# 暗号システムを自作する：サイバーセキュリティプロジェクトドキュメント

---

暗号システムを自作する：サイバーセキュリティプロジェクトドキュメント

はじめに
このプロジェクトでは、Pythonを使用してシンプルな暗号システムを構築します。このシステムは、基本的な置換暗号を使用してメッセージをエンコードおよびデコードすることができます。暗号は、メッセージ内の各文字をアルファベット内で1つの位置ずつシフトすることで機能します。

コードの説明

キー生成
最初のステップはキー文字列を作成することです：

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

この文字列には、スペースと感嘆符を含む、メッセージで使用できるすべての文字が含まれています。

値生成
次に、キーをシフトすることで置換暗号の値を生成します：

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

この行は、キー文字列の最後の文字を先頭に置き、その後に残りの文字を続けます。これにより、各文字が1つの位置ずつシフトされます。

辞書の作成
次に、エンコードおよびデコード用の2つの辞書を作成します：

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

**encrytDict**：キーの各文字を対応する値の文字にマッピングします。
**decryptDict**：値の各文字を対応するキーの文字にマッピングします。

ユーザー入力
ユーザーはメッセージを入力し、モード（エンコードまたはデコード）を選択するように促されます：

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

エンコードとデコード
選択されたモードに基づいて、メッセージがエンコードまたはデコードされます：

```
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")
```

モードが'E'（エンコード）の場合、メッセージ内の各文字が**encrytDict**から対応する値に置き換えられます。モードが'D'（デコード）の場合、メッセージ内の各文字が**decryptDict**から対応する値に置き換えられます。

出力
新しいメッセージが返され、印刷されます：

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

完全なコード
以下は暗号システムの完全なコードです：

```
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())
```

結論
このシンプルな暗号システムは、置換暗号を使用してメッセージをエンコードおよびデコードする基本を示しています。より複雑な暗号化アルゴリズムを追加し、ユーザー入力の検証を改善することで、さらに強化することができます。

---