matrix is then created, and encryption is performed by rearranging the characters based on the specified key.

4 Decryption Process

The decryption process is the reverse of the encryption process, using the same fixed key. The transposition matrix is reconstructed, and the original plaintext is obtained. This process demonstrates the system's ability to securely encrypt and decrypt information.

5 Hash Verification

To validate the integrity of the decrypted text, hash verification is crucial. The check function compares the hash of the decrypted text with the extracted hash value, ensuring the authenticity and correctness of the decrypted data.

6 Brute Force Attack

A brute-force attack is implemented to demonstrate the robustness of the system. Initially designed to try keys of lengths 1 to 9, the key size was reduced to 4 to ensure reasonable running times. The system systematically attempts all permutations of keys to decrypt the ciphertext. The results of each attempt, including the key, deciphered text, and check result, are recorded in a file named `decryption_results.txt`.

7 Methodology

7.1 Custom String Creation

The Cryptography System employs a set of custom strings for encryption and decryption processes. These strings, denoted as $s1$ to $s5$, are manually created with meaningful content, simulating real-world scenarios. These strings represent the plaintext data to be secured.

7.2 Hashing and Custom Mapping

For each custom string, the system utilizes the SHA-256 hash function to generate a 64-character hexadecimal hash value. To improve readability and efficiency, a custom mapping is applied to each hexadecimal character, replacing it with a corresponding character from 'a' to 'p'. This mapping is performed to reduce the length of the hash value while maintaining its fixed 64-character length.

7.3 Combining Plaintext and Hash Value

The hash value is then combined with the original plaintext to create a single string. This combined text serves as the input for the transposition encryption process.

7.4 Transposition Encryption

A fixed key of size 9 is used for transposition encryption. The combined text is encrypted using the transposition_encrypt function, which rearranges the characters based on the specified key. This process introduces complexity and variability to the encryption, enhancing the system's security.

7.5 Brute-Force Attack

The system is subjected to a brute-force attack, attempting to decrypt the encrypted text using all possible key permutations. Initially designed to try keys of lengths 1 to 9, the key size was later reduced to 4 for more practical running times. The brute_force function systematically generates all permutations of keys and decrypts the ciphertext using each permutation. The results, including the key, deciphered text, and check result, are recorded in a file named `decryption_results.txt`.

7.6 Character Range Limitation

It is important to note that the system only considers characters from 'a' to 'p' during the custom mapping process. This limitation is inherent in the mapping scheme and ensures that the resulting ciphertext remains within a defined character range.Also characters in plaintexts limits between 'a' to 'z'

This methodology ensures a clear understanding of the steps involved in the Cryptography System, from custom string creation to encryption and the execution of a brute-force attack.

8 Results

The Cryptography System has been tested and found to work as expected. The implemented security measures, including hash verification and transposition encryption, effectively safeguard sensitive information.

It's worth noting that the key size in the transposition encryption process can be increased to enhance security further. However, this comes at the cost of longer brute-force attack times. The time complexity of the brute force attack is proportional to the factorial of the key size, making larger key sizes computationally more intensive to crack.

9 Security Considerations

The security of the system is discussed, taking into account the strengths of hash functions and transposition encryption. The implementation of a strong hash function and transposition encryption contributes to the system's robustness against various attacks.

While the system is designed to resist brute-force attacks, it's essential to acknowledge that if the key size or the range of possible key sizes is known, the encryption becomes computationally breakable. However, it's important to note that accurately determining the key size in a real-world scenario is often challenging. The system assumes a strong level of secrecy regarding the key size, making it more resilient against attacks.

Potential vulnerabilities associated with brute-force attacks are addressed, highlighting the necessity for strong key generation and verification methods to ensure the overall security of the Cryptography System.