Build Your Own Cryptography System: Cybersecurity Project Documentation

Build Your Own Cryptography System: Cybersecurity Project Documentation

Introduction

This project involves building a simple cryptography system using Python. The system allows users to encode and decode messages using a basic substitution cipher. The cipher works by shifting each character in the message by one position in the alphabet.

Code Explanation

Key Generation

The first step is to create key strings:

```
keys = 'abcdefghijklmnopqrstuvwxyz !'
```

This string includes all the characters that can be used in the message, including a space and an exclamation mark.

Value Generation

Next, we generate the values for the substitution cipher by shifting the keys:

```
values = keys[-1] + keys[0:-1]
```

This line takes the last character of the `keys` string and places it at the beginning, followed by the rest of the characters. This effectively shifts each character by one position.

Dictionary Creation

We then create two dictionaries for encoding and decoding:

```
encrytDict = dict(zip(keys, values))
decryptDict = dict(zip(values, keys))
```

- `encrytDict`: Maps each character in `keys` to the corresponding character in `values`.
- `decryptDict`: Maps each character in `values` back to the corresponding character in `keys`.

User Input

The user is prompted to enter a message and choose a mode (Encode or Decode):

```
message = input("Enter your secret message: ")
mode = input("Crypto Mode : Encode(E) OR Decode(D)")
```

Encoding and Decoding

Based on the chosen mode, the message is either encoded or decoded:

```
if mode.upper() == 'E':
    newMessage = ''.join([encrytDict[letter] for letter in message.lower()])
elif mode.upper() == 'D':
    newMessage = ''.join([decryptDict[letter] for letter in message.lower()])
else:
    print("Please try again, wrong choice entered")
```

- If the mode is 'E' (Encode), each letter in the message is replaced with its corresponding value from `encrytDict`.
- If the mode is 'D' (Decode), each letter in the message is replaced with its corresponding value from `decryptDict`.

Output

The new message is then returned and printed:

```
return newMessage.capitalize()
```

Complete Code

Here is the complete code for the cryptography system:

```
def machine():
    # creating key strings
    keys = 'abcdefghijklmnopqrstuvwxyz !'
    # auto generating the values of strings
    # value will be generated by taking last to first
    # concatenated with the rest of the string
    values = keys[-1] + keys[0:-1]

    # creating two dictionaries
    encrytDict = dict(zip(keys, values))
    decryptDict = dict(zip(values, keys))

    # user input
    message = input("Enter your secret message: ")
    mode = input("Crypto Mode : Encode(E) OR Decode(D)")

    # encode and decode
    if mode.upper() == 'E':
        newMessage = ''.join([encrytDict[letter]
                              for letter in message.lower()])
    elif mode.upper() == 'D':
        newMessage = ''.join([decryptDict[letter]
                              for letter in message.lower()])
    else:
        print("Please try again, wrong choice entered")

    return newMessage.capitalize()

print(machine())
```

Conclusion

This simple cryptography system demonstrates the basics of encoding and decoding messages using a substitution cipher. It can be further enhanced by adding more complex encryption algorithms and improving user input validation.