Concept of privacy

1. Privacy to whom?

-- Identify the stakeholder of system (third-party, user, servicer provider, public…)

-- Which are the stakeholders in system with privacy needs (users, relatives of users, employees, developers, ..)

2. Privacy of what?

-- Identify the data that is privacy-relevant (metadata of users, apps of users, account of users....)

3. Privacy towards whom?

-- Identify the potential adversaries (parents, friends, employers, service providers, developers,...)

Trust assumption

1. identify the trust assumption in system
2. Trusted entity/ component: behave as expected
3. Fewer trust assumption is better

Privacy by design

1. Proactive, preventative
2. Privacy as default
3. Privacy embedded into design
4. Full functionality, positive sum not zero sum
5. End to end security, lifecycle protection
6. Visibility and transparency
7. Respect for user privacy

Privacy engineering strategies

1. Minimize data collection

—implementing the functionality that data really need
— avoid the capture and storage of data in system
— conflict with flexibility developing future products that benefit from collected data

2. Minimize data disclosure

— identify the input and output data
— avoid disclosure of inputs to other entities than those generating the data
— avoid disclosure of outputs to other entities than those intended to obtain the outputs

Privacy engineering stragtegies

3. Minimize data replication

— decide how many copies needed
— avoid replication of data in multiple components of system (Need to balanced with avaliability and recovery from crash)

4. Minimize dara retention in time

— decide how long to keep the data
— data should be deleted if it no longer needed

5. Minimize data centralization

— central database with privacy sensitive info
— favor architectures that personal data under user control
— use multiple entities to collude to violate privacy properties (Multi-party computation)

6. Minimize data identifiability and linkability

— anonymize/pseudonimize the data
— delink the data to prevent unwanted inferences
— aggregate data to minimize inferences about individuals

Measuring anonymity sets

— anonymity depends on the number of subjects in the anonymity set and the probability of each subject in the anonymity set being the one linked to the data

Entropy

— measure of the uncertainty or average unpredictability in a random variable
— Increases with number N of possible values and with the uniformly of the distribution, log2(N)

Simple anonymization

— try to remove names, id numbers, and other unique identifiers
— all queries from a user could be linked together, and the content of the queries revealed information about users.

Ad hoc attack

— hoc是完全有无线连接的移动节点构成的网络， 有动态形成一基有限带宽等特性
— ad hoc网路， 从一个节点到另一个节点的路由可能要经过很多其他的节点的转换（multihop）， 网络中没有固定路由器， 每个节点完成自身的功能外也要充当一个路由器转发其他节点的分组
— ad hoc面临的网络威胁1: 针对网络本身的攻击 ， 破坏网络正常功能， 信道阻塞channel jamming， 非法访问 unauthorized access， 流量分析traffic analysis
— ad hoc 面临的网络威胁2: 神对通信链路的攻击， 破环端到端通信的保密性和完整性， 窃听eavesdropping， 信息伪造 message forgery， 消息重放 message reply， 中间人攻击等

Ad hoc attack

— ad hoc 面临的网络威胁3: 针对移动终端的攻击， 破坏或违法使用移动终端，power攻击， timing攻击等

—防御技术： mutual authentication 双向认证， access control访问控制， data confidentiality数据加密， data integrity数据完整性保护等

— 由于ad hoc网络没有基础设施， 核心网络不见是有对等的节点来充当， 攻击者能够轻易冒充路由器， 通过网络的控制层面或数据层面来破坏网络运行， 所以单条无线链路存在安全弱点， 多条转发路径也有安全隐患。

Anonymizing data is hard

— sparse matrices of features, unique patterns or configurations, diversity of user attributes

— hard to predict which auxiliary info is avaliable

— identifiers may appear at multiple layers

— legal and economic implications

Recommender systems

— base on collaborative system: use patterns learned from user behaviour to make recommendations. Output reveal only relationships between items rather than info about users.

K-anonymity

— 数据发布中隐私保护对象主要是用户敏感数据和个体身份之间对应关系， 通常使用删除部分用户identity无法真正阻止隐私泄露， 攻击者可以通过连接攻击获取个体隐私数据

— 链式攻击 link attack： 攻击者通过发布的数据和其他渠道获取外部数据进行链接操作， 从而推理出隐私数据， 造成信息泄露（简单例子数据库里两张表通过主键关联获取更多信息。

— 个人标识泄露：数据使用人员通过任何方式确认数据中某条数据属于某人， 可以获得具体到个人的敏感信息
— 属性泄露： 数据使用人员根据其他访问的数据了解到某个人新的属性信息
— 成员关系泄露：数据使用人员可以确认某个人的数据存在于数据表中

— 解决方法 k-anonymity（k-匿名）通过概括抽象描述数据和隐藏部分数据项发布精准度较低的数据， 使得每条记录至少与数据表中其他k-1条jilu具有完全相同的quasi-indentifier attribute， 减少链式攻击导致的隐私泄露。

— 首先对用户数据进行分类，explicit identifier（attribute key， 根据这一条可以确定用户记录）， quasi- identifier（结合一定外部信息确定用户记录， 多列信息可以潜在识别某人）， sensitive attributes （需要保护的信息）， non- sensitive attribute（可以直接发布的信息）

— k-匿名算法： 第一部对所有可标识列进行移除/脱敏/数据泛化（邮编47677-> 邮编476**)，表中每一条数据至少喝表中k-1条记录的quasi-identifier一致， 表中任意一条记录至少重复出现k次。

— 攻击者无法知道某个人是否在公开数据中， 无法确定给定人有敏感属性， 无法确认某条数据对应哪个人。

姓名	性别	年龄	邮编	购买偏好
*	男	(20,30]	10008*	电子产品
*	男	(20,30]	10008*	家用电器
*	女	(20,30]	10010*	护肤品
*	女	(20,30]	10010*	厨具
*	男	(30,40]	10220*	电子产品
*	男	(30,40]	10220*	电子产品
*	女	(30,40]	10221*	图书
*	女	(30,40]	10221*	家用电器

此数据用2-anonymity保护
攻击者想确认小明的购买偏好，通过年龄邮编和性别发现至少有两个人相同年龄性别和邮编，无法分辨出哪个是小明， 保证隐私不被泄露。

*： 删除对另数据列， 用星号代替， （，]概括方法把年龄概括成年龄段。

L-divserity

— 在公开数据中， 对于quasi-identifier相同的数据中， 明暗属性必须具有多样化， 才能保证用户隐私不能通过背景知识等方法推测出来。
— 保证相同类型数据至少有L中内容不同的敏感属性
— probabilistic L-diversity： 一个类型中出现频率最高的值概率不大于1/L
— entropy L-diversity： 一个类型中敏感数据分布的墒至少为log(L)
— recursive L-diversity： 保证最经常出现的值，出现频率不能过高

— 弊端： 如果数据仅含有两种结果， 保证2-diversity没有用
如果数据的敏感属性词出现概率相差很大，也容易泄露隐私
Skewness attack：保证敏感属性词出现的频率相同， 但是没有考虑总体的分布， 也会泄露隐私。

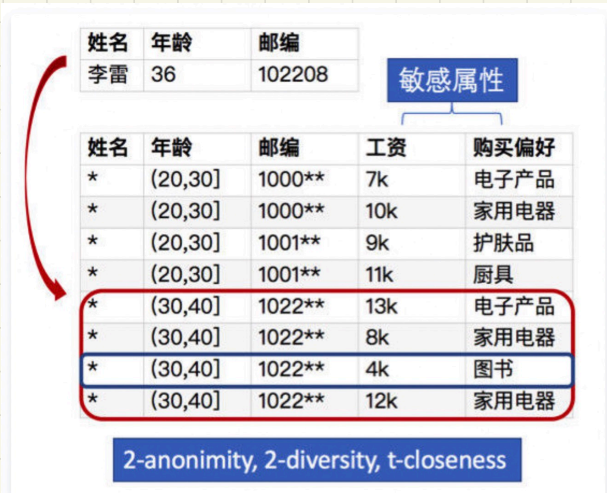
姓名	年龄	邮编
李雷	36	102208

姓名	年龄	邮编	工资	购买偏好
*	(20,30]	10008*	10k	电子产品
*	(20,30]	10008*	10k	家用电器
*	(20,30]	10010*	9k	护肤品
*	(20,30]	10010*	9k	厨具
*	(30,40]	10220*	3k	电子产品
*	(30,40]	10220*	4k	家用电器
*	(30,40]	10221*	12k	图书
*	(30,40]	10221*	12k	家用电器

l-diversity 没有考虑敏感属性的语义

t-closeness

— 保证相同的quasi-identifier类型中， 敏感信息分布与整体数据敏感信息分布接近， 不超过阈值t



很容易用邮编和年龄来判断出李雷的工资和购买偏好
属于针对敏感属性的背景攻击

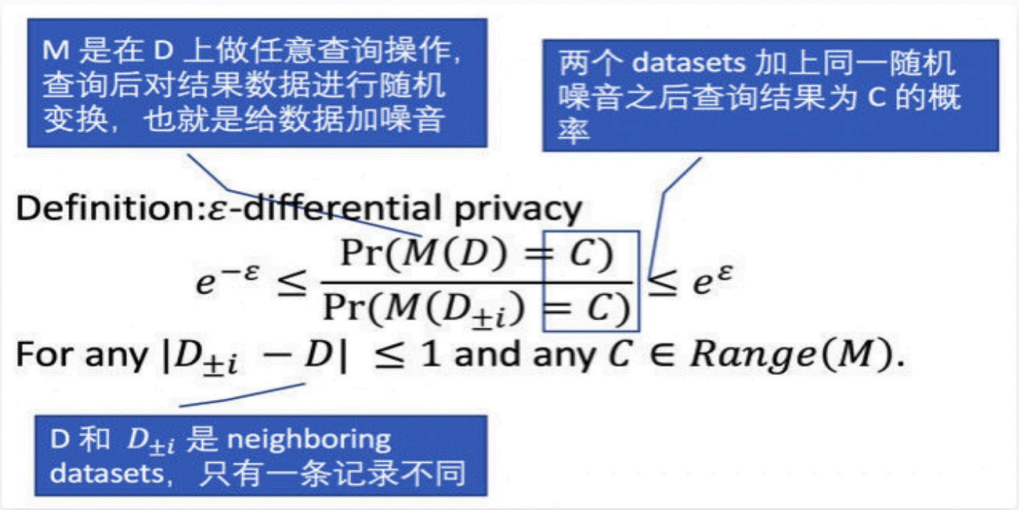
Differential privacy

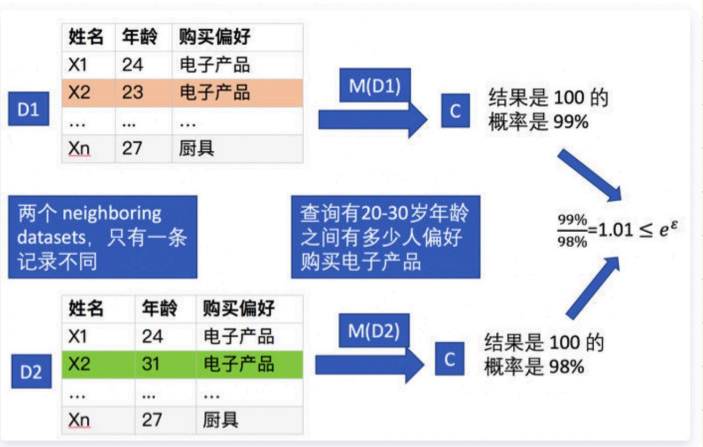
— 差分攻击 （differential attack）， 如果有100个人的购物偏好数据，10个人偏好家电购买， 90个人偏好图书购买， 如果攻击者知道99个人的购物偏好， 那么可以推断第100个人的购物偏好

— 差分隐私： 查询100个信息和查询其中99个信息的结果相对一致(输出的概率相同)， 攻击者无法通过比较数据的不同找出第100个人的信息， 加入随机性。

— M在 D 和 D+i 上做任意查询操作， 对查询结果加入一定随机性（add noise to dataset）加入随机噪音之后对两组数据查询结果为C的概率比小于一个特定值。

ε–差分隐私（ε–differential privacy， ε–DP）可以用下面的定义来表示：





D1 与D2 只有一条记录不一致， 攻击者查询特定的条件， 得到的概率基本相同， 比值小于特定数字， 对于任意查询都可以满足这样的条件

那么如何选择噪声呢?

差分隐私方法中，作者巧妙的利用了拉普拉斯分布的特性，找到了合适的噪声方法。针对数值或向量的查询输出，M(x) = f(x) + 噪声。我们能得出以下结论：

$$M(x) = f(x) + Lap\left(\frac{GS_f}{\varepsilon}\right) \text{ 是 } \varepsilon\text{-DP}$$

其中 Lap 是拉普拉斯分布，GS 表示 global sensitivity：

$$GS_f = max\|f(x) - f(x')\|_1$$

(ε,δ)–differential privacy, (ε, δ)–DP ε–DP 是一种“严格”的隐私保护保证，当在数据库中添加和删除一条数据时候，保证所有查询的输出都类似。但是(ε, δ)–DP 在 ε–DP 的保证中允许了一定概率的错误发生，比如说，用户在 (ε, δ)–DP 的保护下会有 δ 概率的隐私泄露。