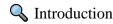
# Monoalphabetic Cipher Encryption



In this project, I implemented a monoalphabetic cipher to encrypt messages using all ASCII characters (0-255). Unlike traditional monoalphabetic ciphers that only shuffle letters, this version replaces every possible ASCII character with another, creating a far more secure and versatile encryption method.

## **\$\text{How It Works}**

Imagine you have a secret code that swaps every letter in your message with another letter. The idea is simple: take the entire alphabet and shuffle it randomly. Then, use this shuffled alphabet to replace each letter in your message.

## **\** Key Idea

Instead of shuffling just the alphabet, we create a random substitution table for all 256 ASCII characters (letters, numbers, symbols, etc.). Each character is mapped to a unique substitute, turning this into a true monoalphabetic cipher.

#### **Monoalphabetic Cipher**

Works like a giant "mix-up" for all characters (letters, numbers, symbols, spaces).

Example: A could become #, 7 could become X, and even spaces turn into random symbols.

Uses a unique random key with more combinations than stars in the universe. Impossible to guess

No pattern. Every character is swapped randomly. Even the letter A might become a symbol in one message and a number in another.

#### **Example Substitution**

With a shuffled key:

$$H(72) \rightarrow \hat{I}(206)$$

, 
$$(44) \rightarrow \ddot{O}(214)$$

$$7(55) \rightarrow 1(161)$$

Space 
$$(32) \rightarrow \dot{U}(217)$$

## Why It's Secure

#### 1. Massive Key Space:

There are 256! ways to shuffle the substitution table. Brute-forcing this is computationally impossible.

#### 2. No Frequency Patterns:

Unlike letter-only ciphers, substitutions include all characters, breaking linguistic patterns (e.g., "E" isn't the most frequent character anymore).

#### 3. Full ASCII Coverage:

Encrypts letters, numbers, symbols, spaces, and even emojis (via their ASCII/Unicode representations).

### **1** Conclusion

This enhanced monoalphabetic cipher provides robust encryption by:

Using all ASCII characters for substitutions.

Leveraging a 256!-sized key space to resist brute-force attacks.

Eliminating patterns in ciphertext through true randomness.