Chap5

隐私保护技术初探

隐私保护技术概述

在大量的网络空间数据中获取有用信息时，如何在保护数据相关者的隐私的同时获取有用信息变得尤为重要。数据隐私保护技术的研究主要分为两个方面：**面向数据发布**和**面向数据挖掘**。

面向数据发布的隐私保护

面向数据发布的隐私保护包含以下几个部分：

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。两种数据存储模式：垂直划分（每个人参与者存储部分属性）和水平划分（每个参与者存储部分数据）。

基于限制发布	基于数据失真	基于数据加密
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	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	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

面向数据隻布的隐私保护技术	优点	缶缺点
基于限制发布的隐私保护技术	发布的数据真实可靠	数据丢失部分信息
基于数据失真的隐私保护技术	算法效率较高	由于干扰使数据丢失部分信息
基于数据加密的隐私保护技术	数据的安全性和准确性均较高	计算开销很大

面向数据挖掘的隐私保护

面向数据挖掘的隐私保护包括以下几个部分：

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：两类方法都会影响对于非敏感规则的挖掘

变换 (distortion)	隐藏（blocking）
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	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，以此使得需要保护或隐藏的规则不被挖掘出来

3. 隐私保护的分类和聚类挖掘 P25，聚类和分类都会暴露数据集中的隐私敏感信息

匿名化

在数据发布过程中，为了保护用户的敏感数据和个人身份之间的对应关系，需要采取匿名化隐私保护模型。传统的匿名化方法往往无法抵抗链接攻击，而 k-anonymity 是一种有效的隐私保护模型，能够解决链接攻击问题。

传统的匿名化方法

传统的匿名化方法包括删除容易关联到个人的属性（如姓名和家庭住址），或者将姓名替换为假名。然而，这些方法存在一定的局限性，无法提供足够的隐私保护，容易受到链接攻击。

链接攻击

链接攻击是指攻击者通过对发布的数据和其他渠道获取的数据进行链接操作，推断出隐藏在匿名化数据中的隐私信息。例如，攻击者可以将医疗数据集与其它公共数据集的准标识符进行联系，从而推断出匿名化数据中的敏感信息。

k-anonymity（PPT 第 34 页）

k-anonymity 是一种数据匿名化技术，当攻击者尝试链接攻击时，由于任意一条记录的攻击，都会同时关联到等价类中的其他 k-1 条记录，因此攻击者无法确定特定用户。k值越大，隐私保护效果越好，但是相应的数据丢失也越严重。

将 k 个记录放入一个等价类中，要求任意一条记录与其他至少 k-1 条记录相似而不可区分，这样数据中的每一条记录都能找到与之相似的记录，降低了数据的识别度，如果一条记录由于样本太少而无法找到 k-1 条相似的记录，那么这条数据不应当被纳入数据集。

同质性攻击和背景知识攻击（PPT 第 35-36 页）

同质性攻击：在数据匿名化过程中可能存在的攻击模式包括**同质性攻击**和**背景知识攻击**。同质性攻击是指没有对敏感属性进行约束，最终结果可能造成隐私泄露。例如，如果一名选民的年龄和邮政编码符合第一个等价类，而第一个等价类里全有心脏病，那么攻击者可推断该选民可能患有心脏病。

背景知识攻击：攻击者可以通过掌握的足够的相关背景知识以很高的概率确定敏感数据与个体的对应关系，得到隐私信息。例如，如果一名选民符合第二个等价类，第二个等价类里面不是糖尿病就是哮喘，且攻击者发现他不像是患有哮喘，那么攻击者可推断该选民可能患有糖尿病。

l-diversity（PPT 第 37-39 页）

l-diversity 在 k-anonymity 的基础上，要求保证每一个等价类的敏感属性至少有 l 个不同的值，即每个用户的敏感属性值在等价类中可以找到与此值不同的至少 l-1 个属性值，使攻击者最多只能以 1/l 的概率确认某个用户的敏感信息，但无法保证隐私不被泄露。

1. 敏感值的分布显著不同

				某疾病检测结果
				阴性
				阳性
				阴性
				…

1000条记录中有1%的阳性记录和99%的阴性记录，阳性检测结果更为敏感

2-diversity

- 每个等价类中疾病检测结果必须包含阴性和阳性两种结果
- 假设某等价类中有一半阳性记录和一半阴性记录，相比于整体1%阳性的概率，该等价类中的个体都有1/2的概率被认为是阳性，具有严重的隐私风险

2. 没有考虑语义信息

l-diversity并没有考虑语义信息也会为隐私信息带来泄露的风险

				工资
				5000
				5500
				6000
				…

敏感属性为工资

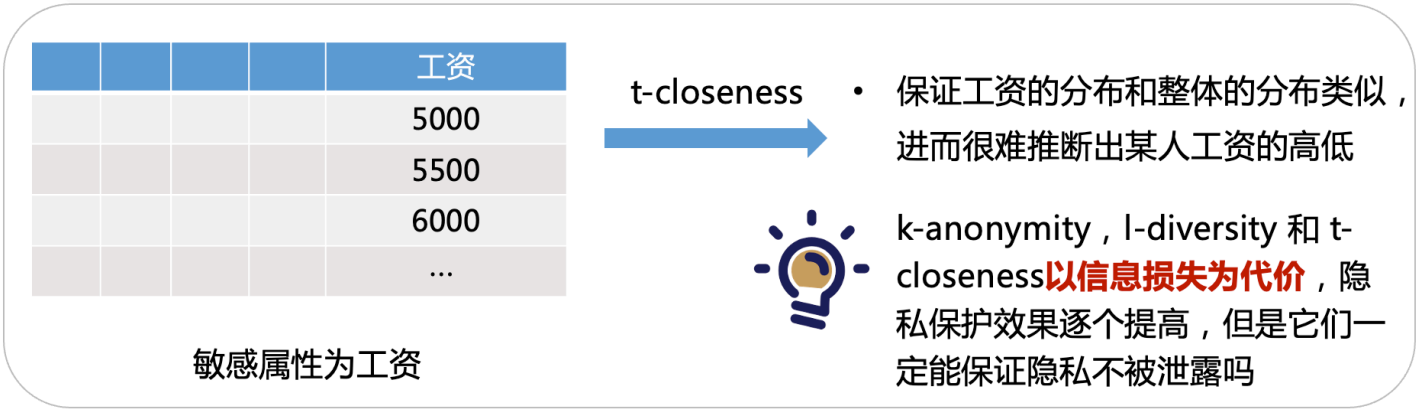
l-diversity

- 若某一个等价类中的工资这一属性的属性值全部在一个固定区间内，那么攻击者并不需要知道详细的属性值就可以通过这个区间就可以判断用户的工资水平

t-closeness（P41）

在 k-anonymity 和 l-diversity 的基础上，t-closeness 考虑了敏感属性的分布问题，**要求所有等价类中的敏感属性的分布尽量接近该敏感属性的全局分布**，差异不能超过阈值 t。

以信息损失为代价，隐私保护效果逐个提高，但是它们不一定能保证隐私不被泄露。



缺点

造成较大的信息损失，信息损失可能会使数据使用者们做出误判；对所有敏感属性提供了相同程度的保护并且没有考虑语义关系，造成了不必要的信息损失；不同的用户对于隐私信息有着程度不同的隐私保护要求；属性与属性之间的重要程度并不相同；没有考虑数据动态更新后重发布的隐私保护问题。

数据匿名化方法（页码：43）

数据匿名化的方法包括**泛化**、**抑制**、**聚类**、**微聚集**、**分解**和**置换**等。目前提出的匿名化主要通过**泛化和抑制**实现，它们能保持发布前后数据的真实性和一致性，但是信息肯定有损失。

匿名化方法	思想
泛化	用更抽象、概括的值或区间代替精确值
抑制	将数据表中的数据直接删除或隐藏
聚类	按照给定的规则将数据集分成各类簇，尽量保证簇内对象相似，不同簇的对象相异
微聚集	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
分解	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	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：44）

泛化是一种常用的数据匿名化方法，其思想是将准标识符的属性用更一般的值或者区间代替。它不会引入错误数据，方法简单，泛化后的数据适用性强，对数据的使用不需要很强的专业知识。但是其预定义泛化树没有统一标准，信息损失大，对不同类型数据的信息损失度量标准不同。

泛化树（页码：45）

泛化树可以看作是一种将底层取值泛化为高层取值的结构，每层的取值构成一个泛化域。底层的取值最具体，顶层的取值最模糊。数值型数据泛化树和分类型数据泛化树是两种常见的泛化树形式。

- **数值型数据泛化树**：值被一个覆盖精确数值的区间代替
- **分类型数据泛化树**：用一个更一般的值代替原值

域泛化（全局泛化）(页码：46）

在数据匿名化过程中，可以采用**域泛化**（全局泛化）和**值泛化**（局部泛化）两种策略。

- **域泛化**：将一个给定的属性域泛化为一般域，将准标识符属性值从底层开始同时向上泛化，一层层泛化直到满足隐私保护要求，然后同时停止泛化

全域泛化	子树泛化	兄弟节点泛化
某个属性的全部值必须在同一层上进行泛化	泛化树中的同一个父亲节点下的所有孩子节点全部泛化或者全部不泛化	在同一个父亲节点下，如果对部分孩子节点进行泛化，其他兄弟节点不要求泛化，父亲节点只能代替泛化了的孩子节点

- 全域泛化：如果“巴黎”和“里昂”泛化到“法国”，则必须同时把“伦敦”和“爱丁堡”泛化到“英国”，保证泛化树中所有路径的泛化粒度相同
- 子树泛化：如果把34泛化到[31,40]，那么37和39也要泛化到[31,40]，而42、48和50可以不泛化到[41,50]
- 兄弟节点泛化：如果“巴黎” 泛化到“法国”，那么“里昂”可以不泛化到“法国”

值泛化（局部泛化）(页码：47）

由于早期的泛化操作都采用整体泛化方案，造成的信息损失大，因此提出了局部泛化方案，将原始属性域中的每个值直接泛化成一般域中的唯一值，将准标识符属性值从底层向上泛化，但是可以泛化到不同的层次。**局部泛化是相同的属性值不全部被泛化，其中一部分值不变。**

单元泛化：如果年龄为34的数据有两条，可以把其中一条泛化到[31,40]，另一个值不变。**区别于子树泛化，子树泛化对于相同的数据必须进行同样的泛化！**

多维泛化：单位泛化是依次将单个属性的属性值转化为相应泛化树中的值，而多维泛化则是在转化过程中同时将多个属性的属性值以笛卡尔乘积的形式转化为相应属性的泛化树中的笛卡尔积形式

单元泛化	多维泛化
对某个属性的一部分值进行泛化，另一部分值保持不变	对多个属性的值同时泛化，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

抑制（P48）

抑制（Suppression），也称为隐藏或隐匿，是一种数据匿名化方法，用于将准标识符属性值从数据集中直接删除或用代表不确定值的符号（如" *"）替代。抑制可以与泛化方法结合使用。

在抑制中，可以采用以下三种方式：

记录抑制	值抑制	单元抑制
对数据表中的某条记录进行抑制处理	对数据表中的某个属性的值全部进行抑制处理	对数据表中某个属性的部分值进行抑制处理

抑制方法可以用于敏感数据的保护，但需要权衡数据可用性和隐私保护之间的平衡。

差分隐私(P49)

差分隐私是一种严格的、可证明的隐私保护模型，相对于传统的匿名化方法，它提供了更严格的隐私保护。差分隐私在数据匿名化方面与传统的匿名化方法有所不同。传统的匿名化方法不能提供足够的数学保障，没有严格定义攻击模型，无法抵抗背景知识攻击，并且难以提供严格和科学的方法来证明其隐私保护水平。相比之下，差分隐私提供了更强的隐私保护，并且能够提供严格的定义和量化评估方法。

- **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。

差分攻击 P51

前三行有两人患病，前四行有三人患病，判断出数据集中的第四行代表的用户患有癌症，如果此时攻击者知道第四行代表的用户是钱六，那么攻击者就可以通过这种方式在没有具体查询特定某人个人信息的前提下获得其隐私数据。为抵抗差分攻击，差分隐私要求保证任意一个个体在数据集中或者不在数据集中时，对最终发布的查询结果几乎没有影响。

差分隐私思想 P52

差分隐私的核心思想是通过引入随机化扰动的方式，在查询结果中添加噪声，从而保护个体隐私。随机算法 M 会对信息进行扰动，使得同一查询在两个数据集上产生相同结果的概率的比值接近于 1。这样可以保证任意一个个体在数据集中或者不在数据集中时，对最终发布的查询结果几乎没有影响。

差分概念 (P53)

隐私保护机制：对数据集 D 的各种映射函数被定义为查询 (Query)，用 $F=\{\mathrm{f1},\mathrm{f2},\cdots\}$ 来表示 一组查询，算法M对查询 F 的结果进行处理，使之满足隐私保护的条件，此过 程称为隐私保护机制（类比查询病患的例子）

邻近数据集：设数据集 D 和 D' 具有相同的属性结构，两者的对称差记作 $D\Delta D'$，$|D\Delta D'|$ 表示 $D\Delta D'$ 中记录的数量，若 $|\mathbf{D}\Delta\mathbf{D}'|=\mathbf{1}$，则称 $\mathbf{D}$ 和 $\mathbf{D}'$ 为邻近数据集（Adjacent Dataset)
例如，设 $D=\{1,2,3,4,5\}$，$D'=\{1,2,4\}$，则 $D\Delta D'=\{3,5\}$，$|D\Delta D'|=2$

差分隐私定义 P54

算法 M 提供 ε- 差分隐私保护，其中参数ε 称为隐私保护预算。

- 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。

差分隐私通过严格定义和数学方法的引入，提供了一种可靠的隐私保护模型，能够在隐私和数据可用性之间取得平衡。

差分隐私的实现(P56)

差分隐私可以通过在查询函数的返回值中加入噪声来实现隐私保护。这种噪声的加入可以使用不同的机制来实现，例如拉普拉斯机制和高斯机制。

全局敏感度和局部敏感度 P57~59

全局敏感度和局部敏感度是衡量查询函数敏感度的重要指标，可以用来确定加入的噪声的大小，从而在隐私保护和数据可用性之间取得平衡。给定的数据集 = 全局敏感度中 1 阶范式距离达最大的数据集时，局部敏感度就等于全局敏感度。

全局敏感度	局部敏感度
对任意的邻近数据集$\mathbf{D}$ 和$\mathbf{D}'$；只由查询函数决定。	对给定的数据集$\mathbf{D}$ 和它的任意邻近数据集$\mathbf{D}'$；由查询函数和给定的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

数值型差分隐私：拉普拉斯和高斯机制

数值型差分隐私的实现机制有拉普拉斯机制和高斯机制，通过在查询结果中加入随机噪声实现隐私保护 拉普拉斯机制提供的是严格的 $(\varepsilon, 0)$ 一差分隐私保护，而高斯机制提供的是松弛的 (ε, δ) 一差分隐私保护。

拉普拉斯机制使用拉普拉斯分布来生成随机噪声，高斯机制使用高斯分布来生成随机噪声，并分别加到查询结果中。在选择拉普拉斯或高斯机制时，需要平衡隐私保护和数据可用性，并根据具体的应用需求选择合适的参数。较小的 ε 值表示更强的隐私保护，但可能导致较大的噪声和较低的数据可用性。

非数值型差分隐私：指数机制

对于非数值型（离散型）的数据，可以使用指数机制实现差分隐私。指数机制使用指数分布来生成随机噪声，并根据查询的可用性函数对输出结果进行加权。可用性函数衡量了输出结果的优劣程度，指数机制提供(ε, 0)-差分隐私保护。

总结起来，差分隐私的实现可以使用拉普拉斯机制和高斯机制处理数值型数据，而非数值型数据可以使用指数机制来保护隐私。选择适当的机制和参数可以在隐私保护和数据可用性之间取得平衡，确保数据的隐私得到有效保护。

同态加密(P69)

同态加密的基本思想是对密文进行操作，而计算结果的解密值与对应明文的计算结果相同。这种特性使得同态加密在安全数据外包的场景中具有重要应用。

通过同态加密技术，可以将数据安全地外包给数据计算方，而不必担心隐私泄露。数据所有者可以对数据进行同态加密，然后将加密后的数据传输给计算方。计算方可以在不知道原始数据的情况下对密文进行计算和处理，并返回处理后的结果。最后，数据所有者使用其私钥对结果进行解密，获得最终的处理结果。

通过同态加密实现安全的数据外包，既保护了数据隐私，又充分利用了数据计算平台的计算能力，实现了数据隐私和计算能力之间的平衡。

组成P76

- **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。

目前，全同态加密仍面临着复杂度问题，离实际应用仍有一定距离。

目前，全同态加密仍面临着复杂度问题，离实际应用仍有一定距离。

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，既保证了投票者的隐私安全，又能够保证投票结果的公正性。

同态加密的优势与挑战 P82

优势：

- **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。

挑战：

- **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。

尽管同态加密正逐步向实用性靠近，但其安全性和实用性方面的研究还有很长的路要走。

半同态加密

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。

在一个加密方案中，如果加密算法和解密算法满足以下条件：

$$D(Enc(a) \otimes Enc(b)) = a \oplus b \tag{1}$$

其中，$\otimes$ 表示在密文域上的运算，$\oplus$ 表示在明文域上的运算，那么该加密方案被称为半同态加密。

半同态加密分为乘法同态加密和加法同态加密：

- 乘法同态加密：RSA 公钥加密算法，ElGamal 公钥加密算法 P84
- 加法同态加密：Paillier 公钥加密算法 P85，是最常用且最具实用性的加法同态加密算法。

全同态加密

全同态加密是指同时满足加法同态性和乘法同态性的加密方案，可以进行任意多次加和乘运算的加密函数。加密算法功能更强大，但由于计算复杂度较高，加密算法设计更加复杂，整体性能远不及半同态加密算法。目前，全同态加密仍然是一个热门的研究领域，尚在不断发展和完善中。

安全多方计算(P93)

安全多方计算(P93)

安全多方计算（Secure Multi-Party Computation，MPC）是一种解决互不信任的多方参与者在保护各自数据的前提下进行合作计算的方法。在安全多方计算中，多个参与者希望共同计算一个函数，同时保护各自的隐私或秘密数据，而不愿意让其他参与者知晓自己提供的信息。它解决了一组互不信任的参与方之间保护隐私的协同计算问题。安全多方计算协议使得参与者能够进行合作计算，而不需要彼此泄露输入信息，从而保护隐私。

安全多方计算的形式化描述 P98

安全多方计算的目标是在无可信第三方的情况下安全地计算一个约定函数，同时要求每个参与方除了计算结果外不能得到其他参与方的任何输入信息。安全多方计算需要满足以下特征：

- **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 P99

- **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 P100

- **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**：随着云计算的发展而发展起来的计算模型，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 P102

在安全多方计算中，基本密码协议是实现安全多方计算的关键工具。它包括茫然传输协议（Oblivious Transfer, OT）、混淆电路协议（Garbled Circuit, GC）和秘密共享协议（Secret Sharing）等。这些协议使用了多种密码学技术，如同态加密技术，来实现安全的计算过程。

安全多方计算的应用 P103

安全多方计算的应用 P103

安全多方计算的优势在于能够在保护隐私的同时进行计算，并具有较高的安全性和准确性。它被广泛应用于以下领域：

- **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。
- 分布式机器学习：联邦学习

安全多方计算牵涉到密码学的各个分支，有着广阔的应用领域，其优势为比较安全和准确，但涉及的加密技术开销、通信开销也很大。目前的研究主要集中于降低计算开销、优化分布式计算协议。

百万富翁问题 P105