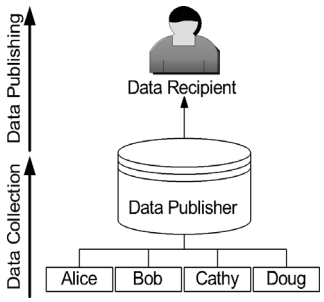
第四章 隐私保护的数据发布

2 课时

4.1 基本概念

我们正生活在一个大数据的时代，越来越多的设备和传感器通过数字网络相连，数据收集者们通过其中的应用程序大量收集个人数据，并将其提供给有需求的数据分析者。分析者可以利用各种工具对获取的数据进行挖掘，以此产生能够支持商业计划、政府决策、科学研究、广告投放等应用的策略，实现商业利益和科研价值，最终使大众受益。

4.1.1 发布框架

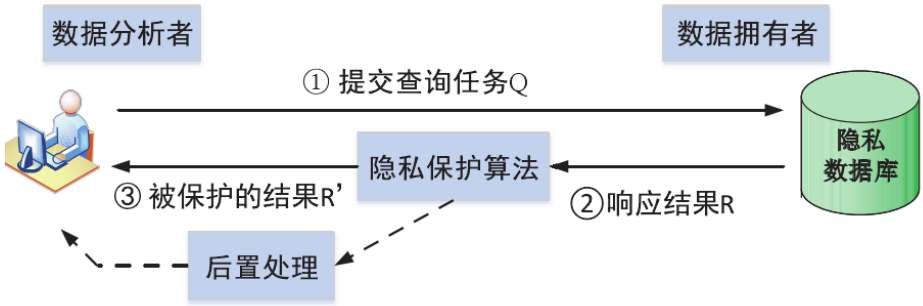


在隐私保护数据发布领域中，数据发布者从数据的拥有者采集到应用中的数据，例如医疗数据、金融数据、电信数据、访问数据、社会调查数据等。然后，将数据发送给数据申请者。这个模式中包括将数据公布于众，或者将数据发送给申请的单位、机构或者个人等，使数据用于科学研究或者支持决策，服务于公众，如图所示。

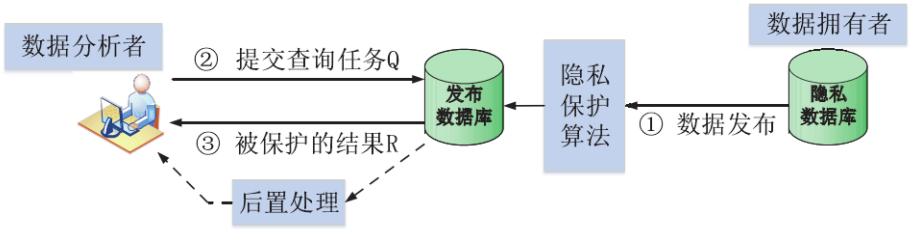
在数据发布应用的第一个阶段数据收集，假设是诚实的模型，数据所有者将数据发送给诚实的数据发布者。然而，在第二个阶段，数据发布阶段是非诚实模型，数据的接收者是不诚实的，数据的接受者可能是一个攻击者。例如，某医药公司获得一份某医院的电子医疗信息，但是无法保证所有的员工中都是诚实的。会有人员通过发布的数据获取其中的敏感信息，称之为攻击者。攻击者设法获取的敏感信息所对应的个体，称为攻击对象。

隐私保护的数据发布技术（PPDP，Privacy Preserving in Data Publishing）是数据发布者将原始数据表进行匿名化操作，然后再对它进行发布，以保护数据中的敏感信息，避免隐私泄露。

数据发布流程框架主要分为两种，交互式和非交互式数据发布框架。



交互式数据发布通常表现为数据的在线查询发布，较多出现在政府机关和研究机构的对外数据发布中，供有兴趣的用户查询。例如美国的联邦经济数据研究网站，能够提供一系列经济数据在不同时间周期内的聚合查询和批量查询。其基本结构如图所示。



非交互式数据发布通常表现为离线发布，例如数据挖掘竞赛发布的公开测试集，交通管理局发布的周期性的路况信息等。数据拥有者先通过隐私保护算法对需要发布的数据集进行完整的匿名处理，然后数据分析者根据已发布的数据集进行各种需要的查询。在非交互式数据发布中，由于数据拥有者并不知道数据分析者会对匿名数据集进行何种查询，因此设计隐私保护算法需要同时满足隐私性以及较高的可用性。

4.1.2 属性分类

假设原始数据是经过预处理的结构化数据，在 PPDP 最基本的格式中，数据发布者有一个格式表：D（显示标识属性，准标识属性，敏感属性，非敏感属性）。

显式标识属性（Identifier Attribute）：也称为显式标识符或标识符，是能唯一标识单一个体的属性，比如姓名、身份证号码。

准标识属性（Quasi-identifier Attribute, QI）：是组合起来能唯一标识单一个体的属性，如性别和年龄的组合等。

等价类：准标识属性完全相同的多条记录，称为一个等价类。

敏感属性（Sensitive Attribute）：包含敏感数据的属性，尤其是涉及个体隐私的细节信息，比如疾病、病人患病记录、个人薪资、地理位置等。

数据的发布者不能把原始数据直接发布，要避免数据接收者把数据表中的敏感属性与个体链接起来。敏感属性包含个体隐私的信息，是数据接收者进行数据挖掘、数据分析的对象，不能被移除。

4.1.3 背景知识

数据发布隐私保护需要关注的一个重要问题就是攻击者可能拥有的各种背景知识，这些知识可以包括外部数据、常用知识、有关匿名算法的知识和过去发布的数据，这些信息可以通过关联已发布的数据集来推测匿名数据集中的个人敏感属性。

（1）外部数据。主要包括公开可获得的数据，如选民登记记录，电影评分统计等；攻击者容易获得的关联数据，如目标用户隔壁邻居的年龄和地址等。这些外部数据可能包含除原始数据中敏感属性外的所有类型的信息。通过这些从外部数据获得的额外信息，攻击者可以在匿名数据中推敲目标个体存在的元组，并进一步发现目标个体的敏感值。

（2）常用知识。这是关于目标个体敏感信息分布的额外信息，可以从许多来源获得。例如，攻击者可能有一个常识：冬天很容易感冒，或者对手可能从他的同事那里听说另一位同事的工资超过 10K。如果目标个体可能患有某些疾病或其工资数目在某一个固定的范围内，那么攻击者就可以利用这些非关联的常识信息排除匿名数据集中的一些个体，从而以更高的概率推断出目标个体。

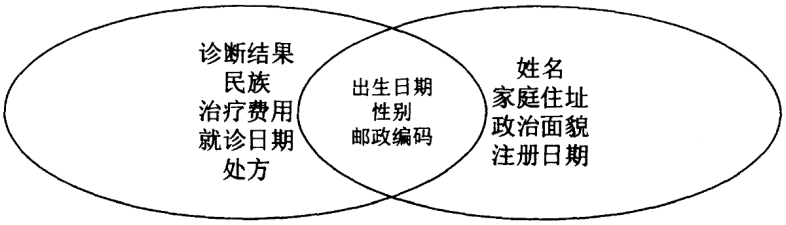
（3）基于隐私保护算法的知识。攻击者可能知道当前匿名数据集所使用匿名算法的机制，因为生成匿名数据的算法很可能会在数据发布时公布。在某些情况下，这些算法本身就可能披露敏感信息。

（4）过时数据。在数据发布的场景中，有些需要数据拥有者在固定时间周期中进行多次发布，以确保数据集的实时性。那么这种方式下攻击者可以获得所有先前发布的数据，并使用这些数据来排除目标个体的可能候选元组或敏感属性值。

4.1.4 相关攻击

很显然，攻击者有了背景知识，如果发布数据表仅仅简单移除了显式标识属性是不够的，隐私信息仍然有可能被准标识属性联合起来定位获得。

Sweeny 等人在 2002 年说明了在美国，公众可以从公开的选民数据集获取姓名、社会保障号、年龄、邮政编码这些人口统计信息。这将导致 87%的美国人遭受“链接攻击”。这意味着他们能够被准标识属性联合起来唯一确定。



链接攻击示例

如图所示，公开数据集中包含姓名、家庭住址、政治面貌、注册日期、出生日期、性别、邮政编码。公众可以获得数据集所含个体的这八个属性信息。另外一张表，是医院的医疗记录，它仅仅从原始的医疗记录中移除了显式标识属性，公开了诊断结果、就诊日期、处方、出生日期、性别、邮政编码等属性。由于人们可以从公开数据集中获取与医疗记录相重叠的属性出生日期、性别、邮政编码，从而可以唯一确定个体的敏感属性，造成隐私的泄露，即诊断结果、就诊日期等。

总之，如果攻击者有包含背景知识的数据，包含了个体的准标识属性值，通过连接这两张表，他能推断出一些敏感属性值，可以细分为三种类型的攻击。

记录链接：当攻击者能够将记录的所有者与发布的数据表中的相应的记录相对应时，称为记录链接。例如，通过准标识符确定一条记录的所有者身份。

属性链接：当攻击者能够将记录的所有者与发布的数据表中的敏感属性相对应时，称为属性链接。

表格链接：当攻击者能够将记录的所有者与发布的数据表本身相对应时，称为表格链接。

目前，已有的各种隐私保护方法都是为降低某些隐私泄露危险、抵御攻击者的攻击模型而产生的，在数据的发布过程中，数据集可能遭受到来自攻击者的隐私威胁除了链接攻击（Link Attack）之外，还有同质性攻击（Homogeneous Attack）、敏感性攻击（Sensitivity Attack）、概率攻击（Probability Attack）等。

4.1.5 基本匿名化方法

为了完成数据表的隐私保护的安全发布，需要对其数据进行匿名化操作，常用的方法有泛化、抑制、解剖、扰动等。

泛化是用一个更加泛化的值代替具体的值。对于分类型属性，泛化是根据分类树用属性值所在的类别来代替具体值。对于数值型属性，泛化是用数值所在的区间代替具体的数值。

抑制是抑制某个数据项，不发布这个数据项。对于分类型属性，抑制是泛化到分类树的根节点这种特殊的情况；对于数值型属性，抑制是泛化到属性值域这个最大的区间的特殊情况。泛化的逆操作称为细化，抑制的逆操作称为公开。

解剖是指不修改原始数据表中的准标识属性或者敏感属性，而是将数据表分割成两张表发布，一张是准标识属性表，一张是敏感属性表。这两张表中的数据通过等价类的标号链接，两张表中属于同一个等价类的记录具有相同的等价类标号。同一个等价类的敏感属性值如果相同，那么在敏感属性表中只出现一次，也就是敏感属性表中属于同一等价类的数值都是不同的。因而，同一个等价类中的记录链接到类内的敏感属性值的概率是相等的。

扰动是防止统计泄露中的一种针对数据的操作。它是保持数据的一些统计性质不变的前提下，对数据进行添加噪声，数据交换，或者人工数据合成操作。生成的数据已经不再是真实数据，它不会与真实的数据链接起来，从而保护数据的隐私信息。扰动对于数值型统计查询（例如聚合查询）很有用，因为它可以保留原始数据的统计信息。而且基于差分隐私 （DP，Differential Privacy）保护算法的扰动数据集能够达成最理想的隐私保护效果。但在非数值型数据集中，由于准标识符和敏感信息之间的关系失真太多，因此数据挖掘算法从扰动数据中学习的知识模型可能精度较差。

4.2 K-匿名模型

在实际应用中，数据发布的情况是十分丰富并且复杂。基本的数据发布是数据表只发布一次，并且数据表中只含有一个敏感属性。其他的数据发布属于在基本情况基础上的拓展，包括动态的数据发布、多敏感属性的数据发布等。例如，医疗机构产生的医疗数据中含有很多属性列，数据表如果用于医学的科学研究用途，那么就不需要发布关于医疗费用的属性列。而如果数据表记录用于给保险公司进行审核账目，就需要发布患者在医院期间产生各类医疗费用。需要发布的医疗数据随着时间会不断更新，会有修改或者新增医疗记录的情况。

本节以基本的 K-匿名模型为例，讲解数据发布过程中的攻防博弈。

4.2.1 K 匿名

表 2-2 医疗数据表

Table 2-2 A medical table

准标识属性			敏感属性
性别	年龄	邮政编码	疾病
女	21	100041	咽炎
女	21	100041	扁桃体炎
女	23	100042	胃癌
女	25	100043	咽炎
女	25	100043	口腔溃疡
男	28	100030	胃癌
男	28	100030	胃癌
男	29	100032	胃癌
男	29	100032	咽炎

表 2-3 公开数据表

Table 2-3 A publishing table

姓名	性别	年龄	邮政编码
Alice	女	21	100041
Bob	男	28	100030
Carol	女	21	100041
David	男	28	100030
Emily	女	23	100042
Ford	男	29	100032
Gita	女	25	100043
Hale	男	29	100032
Ivy	女	25	100043
Jack	男	30	100033

如果仅仅是将显示标识属性删除，是不够的。如表 2-2 所示，是医疗机构发送给医疗科研机构的医疗数据，它只是从原始数据表中删除了显式标识属性。如表 2-3 所示，是另外一个公开的数据集，它包含有表中全部个体的显式标识属性和准标识属性值的详细信息。科研机构可以将这两张表链接，从共同的准标识属性性别、年龄、邮政编码得出一些

背景知识，从而可能推断出医疗机构发布的数据表中的个体。例如，是一个女性，23 岁，邮政编码是 100042。将表 2-2 和表 2-3 链接起来，就可以推出患有胃癌，造成疾病泄漏。

K 匿名模型要求在所发布的数据表中，对于每条记录都至少存在其他 K-1 条记录，使得它们在全体准标识属性上取值相等，即这个模型要求每个等价类的记录不少于 K。

实现 K-匿名的方法就是泛化或者抑制。

例 1: 将准标识属性性别、年龄、邮政编码按照图 2-3 中的分类树进行泛化，得到了组编号分别为 1 和 2 的两个等价类，如表 2-4 所示。第一个等价类中有 5 条记录，第二个等价类有 4 条记录，即每个等价类至少含有 4 条记录，这个数据表满足 4-匿名。

表 2-4 4-匿名数据表

Table 2-4 4-anonymity table

组编号	性别	年龄	邮政编码	疾病
1	女	21-25	10004*	咽炎
1	女	21-25	10004*	扁桃体炎
1	女	21-25	10004*	胃癌
1	女	21-25	10004*	咽炎
1	女	21-25	10004*	口腔溃疡
2	男	26-30	10003*	胃癌
2	男	26-30	10003*	胃癌
2	男	26-30	10003*	胃癌
2	男	26-30	10003*	咽炎

为了满足匿名模型，需要使等价类中记录的数量至少为 k 条，因此 k 越大，隐私保护越好，由此带来的数据损失也就越大。然而，这个匿名模型只针对准标识属性有约束，并没有约束敏感属性。

4.2.2 l-多样化

同质性攻击。如果在一个等价类中全部敏感属性的取值相等，那么虽然攻击者不能确定哪条记录属于攻击对象，但是，能以 100%的概率确定攻击对象的记录的敏感属性。因此，这个模型仅能够从一定程度上抵御记录链接，不能够抵御属性链接。同质性攻击是等价类中的敏感值都相等，而导致的属性链接。它是由于等价类中的敏感值缺少多样性而造成的。

如表 2-2 所示的医疗数据表。攻击者知道是一名男性，年龄 28，邮政编码是 100030。虽然攻击者不能推断出表中哪条记录是他的医疗记录，但是可以得出患有胃癌，成功地进行了属性链接。

***l*-多样性匿名模型**。如果数据表中的每个等价类有至少 *l* 个敏感属性值，那么称数据表是 *l*-多样性的。

虽然 *l*-多样性原则 *k*-匿名性在有关防止属性泄露方面上迈出了关键性的一步，但它不足以防止（敏感）属性泄露，因为它容易遭受倾斜攻击和相似性攻击。

以倾斜攻击为例，在满足多样性的一个匿名表中，如果某个敏感属性值在全局出现的频率很低，而在某个等价类中出现的频率远高于全局的频率，那么这个等价类中被攻击者链接为此敏感属性值的概率远高于全局的概率，这就是倾斜攻击。表 2-4 中的表格满足了 2-多样性匿名，胃癌在全局出现的频率是 50%，但是在第二个等价类中胃癌出现的频率是 75%，因而使得第二个等价类中的记录更容易被链接到胃癌这种疾病。

可以看到，当总体分布是偏态分布时，满足 *l*-多样性并不会阻止属性公开。

4.2.3 T-相近

t-相近模型是一个首次提出敏感属性值的分布的隐私保护方法，它考虑了等价类内敏感属性的分布，要求每个 k-匿名组中敏感属性值的统计分布与该属性在整个数据集中的总体分布“接近”。

一个等价类满足 t-相近模型，则等价类中敏感属性值的分布与在数据表的分布差异不超过 t。如果数据表的每个等价类都满足 t-相近，则称这个数据表满足 t-相近。

t-相近是基于 *l*-多样性组的匿名化的进一步细化，用于通过降低数据表示的粒度来保护数据集中的隐私。这种减少是一种折衷，它会导致数据管理或挖掘算法的一些有效性损失，从而获得一些隐私。因为，满足这个模型的匿名表中，由于每个等价类与全局等价类的分布的差异不大（不超过阈值 t），使得匿名表丢失了很多准标识属性与敏感属性之间的相关信息，这可能正是数据接收者进行数据挖掘和科学研究所需要的信息。

4.2.4 其它模型

数据发布的过程中，如何保护隐私和确保可用性，总是存在矛盾，而相关研究也是在这个矛盾中逐步前进。

传统的数据发布隐私保护技术通过删除能够唯一识别个体身份的信息(标识符属性)实现匿名发布，典型的解决办法就是 K-匿名模型。如前面所述，虽然 k-anonymity 隐私模型切断了个体与数据表中某条记录之间的联系，但是却没有切断个体与敏感信息之间的联系，因此 *l*-多样性模型、*t*-相近模型等相继提出。

差分隐私模型。基于 K-匿名模型及其改进策略的匿名保护模型大都沿用了属性的泛化操作，对发布数据的可用性造成较大影响。同时，大数据发布环境下的组合攻击、前景知识攻击等新型攻击方式对 K-匿名模型及其改进方法提出了严峻挑战。Dwork 等人提出的差分隐私模型借鉴了密码学中语义安全的概念，通过在发布数据或查询结果中添加随机噪声来达到隐私保护的效果。差分隐私模型允许攻击者拥有无穷的计算能力和任何有用的背景知识，而且不需要关心攻击者的具体攻击策略。在最坏的情况下，即使攻击者获得了除

某一条记录之外的所有敏感数据，差分隐私模型仍然可以保证攻击者无法从查询输出结果判断该条记录是否在数据集内。由于具备严格的数学特性，差分隐私被认为是一种非常可靠的保护机制，得到了大量研究学者的关注。基于差分隐私模型的数据发布主要针对敏感数据的统计信息进行保护。

***m*-不变性模型。**传统的静态数据集隐私保护方法无法直接应用于动态数据集重发布过程中，因此，需要研究适用性较强且能够保护动态数据集隐私安全的数据匿名方法。*k*-匿名、*l*-多样性等模型都是面向静态数据集的隐私保护而提出的，无法保证动态数据集的隐私安全。动态数据集的隐私保护问题所面临的挑战是：隐私保护模型不仅要保护数据集的当前快照和以往发布的快照，而且在攻击者将所有发布数据集联合后也能保护数据集的隐私安全。针对动态数据集的重发布的隐私保护问题，*m*-不变性模型被提了出来。该模型要求数据拥有者每个周期发布的匿名数据表中，每个等价类都包含至少 *m* 条记录，且他们的敏感值各不相同，且每条记录 t 在其发布周期$[t_1, t_2]$（$t_1<=t_2$）内的归属等价类具有相同的敏感属性值集合。

虽然 *m*-不变性模型能够维护数据重发布下的隐私安全，但该模型仅关注了数据集对记录的插入和删除两种操作，但在动态更新记录属性值时，*m*-不变性模型便无法较好地保持数据集的隐私安全；此外，*m*-不变性匿名模型还要考虑 *m* 值选取的合理性问题，*m* 值选取不当便会导致向数据集中添加假数据降低数据的可用性。

4.3 数据脱敏与溯源

4.3.1 数据脱敏

数据脱敏是指对某些敏感信息通过脱敏规则进行数据的变形，实现敏感隐私数据的可靠保护。在涉及客户安全数据或者一些商业性敏感数据的情况下，在不违反系统规则条件下，对真实数据进行改造并提供测试使用，如身份证号、手机号、卡号、客户号等个人信息都需要进行数据脱敏。

1989 年，Adam 等就提出数据脱敏（Data masking）的概念，脱敏的方法有替换、遮蔽、加密等，比如，将手机号部分数字通过用*号替换实现脱敏等。上节讲述的一些匿名化方法也可以用来脱敏。

名称	描述	示例
掩码	利用“*”等符号遮掩部分信息，并且保证数据长度不变，容易识别出原来的信息格式，常用于身份号、手机号等	12300001234 →123****1234
替换	一般会有一个字典表，通过查表进行替换	张三→X 李同→Y

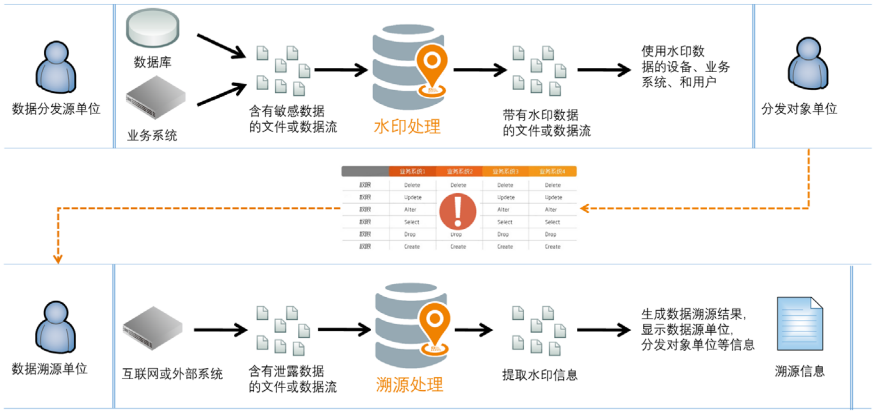
混合掩码	将相关的列作为一个组进行屏蔽，以保证这些相关列中被屏蔽的数据保持同样的关系，例如，城市、省、邮编在屏蔽后保持一致	
截断	舍弃某些必要信息保证数据的模糊性	13800001234→13800
加密	利用加密算法对数据进行变化	13800001→IQ5XRW==

数据脱敏按模式可以分成静态数据脱敏和动态数据脱敏。其主要分别取决于，是否对敏感数据信息采取实时的脱敏操作：

- 静态数据脱敏是数据存储时脱敏，存储的是脱敏数据。一般用在非生产环境，如开发、测试、外包和数据分析等环境。
- 动态数据脱敏在数据使用时脱敏，存储的是明文数据。一般用在生产环境，动态脱敏可以实现不同用户拥有不同的脱敏策略。

4.3.2 数据溯源

数据溯源是数据发布后流转过程中发生泄密后的回溯泄密节点的操作。数据溯源通常通过向数据中加入水印，在数据泄密后，通过提取数据中的水印来完成泄密节点的溯源。很显然，实现数据溯源的关键就是水印不能被攻击者检查出来或者破坏掉，也就是水印的鲁棒性要好。如下图所示。



1. 基于标注技术的溯源方法

对于文件而言，有很多冗余空间，可以隐秘的写入一些流转过程产生的标注信息。

具体的来说，可以按时间序，在每次文件流转或修改的时候增加标注信息，标注信息包含当前文件的哈希值等鉴别信息、时间、源属主等。

根据应用场景选择标注信息嵌入机制：

✓ 对于非文本型具有特定格式的文件，可采用信息隐藏技术嵌入到文件中，随文件流转。文件无论修改与否均适用。

✓ 将标注信息存储到第三方存储系统中，只适用于文件修改的。

2. 基于数据库水印的溯源方法

对于数据库存储的数据而言，很难找到冗余空间，添加水印的难度很大，而且鲁棒性不够高，容易被擦除。因为，数据库存储对数据提出了严苛的限制。

即使如此，仍有一些数据库水印算法提出，包括伪行、伪列等，如下表所示。

应用场景	算法名称	原理说明	重点突破
对单条数据的查询	**伪行算法**	增加伪行实现水印嵌入	原始数据规模、数据属性关系、数据仿真度
对数据的统计查询	**伪列算法**	增加伪列实现水印嵌入	数据重复性、数据仿真度
文本型数据查询	**文本属性算法**	添加不可见字符实现水印嵌入	规则确定、防擦除
数值数据查询	**数值属性可逆算法**	替换最低有效位实现水印嵌入	精度失真比例、执行性能

4.4 保留格式加密及应用

脱敏后的数据通常会被用来做数据分析等任务，为了满足数据分析后结果脱敏的需求，需要有可逆脱敏的技术支持。保留格式加密（format-preserving encryption，简称FPE）能确保密文与明文具有相同的格式，可以提供可逆脱敏的能力。

目前，NIST 已经接受 FPE 算法，并颁布了两种标准算法：FF1 算法和 FF3 算法。

4.4.1 基本定义

基于 FPE 已有的研究成果，从两个角度对 FPE 进行了定义：基本 FPE 和一般化 FPE。基本 FPE 描述了 FPE 要解决的问题，即确保密文属于明文所在的消息空间；一般化 FPE 则强调 FPE 问题的复杂性在于待加密消息空间的复杂性。

定义 1(基本 FPE)。FPE 可以简单描述为一个密码 $E:K\times X\to X$，其中，K 为密钥空间，X 为消息空间。

基本 FPE 强调明文和密文处于相同的消息空间，因此具有相同的格式。以 n 位信用卡号的保留格式加密为例，密文要求和明文一样都是由十进制数字组成的长度为 n 的字符串，即两者均为消息空间 $\{0,\ldots,9\}^n$ 内的元素。根据基本 FPE 的定义，分组密码也是一种特殊的 FPE，它是由分组长度 n 决定的 $\{0,1\}^n$ 字符串集合上的置换。然而，FPE 要处理的消息空间远比分组密码复杂的多，比如格式为“YYYY-MM-DD”的日期型消息空间，不仅有长度为 10 的字符串长度限制，还需要满足特定位置是字符‘-’、年、月、日在合理范围内等格式要求。

为了更准确地描述 FPE 问题，定义集合 Ω 为格式空间，任意一个格式 $\omega\in\Omega$，可确定消息空间的一个与格式 ω 相关的子空间 X_ω。FPE 与集合 $\{X_\omega\}_{\omega\in\Omega}$ 有关，称 X_ω 为由格式 ω 确定的消息空间的一个分片，每个分片都是一个有限集。当给定密钥 k，格式 ω 和调整因子 t 后，FPE 就是一个定义在 X_ω 上的置换 $E_k^{\omega,t}$.

定义 2(一般化 FPE). FPE 可以描述为一个密码 $E:K\times\Omega\times T\times X\to X\cup\{\perp\}$，其中，$K$ 为密钥空间，Ω为格式空间，T 为调整空间，X 为消息空间。所有空间都非空，且$\perp\notin X$。

为了有效地研究分析加密模型，可通过算法三元组$E_{FPE}=(Gen,Enc,Dec)$来描述一般化 FPE，其中：

- 算法 Gen：初始化系统参数 *params*。
- 算法 Enc：输入为调整因子 t 和明文 x，返回在分片 X_ω内的密文 y 或者$\perp$。该算法执行 $E_K^{\Omega,T}(X)=E(K,\Omega,T,X)$过程，$E_K^{\Omega,T}(\cdot)$ 是 X_Ω上的一个置换。如果 $x\in X_\omega$,则返回 $y=E_k^{\omega,t}(x)$；否则，返回$\perp$。
- 算法 Dec：输入为调整因子 t 和密文 y，返回相同分片 X_ω内的明文 x 或者$\perp$。该算法是算法 Enc 的逆运算，定义如下:如果 $y\in X_\omega$，则返回 $x=D_k^{\omega,t}(y)$；否则，返回$\perp$。

安全性。保留格式加密是一种特殊的对称密码，基础模块是分组密码和伪随机函数。由于安全性通常可以归约到基础模块的安全性上，因此，保留格式加密的一个重要的安全目标是伪随机性。

2002 年，Black 和 Rogaway 首次描述了保留格式加密的安全性，认为标准的安全目标就是伪随机置换(pseudorandom permutation,简称 PRP)安全。

根据基本 FPE 的定义，对任意 $k\in K$，$E(k,\cdot)=E_k(\cdot)$是消息空间 X 上由对称密钥 k 决定的一种置换。设 $Perm_k(\cdot)$表示消息空间 X 上所有置换的集合，$P\xleftarrow{\$}Perm_k(\cdot)$ 表示从 $Perm_k(\cdot)$中随机抽取一个置换 P。设 A^f 是一个可以查询预言机 f 的攻击者，f 要么是加密预言机 $E_k(\cdot)$，要么是一个随机置换预言机 $P(\cdot)$。假定攻击者从不执行消息空间之外的查询，而且不重复相同的查询，这样的攻击者 A 可以认为是保留格式加密方案E_{FPE} 的 PRP 攻击者，并且定义其在攻击中可获得的优势为

$$\mathrm{Adv}^{\mathrm{PRP}}_{\mathsf{E}_{\mathrm{FPE}}}(A)\overset{def}{=}|Pr[k\xleftarrow{\$}K:A^{E_k(\cdot)}=1]-Pr[P\xleftarrow{\$}Perm_k(\cdot):A^{P(\cdot)}=1]|.$$

该式度量了攻击者 A 区分保留格式加密和随机置换的概率优势。

定义(PRP 安全). 令$\mathrm{Adv}^{\mathrm{PRP}}_{\mathscr{E}_{\mathrm{FPE}}}(t,q)\triangleq\max_A\mathrm{Adv}^{\mathrm{PRP}}_{\mathscr{E}_{\mathrm{FPE}}}(A)$，其中，$t$ 为攻击者执行破解算法的时间，q 为攻击者查询的次数。如果 $\mathrm{Adv}^{\mathrm{PRP}}_{\mathsf{E}_{\mathrm{FPE}}}(\cdot,\cdot)$ 是一个可忽略的量，称该保留格式加密方案 E_{FPE} 为伪随机置换，也就是达到了 PRP 安全。

4.4.2 基本方法

2002 年，Black 和 Rogaway 提出了 3 种 FPE 构建方法：Prefix，Cycle-walking 和 Generalized-Feistel。这 3 种方法不仅在一定程度上解决了整数集上的 FPE 问题，而且成为构造 FPE 模型的基本方法。

（一）Prefix 方法

Prefix 方法很简单，首先在内存中建立一个随机的置换表，然后基于该置换表对数据进行加解密。这意味着加解密速度非常快，但是在较大消息空间上建立置换表将会耗费更多的

时间，因此只适用于小的有限集 $X=\{0，1，\ldots，n-1\}$，$n<10^6$。

将 Prefix 方法记为密码Π_{FPE}，其密钥空间为 K，消息空间为整数集 $X=\{0，1，\ldots，n-1\}$，$n<10^6$。为了建立置换表，采用基础分组密码 E，其对称密钥为 $k\in K$，计算如下元组:$I=(E_k(0)，E_k(1)，\ldots，E_k(n-1))$。由于 I 中每个分量 $E_k(i)$，$i\in X$ 是长度为分组长度的不同二进制符号串，可以按照数值关系对其进行排序，由此得到 $E_k(i)$对应的排序值 r_i。进一步对 I 进行操作，将分量 $E_k(i)$换成其对应的排序值并得到元组 $J=(r_0，r_1，\ldots，r_n)$，这样就建立了消息空间 X 上的一个置换表:给定任意明文 $x\in X$，返回元组 J 中相同序号的分量 r_x，就得到了对应的密文。

举例:消息空间为 $X=\{0，1，2，3，4\}$，为了建立置换表，选择基础分组密码 E 为 AES，计算 $E_k(0)=166;E_k(1)=6;\ E_k(2)=130;E_k(3)=201;E_k(4)=78$(这里，AES 的加密结果本为分组长度的二进制字符串，为方便起见，将其用十进制数表示)，得到元组 $I=(166，6，130，201，78)$，将每个分量替换为其对应的排序值得到元组 $J=(3，0，2，4，1)$。从而，假设要加密明文 0，返回元组 J 中序号为 0 的分量为其密文，即 3。

实际应用中，通常会有对密钥进行更新的要求，然而对于 Prefix 方法来说，密钥的更新意味着重新建立置换表，需要消耗较高的代价(重新加密整个消息空间并进行排序和替换)。因此，有必要在特定环境里对密码应用调整因子，可以使其不需要密钥更新而更改加密函数。文献[7]已经提出一些构建可调整密码的方法:为分组密码 E 引入调整因子 t 来加密明文 x，可执行操作 $y=E_k((E_k(x)+t)\ \mathrm{MOD}\ n)$。可见，引入调整因子后的加解密过程执行了两次加密，但是对 Prefix 方法而言，加解密是在内存中查表的操作，因此不会影响效率。

Prefix 方法不会降低基础分组密码的安全性，即当 E 是 PRP 安全的时候，Prefix 也能达到相同的安全性。

（二）Cycle-Walking 方法

Cycle-Walking 方法为确保密文为消息空间内的元素提供了一种通用的解决思路，其加密的原理是利用基础分组密码(AES 或 3DES 等)对中间输出值不断进行处理，直至其在可接受的输出范围内。

设 $\mathrm{Cycle}_k(x)$表示使用 Cycle-Walking 方法对明文 x 加密，密钥为 k，加密过程为:要加密明文 $x\in\{0，\ldots，n-1\}$，选用分组密码 E(如 AES)，设 $y=E_k(x)$，如果 $y\in\{0，\ldots，n-1\}$则返回 y;否则，循环执行 $y=E_k(y)$，直到有$\{0，\ldots，n-1\}$范围内的 y 产生为止。Cycle-walking 可以将不在期望范围内的密文加密到此范围内，但是需要多次调用 E。

举例:设 $X=\{0，1，\ldots，10^6-1\}$，首先确定所采用的基础分组密码，由于 $10^6<2^{64}$，选用 64 位的 DES 来处理，可以保证其输出范围始终包含 X。假设现在要加密明文 $x=314159$，计算得到 $c_1=E_k(314159)=1040401$(这里，E 采用 DES 算法，为方便起见，将其 E 的加密输出用十进制数表示)，因为 $c_1\notin X$，迭代计算 $c_2=E_k(1040401)=1729$。因为 $c_2\in X$，所以 $\mathrm{Cycle}_k(314159)=1729$。

Cycle-walking 不会降低传统分组密码的安全性[4]，但是在效率方面，一次加密可能需要多次调用基础分组密码，当明文的二进制位数远小于分组长度时，会因为迭代次数增加而导致性能降低。因此，Cycle-walking 方法适合大小接近分组长度的整数集。比如，如果采用 DES 算法，适合的范围是 54~64 二进制位的整数集。

（三）Generalized-Feistel 方法

Generalized-Feistel 方法要比 Prefix 和 Cycle-walking 复杂，可以适用于更广泛的加密范围。

由于 Cycle-walking 方法对于接近分组密码大小的整数集完成保留格式加密时具有较高的性能，因此 Generalized-Feistel 方法的核心思想是基于 Feistel 网络来构建符合整数集大小的分组密码，并结合 Cycle- walking 方法使最终密文输出在合理范围内。Generalized-Feistel 方法由两部分组成:① 由 Feistel 网络构造的分组密码 E，假设消息空间元素个数为 n，则 E 的分组长度要略大于 $\log_2 n$;② Cycle-walking 方法，确保数据被加密到合理范围内。

Feistel 网络是目前主流的分组密码设计模式之一，基于 Feistel 网络，可以通过自定义的分组大小、密钥长度、轮次数、子密钥生成、轮函数等来构造一个分组密码。如下图所示。输入长度为 $2m$ 的比特串，首先将其等分为长度相等的两部分 L 和 R，这里它们的长度 $|L|=|R|=m$，然后在对 R 执行伪随机函数 $\mathrm{F}_k(R)$后与 L 异或得到 $L'=\mathrm{F}_k(R)\oplus L$，最后将 L' 与 R 交换位置后所连接成的新的比特串 $R\,||\,L'$ 作为下一轮迭代的输入。

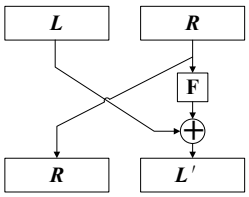


Fig. 1 Classical Feistel network

图 1 传统 Feistel 网络

为了构建 Generalized-Feistel 密码$\Gamma\Phi_{n,f,r}(x)$，使用一个基本的伪随机函数 f 和 r 轮 Feistel 运算来加密整数集 $X=\{0，1，\ldots，n-1\}$内的值 x，首先定义一个基于 Feistel 网络的对称密码 $E_{n,f,r}(x)$:

1) 寻找最小的 w 使得 $2^{2w}\geqslant n$，$2w$ 就是需要构建的分组密码的分组长度;
2) 定义 $f'(x)=trunc(f(x)，w)$表示截取 $f(x)$的低 w 位数据;
3) 定义轮运算 $Round(R，L)=L$ XOR $f'(R)$;
4) 计算 $E_{n,f,r}(\cdot)$:① 寻找 R，L，使得 $x=L\times 2^w+R$;② 重复 r 次:$\{T=Round(R，L)，L=R，R=T\}$;③ 输出 $L\times 2^w+R$。

$\Gamma\Phi_{n,f,r}(x)$将使用所构建的分组密码 $E_{n,f,r}(x)$进行如下计算:① $y=E_{n,f,r}(x)$;② while $(y\geqslant n)\{y=E_{n,f,r}(y)\}$。很显然，$\Gamma\Phi_{n,f,r}(x)$使用了 Cycle-walking 方法，确保数据加密到合理的范围内。由于其构建的分组密码的分组长度与待加密消息空间大小的二进制位数接近，因此具备较好的性能。

4.4.3 基本模型

Generalized-Feistel 模型及其思想成为后来 FPE 模型设计的主要参照，其中包括 FFSEM 模型、RtE 方法及 FFX 模型。这些模型都将基础分组密码作为伪随机函数，基于 Feistel 网络来构造满足要求的对称加密模型，将安全性转移到基础分组密码相关的安全性上，具备可

证明的安全性和实用性。

（一）FFSEM 模型

FFSEM 是基于 Generalized-Feistel 方法的整数集上的典型 FPE 方案，它由两个基本部分组成：①平衡 Feistel 网络，用来产生指定分组长度的分组密码；②Cycle-walking，用 2*m* 位的分组密码对大小为 $n(n<2^{2m})$ 的集合进行加解密的普遍方法。

算法 Gen：FFESM 初始化阶段主要定义：①FFSEM-PRF，即平衡 Feistel 网络中所使用的伪随机函数，FFSEM 使用截断基础分组密码 AES 输出的方式构造了 FFSEM-PRF；②基础分组密码的密钥 *k*、消息空间的大小 *n* 和轮次数 *r* 等。详细的参数信息参阅文献**错误!未找到引用源。**。

算法 Enc：输入为明文 *x*，输出为满足格式要求的密文 *y*。

首先，将明文 *x* 编码为 $l=\lfloor \log_2 n \rfloor+1$ 位的二进制数(这里 $\lfloor x \rfloor$ 表示不超过 *x* 的最大整数)，不足的二进制位以 0 填充；然后，执行 Cycle-walking 过程，每次 Cycle-walking 都将执行 *r* 轮 Feistel 轮运算，直到产生合适的密文；最后，对密文进行二进制解码得到对应数值的整数，其加密过程如图 1 所示。

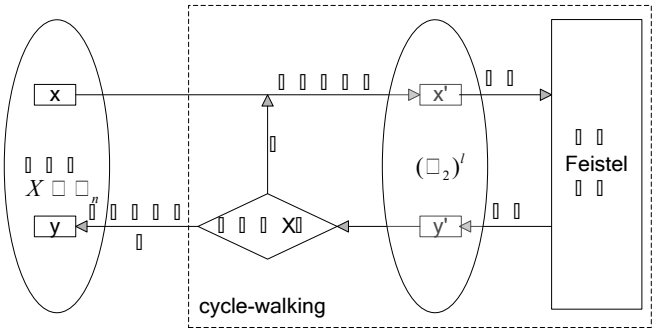


Fig.1　The Feistel finite set encryption mode

图 1　FFSEM 模型

目前，该加密模型已经被 Voltage 公司广泛应用，而且被 NIST 所采纳。然而，FFSEM 仅解决了整数集上的 FPE 问题，并不能成为一种普遍适用的 FPE 模型；而且 Cycle-walking 过程需要多次调用基础分组密码，存在不确定的性能问题。

（二）FFX 模型

Bellare 在消息空间、Feistel 网络和运算等方面对 FFSEM 进行了扩展，提出了扩展机制 FFX 模型。该模型使用了非平衡 Feistel 网络，通过自定义运算可以解决 *n* 位字符串所构成的消息空间 $chars^n$ 的 FPE 问题。

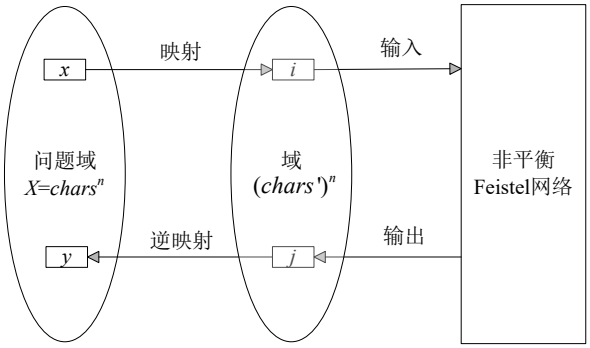


图 3　FFX 模型

算法 Gen： FFX 模型初始化阶段主要定义：

1) 字母表 $chars=\{char_0,char_1,char_2,char_3,\cdots,char_{radix-1}\}$ 及其基数 $radix = |chars|$；

2) 所使用的非平衡 Feistel 网络类型；

3) 消息空间中元素的长度 n；

4) 每轮中所用到的伪随机函数 f 、所采用的运算类型、轮次数 r 和调整因子 t 等。

算法 Enc：输入为基础分组密码的密钥 k、调整因子 t 和字符串 x，输出为满足格式要求的字符串 y。字符串 x 和 y 都是由字母表 $chars=\{char_0,char_1,char_2,char_3,\cdots,char_{radix-1}\}$ 中的字符组成的长度为 n 的字符串。

首先，将字符串 x 中的字符编码替换为数字：建立字母表 $chars=\{char_0,char_1,char_2,char_3,\cdots,char_{radix-1}\}$ 与 $chars'=\{0,1,2,3,\cdots,radix-1\}$ 的一一映射，将每个字符 $char_i$ 编码为对应的第 i 个数字。需要注意的是 $chars'$ 中每个数字前面的 0 和其它字符一样计入长度，例如，$chars=\{a,b,c,\cdots,z\}$，$x=acz$，将 x 编码后得到 $x=010326$。

然后，执行 r 轮指定非平衡 Feistel 网络的运算：首先，将输入(字符串 x 的编码)分割为左右两部分 L 和 R，$|L|\neq|R|$；然后，执行伪随机函数 f，对 L 和 $f_k(R)$ 执行选择的类型的运算得到 L'：①$c_i = (a_i + b_i)$ MOD $radix$，当 $radix = 2$ 时，该运算就是异或运算；②$\sum c_i radix^{n-i}=(\sum a_i radix^{n-i}+\sum b_i radix^{n-i})$ MOD $radix^n$；最后，连接 L' 与 R 得到输出 $L'\|R$,并将其作为下一轮非平衡 Feistel 网络运算的输入。

可见，FFX 模型通过将非数字字母表与数字集合建立双射，将每个字符映射为对应的数字参与加密运算，实现对消息空间 $chars^n$ 的保留格式加密。与 FFSEM 相比，FFX 适用的范围更广，而且在处理信用卡号、社会保险号等 FPE 问题时避免了 Cycle-walking，具有较高的效率。

4.4.4 数据库水印应用

保留格式加密可以用于水印的生成。

DepartID	DepartName	Phone	OutpatientAddress	InpatientAddress
001	耳鼻咽喉科	022-87529574	门诊楼 A 区 3 层	一号住院楼 7 层
002	产科	022-87529564	门诊楼 C 区 2 层	一号住院楼 2 层
003	皮肤科	022-87529565	门诊楼 B 区 4 层	一号住院楼 11 层
004	普通外科	022-87529598	门诊楼 A 区 2 层	一号住院楼 4 层
005	中医科	022-87529603	门诊楼 B 区 3 层	二号住院楼 3 层
006	心血管科	022-87529597	门诊楼 B 区 2 层	一号住院楼 14 层
007	内分泌科	022-87529562	门诊楼 B 区 4 层	二号住院楼 5 层

如上表所示，DepartID 和 DepartName 是松耦合，不同的 DepartID 和 DepartName 的对应关系，可以作为一种隐性的水印。在不同的流转环节，产生水印的时候，我们可以选择一个不同的密钥，用于对 DepartID 的整数进行加密，进而产生不同的序。

添加水印的过程为，基于密钥置换主键 DepartID 列，其余列保持不变，进而按照主键 DepartID 新的序列重新排序，即可产生带有水印的数据。

一旦泄密，因为密码机制的健壮性，只需要少数几条，就可以进行鉴别和溯源。