**Векторная схема разделения секрета Блэкли**

**Теоретические сведения.**

По-прежнему будем строить (n, k) – пороговую схему разделения секрета M, где n ≥ k, n — общее число долей, k — число долей, достаточное для восстановления секрета.

**Разделение секрета**. Доверенный центр (Трент) генерирует случайную точку в k-мерном аффинном пространстве, одной из координат которой является секрет M. Пусть это будет точка A(M, a1, a2, …, a k-1). Далее генерируются n k- мерных векторов:

Эти векторы должны обладать тем свойством, что любые k векторов из них являются линейно независимыми. По этим векторам строятся n k-мерных гиперплоскостей, проходящих через точку A:

Эти гиперплоскости αi (1 ≤ i ≤ n) являются долями.

В качестве аффинного пространства можно также рассматривать арифметическое пространство над простым полем Галуа GF(p), где выполняется условие .

**Восстановление секрета**. Любые k гиперплоскостей из αi (1 ≤ i ≤ n) пересекаются в единственной точке A, одна из координат которой является секретом M. Любые меньше чем k гиперплоскостей имеют в пересечении не менее, чем одну прямую. Для вычисления пересечения достаточно решить линейную систему уравнений из уравнений гиперплоскостей, которые предложены в качестве k долей, и из полученной точки выделить координату M.

**Листинг кода.**

import random  
import galois  
import numpy as np  
import sys  
#p - количество сторон, которые могут восстановить секрет  
#n - общее количество сторон  
#mod - поле  
#m - секрет m  
  
#class Point:  
# def \_\_init\_\_(self, m, points):  
# self.m = m  
# self.points = points  
  
def generation\_point(m, p, mod):  
 point = [m]  
 for i in range(p - 1):  
 num\_generation = random.randint(0, mod - 1)  
 point.append(num\_generation)  
 f = open('Точка.txt', 'w')  
 for i in point:  
 f.write(str(i) + ' ')  
  
  
  
  
def generation\_dol(n, p, mod):  
  
 f = open('Точка.txt', 'r')  
 point = list(map(int, f.read().split()))  
 print(point)  
  
 side = []  
 result\_secret = []  
 for i in range(n):  
 side.append([])  
 for j in range(p):  
 num\_generation = random.randint(0, mod - 1)  
 side[i].append(num\_generation)  
  
 for i in side:  
 result = 0  
 for j in range(len(i)):  
 result += i[j] \* point[j]  
 result = -result % 11  
 result\_secret.append(result)  
 i.append(result)  
 print(side)  
 f = open('Система уравнений.txt', 'w')  
 for i in range(len(side)):  
 for j in range(len(side[i])):  
 f.write(str(side[i][j]) + ' ')  
 f.write('\n')  
 f.close()  
 f = open('Секреты сторон.txt', 'w')  
 f.write("Секреты сторон:" + '\n')  
 for i in result\_secret:  
 f.write(str(i) + '\n')  
 f.close()  
  
#print(generation\_dol(4, 3, 11, [6, 4, 2]))  
  
  
def restoring\_the\_secret(mod, p):  
 side = []  
 f = open('Система уравнений.txt', 'r')  
 text = f.read()  
 text = text.split('\n')  
 for i in range(len(text)):  
 side.append(list(map(int,text[i].split())))  
  
 GF = galois.GF(mod)  
 side\_secret = []  
 side\_x = []  
 i = 0  
 while i < p:  
 lin\_side = random.choice(side)  
 if not lin\_side in side\_secret:  
 side\_secret.append(lin\_side)  
 i += 1  
 print(side\_secret)  
 for i in range(len(side\_secret)):  
 side\_x.append(side\_secret[i][-1])  
 side\_secret[i] = side\_secret[i][:-1]  
 #print(side\_secret)  
 #print(side\_x)  
 side\_x = [(i \* -1) % mod for i in side\_x]  
 #print(side\_x)  
 A = GF(side\_secret)  
 b = GF(side\_x)  
 x = np.linalg.solve(A, b)  
 print(f'Наш секрет равен = {x[0]}')  
 f = open('Сообщение.txt', 'w')  
 f.write("Сообщение = " + str(x[0]) + '\n')  
 #Необходимо добавить проверку на линейную зависимость векторов  
  
print("Введите секрет = ", end ='')  
m = int(input())  
print("Введите модуль = ", end ='')  
mod = int(input())  
print("Введите количество сторон, которые могут восстановить секрет = ", end= '')  
p = int(input())  
print("Введите общее количество сторон = ", end ='')  
n = int(input())  
  
#1 шаг мы создаем точку в пространстве  
#point = generation\_point(m, p, mod)  
#2 шаг мы создаем уравнения системы  
#side = generation\_dol(n, p, mod, point)  
#restoring\_the\_secret(side, mod, p)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 action = input(f"Введите 0 - для создания точки, 1 - для составление гиперплоскостей, 2 - для нахождения "  
 f"секрета по {p} сторонам, 3 - для выхода: ")  
 if action == "0":  
 generation\_point(m, p, mod)  
 elif action == "1":  
 generation\_dol(n, p, mod)  
 elif action == "2":  
 restoring\_the\_secret(mod, p)  
 elif action == "3":  
 sys.exit(-1)