- Chap5
  - 隐私保护技术初探
    - 隐私保护技术概述
    - 面向数据发布的隐私保护
      - 1. **基于限制发布的隐私保护**：这种技术在将数据公布给数据挖掘者之前，对数据进行扰动、加密、匿名等处理，将数据中的隐私藏起来。研究主要集中于数据匿名化，例如有选择的发布原始数据、不发布或者发布精度较低的敏感数据。
      - 2. **基于数据失真的隐私保护**：该技术通过对原始数据进行扰动，目的是隐藏真实数据，只呈现出数据的统计学特征。失真后的数据满足两个条件：保持原本的某些特性不变，且攻击者不能根据失真数据重构出真实的原始数据。此技术主要包括随机化、阻塞、变形、交换等。
      - 3. **基于数据加密的隐私保护**：这种技术对原始数据进行加密，通过密码机制实现其他参与方对原始数据的不可见性以及数据的无损失性。由于加密技术可解决安全通信的问题，因此多应用于分布式应用。
    - 面向数据挖掘的隐私保护
      - 1. **关联规则的数据挖掘**：关联规则挖掘是数据挖掘领域研究的重点之一，是从大量数据中挖掘数据项之间隐藏的关系，发现数据集中项集之间的关联和规则的过程。例如购物篮分析，寻找商品之间隐藏的关联规则。
      - 2. **隐私保护的关联规则挖掘**：包含两类方法，一种是修改支持敏感规则的数据，使得规则的置信度和支持度小于一定的阈值而实现规则的隐藏；另一种是不修改数据，而是隐藏生成敏感规则的频繁项集，尽可能降低敏感规则的置信度或者支持度
    - 匿名化隐私保护模型
      - 传统的匿名化方法
      - 链接攻击
      - k-anonymity（PPT第34页）
      - 同质性攻击和背景知识攻击（PPT第35-36页）
      - l-diversity（PPT第37-39页）
      - t-closeness（P41）
  - 数据匿名化方法（页码：42）
    - **泛化（Generalization）**：这是通过使用更抽象、概括的值或区间来替代精确值的方法。
    - **抑制（Suppression）**：这种方法是将数据表中的数据直接删除或隐藏。
    - **聚类（Clustering）**：这种方法根据给定的规则将数据集分成各种类簇，尽可能保证类簇内对象相似，不同类簇的对象相异。
    - **微聚集（Micro-aggregation）**：这种方法是将相似的数据划分在同一个类中，每个类至少有k条记录，用类质心代替类中所有记录的准标识符属性值。
    - **分解（Decomposition）**：这种方法根据敏感属性值对数据表分组，尽量使得同一组的敏感属性值不同，将分组后的数据表拆分为分别包含准标识符属性信息和包含敏感属性信息的两张表。
    - **置换（Permutation）**：这种方法是对数据表分组，把每组内的敏感属性值随机交换，打乱顺序，再拆分数据表，对外发布。
    - 泛化（页码：43-44）
      - 泛化树（页码：44）
        - **数值型数据泛化树**：对于数值型属性，可以使用泛化树进行泛化。底层的具体取值被一个覆盖精确数值的区间代替，而上层的取值则逐渐变得更加抽象和概括。这样，可以保持数据的真实性和一致性，同时减少敏感信息的泄露。
        - **分类型数据泛化树**：对于分类型属性，可以使用泛化树进行泛化。底层的具体取值被一个更一般的值代替，而上层的取值则逐渐变得更加抽象和概括。这样，可以保持数据的真实性和一致性，同时降低敏感信息的泄露风险。
      - 域泛化（全局泛化）与值泛化（局部泛化）（页码：45-46）
        - **域泛化**：域泛化将一个给定的属性域从底层开始同时向上泛化，直到满足隐私保护要求，然后停止泛化。在域泛化中，同一个属性的全部值必须在同一层上进行泛化。域泛化分为全域泛化和子树泛化两种方式。
          - **全域泛化**：在泛化树中，同一个父节点下的所有子节点要么全部泛化，要么全部不泛化。全域泛化可以确保一致性，但可能导致较大的信息损失。
          - **子树泛化**：在泛化树中，对部分子节点进行泛化，而其他兄弟节点不要求泛化。父节点代替泛化的子节点进行数据发布。子树泛化相比全域泛化可以减少信息损失，但可能引入一些不一致性。
        - **值泛化**：值泛化将原始属性域中的每个值直接泛化成一般域中的唯一值。在值泛化中，可以对多个属性的值同时进行泛化，只需要对不符合限制要求的等价类进行泛化，要求一个等价类中所有记录都泛化成相同的值。值泛化可以保持数据的一致性，但可能引入一些不一致性。
    - 抑制（P48）
      - **记录抑制**：对数据表中的某条记录进行抑制处理，即完全删除该记录，使其无法识别。
      - **值抑制**：对数据表中某个属性的部分值进行抑制处理，即用符号或其他模糊化方法替换一部分属性值，以减少敏感信息的泄露。
      - **单元抑制**：对数据表中某个属性的所有值进行抑制处理，即完全删除该属性，使其无法被使用或分析。
  - 差分隐私(P49)
    - **严格定义的隐私保护**：差分隐私对隐私保护进行了严格的定义，并提供了量化评估方法，使不同参数处理下的数据集所提供的隐私保护水平具有可比较性。
    - **抵抗背景知识攻击**：差分隐私假设攻击者掌握最大的知识背景，即能够获得除目标记录外所有其他记录的信息。因此，差分隐私的设计考虑了更强的攻击模型，能够更好地保护隐私。
    - **严格的隐私保护证明**：差分隐私提供了严格和科学的方法来证明其隐私保护水平。当模型参数改变时，可以对其隐私保护水平进行定量分析，以便在隐私与数据可用性之间进行权衡。
    - 差分隐私基础(P53)
    - 数值型差分隐私和非数值型差分隐私匿名化与差分隐私
    - 差分攻击
    - 差分隐私思想
    - 差分隐私定义
      - 当ε = 0时，攻击者无法区分相邻数据集，保护程度最高，但数据可用性最差。
      - 当ε增大时，保护程度逐渐降低，ε过大会造成隐私泄露。
      - 通常，ε取较小的值，如0.01、0.1或ln 2、ln 3等。ε的取值应结合具体需求设定，以平衡输出结果的安全性和可用性。
    - 差分隐私的实现(P56)
      - 拉普拉斯机制
      - 高斯机制
      - 全局敏感度和局部敏感度
        - 全局敏感度是指在任意一对邻近数据集D和D'上，查询函数f的输出结果之间的最大变化范围。它与数据集无关，由查询函数本身决定。
        - 局部敏感度是指对于给定的数据集D和它的任意邻近数据集D'，查询函数f在D上的局部敏感度。它由查询函数和给定数据集中的数据共同决定。
      - 数值型差分隐私：拉普拉斯和高斯机制
      - 非数值型差分隐私：指数机制
  - 同态加密(P69)
    - 同态加密基础
    - 同态加密的应用场景：安全的数据外包
    - 组成
      - **KeyGen算法**：通过计算安全参数生成一对公私钥。
      - **Encrypt算法**：使用公钥将明文加密为密文。
      - **Evaluate算法**：在密文上进行运算，例如加法或乘法。
      - **Decrypt算法**：使用私钥将密文解密为明文。
    - 同态加密的发展
      - 1. **仅支持加法同态的加密体制**：最早的同态加密体制只支持加法同态或乘法同态，但不能同时满足两者。
      - 2. **半同态加密**（Partially Homomorphic Encryption，PHE）：半同态加密体制同时满足加法同态和乘法同态的性质，但只能进行有限次的加和乘运算。
      - 3. **浅同态加密**（Somewhat Homomorphic Encryption，SWHE）：浅同态加密体制也同时满足加法同态和乘法同态的性质，但可以进行任意多次加和乘运算。
      - 4. **全同态加密**（Fully Homomorphic Encryption，FHE）：全同态加密体制是最理想的同态加密形式，它可以在不解密的条件下对加密数据进行任何可以在明文上进行的运算，实现了深度和无限的数据分析，对加密信息进行深入分析而不影响其保密性。
    - 半同态加密
      - ⊗
        - 乘法同态加密：当 表示乘法时，称为乘法同态加密。典型的乘法同态加密算法有：
          - RSA公钥加密算法（1977年提出）
          - ElGamal公钥加密算法（1985年提出）
        - 加法同态加密：当 表示加法时，称为加法同态加密。典型的加法同态加密算法有：
          - Paillier公钥加密算法（1999年提出），是最常用且最具实用性的加法同态加密算法。
      - ElGamal乘法同态加密
      - Paillier加法同态加密
      - 全同态加密
    - 同态加密的应用
      - **医疗机构的数据分析**：在医疗机构中，数据处理能力较弱，可以借助云服务商提供的计算服务。使用同态加密，医疗数据可以在加密的状态下存储和计算，而不泄露隐私信息。云服务商可以进行数据搜索、分析和处理等功能，同时保护数据隐私。
      - **电子投票**：同态加密可以用于设计安全的电子选举系统。统计方可以在不知道投票者投票内容的情况下对投票结果进行统计，既保证
    - 同态加密的优势与挑战
      - **降低计算代价**：同态加密可以对多个密文进行计算后再解密，降低了计算代价。
      - **降低通信代价**：同态加密可实现无密钥方对密文的计算，无需经过密钥方，降低了通信代价。
      - **保证数据安全性**：同态加密可以实现让解密方只能获知最终的结果，而无法获得每个密文的消息，从而保证了信息的安全性。
      - **计算效率**：当前的同态加密方案的计算复杂度较高，如何设计高效的全同态加密方案仍然是一个问题。
      - **安全性**：同态加密方案大多基于未论证的困难问题，寻找可论证的困难问题仍然是一个挑战。
      - **噪音消除**：同态加密需要额外的消除噪音算法，如何设计具有自然同态性的全同态加密方案仍然是一个问题。
  - 安全多方计算(P93)
    - 安全多方计算基础
    - 安全多方计算的提出
    - 安全多方计算的形式化描述
      - **输入独立性**：各方能独立输入数据，计算时不泄露本地数据。
      - **计算正确性**：计算结束后各方能够得到正确的计算结果。
      - **去中心化性**：各参与方地位平等，提供了去中心化的计算模式。
    - 安全多方计算的威胁模型
      - **诚实模型**：参与者按照协议要求行动，不提供虚假数据，不泄露、窃听数据，不终止协议，完全按照协议执行。
      - **半诚实模型**：在诚实模型基础上保留所有收集到的信息，推断其他参与者的秘密信息。
      - **恶意模型**：无视协议要求，可能提供虚假数据、泄露数据、窃听甚至终止协议。
    - 安全多方计算的计算模型
      - **基于"可信第三方"的计算模型**：参与方得到计算结果，可信第三方得到参与方的输入信息和计算结果，信息的保密性由可信第三方来保证。然而，在实际情况下很难找到完全可信的第三方，所以这种模型很少使用。
      - **交互计算模型**：参与方按照协议步骤执行计算，按协议的要求将中间结果发送给其他参与方，同时接收其他参与方计算的中间结果，信息的保密性由协议的安全性来保证。这是安全多方计算中最常用的模型，提供了一种去中心化的计算方式。
      - **外包计算模型**：各个参与方希望使用云计算提供的计算资源，但不想直接将信息委托给云计算服务提供商，也不想让其得知计算结果。参与方将信息处理后存储在外包服务器上，由外包处理器对所有参与方的秘密信息进行计算，并将结果发送给各参与方。信息的保密性由协议的安全性来保证。
    - 基本密码协议
    - 安全多方计算的应用
      - **门限签名**：将私钥拆分为多个秘密分片，只有在达到门限值的参与者共同协作时才能生成有效的签名。
      - **电子拍卖**：在不直接公开竞拍者的出价情况下，能够计算出所有参与者输入的最大值或最小值，使得在线拍卖成为现实。
      - **联合数据查询**：多个数据库可以共同进行数据查询，使用安全多方计算保护各数据库的私有信息或知识版权。
      - 其他领域：安全多方计算涉