

# サイバーセキュリティプロジェクトドキュメント：暗号化

## サイバーセキュリティプロジェクトドキュメント：暗号化

### 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化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つの位置に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、メッセージで使用できるすべての文字（スペースと感嘆符を含む）が含まれています。

#### 値生成

次に、キーをシフトして置換暗号の値を生成します：

```
value = 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、残りの文字を続けます。これにより、各文字が1つの位置にシフトされます。

#### 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：

```
encryptDict = dict(zip(keys, value))
decryptDict = dict(zip(value, keys))
```

- `encryptDict` : キー内の各文字を値内の対応する文字にマッピングします。
- `decryptDict` : 値内の各文字をキー内の対応する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 message")
mode = input("enter the 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([encryptDict[letter] for letter in message.lower()])
elif mode.upper() == 'D':
    newMessage = ''.join([decryptDict[letter] for letter in message.lower()])
else:
    print("enter the correct form")
```

- モードが'E'（エンコード）の場合、メッセージ内の各文字が `encryptDict` 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、表示されます：

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

完全なコード

以下は暗号化システムの完全なコードです：

```
def machine():
    keys = 'abcdefghijklmnopqrstuvwxyz !'
    value = keys[-1] + keys[0:-1]
    encryptDict = dict(zip(keys, value))
    decryptDict = dict(zip(value, keys))
    message = input("enter your message")
    mode = input("enter the mode : encode(E) or Decode(D)")
    if mode.upper() == 'E':
        newMessage = ''.join([encryptDict[letter] for letter in message.lower()])
    elif mode.upper() == 'D':
        newMessage = ''.join([decryptDict[letter] for letter in message.lower()])
    else:
        print("enter the correct form")
    return newMessage.capitalize()

print(machine())
```

## 結論

このシンプルな暗号化システムは、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。

---