دانشگاه اصفهان

نام درس :

امنیت و رمزنگاری

عنوان پروژه :

تسهیم راز شمیر

حمید مهران فر

۴۰۰۳۶۱۳۰۵۸

ماژول make function :

در این ماژول با استفاده از کلاس MakeFunction پارامتر های مورد نیاز را دریافت کرده و تابع حداکثر از درجه t را می سازد . نحوه ساخت به این صورت است که ضریب همراه با توان در tupple ذخیره می شود . برای مثال (S ، 0) یعنی x به توان 0 ضربدر S که همان S می شود . تابع مورد نیاز لیستی از این tupple هاست . ضرایب هم به یک عدد رندوم بین ۱ و p هستند .

```
def generateFunction(self):
    for i in range(1, self.t):
        num = self.__randomGenerator()
        self.func.append((num, i))
```

سپس هر x به توان مقدار موجود در tupple میرسد و در ضریب ضرب می شود .

```
def getShares(self):
    values = []
    for x in range(self.n):
        temp = 0
        for val in self.func:
            temp += val[0]*(pow(x+1, val[1]))
        values.append(temp % self.p)
    return values
```

مقدار n شئ محاسبه میشود و برگردانده میشود .

تابع randomGenerator :

```
def __randomGenerator(self):
    return random.randint(1, self.p)
```

ماژول findSecret :

در این ماژول با استفاده از کلاس FindSecret مقادیر مورد نیاز دریافت می شود و راز S برگردانده می شود . میتوان پارامتر x را هم به عنوان ورودی وارد کرد . با استفاده از تابع زیر مقدار S محاسبه می شود .

$$f(0) = \sum_{j=0}^{k-1} y_j \prod_{\substack{m=0 \\ m \neq j}}^{k-1} \frac{x_m}{x_m - x_j}$$

تابع inverse ، معکوس ضربی xm-xj را برمیگرداند و تابع calculateMultiplication قسمت دوم فرمول بالا را محاسبه می کند . سپس تابع calculateSecret ، yi را در قسمت دوم فرمول ضرب کرده و با یکدیگر جمع می کند . در نهایت مقدار S برگردانده می شود .

```python
def inverse(self, number):
    number = number % self.p
    if number == 0:
        return 0
    for i in range(1, self.p):
        if (i * number) % self.p == 1:
            return i
    return -1

hhaa1382
def calculateMultiplication(self, i):
    temp = 1
    for j in range(self.t):
        if i != j:
            inv = self.inverse(self.x[j]-self.x[i])
            if inv == -1:
                raise Exception(f"{self.p} is not invertible for some numbers !!")
            temp *= inv * self.x[j]
    return temp % self.p

hhaa1382
def calculateSecret(self):
    s = 0
    for i in range(len(self.Y)):
        s += self.Y[i] * self.calculateMultiplication(i)
    return s % self.p
```