

# **SUBSTITUTION CYPHER**

Ciphers are essential for protecting digital information in today's world.

- **Encryption**: Transforms readable data (plaintext) into unreadable format (ciphertext).
- **Decryption**: Reverses encryption, turning ciphertext back into plaintext.
- **Confidentiality**: Protects sensitive information from unauthorized access.
- **Privacy**: Ensures personal and sensitive data remain secure.
- **Data Integrity**: Maintains the accuracy and consistency of data.
- **Authentication**: Verifies the identity of users or devices.
- **Cybersecurity**: Defends against hacking and data breaches.
- **Secure Communication**: Enables safe online transactions and data exchange.

### **Substitution Cipher**

A **Substitution Cipher** is a type of encryption method where each letter in the plaintext is replaced by a different letter according to a fixed system. The key to the cipher is the set of substitutions used to replace the letters. This type of cipher can be either simple.

### **Programme for cypher**

```
import string

def substitution_cipher(text, key):

    alphabet = string.ascii_uppercase

    key_map = {alphabet[i]: key[i] for i in range(26)}

    result = ""

    for char in text.upper():

        if char in key_map:

            result += key_map[char]

        else:

            result += char # Non-alphabet characters remain unchanged

    return result
```

### **# Example Usage**

```
key = "MNBVCXZLKJHGFDSAPOIUYTREWQ"
```

```
plaintext = "HELLO"
```

```
ciphertext = substitution_cipher (plaintext, key)
```

```
print("Substitution Cipher:", ciphertext) # Output: URYYB
```

### **Applications:**

Substitution ciphers are used for basic encryption, but due to their simplicity, they are vulnerable to frequency analysis attacks, where the most common letters in the ciphertext are matched to the most common letters in the language of the plaintext.