|  |  |  |
| --- | --- | --- |
| 姓名 | 侯晨 |  |
| 学号 | 18150100002 |

**《大数据安全与隐私》课程**

**实验报告**

|  |  |
| --- | --- |
| **学号：** | 18150100002 |
| **姓名：** | 侯晨 |
| **指导教师：** | 彭延国 |
| **日期：** | 2021年5月25日 |

**实验报告撰写要求**

1. 不得抄袭、复制，一经发现取消课程实验成绩；
2. 正文小四字号，固定行距20磅，中文宋体，英文Times New Roman；
3. 每次实验起新页；
4. 实验报告电子档命名规则：“《大数据安全与隐私》实验报告-学号-姓名.doc/.docx/.pdf”；
5. A4纸单面打印，一份；
6. 实验报告作为实验验收的补充，仅提交实验报告未进行现场验收的没有实验部分成绩。

**实验一：现代密码学基础技能**

1. **实验原理：**

本实验模拟真实网络环境下安全传输加密文件，网络传输部分采用SOCKET,对文件的加密采用DES加密，采用DH密钥交换协议交换密钥；本实验采用c语言实现，模拟真实网络环境为本机向本机（自环）安全传输文件，并实现正确的加解密。

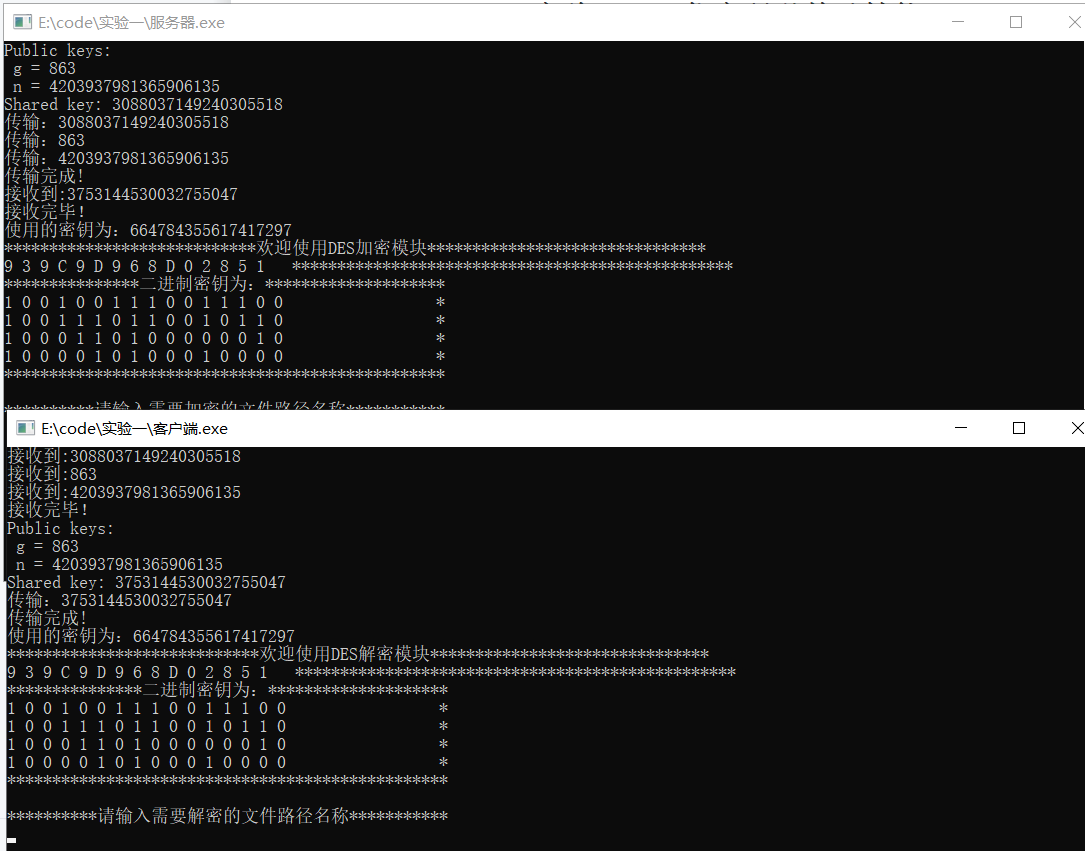
1. **实验结果：**

1 加密txt文档

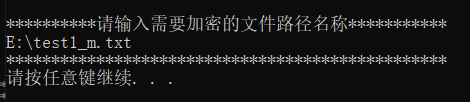
初始文档为：



运行服务器.cpp与客户端.cpp 产生并交换密钥如图：



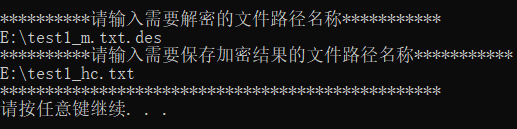
输入需要加密的文件路径



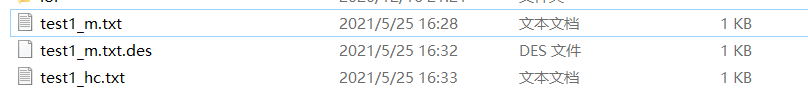
发现在当前目录下有已经加密（带有.des）后缀的文件



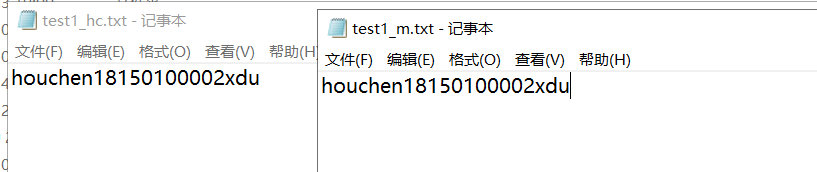
在客户端.exe中输出需要解密的加密文件路径，并输入解密后文件保存路径：



在文件中可以看到：



打开test1\_hc.txt:



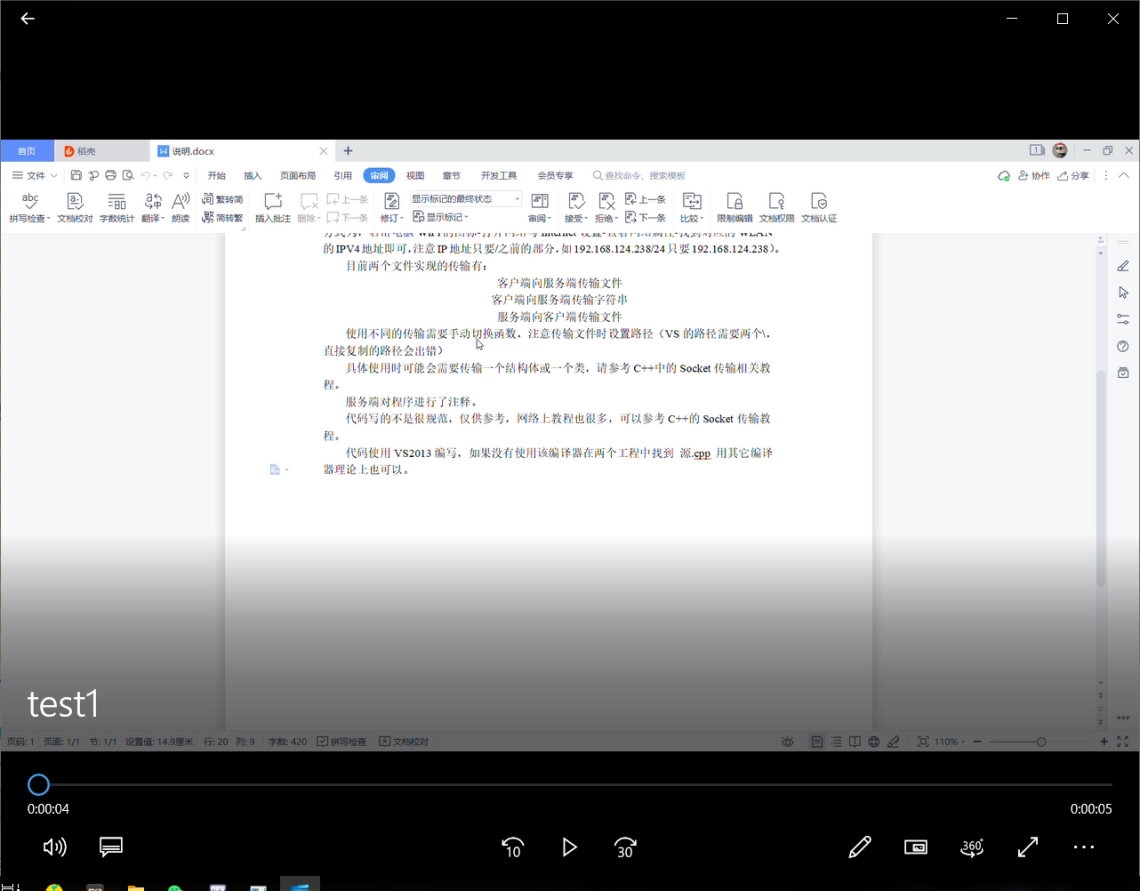
发现与原文档相同，本次实验成功！

2 加密MP4文件

初始文件为



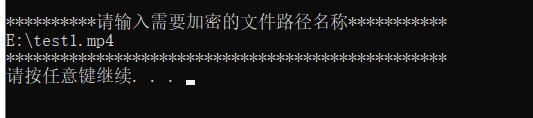
打开为：



运行服务器.cpp与客户端.cpp 产生并交换密钥如图：



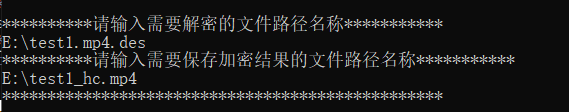
输入需要加密的文件路径：



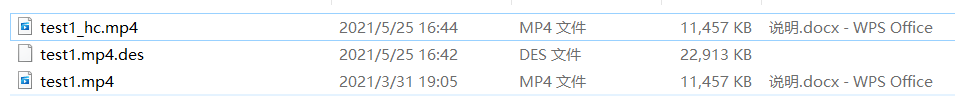
加密之后，在源文件路径下有：



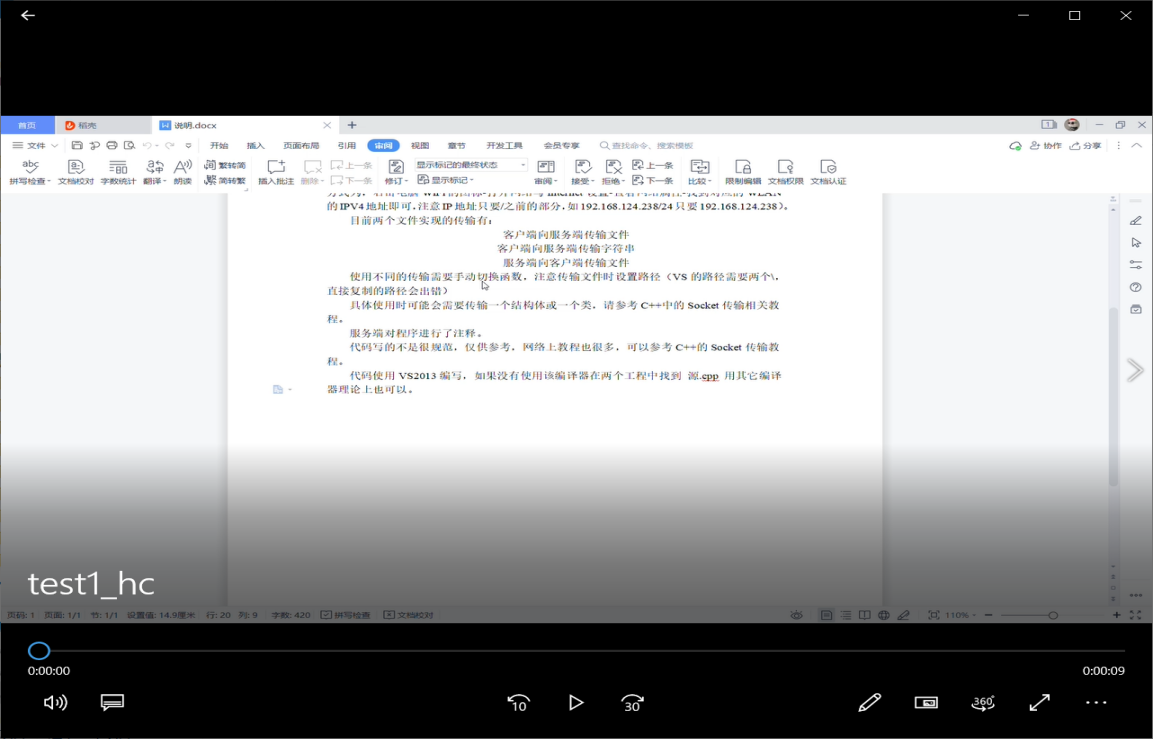
在客户端.exe中输入需要解密的des加密文件路径与解密文件存放路径：



可以看到该路径下有解密文件：



打开解密文件：



与源文件相同。

实验成功！

由此可得，该程序可以加密任何类型文件，实现真实网络环境的安全传输。

1. **实验总结与心得：**

本次实验让熟悉了真实网络传输的具体流程以及加密的具体过程，对网络安全传输文件有了更加清晰的认识。

在代码方面，对于大一学习的c语言，现在有些许生疏，例如，在读取文件内容转换为ascii码输出那个函数纠结了好久；在运行时，有野指针空指针报错等问题，不用最后通过上网查询相关知识，使得问题得以解决。

在加密知识方面，通过代码的参考与修改，我对DES加密以及DH密钥交换协议的流程有了更清晰的认识。

**实验二：数据安全检索**

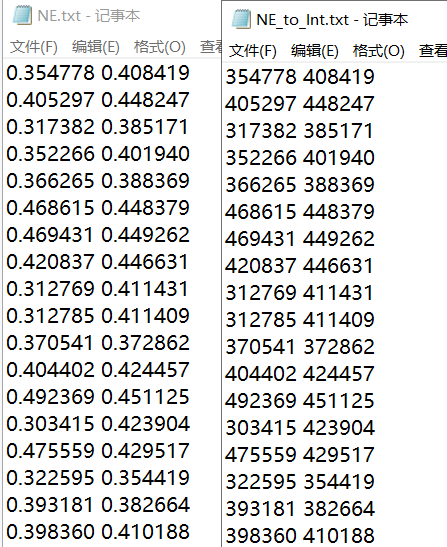
1. **实验原理：**

在NE二维乱序数据集明文上构建KD树，并采用OPE对数据进行加密（每一维分别加密），并在密文上构建KD树；分别在明文和密文上进行knn检索，以达到安全全(近似)近邻查询。

本实验采用Python3.7实现。

1. **实验结果：**

数据预处理：首先将NE数据集转换为整数集合：



以便于OPE加密。

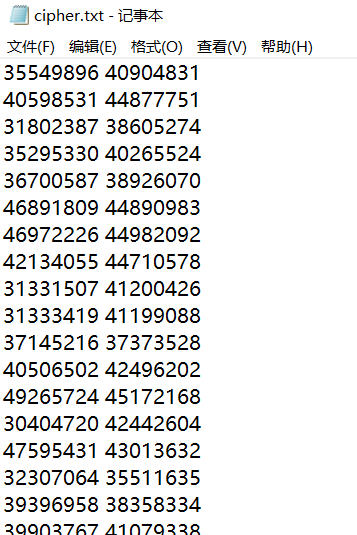
在程序中构建明文的KDTree



对明文进行OPE加密：



将明文（0-100000）映射到（0-100000000）的密文上并保存在文件中。



对密文建立KDTree :

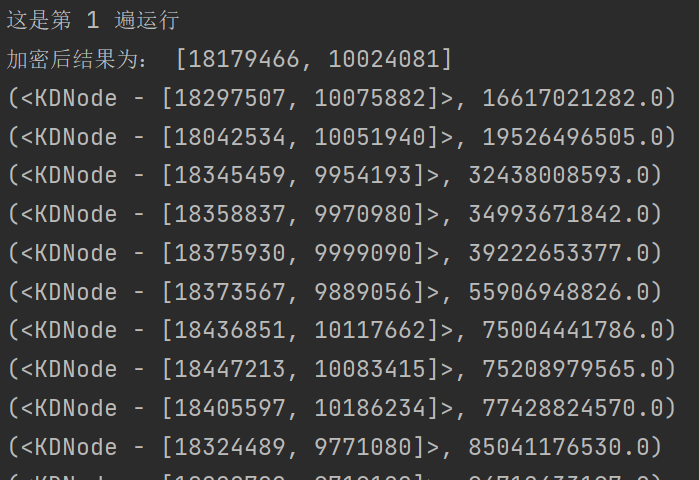


搜索（明文与密文）与（181501，100002）这个点距离最近的100个点，进行20搜索：



运行程序：

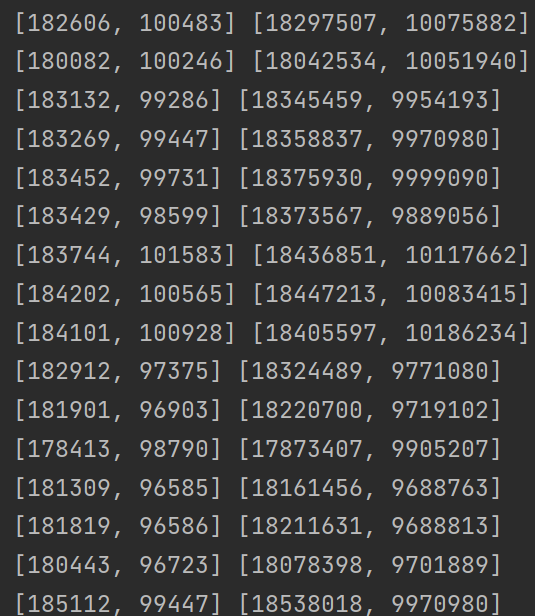
可知（181501，100002）加密结果为（18179466，10024081）



下面100个点为在密文中搜索100个最近的点的加密位置数据（按距离递增），后面数据为密文之间的欧氏距离。

对照：

前者为明文，后者为密文（按欧氏距离递增）



结果：



Ratio结果与1接近，实验取得成功。

**实验总结与心得：**

在本次实验中，我学习到了保序加密的具体流程以及代码的具体实现，KD树的索引建立以及最近邻算法的搜索与优化。

对于加密算法方面，在搜集实验资料中，我意识到了保需加密技术需要能保护用户数据机密性,也能够实现密文数据高效查询这一特点与优势，并随着一些伴随着保序加密算法的索引结构被提出，在未来，算法将更加的安全与高效。

**实验三：大数据隐私保护**

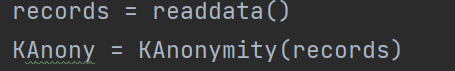
1. **实验原理：**

在对大量数据集Adult Data Set进行K匿名加密与差分隐私加密，并对加密结果的某条敏感属性（年龄）进行查询，与明文对照，分析加密的整体隐私保护可行性；在删除某条数据后，再用相同方法进行查询，分析加密算法对单个（已删除）数据隐私保护的可行性。

本实验使用Python 3.7实现。

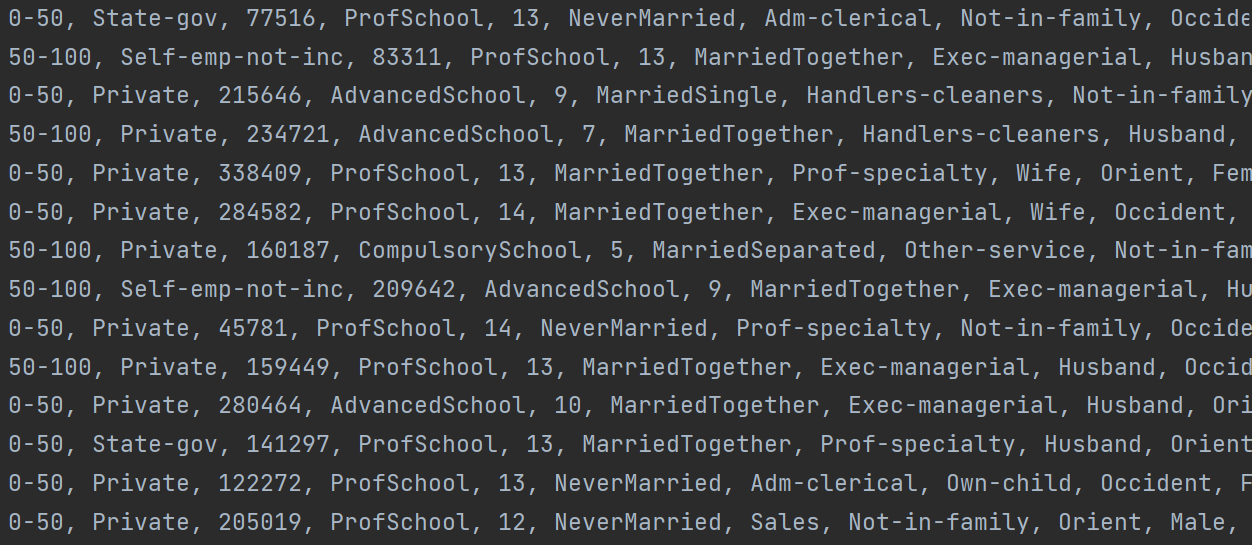
1. **实验结果：**

读取数据：

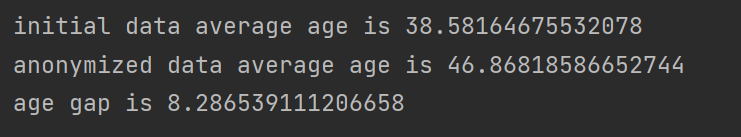




对于k=5加密数据（节选）：



计算明密文平均年龄及俩者差：

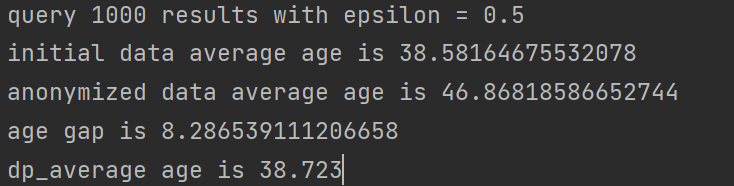


可以看出对于k=5时，两者差平均年龄差在8岁左右，可以较好的保护整体数据的隐私。

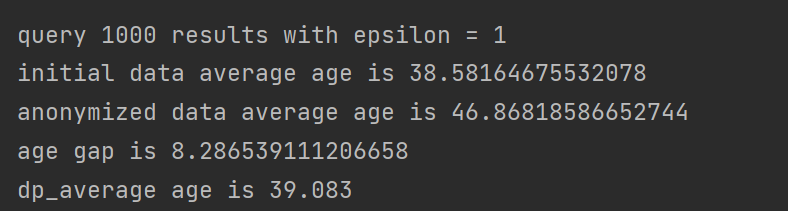
对与差分隐私查询加密：

在epsilon取0.5 与 1进行实验：

在epsilon取0.5时，查询1000次差分隐私加密结果平均年龄为38.723



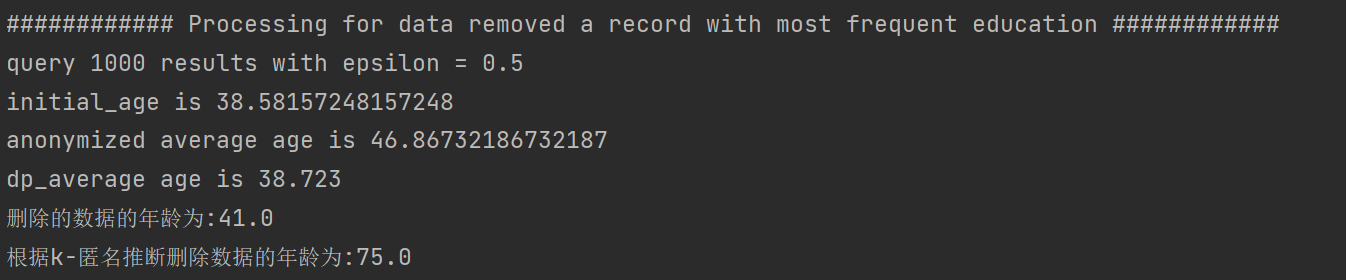
在epsilon取1时，查询1000次差分隐私加密结果平均年龄为39.083



这俩结果都与明文平均年龄想接近。

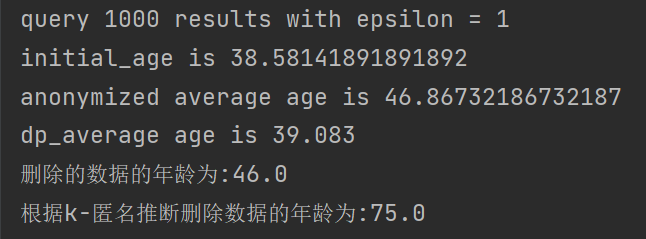
在移除一个数据后，分别在不同epsilon下运行

在epsilon取0.5时：



移除真实年龄为41岁，而攻击者基于k匿名推断的年龄为75岁，这个的原因是对于年龄的泛化，只有0-50与50-100两个泛化维度，使得攻击者只可以猜测出移除数据的敏感信息（年龄）只有75与25两个值，而且不一定会正确；差分隐私平均年龄查询结果依然是38.723，与epsilon=0.5时，原始数据查封隐私查询结果相同，可以看出差分隐私加密可以有效保护单条个人敏感信息。

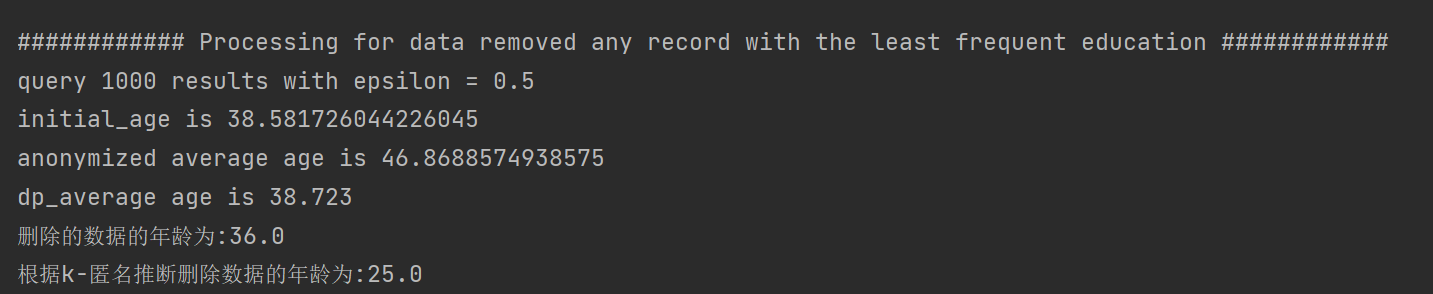
同理 在epsilon=1时，进行查询



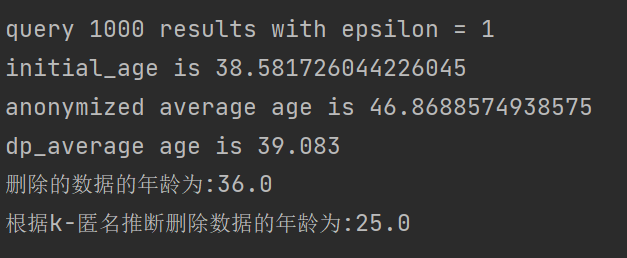
差分隐私查询结果与epsilon=1的原始数据查询结果相同。

移除不同种类的单个数据作为重复实验：

epsilon=0.5



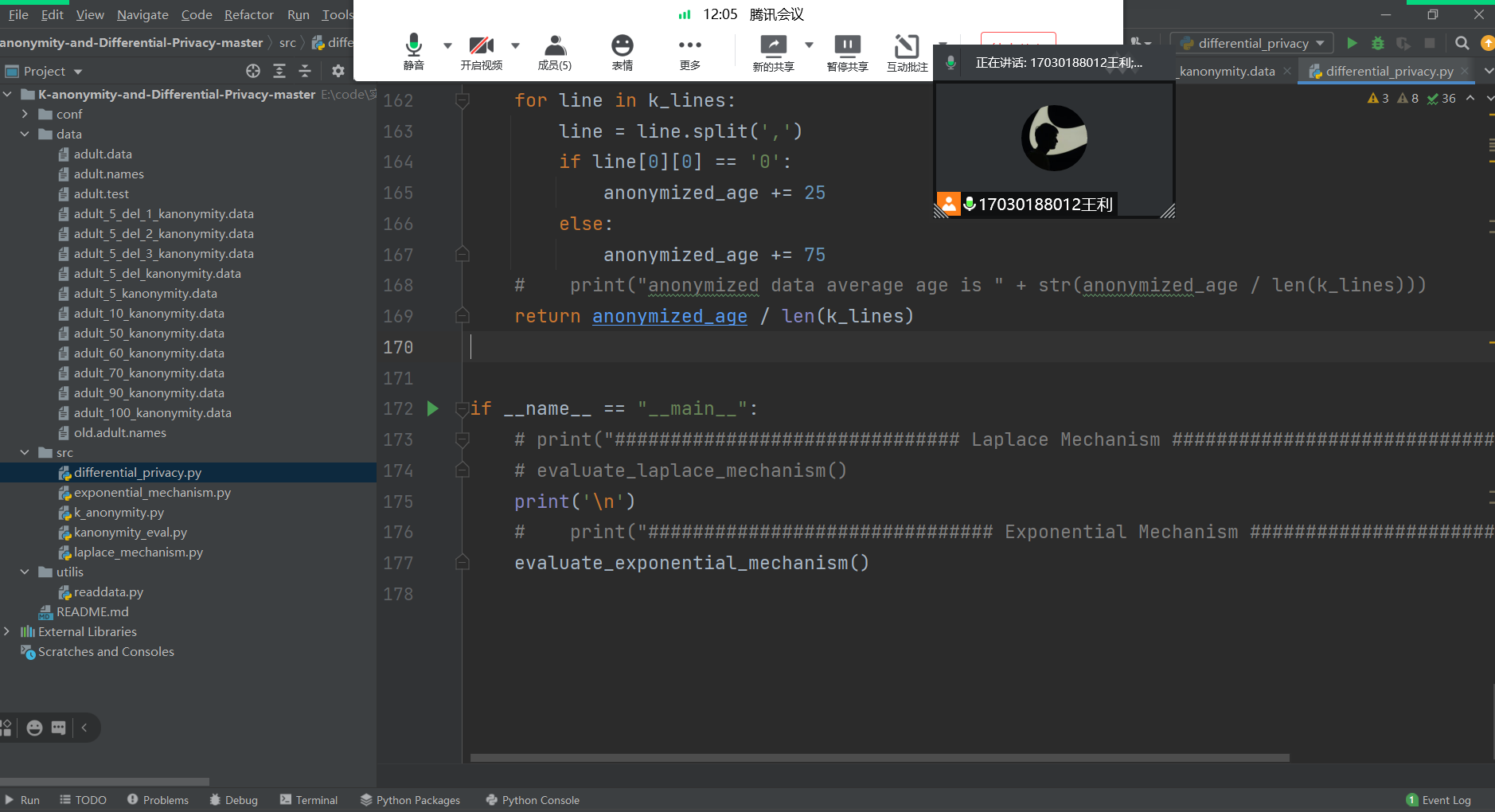
epsilon=1



依然可以得到相同结论。

实验成功！

验收截图：



**实验总结与心得：**

在上网搜索相关资料的过程中，我意识到整体以及单条数据加密的必要性与安全性，并对k\_anonymity加密算法以及difference\_privacy加密算法有了更深刻地了解。既要保证数据查询地正确性和高效性，又必须使得数据不会被攻击者所非法获取与推断出，这就是匿名加密算法与差分隐私算法存在的必要性。

从整体上来看，可搜索的加密方案允许用户将数据以一种能够保护隐私的形式外包给第三方，同时保证用户通过特定的途径能够在外包的数据上进行检索。