数据脱敏与个人隐私保护

——白玥 黄弈萧 姜佳祺 蒋泽彬 刘禹含

1 经典综述

2 数据加密算法

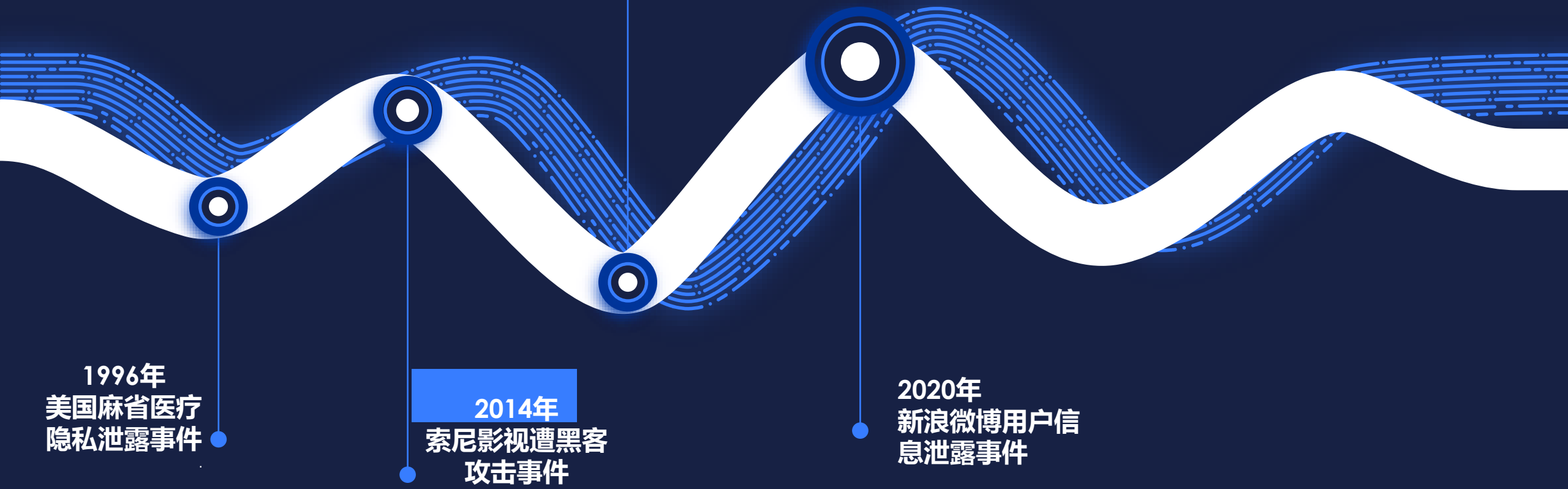
3 隐私计算技术

4 技术 demo

数据脱敏与个人隐私保护

蒋泽彬

个人隐私泄露实例



2018年
Facebook泄露
门事件

1996年
美国麻省医疗
隐私泄露事件

2014年
索尼影视遭黑客
攻击事件

2020年
新浪微博用户信
息泄露事件

相关法规

1

欧盟《通用数据保护条例》

欧盟于2018年5月25日正式实施了《通用数据保护条例》，简称《GDPR》），是一项保护欧盟公民个人隐私和数据的法律，其适用范围包括欧盟成员国境内企业的个人数据、也包括欧盟境外企业处理欧盟公民的个人数据。

2

美国加州消费者隐私法

美国已有多个州先在数据安全与隐私保护进行了立法，其中最著名的要数2018年6月加州通过《加州消费者隐私法案》（简称《CCPA》）。该法案被称为美国“最严厉和最全面的个人隐私保护法案”，于2020年1月1日生效。

3

中华人民共和国网络安全法

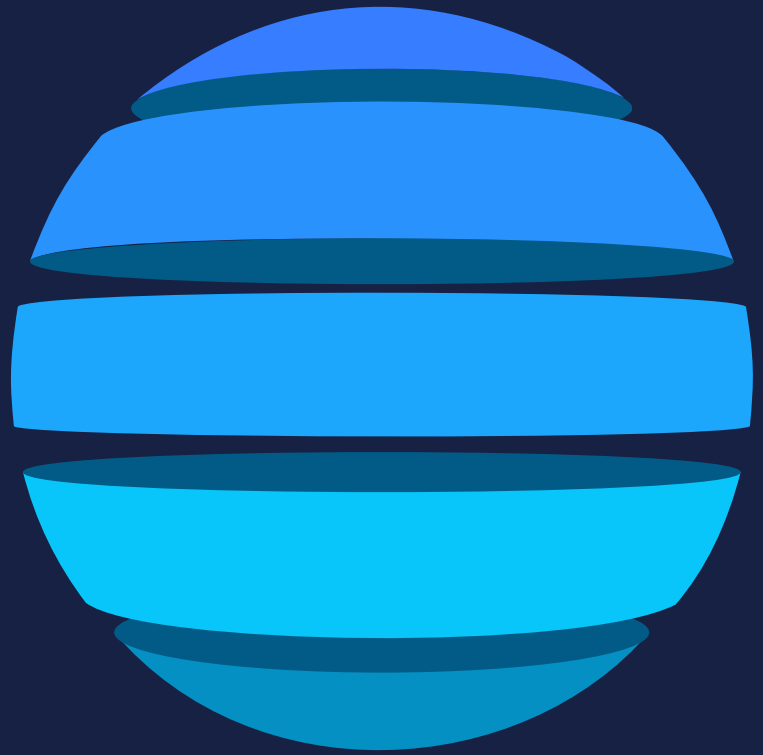
我国于2017年6月1日正式实施《中华人民共和国网络安全法》。《网安法》是我国首部全面规范网络空间安全管理方面问题的基础性法律，一共包括7章79条，包含网络运行安全、关键信息基础设施的运行安全、网络信息安全等内容。

数据安全技术

1 数据加密

2 匿名化

3 隐私计算

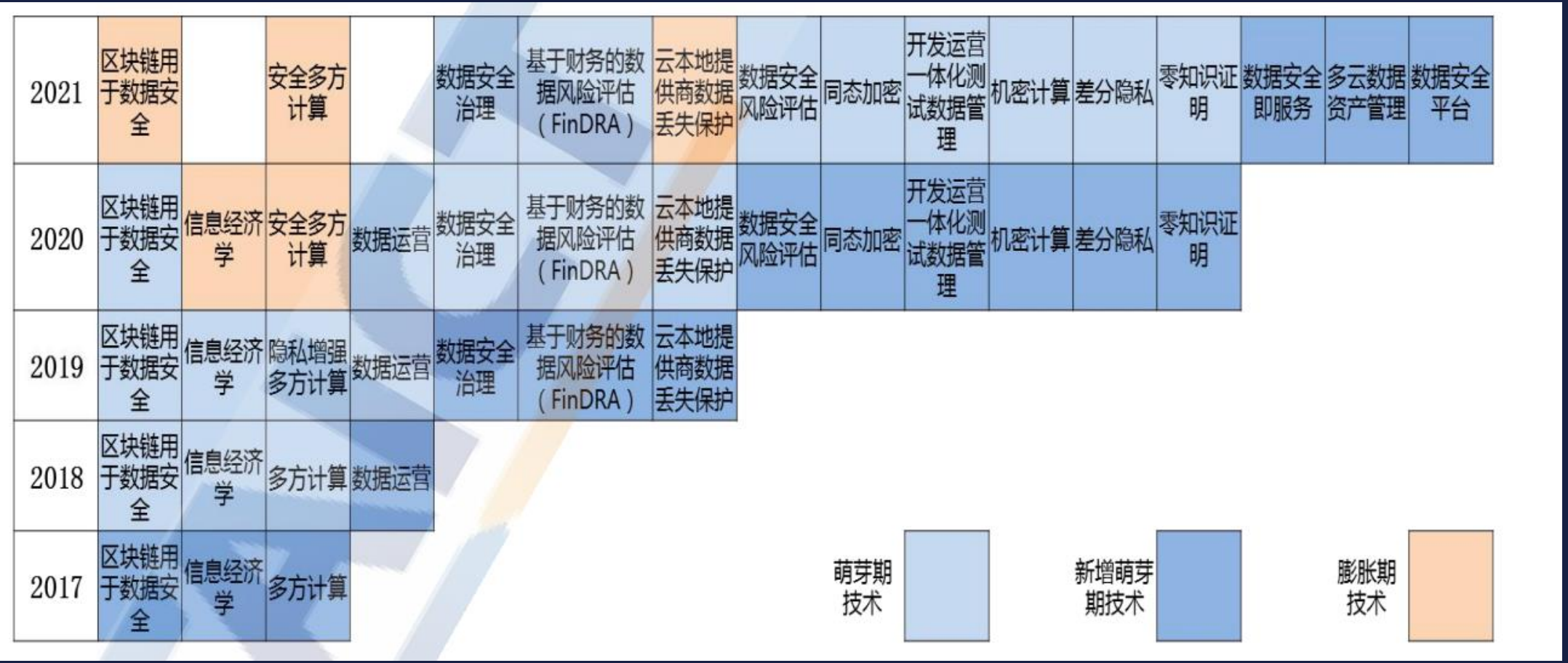


4 数据识别

5 数据脱敏

6 数字水印

新兴数据安全技术呈逐年增长趋势



Massive X

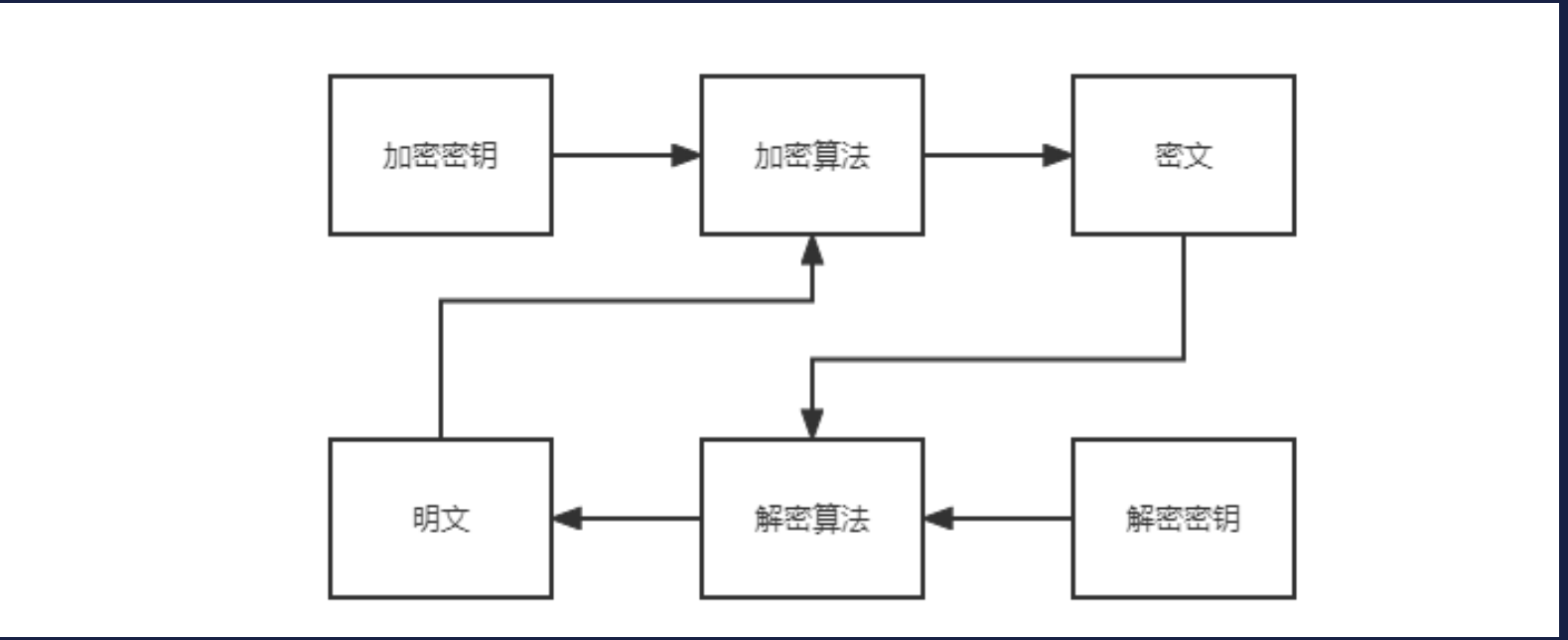
数据加密算法

刘禹含

cbc

数据加密算法

数据加密，指通过加密算法和加密密钥将明文转变为密文，而解密则是通过解密算法和解密密钥将密文恢复为明文



简单数据加密技术

- MD5
- 对称加密AES
- 非对称加密RSA

MD5

时间：1992年

论文：MIT Laboratory for Computer Science and RSA Data Security Inc.

作者：R.Rivest

MD5

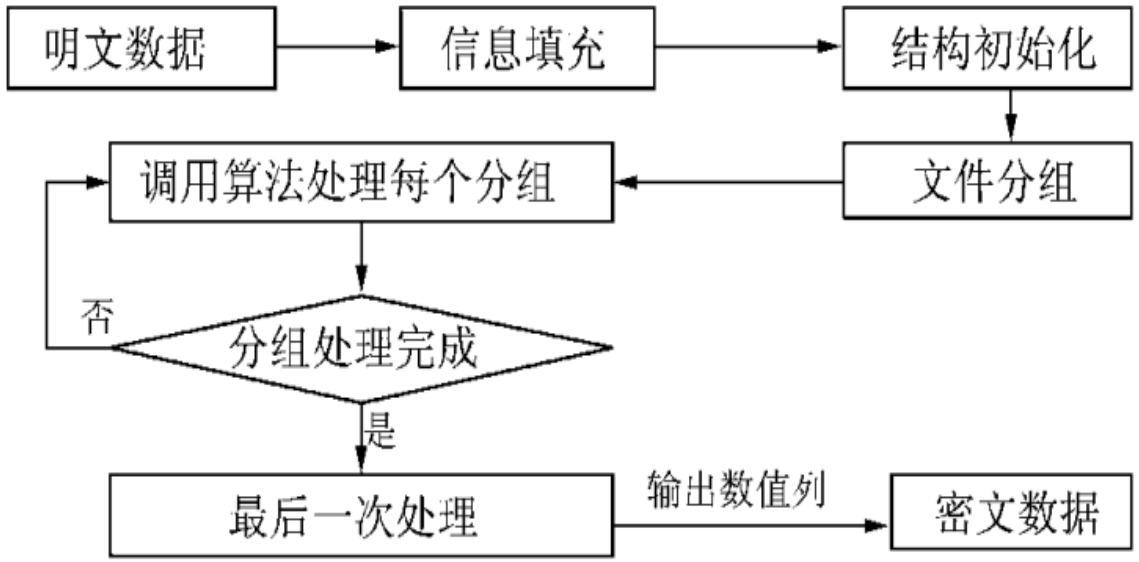


图 1　MD5 算法流程图

算法流程：

(1)信息填充

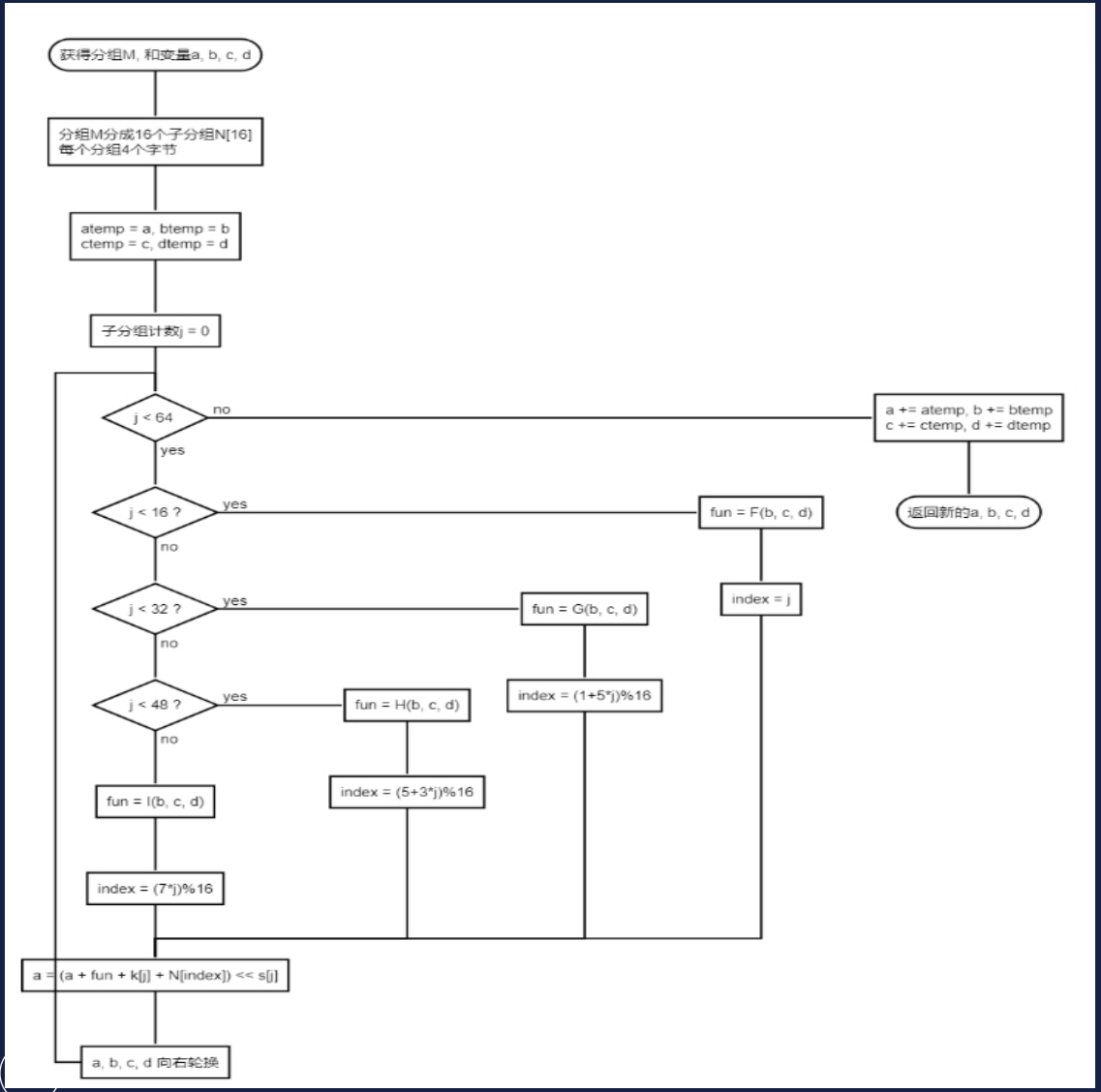
(2)结构初始化

(3)分组文件

(4)对每组数据进行处理

(5)输出结果

MD5



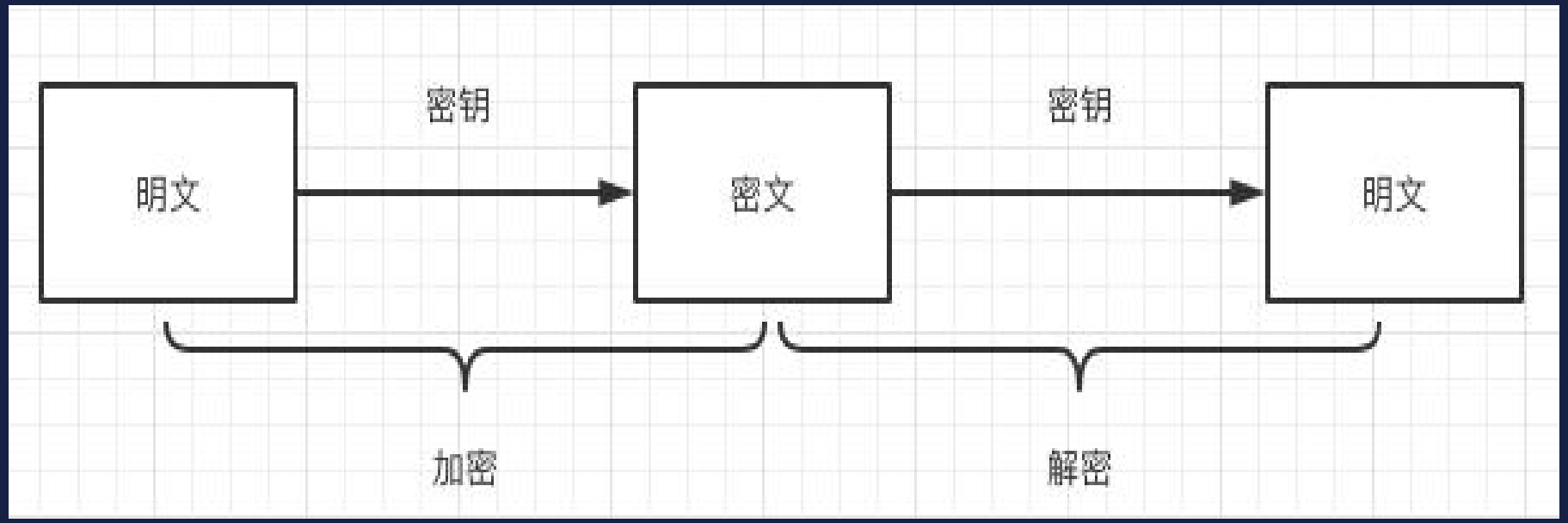
分组处理过程：

①链接变量A、B、C、D复制到外四个变量a、b、c、d中

②进入算法的四轮循环运算

③输出结果与A、B、C、D相加

对称加密算法



特点：密钥唯一

优点：算法公开、计算量小、加密速度快、加密效率高；

缺点：密钥泄露之后，数据就会被破解。

对称加密AES算法

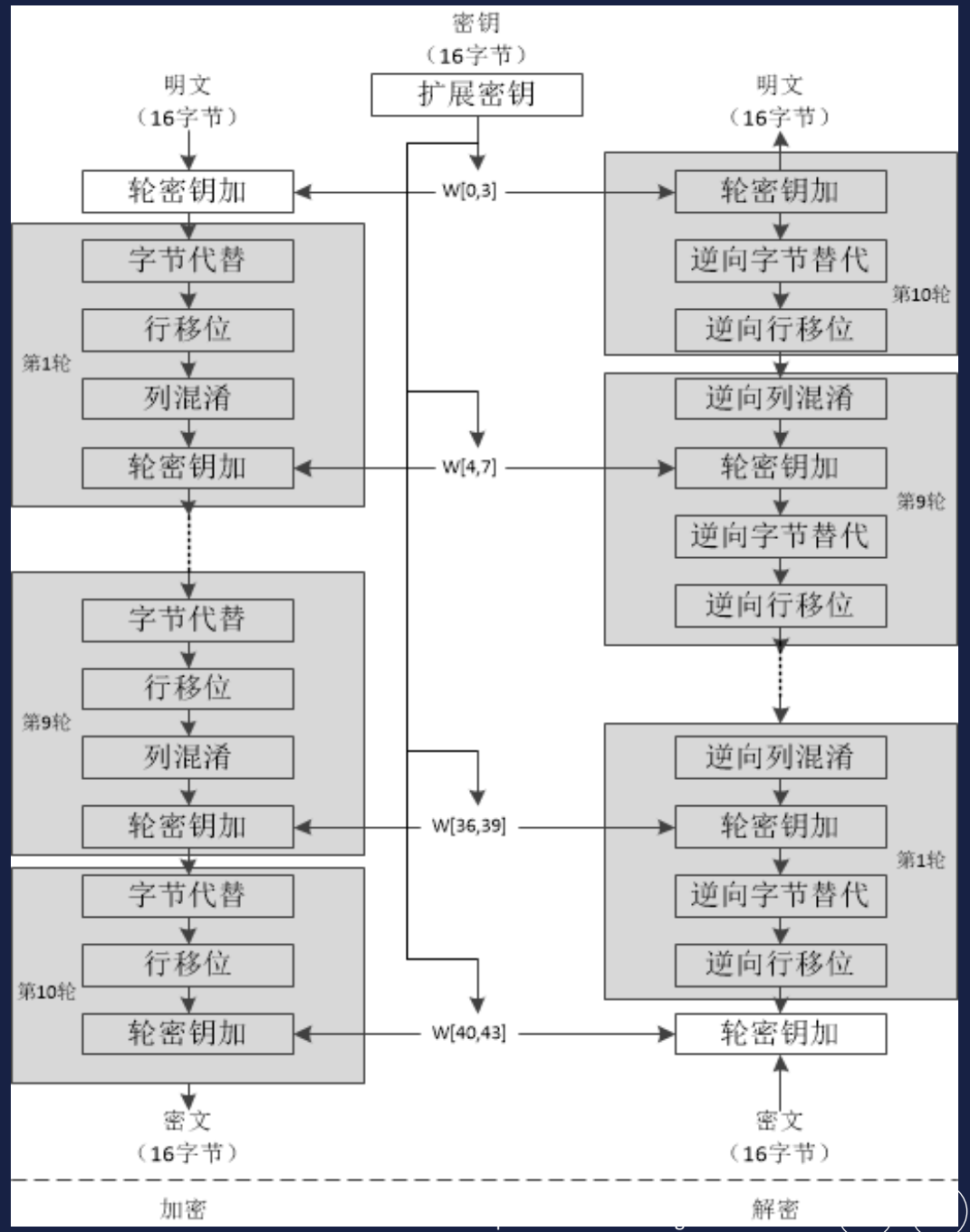
特点：

解密算法对应加密算法的逆操作

加解密顺序相反

AES加密算法涉及4种操作：

- 字节替代
- 行移位
- 列混淆
- 轮密钥加



AES算法4种操作

- 字节替代

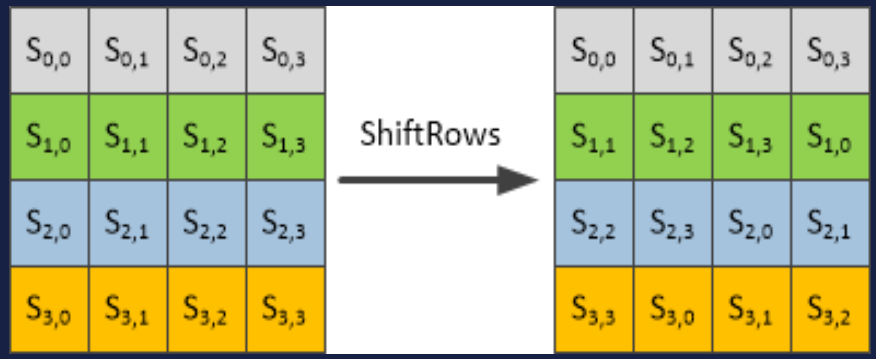
主要功能是通过S盒完成一个字节到另外一个字节的映射。

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

(a) S-box

- 字节替代

4x4的矩阵内部字节之间的置换，用于提供算法的扩扩散性。



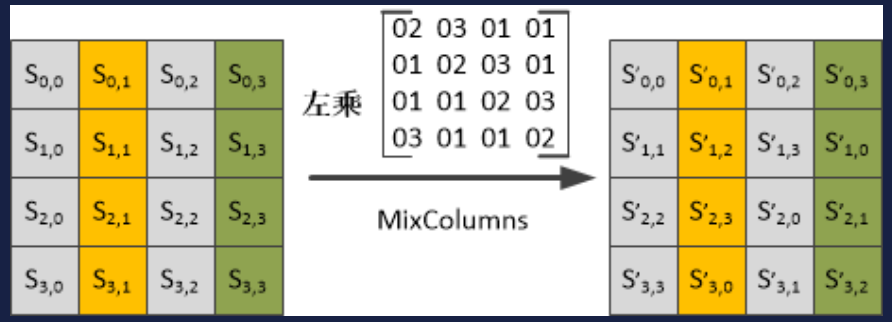
AES算法4种操作

- 列混淆

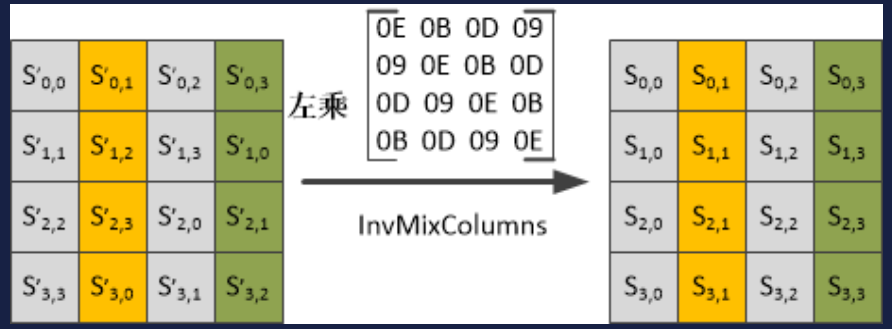
利用GF(28)域上算术特性的一个代替，

同样用于提供算法的扩散性。

正向列混淆



逆向列混淆



- 轮密钥加

加密过程中，每轮的输入与轮子密钥异或一次；因此，解密时再异或上该轮的轮子密钥即可恢复

非对称加密RSA算法

加密过程：

$$密文 = 明文^{E} mod N$$

$$公钥 = (E, N)$$

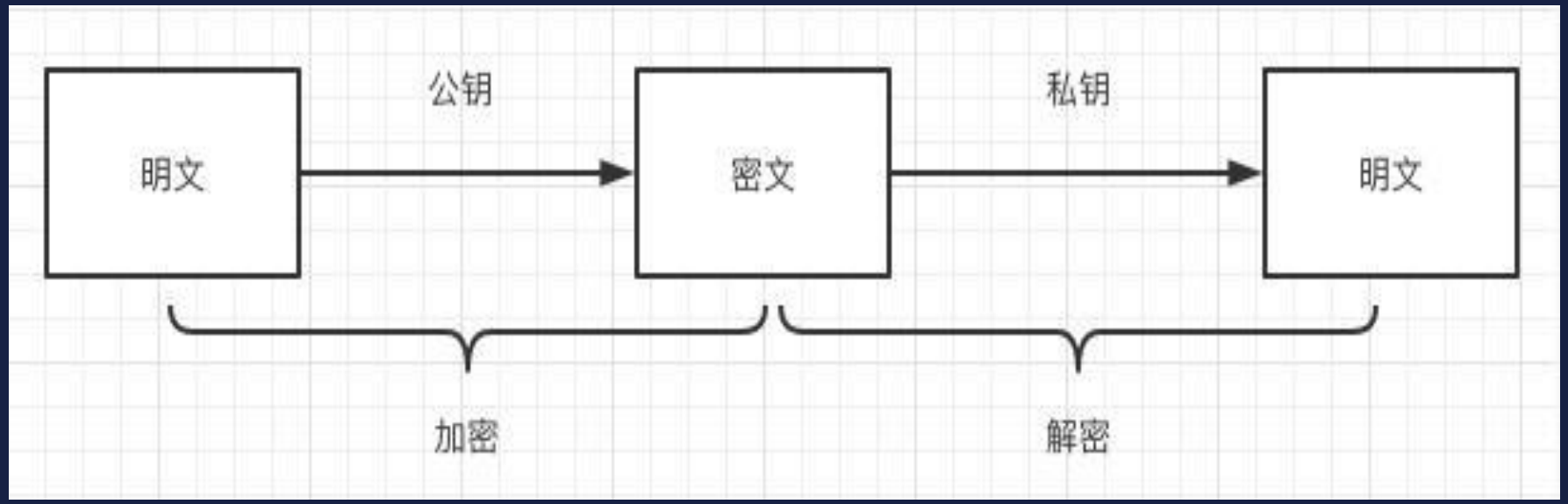
解密过程：

$$密文 = 明文^{D} mod N$$

$$私钥 = (D, N)$$

密钥对即为（E，D，N）

非对称加密算法



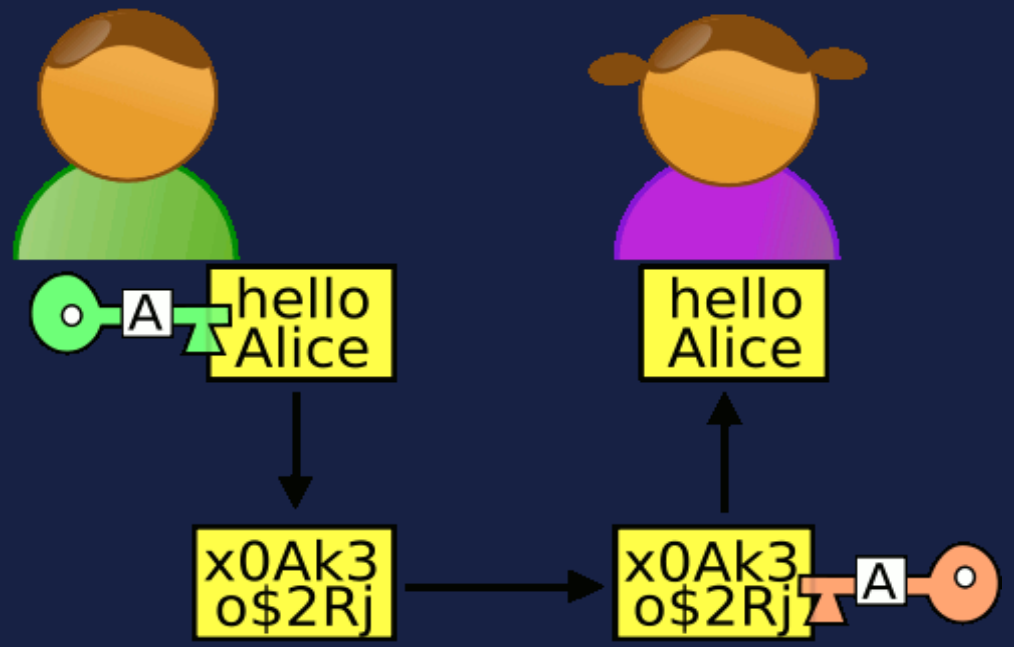
特点：两个密钥：公钥和私钥

优点：算法强度复杂、安全性高；

缺点：加解密速度没有对称加密算法快。

RSA算法生成密钥对

- 随机选择两个不相等的质数p和q：61，53
- 计算p和q的乘积n = 3233（n的长度就是密钥长度）
- 计算n的欧拉函数ϕ(n) = (p-1)(q-1) =3120
- 随机选择一个整数e，条件是1< e < ϕ(n)，且e与ϕ(n) 互质（17）
- 计算e对于ϕ(n)的模反元素d=2753



数据匿名化算法

数据匿名化算法属于基于限制发布的技术。

两种基本操作：
（1）抑制：不发布该数据项；
（2）泛化：对整数 5 的一种泛化形式是［3，6］，因为 5 在区间［3，6］内.

传统数据匿名技术

- k-anonymity
- 1-diversity
- t-Closeness

k-anonymity

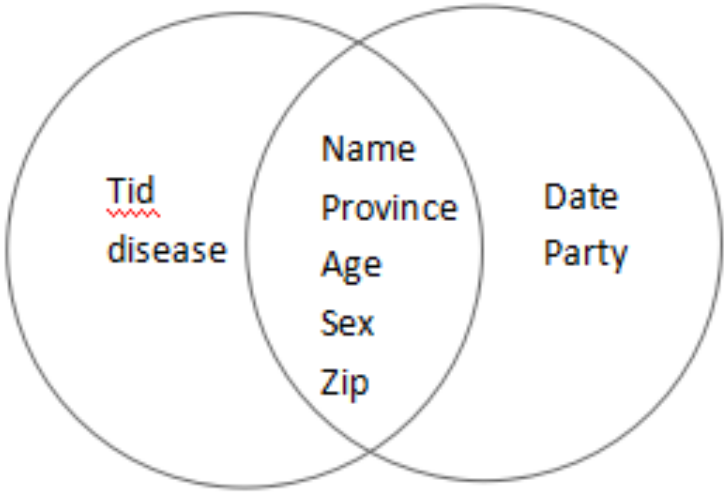
时间：1998年

论文：Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression

作者：拉坦亚·斯威尼和皮兰格拉·萨马拉蒂

4个基本概念

- 显示标识符：身份证号、学号等
- 准标识符：生日、性别、民族、邮编等
- 敏感属性：需要保护的信息
- 链式攻击



表一和表二重叠属性

tId	name	Province	age	sex	disease	Zip
1	李青	安徽	32	女	肺炎	100870
2	张雨	安徽	32	女	流感	100871
3	夏科	安徽	32	男	哮喘	100876
4	马浩	安徽	32	男	肺结核	100875
5	李云	重庆	37	女	脂肪肝	101230
6	刘然	重庆	37	女	脂肪肝	101231

表一 医疗信息表

Name	Sex	Age	Province	Zip	Date	Party
李青	女	32	安徽	100870	2016.07	共产党
刘然	女	37	重庆	101231	2016.07	共产党
马方	男	34	湖南	436570	2016.07	共产党
沈思	女	37	重庆	101231	2016.07	共产党
张佳	男	33	天津	106470	2016.07	共产党
方淮	男	38	河北	104670	2016.07	共产党

表二 选民信息表

k-anonymity

基本要求：方法要求每条记录在发布数据前，都至少与表中K-1条记录无法区分开来。

tId	name	Province	age	sex	disease	Zip
1	李青	安徽	32	女	肺炎	100870
2	张雨	安徽	32	女	流感	100871
3	夏科	安徽	32	男	哮喘	100876
4	马浩	安徽	32	男	肺结核	100875
5	李云	重庆	37	女	脂肪肝	101230
6	刘然	重庆	37	女	脂肪肝	101231

抑制和泛化

tid	Province	age	sex	disease	zip
1	安徽	（30,40]	女	肺炎	1008**
2	安徽	（30,40]	女	流感	1008**
3	安徽	（30,40]	男	哮喘	1008**
4	安徽	（30,40]	男	肺结核	1008**
5	重庆	（30,40]	女	脂肪肝	1012**
6	重庆	（30,40]	女	脂肪肝	1012**

k-anonymity缺陷

tid	Province	age	sex	disease	zip
1	安徽	（30,40]	女	肺炎	1008**
2	安徽	（30,40]	女	流感	1008**
3	安徽	（30,40]	男	哮喘	1008**
4	安徽	（30,40]	男	肺结核	1008**
5	重庆	（30,40]	女	脂肪肝	1012**
6	重庆	（30,40]	女	脂肪肝	1012**

- 同质化攻击
- 背景知识攻击
- 未排序匹配攻击
- 补充数据攻击

l-diversity

时间：2006年

论文：privacy beyond K-anonymity

作者：Machanavajjhala等

l-diversity

基本要求：在公开的数据中，每一个等价类里的敏感属性必须具有多样性，即保证每一个等价类里,敏感属性至少有L个不同的取值。

tid	Province	age	sex	disease	zip
1	安徽	（30，40]	女	肺炎	1008**
2	安徽	（30，40]	女	流感	1008**
3	安徽	（30，40]	男	哮喘	1008**
4	安徽	（30，40]	男	肺结核	1008**
5	重庆	（30，40]	女	脂肪肝	1012**
6	重庆	（30，40]	女	哮喘	1012**

l-diversity缺陷

tid	Province	age	sex	disease	zip
1	安徽	（30，40]	女	肺炎	1008**
2	安徽	（30，40]	女	流感	1008**
3	安徽	（30，40]	男	哮喘	1008**
4	安徽	（30，40]	男	肺结核	1008**
5	重庆	（30，40]	女	脂肪肝	1012**
6	重庆	（30，40]	女	哮喘	1012**

- 敏感属性比例的严重不均衡导致L-多样性难以实现
- 偏斜性攻击
- L-多样性没有考虑敏感属性的语义，而导致敏感信息的泄露。

t-Closeness

时间：2007年

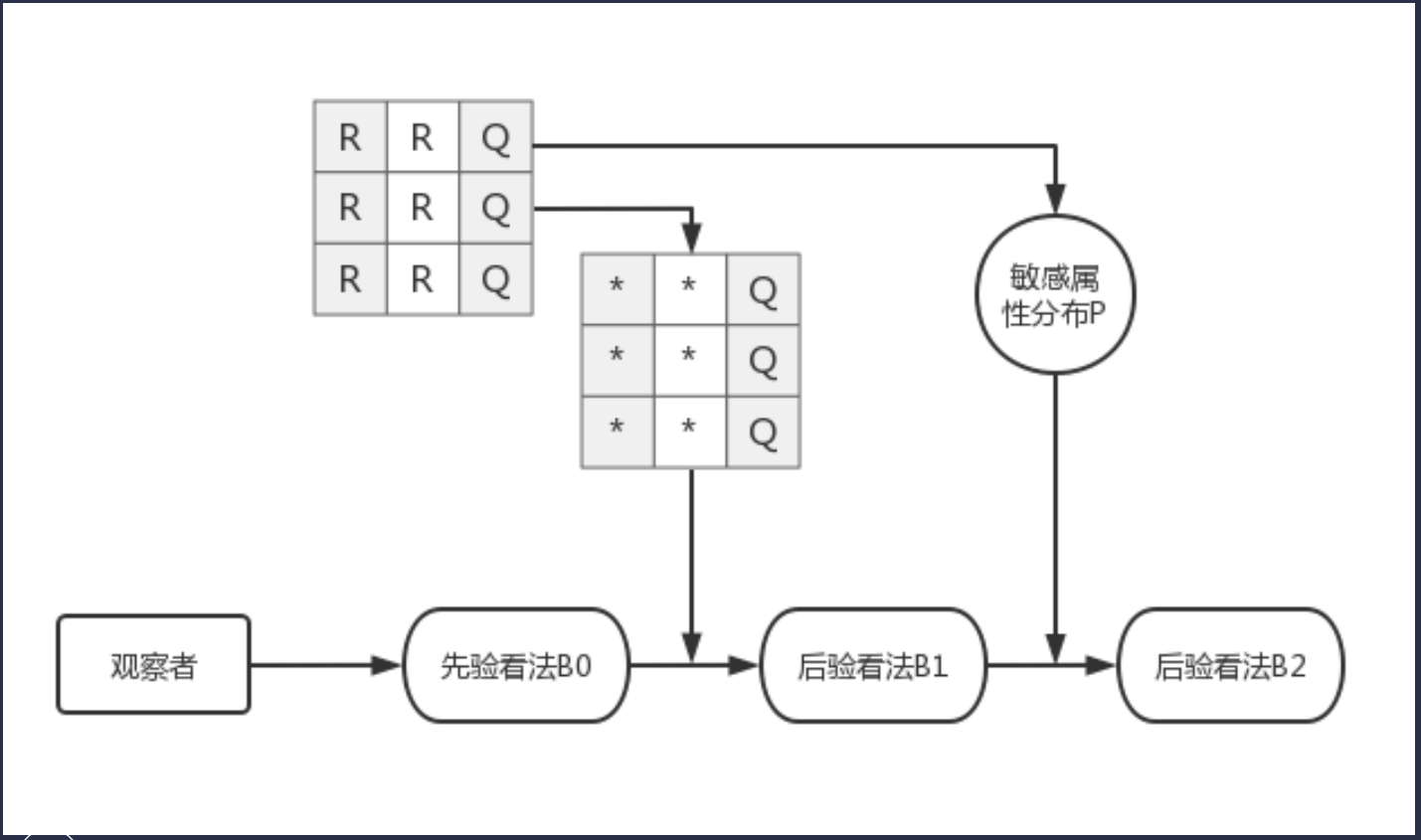
论文：t-Closeness: privacy beyond K-anonymity and L-diversity

作者：LI N H，LI T C，SURESH V S

t-Closeness解读

基本要求：所有等价类中敏感属性的分布尽量接近该属性的全局分布

观察者获得信息的过程:



l-多样化——限制B0和B2之间的差异

t-近似——限制B1和 B2之间的差异

匿名模型实现算法

Mondrian（蒙德里安）算法

时间：2006年

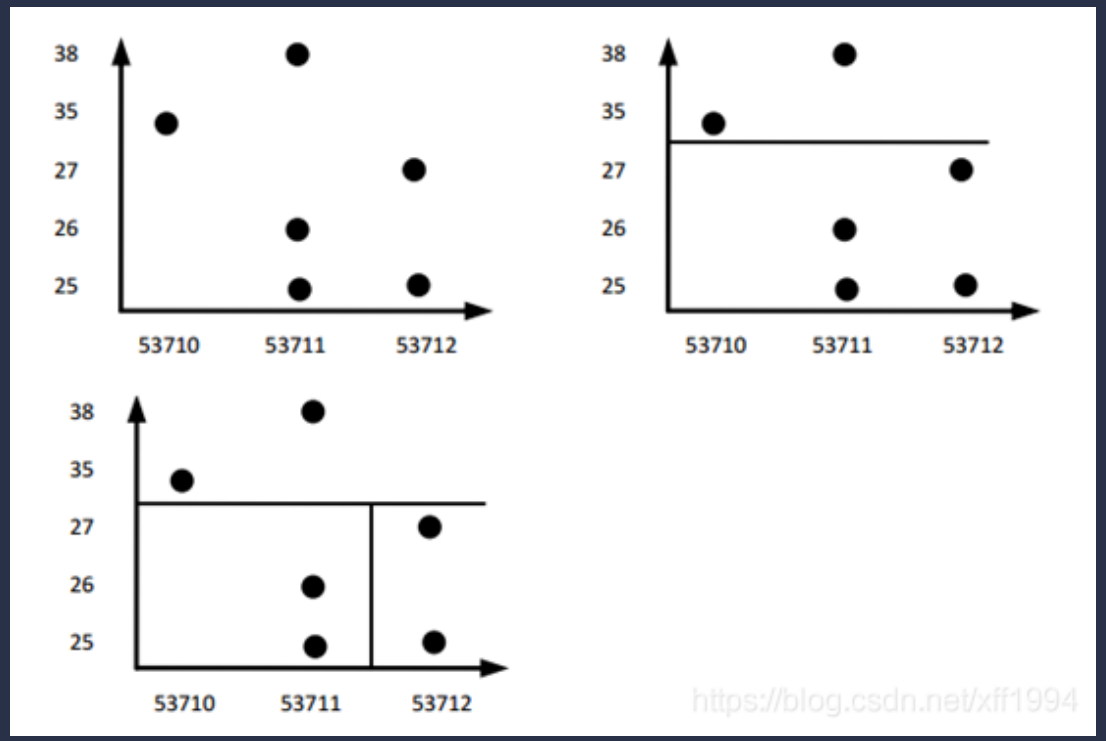
论文：Mondrian multidimensional k-anonymity

作者：Kristen LeFevre, David J DeWitt and Raghu Ramakrishnan

Mondrian（蒙德里安）算法

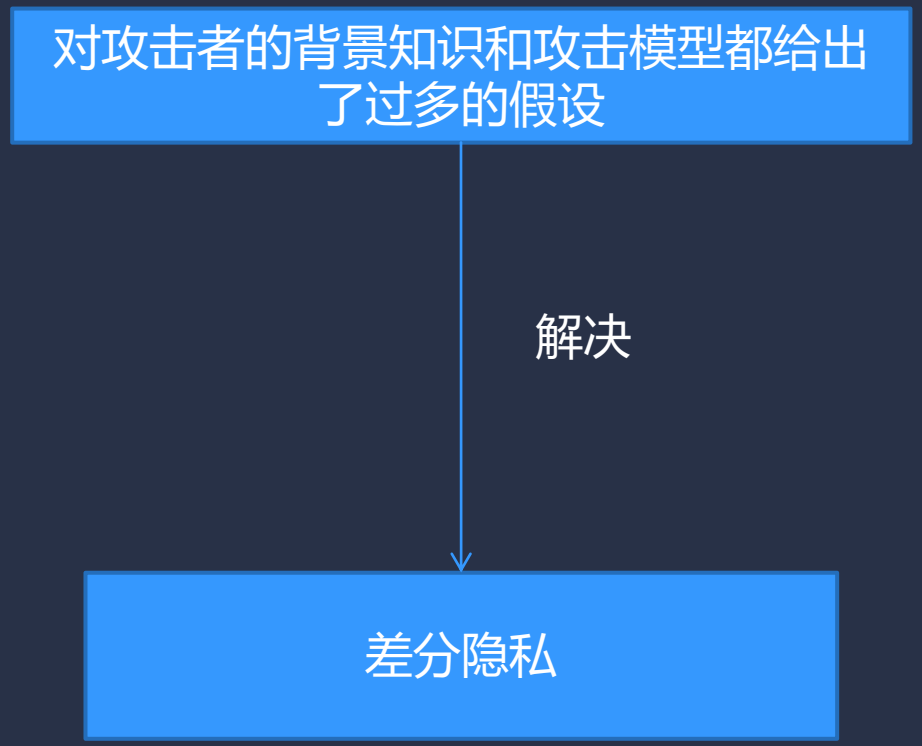
Mondrian算法针对数值型数据。其核心思想是将所有QID属性看成是一样的，即只有一个等价组，然后自上而下，启发式选择QID的某个属性逐次划分，直到满足条件无法划分。

以单个属性为例：



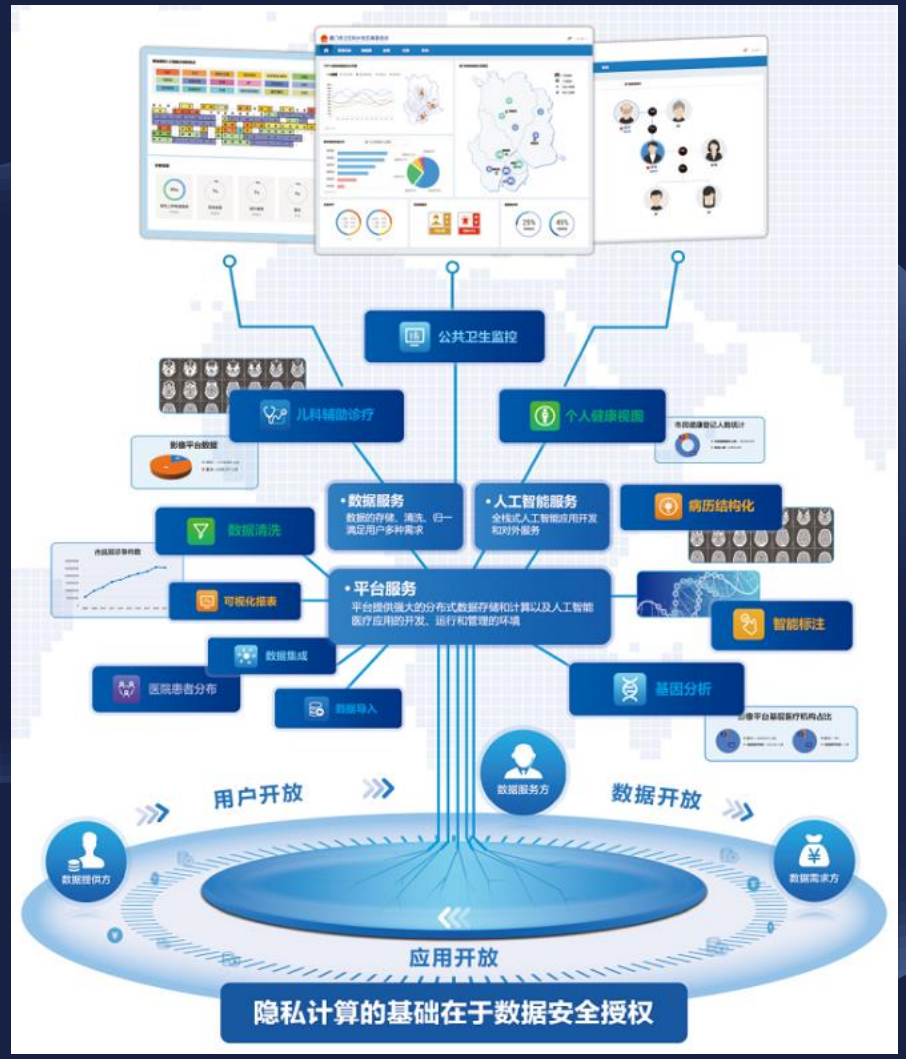
https://blog.csdn.net/xff1994

匿名化算法缺陷



隐私计算

姜佳祺



定义

隐私计算(Privacy compute 或 Privacy computing)是指在保护数据本身不对外泄露的前提下实现数据分析计算的技术集合，达到对数据“可用、不可见”的目的；在充分保护数据和隐私安全的前提下，实现数据价值的转化和释放。

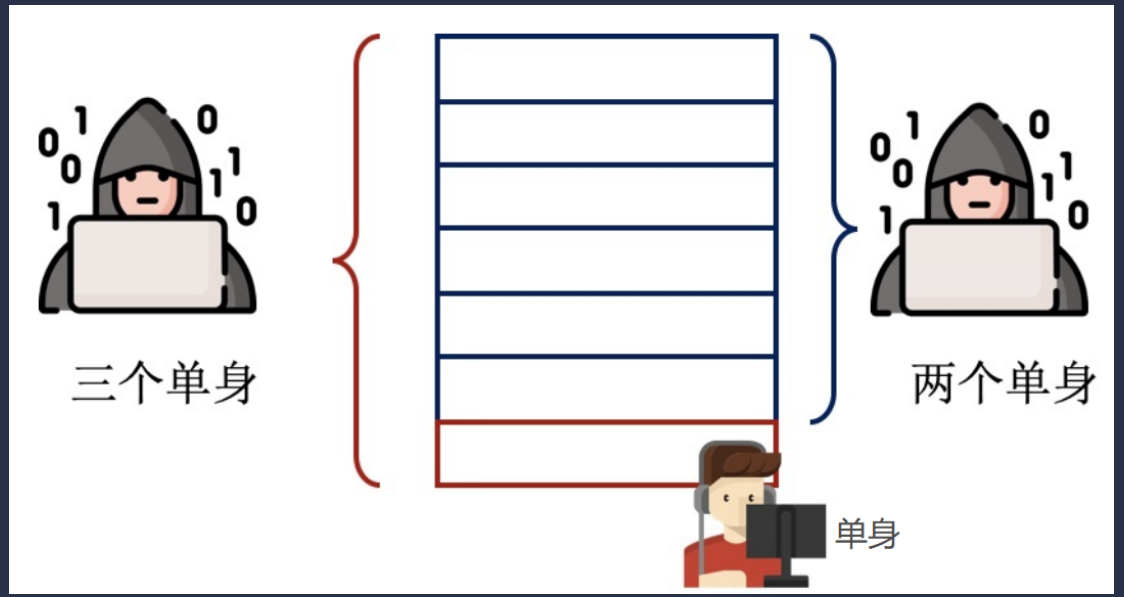
差分隐私

联邦学习

隐私技术

差分隐私的提出

假设现在有一个婚恋数据库，2个单身4个已婚，只能查有多少人单身。刚开始的时候查询发现，2个人单身；现在张三跑去登记了自己婚姻状况，再一查，发现3个人单身。所以张三单身。

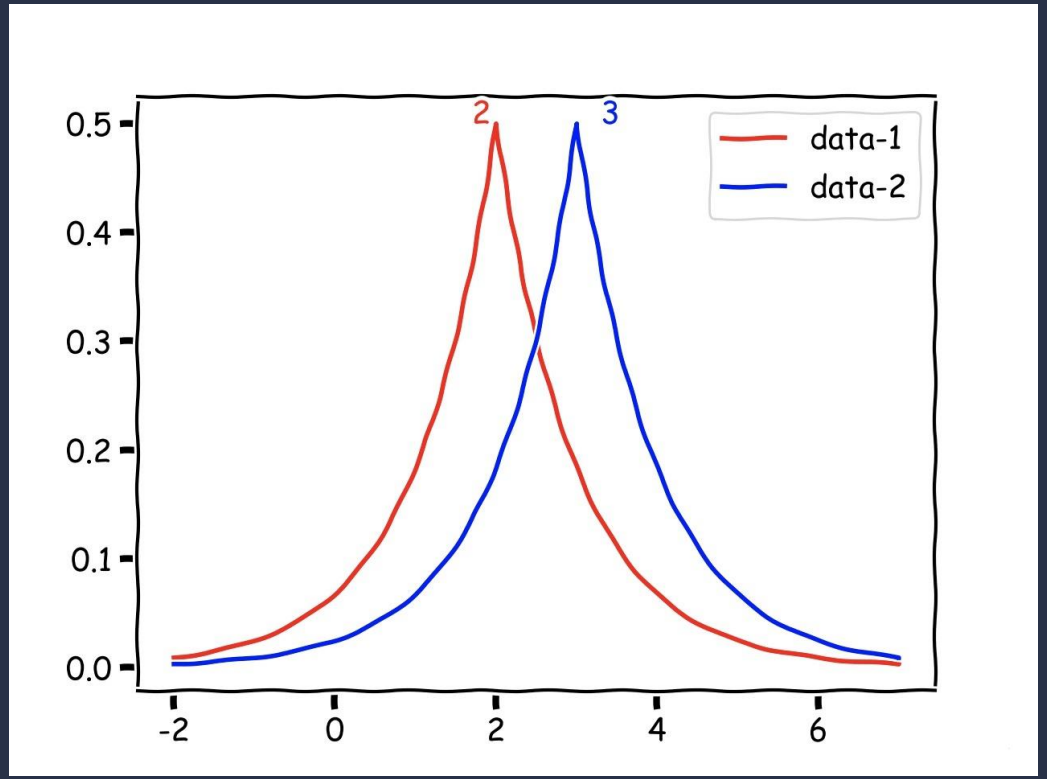


加入随机噪声

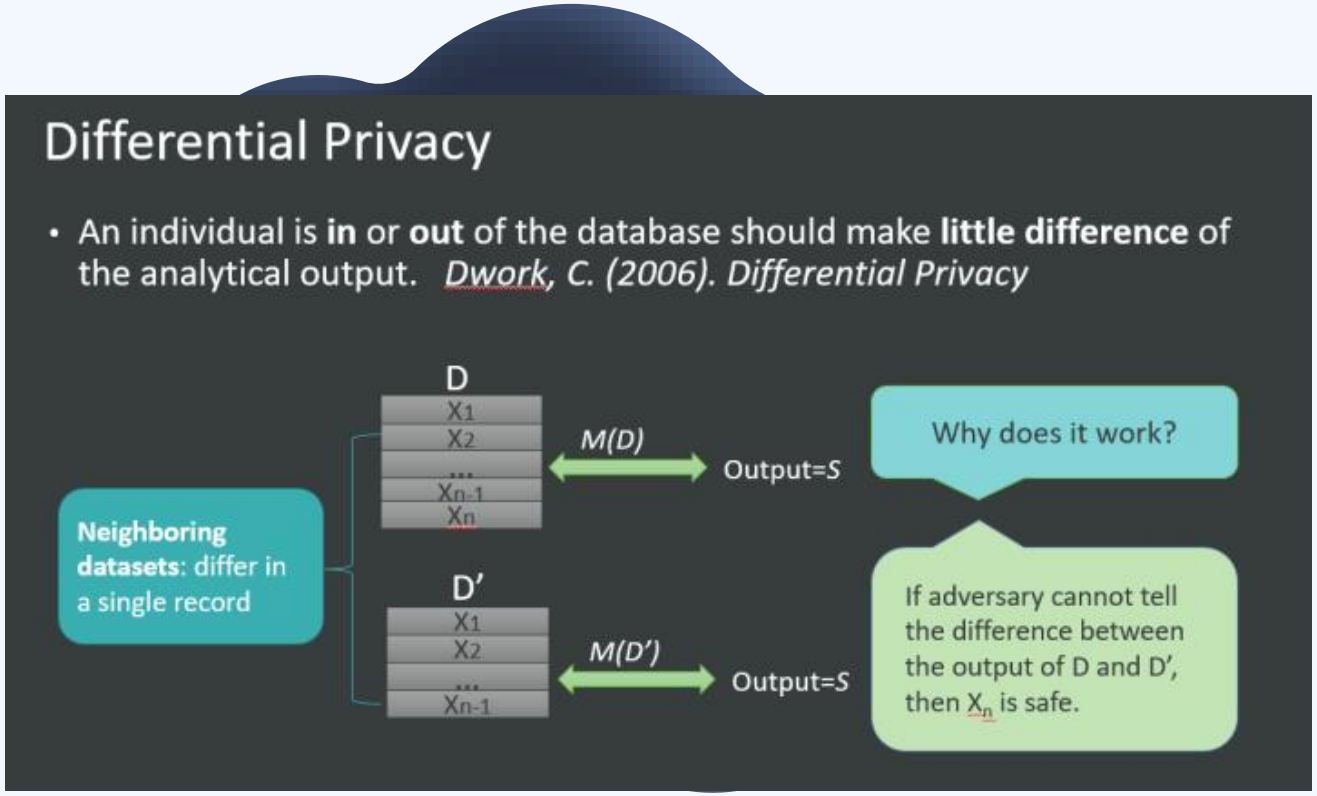
比如刚才的例子，本来两次查询结构是确定的2和3，现在加入随机噪声后，变成了两个随机变量，画出它们概率分布图。

现在，如果张三不在数据库的话，得到结果可能是2.5；张三在的话，得到的结果也可能是2.5；两个数据集查询得到某一个结果的概率很接近，以至于我们根本分不清这个结果来自于哪一个数据集，这样也就实现了攻击者的知识不会因为张三这个样本的出现与否而发生变化。

怎么解决



核心思想



让攻击者查询到对于差别只有一条记录的两个数据集，查询它们获得相同值的概率非常非常的接近。从而无法推测出那个单独的数据结果。

Laplace Mechanism

拉普拉斯机制

Index Mechanism

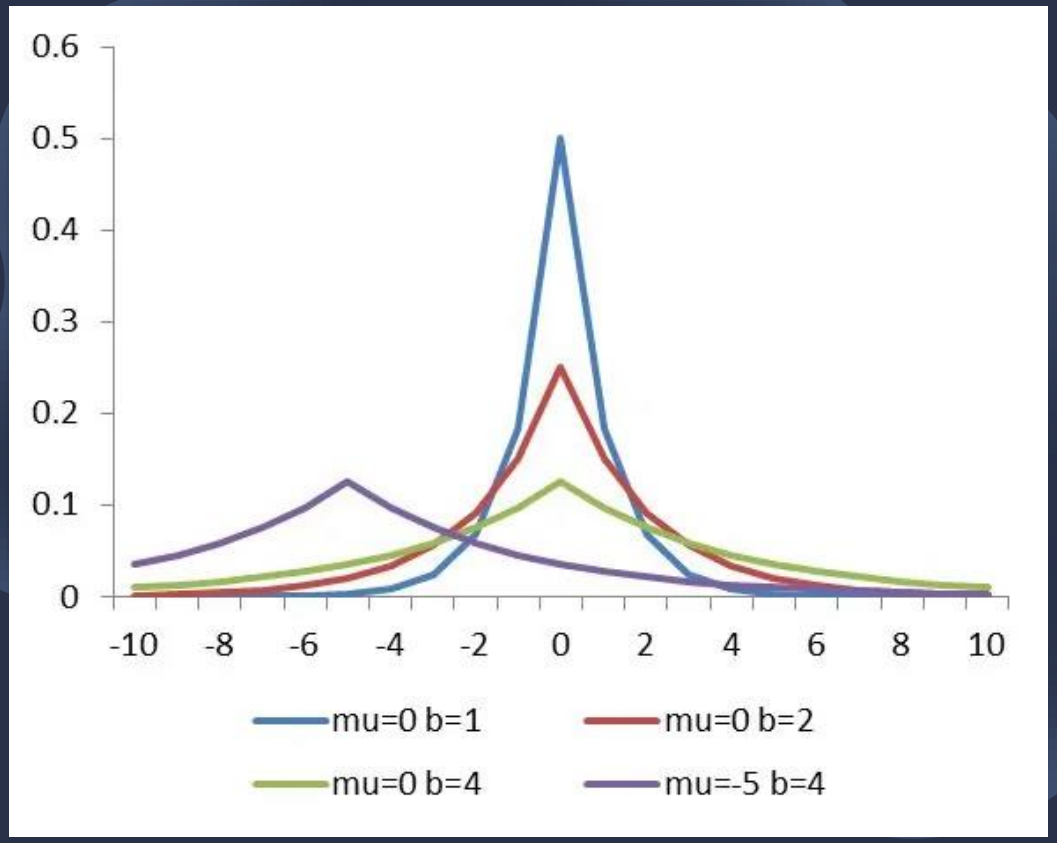
指数机制

常用的两种差分隐私机制

拉普拉斯分布

噪声就给我们提供了一个很好的概率化的方法。举个简单的例子，假如查询为“查询数据集中年龄小于20的人数”，并且查询结果为“50”：在传统模式下，输出就是50；在差分隐私模式下，会以比较大概率输出50左右的结果，也会以比较小的概率输出和50差别比较大的结果。但是，我们需要保证输出的期望为50（保证数据有效性）。

那么这个概率怎么用Laplace分布呢？我们可以直接在输出结果50上加均值为0的噪声。直观上来说，我们通过Laplace将查询结果概率化了。



存在问题

研究这个问题是因为我们不限定用户对数据集做出什么样的查询，直观上来说，如果查询的是人数，那么“有你”和“没有你”相差不大（只会相差1），只需要加一个小一点的噪声即可造成两个结果的混淆；那如果我们查询的是人的工资呢，加一个很小的噪声显然是无法满足应用需求的（因为数据相差太大，稍微对数据的改变依然可以看出数据差别很大）。

提出“敏感度”概念

由于数据集中少一条记录就会对数据查询f的结果造成一定的影响，我们自然想知道，这个影响最大是多少呢?
于是有了敏感度Δf。

$$\Delta f = \max_{x,y\in\mathbb{N}^{|x|},\|x-y\|_1=1} \|f(x)-f(y)\|_1$$

有了Δf之后，就认为Δf越大，噪声也应越大， Δf越小，噪声也应该越小。

指数机制

指数机制的提出是为了防止添加噪声破坏其价值。

例如：我们有大量的瓜和四个投标人：A，B，C，D，其中A，B，C各自出价1.00元，D出价3.01元。如何制定最优出售价格？假如以 3.01元出售，则收入为3.01元（只有D买）；以1.00元出售，收入为3.00元（ABC购买）；以3.02元出售，收入为零！

如果加入噪声可能会造成出售价的不确定性，从而与实际出售价产生一定的差距。

如何实现

指数机制适用于回答具有任意实用程序（和任意非数字范围），同时保留差异隐私。给定一些任意范围$\mathcal{R}$，首先我们需要定义一个实用性函数μ：

$$\mu: \mathbb{N}^{|x|} \times \mathcal{R} \to \mathbb{R}$$

简单来说，这里μ的作用就是给用户拥有的数据打个分，分越高，表示这个数据越重要（比如上述案例中的价格，可以直接作为μ看待）。因此对于数据集 X，我们希望选取的元素有最大效益。

$$\Delta\mu = \max_{r \in \mathcal{R}, \|x-y\|_1 \leq 1} |\mu(x,r) - \mu(y,r)|$$

指数机制

对任意的r按照exp($\epsilon u(x,r)/\Delta\mu$)的概率选取最优解即可满足差分隐私要求。

$$\ln\frac{\exp(\frac{\epsilon\mu(x,r)}{\Delta\mu})}{\exp(\frac{\epsilon\mu(y,r)}{\Delta\mu})}=\frac{\epsilon[\mu(x,r)-\mu(y,r)]}{\Delta\mu}\leq\epsilon$$

设随机化算法 M 输入为数据集 X，输出为一个实体对象 $r\in\mathcal{R}$,$\mu(X,\mathcal{R})$为可用性函数，$\Delta\mu$ 为函数$\mu(x,\mathcal{R})$ 的敏感度，若以正比于exp(($\epsilon\mu(x,r)$)/2$\Delta\mu$)的概率从输入中选择并输出r，则算法 M 是ϵ-DP的。

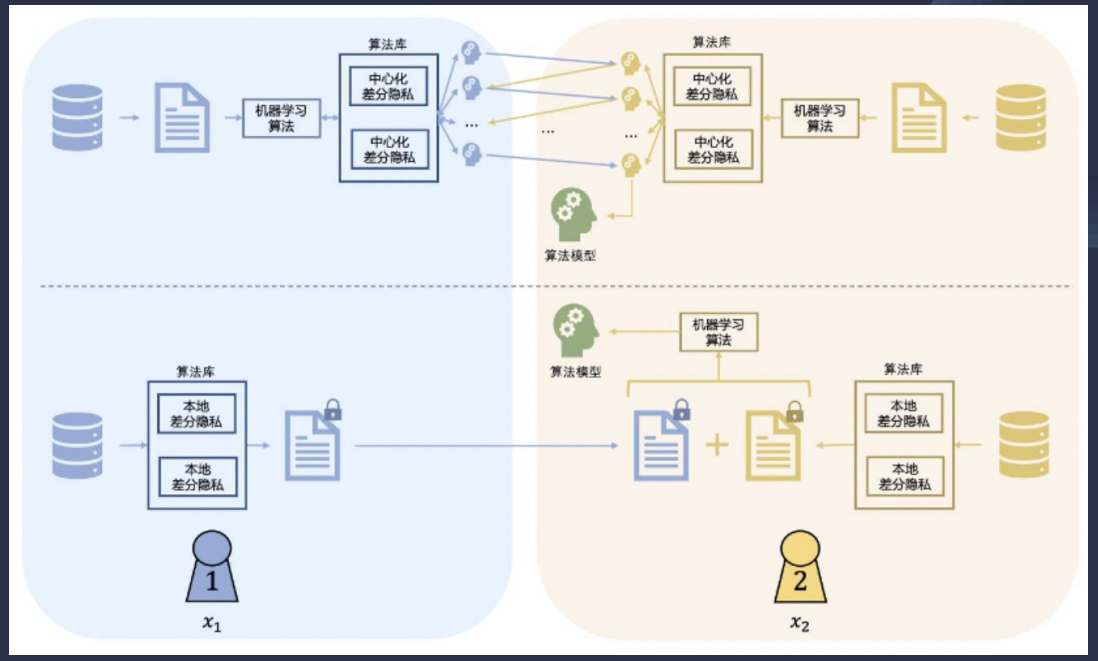
差分隐私应用

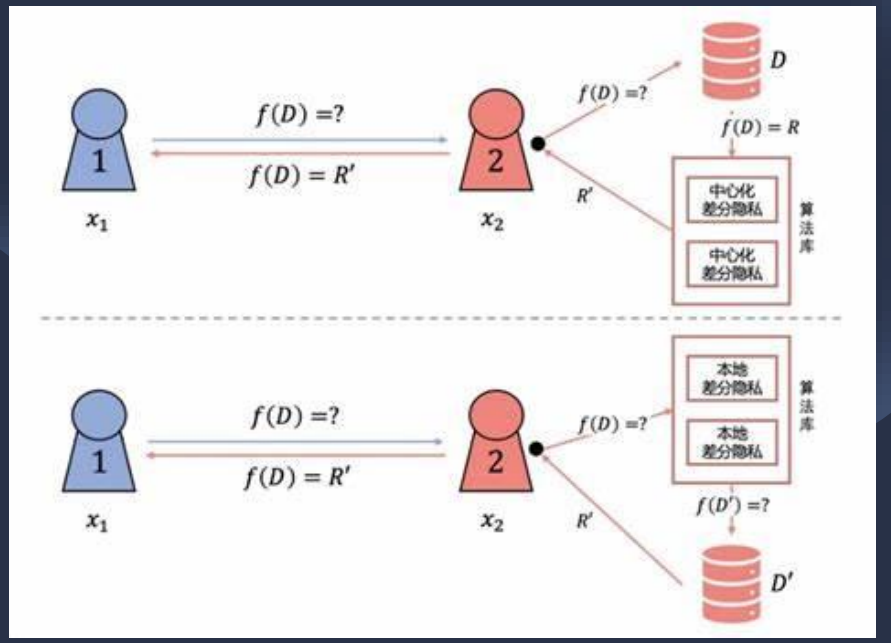
美国人口普查：通勤
模式识别
谷歌：恶意软件识别
苹果：IOS 10智能
助手改进
Uber：用户数据
采集分析
IBM：开源库
Diffprivlib
美国人口普查：
微观普查数据
2008
2014
2015
2016
2017
2019
2020
2021
谷歌：城市交通统计
微软：Windows数
据采集分析
Tensorflow：隐私
保护机器学习开源库
领英：广告
客户查询
阿里巴巴：
DataTrust平台

阿里巴巴差分隐私应用

阿里巴巴将差分隐私技术部署到了数据平台，辐射多种数据分析任务。比如对多维度的用户数据进行差分隐私处理，以生成人群画像，用于营销分析和广告投放等任务。

与浙大网安合作的Datatrust隐私增强计算平台中已经部署落地差分隐私保护的联邦学习决策树训练。





联邦学习提出的原因

数据孤岛问题

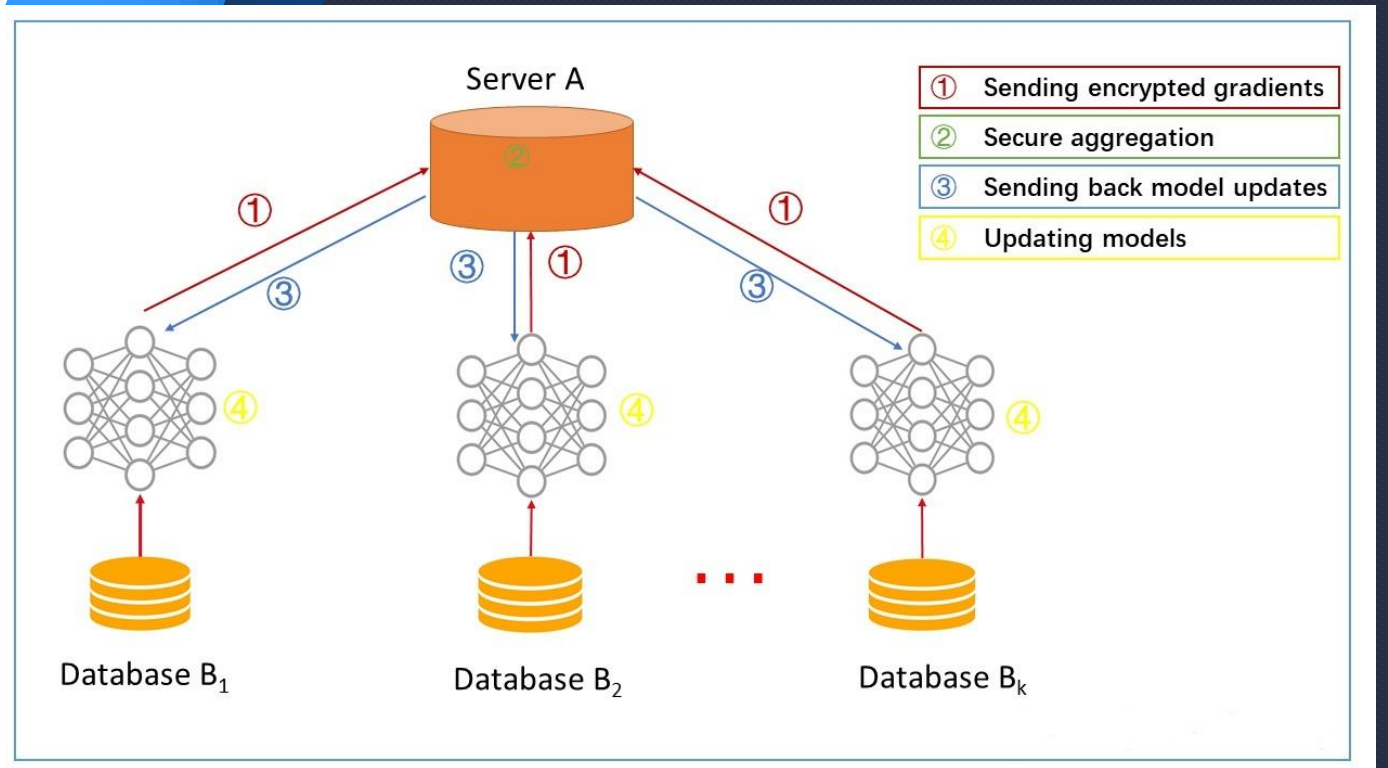
一个AI项目可能涉及多个领域，需要融合各个公司、各个部门的数据。（比如研究居民线上消费问题，需要各个消费平台的数据，可能还需要银行数据等等）但在现实中想要将分散在各地、各个机构的数据进行整合几乎是不可能的。

数据隐私问题

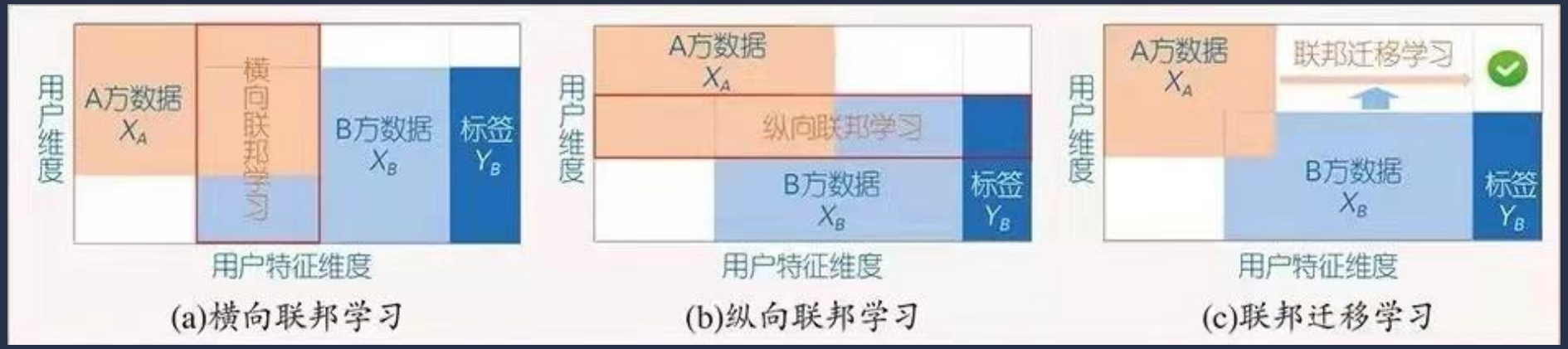
GDPR的出台，使得各方对数据所有权和隐私性的关注越来越多，对用户隐私和安全管理日趋严格，拿不到赖以生存的数据集。

定义

搭建一个虚拟的“联邦国家”，把大大小小的“数据孤岛”联合统一进来。他们就像这个“联邦国家”里的一个州，既保持一定的独立自主（比如商业机密，用户隐私），又能在数据不共享出去的情况下，共同建模，提升AI模型效果。



联邦学习分类



- 业态相同或相似
- 特征重叠多，用户重叠少
- 样本学习
- 触达用户相同或相似

- 特征重叠少，用户重叠多
- 特征学习
- 业态、用户均交集较少

- 特征重叠少，用户重叠少
- 迁移学习

联邦学习
面临的问题

Problems faced by federal learning

01 代价昂贵

02 系统异构型

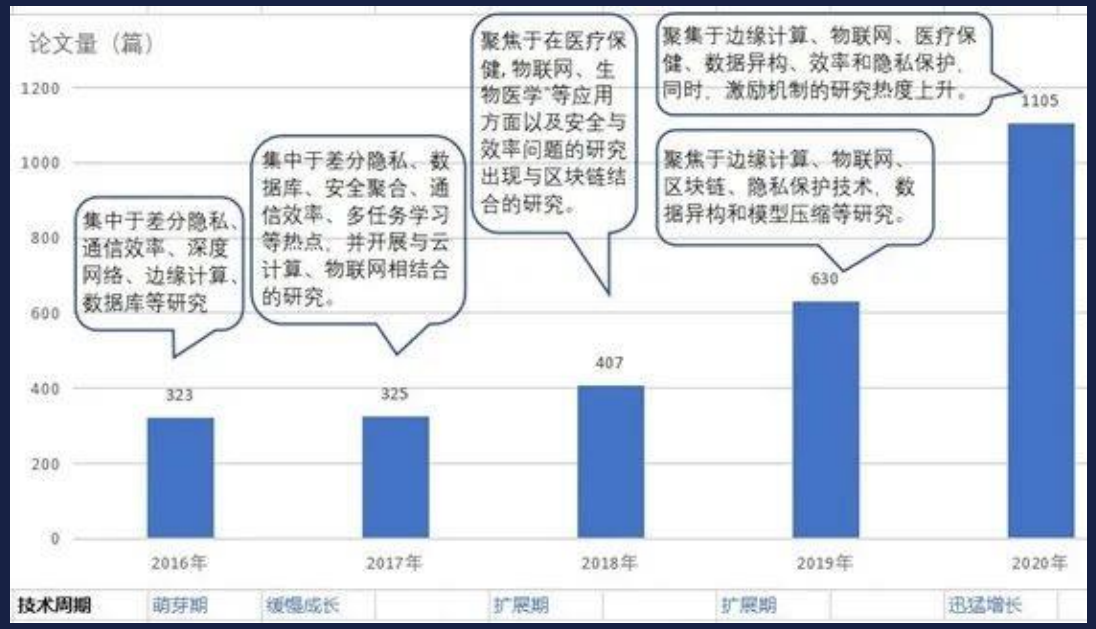
03 统计异质性

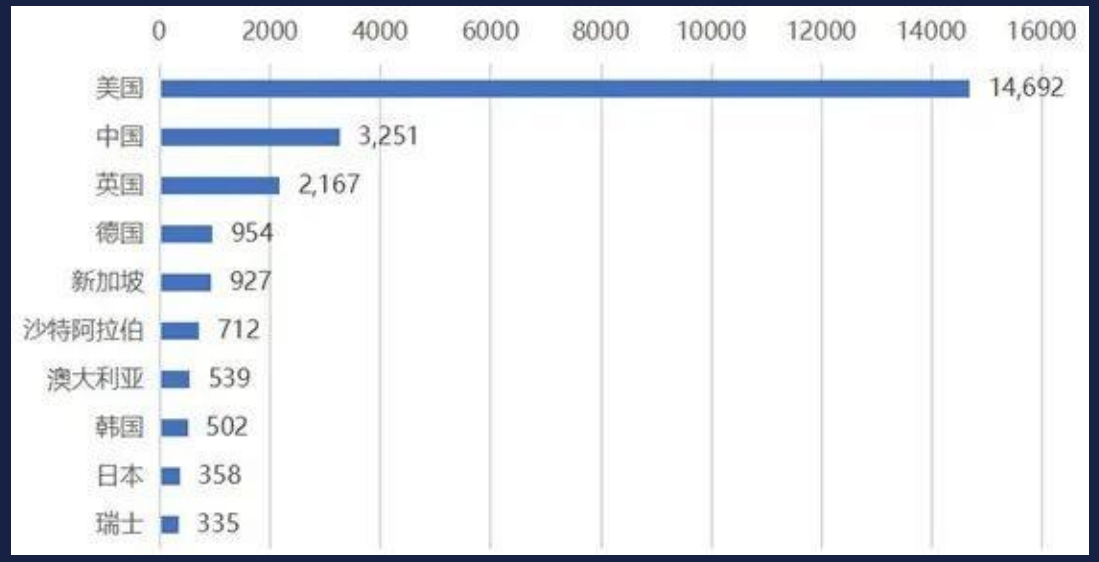
04 隐私问题

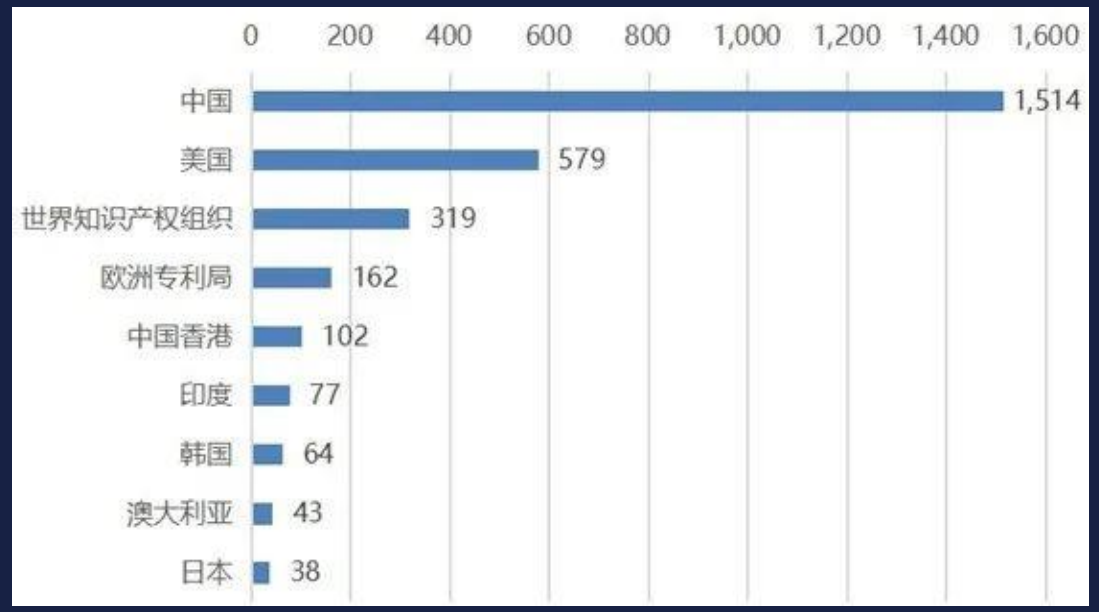
联邦开源框架

开源框架	受众定位	牵头公司	联邦学习类型	机器学习算法	安全协议
FATE	工业产品 学术研究	微众银行	横向联邦 纵向联邦	LR 、 GBDT 、 DNN 等	同态加密 RSA SecretShare
Tensorflow Federated	学术研究	Google	横向联邦	LR 、 DNN 等	DP
PaddleFL	学术研究	百度	横向联邦 纵向联邦	LR 、 DNN 等	同态加密 SecretShare
Pysyft	学术研究	OpenMined	横向联邦	LR 、 DNN 等	同态加密 SecretShare

联邦学习现状


论文量（篇）
1200
1000
800
600
400
200
0
集中于差分隐私、通信效率、深度网络、边缘计算、数据库等研究
集中于差分隐私、数据库、安全聚合、通信效率、多任务学习等热点，并开展与云计算、物联网相结合的研究。
聚焦于在医疗保健，物联网、生物医学"等应用方面以及安全与效率问题的研究出现与区块链结合的研究。
聚集于边缘计算、物联网、区块链、隐私保护技术，数据异构和模型压缩等研究。
聚集于边缘计算、物联网、医疗保健、数据异构、效率和隐私保护，同时，激励机制的研究热度上升。
323
325
407
630
1105
2016年
2017年
2018年
2019年
2020年
技术周期
萌芽期
缓慢成长
扩展期
扩展期
迅猛增长


0
2000
4000
6000
8000
10000
12000
14000
16000
美国 14,692
中国 3,251
英国 2,167
德国 954
新加坡 927
沙特阿拉伯 712
澳大利亚 539
韩国 502
日本 358
瑞士 335


0
200
400
600
800
1,000
1,200
1,400
1,600
中国 1,514
美国 579
世界知识产权组织 319
欧洲专利局 162
中国香港 102
印度 77
韩国 64
澳大利亚 43
日本 38

联邦学习应用

电子病历(HER)，已经成为现实世界医疗数据的一个重要来源，被用于重要的生物医学研究，包括机器学习研究。联邦学习是实现EHR数据大规模代表性机器学习的工具。

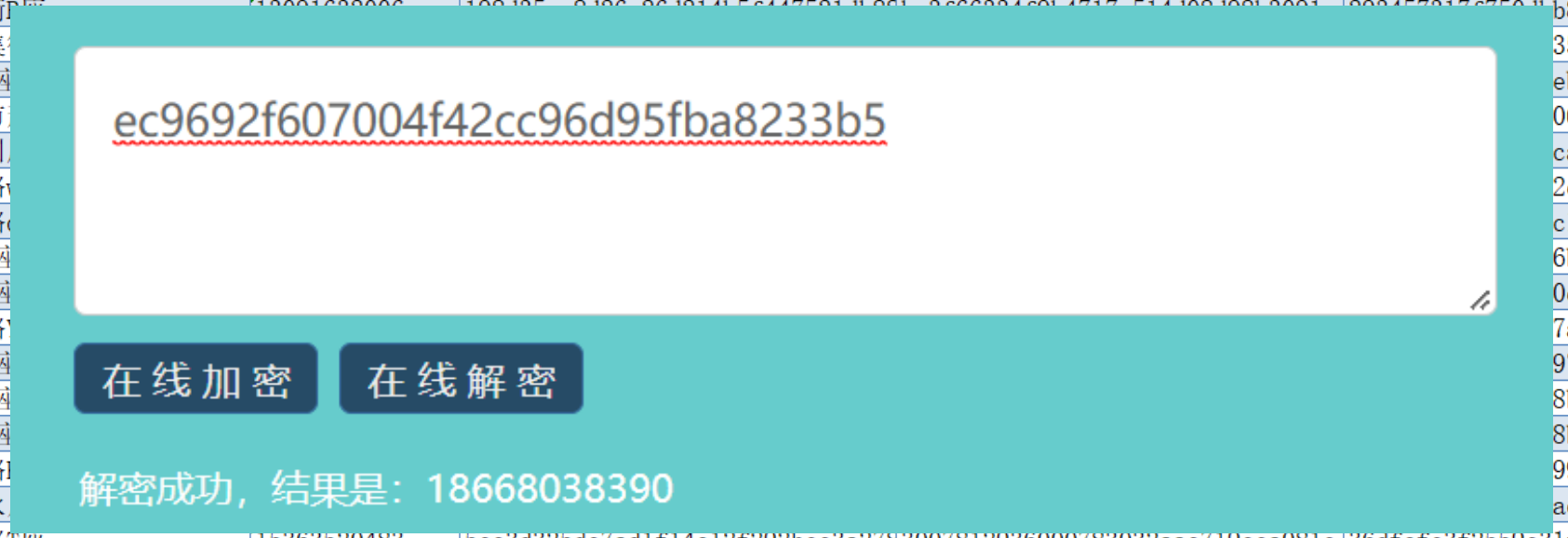
技术demo

黄弈萧&白玥

单向加密-MD5方法

Python的hashlib模块支持md5(),sha1(),sha224()等多种hash构造方法,利用update(arg)更新hash的对象，通过hexdigest()方法获得当前的摘要。

	姓名	地址	电话号码	姓名_md5	地址_md5	电话号码_md5
2	姜志强	新疆维吾尔自治区马鞍山县华龙合肥街X座	18668038390	cec9e66ed2c0eedbb8493a50f21dadb7	dbac944e86238ee21a32e5ab3b565a08	ec9692f607004f42cc96d95fba8233b5
3	王琴	新疆维吾尔自治区沈阳县牧野赵路Q座	14500910085	66e8d1451ceb0f53672366eeed5c96b1	295788fafd86f9f5aca714e144e471da	0e52e3f7ee98ca3fa0eda1a57fe7028d
4	黄建国	河南省昆明县璧山刘街				b891c4dab4ba9e4aa
5	王岩	重庆市六安县花溪辛集				35ffe0f8158bf11f5
6	刘秀云	江苏省楠县魏都朱街F座				ebd0c46ba13c8f446
7	王芳	新疆维吾尔自治区勇市				0631b6eeeb55ab599
8	龙晶	辽宁省乌鲁木齐市永川				cae0cd7b19068bb97
9	洪金凤	山东省潜江县南长曾路				2d46b43d15f2484b3
10	张帆	山西省林市西峰银川路				c163700eb2e17ed01
11	吴雷	河南省鹏县南湖李街A座				6bf2a8338d0f102a5
12	郭俊	台湾省玉县丰都郝路f座				0d5345d1273eaae7c
13	连桂英	浙江省六安县闵行侯路				7a19d8349fc38a2ed
14	温桂珍	河南省鑫市怀柔冯路n座				97bcdbb75366813ad
15	何倩	重庆市畅市清河马路V座				8b324ab78366de2c9
16	何淑华	山东省帅市东城葛街a座				8bec8e92a5b961975
17	林敏	浙江省重庆市城东赖路				99d596237aee66ced
18	刘荣	内蒙古自治区沈阳县永				ac3aeebd64a75d8d7
19	杨雪	广东省帅市友好台北路T座	15363520483	5ee3d325de7ad1f14c12f292bee3a278	3007812936000783032aae719eea081e	36dfcfe3f2559e31f417700d6277c470
20	杨冬梅	西藏自治区强市萧山宜都街p座	18052974738	6cb25c4094c4e3355408b34a1dd749c1	5c5230a519a5f14ab49a2954edcc9f71	9cc2b3d50069f6d43509d52a14b230a1



MD5密码认证，带“加盐”方法

Python的hashlib模块支持md5(),sha1(),sha224()等多种hash构造方法,利用update(arg)更新hash的对象，通过hexdigest()方法获得当前的摘要。

	电话号码	姓名_md5——加盐	地址_md5——加盐	电话号码_md5——加盐
2	18668038390	cec9e66ed2c0eedbb8493a50f2		b5kM7ts09pfT8N:iDtB0K?
3	14500910085	66e8d1451ceb0f53672366eeec		8dkM7ts09pfT8N:iDtB0K?
4	13091632006	198d35cc8d26e26d214b5f4475		aakM7ts09pfT8N:iDtB0K?
5	15874393614	12498e3f7d687aeb6c504e1cb9		f5kM7ts09pfT8N:iDtB0K?
6	15371166825	c3a410451c5c27589220285c42		46kM7ts09pfT8N:iDtB0K?
7	14761535765	8b9d24beb7afd45c595e51597f		99kM7ts09pfT8N:iDtB0K?
8	14757463213	c5d0124f736dcab6f532c1deb6		97kM7ts09pfT8N:iDtB0K?
9	18095092049	2ba6384ebfc475ec19a31ed2f7		b3kM7ts09pfT8N:iDtB0K?
10	18267257467	243dc461dad55e8d51fa1bd487		01kM7ts09pfT8N:iDtB0K?
11	18293885886	73fe68735a2d56fc1543e42527		a5kM7ts09pfT8N:iDtB0K?
12	13875810826	0efa865207c24d6339b5e158dc		7ckM7ts09pfT8N:iDtB0K?
13	18803953434	32f4b4d3d5703f5f938c16d8ef		edkM7ts09pfT8N:iDtB0K?
14	18688445358	8915e33168ddcf76d23bd03f7c		adkM7ts09pfT8N:iDtB0K?
15	15074990036	14e8ca1755e5d9104bf4c0253b		c9kM7ts09pfT8N:iDtB0K?
16	15216434650	118402803b2d7d091e7429e25c		75kM7ts09pfT8N:iDtB0K?
17	15769779471	523dd5cfd4b0e1da5002766674		edkM7ts09pfT8N:iDtB0K?
18	15104828095	4b648184417daa15b84e461cf5		d7kM7ts09pfT8N:iDtB0K?
19	15363520483	5ee3d325de7ad1f14c12f292bee		70kM7ts09pfT8N:iDtB0K?
20	18052974738	6cb25c4094c4e3355408b34a1dd749c1kM7ts09pfT8N:iDtB0K?	5c5230a519a5f14ab49a2954edcc9f71kM7ts09pfT8N:iDtB0K?	9cc2b3d50069f6d43509d52a14b230a1kM7ts09pfT8N:iDtB0K?

对称加密-AES

填充：pkcs7padding

AES加密模式：CBC

key：16位

iv偏移量：16位

字符集：utf-8

输出：base64

1	AES加密电话号码	解密	原电话号码
2	b'StEa7k+NIygaBJ8WIRVQRQ=='	b'18668038390'	18668038390
3	b'dcjAZnGLxGWPGmbkrX7xUA=='	b'14500910085'	14500910085
4	b'Q/AlCuLrCIuC/ZJ4Ua3Ejg=='	b'13091632006'	13091632006
5	b'fCfwG6eiYRHoIT2BKc/BEg=='	b'15874393614'	15874393614
6	b'bv0bZogOH5aWOrF/Z6KLFw=='	b'15371166825'	15371166825
7	b'IAS88SFcW7Ggy23xD+y3/g=='	b'14761535765'	14761535765
8	b'BurCjcaW7u5P5TacJFL11A=='	b'14757463213'	14757463213
9	b'UapUiCgKRUPCD9L6UOOvWA=='	b'18095092049'	18095092049
10	b'dJkMww4zK4+aN68VIx5IVQ=='	b'18267257467'	18267257467
11	b'YRajCyxXL8es6yedyKbEqA=='	b'18293885886'	18293885886
12	b'txah+ZScxh4jWCV3xFAhAQ=='	b'13875810826'	13875810826
13	b'Z7J3sGkyRRWI2YSajbF/9w=='	b'18803953434'	18803953434
14	b'HMULVOPRDkzKpK1CO3Wd/g=='	b'18688445358'	18688445358
15	b'FyaLvJcnzxzlVU8wVpaAGg=='	b'15074990036'	15074990036
16	b'BNRGzCTV6R6zWGtWLyDGvg=='	b'15216434650'	15216434650
17	b'CgAOAfOET69sE4h84EOUOQ=='	b'15769779471'	15769779471
18	b'ddg56y+A+9r0AIqQ9BuAqA=='	b'15104828095'	15104828095
19	b'bBapr+QXWK6mw9X1NuC1kQ=='	b'15363520483'	15363520483
20	b'z/1idp5kAefR4zzxU6ILkg=='	b'18052974738'	18052974738

1. 生成pubkey和privkey， 以文件的形式保存到本地；

2. 从文件中读取公钥或者私钥；

3. 加密数据使用pubkey， 利用rsa.encrypt()加密；

4. 解密数据使用privkey， 利用rsa.decrypt()解密；

非对称加密-RSA

1	公钥	私钥	rsa加密电话号码	解密电话号码	原电话号
2	PublicKey(730613172942348582152555823669479489629192999036545004366727924593650743366993456513748578869702966422141110048026118484601461349065176920688108239632819 1, 65537)	PrivateKey(730613172942348582152555823669479489629192999036545004366727924593650743366993456513748578869702966422141110048026118484601461349065176920688108239632819 1, 65537, 128715987835762343859703824405874699593491651538458407013110771279709041878953764816604731147902696622625678437276717073489283147656376736010371076262204 9, 458540928657737962062100512089729870143281883838958861768314255241563668514129546 3, 159334342319459458614671842040219359637391503993711603980535171647814365 7)	b"{\x9a\x92'\x893$\x11\x9e>\xe1\xe2\x08\x89\xb2>\xb3\xe0\xc1(\x9f\x0c\x11T\xc5\xc4'\xbaT<\x88C\x8f\x9a\x16E@\x11\xa6\xf4\xd4\x0eZ-\x8f0\xa8;b|\xcc\xe0\x97e\xa4\x12_\x85c6KH\r"	18668038390	18668038390
3	PublicKey(112582410264254994045204284811566222052388240686995962427550961894522089372152009339088667797077932443751390995261943679224289869330292057586539384856475 57, 65537)	PrivateKey(112582410264254994045204284811566222052388240686995962427550961894522089372152009339088667797077932443751390995261943679224289869330292057586539384856475 57, 65537, 760335392328942473936064642880316852199636223972279520424467571663860444410053426128767139653455172691651742301803818410052854743506481403276032675267904 1, 716353662556583625829139370240935816509873395807732578786280041490225541957501891 3, 157160375033836432977157379151760368560940270849722224520074199207232118 9)	b'U\x9a\xfc9\xd9\xb4i.\xcd\x1d\xb3dH\xab\xacN\x92\xe9\x81\xedWX\xb1\x9e\x14k\x19\xa6\xe81"%*\xfe\x1d\xe69\xb2\x10\x1d\xb9z$\xed1\xa0\xc7\xb2;.\x17\x7f\x07\x85\x86\xa5\x87\xd0\xa9\xee\xf6Q\x16\x1d'	14500910085	14500910085

k=10，测试编写的mondrian算法的正确性

K匿名-Mondrian算法

```
before cut:
19       0
20       0
47       0
52       0
63       0
        ..
32480    0
32504    1
32513    0
32548    1
32554    1
Name: age, Length: 1675, dtype: int64
after cut:
844,831,
       age  education_num       occupation
19      43             14  Exec-managerial
20      40             16   Prof-specialty
47      44             14  Exec-managerial
52      47             15   Prof-specialty
63      42             16   Prof-specialty
...    ...            ...              ...
32326   48             16   Prof-specialty
32365   41             14   Prof-specialty
32473   41             14   Prof-specialty
32480   41             14  Exec-managerial
32513   46             14   Prof-specialty

[844 rows x 3 columns]
       age  education_num       occupation
100     76             14  Exec-managerial
189     58             16   Prof-specialty
272     50             14  Farming-fishing
285     57             14            Sales
324     76             14     Craft-repair
...    ...            ...              ...
32438   66             14  Exec-managerial
32470   50             16   Prof-specialty
32504   50             14            Sales
32548   65             15   Prof-specialty
32554   53             14  Exec-managerial
```

```
=>begin searching: k=10
       age  education_num         occupation
0       39             13       Adm-clerical
1       50             13    Exec-managerial
2       38              9  Handlers-cleaners
3       53              7  Handlers-cleaners
4       28             13     Prof-specialty
...    ...            ...                ...
32556   27             12       Tech-support
32557   40              9  Machine-op-inspct
32558   58              9       Adm-clerical
32559   22              9       Adm-clerical
32560   52              9    Exec-managerial

[30162 rows x 3 columns]
=>end searching, total search time is:8.392462253570557(s)

=>begin publishing:
===>after generalization:
           age education_num         occupation
0      [39,39]       [13,13]       Adm-clerical
1      [50,50]       [13,13]    Exec-managerial
2      [38,38]         [7,9]  Handlers-cleaners
3      [53,53]         [6,7]  Handlers-cleaners
4      [28,28]       [13,13]     Prof-specialty
...        ...           ...                ...
32556  [27,27]       [12,13]       Tech-support
32557  [40,41]         [1,9]  Machine-op-inspct
32558  [58,59]         [6,9]       Adm-clerical
32559  [22,22]        [9,10]       Adm-clerical
32560  [52,53]         [6,9]    Exec-managerial

[30162 rows x 3 columns]
===>publish file path:./data/adult_mondrian.data
=>end publishing, total lossmetric is:0.02320903952749525, total publishing time is:6.602484226226807
mondrian end.
```

mondrian算法结果的分析

对相同的QI = [‘age', 'gender', 'race', 'marital_status'], S = ['occupation']，变换不同的匿名属性值，观察结果的变化。

k=5，LM=0.00974

```
===>k=5
=>end publishing, total lossmetric is:0.009743458384086665, total publishing time is:12.828240871429443
```

k=10，LM=0.00265

```
===>k=10
=>end publishing, total lossmetric is:0.026514265886586 89, total publishing time is:6.278696298599243
```

k=20，LM=0.0503

```
===>k=20
=>end publishing, total lossmetric is:0.050286867960199086, total publishing time is:3.1341333389282227
```

Mondrian算法处理标签型属性

- 在原始数据集中的取值实例集合，一一映射到连续的整数集合上，使得标签型属性变成数值型属性；最后在发布泛化后数据时，将该属性的整数与标签的对应关系也一并发布出去；

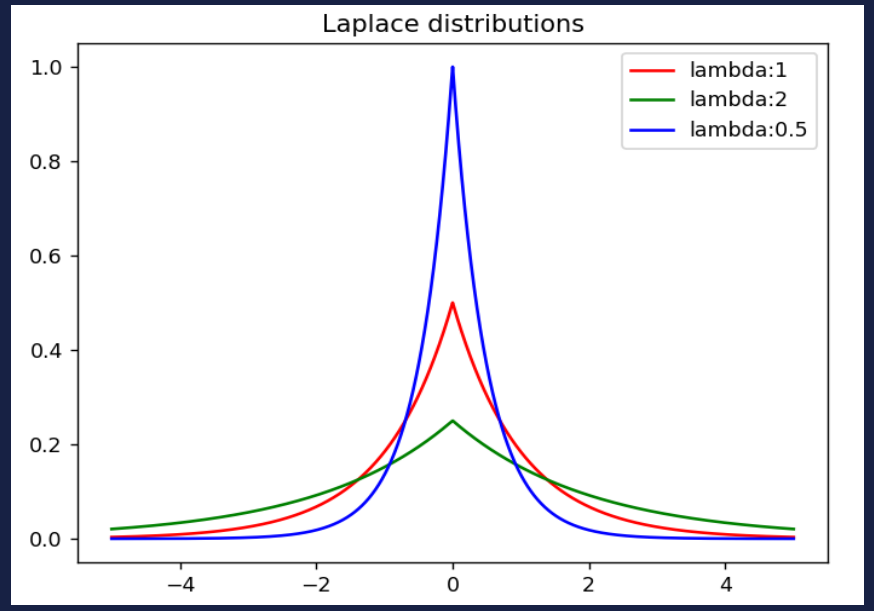
测试QI=['age', 'gender']，k=10

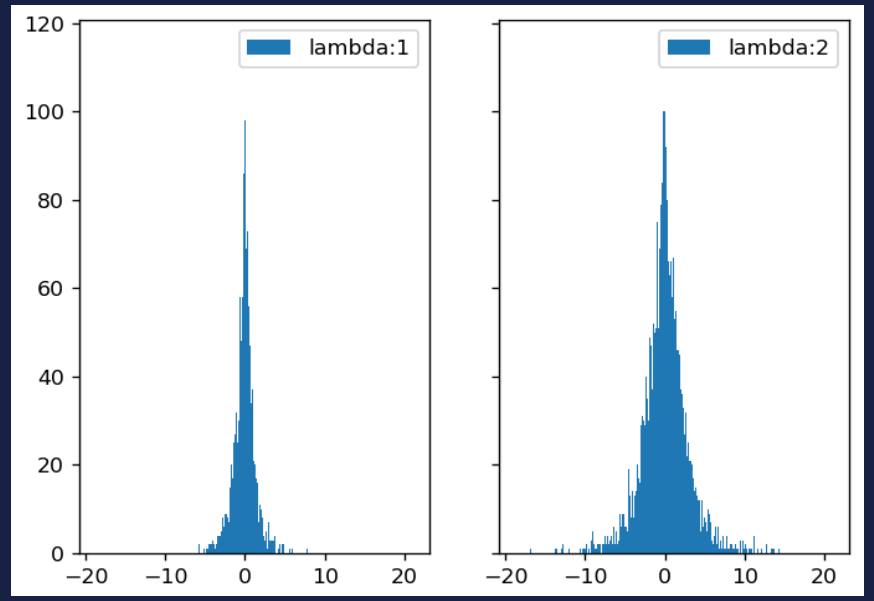
```
===>publish file path:./data/adult_mondrian.data
===>k=10
gender:
  Female: 1
  Male: 0

=>end publishing, total lossmetric is:0.046232536085957778, total publishing time is:6.410412311553955
mondrian end.
```

yaml映射关系：

data_privacy > k_anonymity > data > adult_mondrian.yaml

```yaml
gender:
  Female: 1
  Male: 0

```

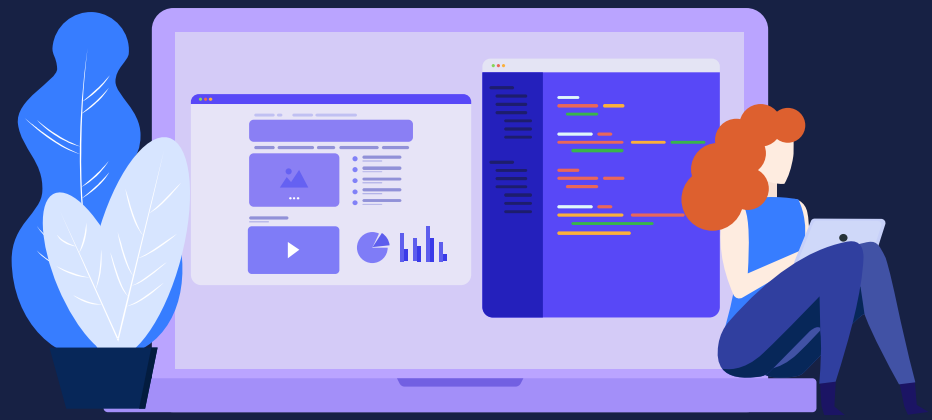


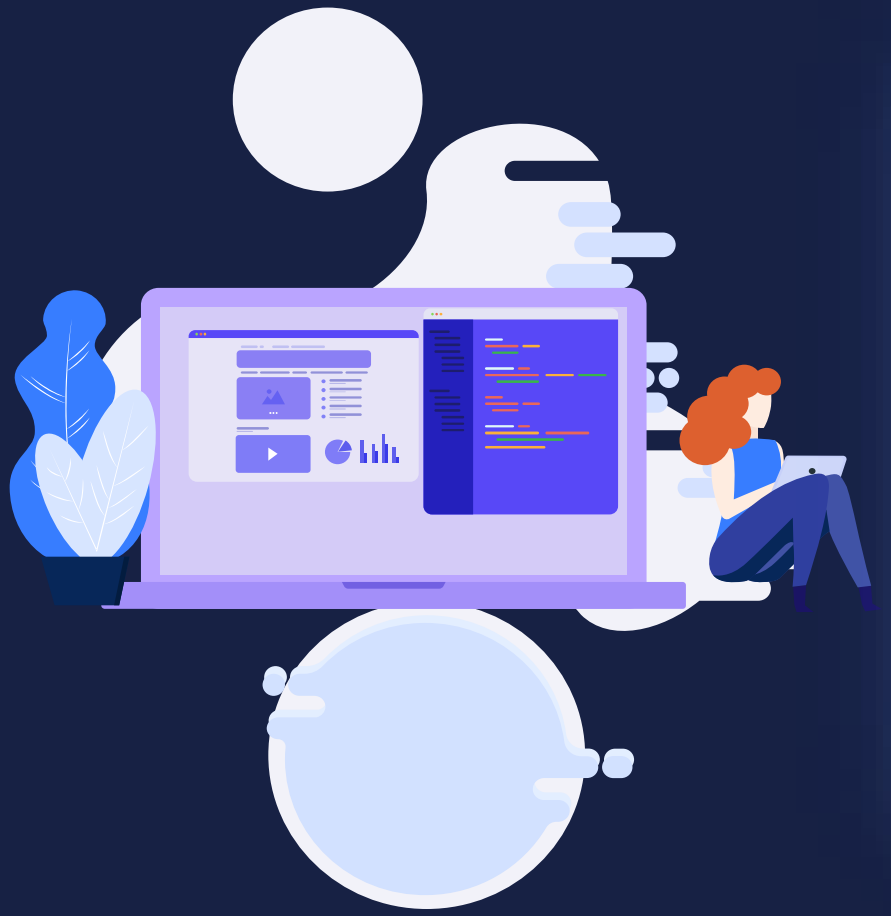


差分隐私——拉普拉斯机制（针对数值型数据）

Laplace分布的概率密度函数：

$$p(x) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}}$$





算法实现

I. 读取数据集中的数据
II. 计算基于拉普拉斯分布的噪声
III. 计算基于拉普拉斯加噪的混淆值

脱敏后的数据（部分）

潘淑兰：5720.76
方　敏：6450.95
顾　婷：5671.89
王志强：7180.84
周　明：9330.91
徐　坤：9272.70
席　岩：5389.17

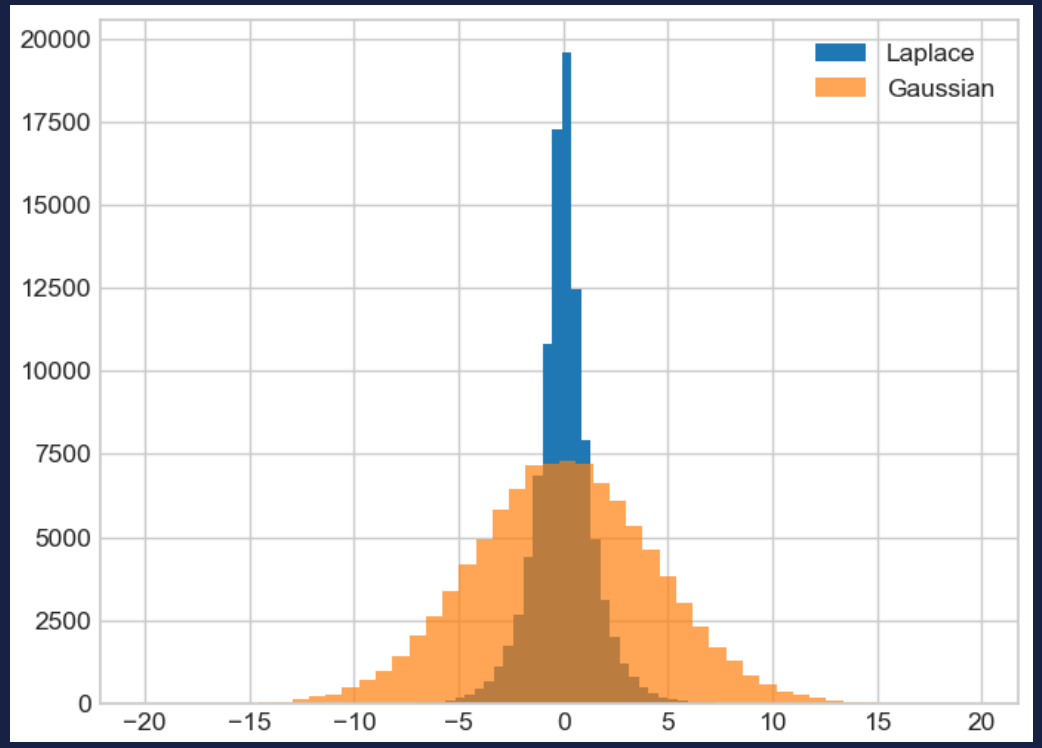
差分隐私——高斯机制（针对数值型数据）

拉普拉斯机制VS高斯机制

（f:D→R）

经验概率密度函数图

(epsilon=1，delta=10^(-5))



差分隐私——指数机制（针对非数值型数据）

有限集合的指数机制

评估函数：

```
def score(data, option):
    return data.value_counts()[option]/1000
```

I. 计算集合中所有元素的评估值
II. 根据评估值计算可能性，再归一化
III. 根据可能性选择元素

结果：

Married-civ-spouse: 184
Never-married: 14
Divorced: 2

感谢聆听

Thanks for listening