**摘要**

我们报告了用于隐私保护数据分析的隐私集成查询（PINQ）平台的设计和实现。 PINQ通过类似SQL的语言为分析人员提供了一个原始数据编程接口。同时，PINQ的分析语言的设计及其精心实现为平台的任何和所有用途提供了差异隐私的正式保证。PINQ的无条件结构保证不依赖分析师的专业知识或努力，大大扩大了隐私保护数据分析设计和部署的范围，特别是非专家。

**类别和主题描述符：**H.3 [在线信息服务]：[数据共享]

**一般条款：**算法，安全性，理论

**关键词：**匿名化，机密性，差异化隐私，LINQ

**1.引言**

目前，广泛的组织收集和分析了大量的个人信息。虽然这些数据显然具有很大的分析潜力，但它们通常是在隐私的前提下收集的。无意的披露可能会对数据主体造成损害，并危及未来对此类敏感信息的访问。

这导致了对保证基础记录隐私的数据分析技术的极大兴趣。尽管在此类算法的设计方面取得了重大进展，但隐私结果仍然微妙，众多且完全不同。无数的定义，假设和保证甚至使得评估和调整新技术对隐私专家来说都是一个挑战。非专业数据分析师和数据提供者之间的谨慎合作几乎不可能。

这项工作提供了一个针对实时数据进行交互式数据分析的平台，该平台强制执行最强大的已知无条件隐私保证之一：差异隐私[1,2]。差异隐私要求在有和没有任何一条记录运行时，计算结果无法正式区分，几乎就像每个参与者都选择退出数据集。该平台包括声明性编程语言，其中所有书面语句提供差异隐私，并且精心实施的执行环境以实现差异隐私的形式要求。

该方法的重要特征是隐私保证由平台本身提供;它们不需要平台用户考虑隐私复杂性。这与许多先前的隐私研究实例不同，后者严重依赖专家设计和分析来创建分析，并通过专家评估来适当地审查提出的方法。在这种模式下，非专家分析师无法清楚或令人信服地表达自己，非专家提供者无法验证或解释他们的隐私保障。这个平台本身就是信任的共同基础，即使对于分析师和提供者也是如此没有以前的隐私经验，甚至没有彼此的经验。

我们优于现有平台的优势在于差异隐私;其强大的保证在于许多声明性操作兼容，并允许对此类程序进行端到端分析。它保证了在存在任意先验知识和任意后续行为的情况下，可以简化攻击模型并允许实际的增量部署。它的形式本身也能实现意想不到的新功能，包括使用groupby和连接敏感属性，分析文本和非结构化二进制数据，模块化算法设计（即没有整个程序知识），并分析整合多个独立的数据源。也许最重要的是，差异隐私不需要对底层记录的语义做出任何假设; 分析师将能够以原始形式直接针对甚至是最敏感和最微妙的数据集编写分析，而无需担心数据的披露

我们的方法的主要限制是分析师只能从远处操作数据：操作仅限于声明性转换和聚合;没有源或派生数据返回给分析师。这种限制对于许多分析师来说并不完全陌生他们无法亲自检查大量数据。相反，他们编写计算机程序将数据提取到可管理的聚合，然后他们进一步分析。虽然提出的平台在分析师和数据之间引入了更严格的边界，但它不是全新的。

**1.1 PINQ概述**

我们在一个称为Privacy Integrated Queries的平台上实现了我们提出的体系结构的原型。实现基于C＃的LINQ语言，这是.NET的一个良好集成的声明性语言扩展。数据提供者可以使用PINQ为每个分析人员提供指定的隐私分配的任意LINQ数据源。分析师使用PINQ数据源编写任意C＃程序，好像他们使用的是不受保护的LINQ数据源.PINQ的受限语言和运行时检查确保提供商的差异隐私要求得到尊重，无论分析师如何使用这些受保护的数据集。

PINQ被设计为现有分析引擎前面的薄层;它不管理数据或执行查询。相反，它提供了LINQ编写的常见转换和聚合的差异私有实现，由底层数据集的LINQ提供程序执行。这种方法大大简化了我们的实现，但也允许很大程度上部署的灵活性：数据源只需要一个LINQ接口来支持PINQ。

**1.1.1 PINQ的数学**

差异隐私要求有或没有任何一个输入记录的计算结果几乎与一样;我们现在推迟定义的定量形式。从每个参与者的角度来看，具有此保证的计算就好像他们的数据一样从来没有使用过。这是一个非常强烈的要求，部分是因为它没有对先验知识或数据的语义做出任何假设。它也是可实现的;最简单的例子是噪声计数：释放由对称指数（拉普拉斯）噪声扰动的数据集中的记录数。许多其他简单聚合具有类似的准确随机类似。

我们可以通过引入数据集的变换来显著扩展差异私有计算集。我们对许多关系转换的“稳定性”进行了分析，表明对其输入的微小更改总是导致对其输出的相对较小的更改。应用于变换数据的差分 - 私有分析掩盖了变换输出中的微小变化，因此也掩盖了其输入中的微小变化。组合转换和分析将提供差异隐私。这些转换可以任意组合，将差异隐私保证传播到其源数据。

最后，一系列差异私有计算也提供差异隐私;隐私消耗在最坏的情况下是累加的，因此可以在线跟踪。分析师可以在查询后进行查询，将其结果折叠到后续查询中，而不会影响我们描述和约束端到端隐私属性的能力。

**1.1.2 PINQ的实施**

我们已经将PINQ实现为基于功能的系统.PINQ允许数据提供者将LINQ数据源包装在具有编码差异隐私限制的受保护对象中，其中实现对象以评估查询的差异隐私属性并遵守强加的限制。评估由任何聚合触发，通过查询中使用的转换进行跟踪，并与参与数据源的当前限制进行比较。如果评估通过，则有效限制将减少，并且针对源LINQ提供程序执行查询，通过PINQ的差异私有实现返回。

我们强调PINQ代表了一个非常适度的代码库;在其当前实现中它只有613行C＃代码。数学后的评估逻辑并不复杂。必须仔细实现聚合以提供隐私，但这些通常是只需要对正确的聚合进行后处理（例如添加噪声）.PINQ还必须确保提交的查询符合我们的数学模型。 LINQ通过允许在Where谓词、Select函数和其他操作中使用通用c#计算来实现强大的功能。PINQ限制并管理这些计算以减少利用边信道的可能性。

**1.1.3 PINQ的应用**

使用PINQ进行编程是通过声明性LINQ语言完成的，在其他无约束的C＃程序中。分析师不能直接访问基础数据;相反，信息是通过PINQ的聚合提取的。为了交换这种间接，允许分析在未经掩盖的，未经改动的实时记录上运行。

除了一些重要的例外，使用PINQ编写的程序看起来几乎与LINQ中的对应程序相同。分析师从允许的转换中组装任意查询，并指定聚合的准确性。示例1包含一个C PINQ片段，用于计算发出输入查询短语搜索的不同IP地址。

我们将这个示例开发成一个更复杂的搜索日志可视化应用程序，展示了PINQ相对于其他方法的一些优势：丰富的数据类型，复杂的转换以及与更高级别应用程序的集成等。 完整的应用程序不到一百行代码，写了不到一天。

我们在PINQ中编写了几个其他数据分析的例子，包括k-means聚类，感知器分类和列联表测量。 这些例子是对现有方法的相对容易的改编[3,4]。我们还在PINQ中实现了关联规则挖掘，展示了许多有趣的权衡; 测量相同数量（在没有噪音的情况下）的不同方法可能导致惊人的不同结果，无法比拟的质量。

**1.2相关工作**

对隐私保护数据分析进行了大量研究，产生了许多不同的方法，几乎​​同样多的隐私定义。尽管我们可以在PINQ中复制许多（但不是全部）这些结果，但它的贡献并不在于存在这样的复制品，而在于它们的复制方式。每个先前工作的实例都需要专家的大量努力。研究人员在设计，分析和实施方面。用PINQ写的那些没有。

已经提出了几个用于交互式数据访问的平台，从简单的远程访问到查询审计方案[5]，以及密切相关的安全查询[6]。我们的主要偏离这些先前的工作在于我们提供正式结束的目的。 - 任意使用下的差异保密;我们不知道任何提供此类保证的现有分析平台。

发布是交互式数据访问的一种替代方式，其中数据集被清理，扰动，聚合，抑制和以其他方式改变以掩盖特定信息。遗憾的是，清理既不容易做，也不定义;该领域的研究尚未稳定在没有已知漏洞的强大定义[7,8,9]。此外，这些方法在发布之前故意扭曲数据，要求分析师在进行有效的统计推断之前理解这个（通常是故意保密的）过程。相比之下，PINQ中的分析是针对原始源数据进行的，生成的结果精确到定义良好（且较小）的加性误差。

密码学社区使用短语“隐私保护数据挖掘”，参见[10]，但是用于加密安全功能评估，它揭示了计算的结果，但没有关于输入的更多信息，而不是结果可能暗示的。这个定义回避了什么结果是安全释放的重要问题。即使是简单的计数也能准确地报告先前知识或重复使用的信息。我们在本文中的关注不仅是可能发生安全漏洞，而且即使通过忠实和安全地执行计算，也可以直接公开敏感信息。

信息流控制[11]通过一般计算跟踪敏感信息的流动。这种分析具有依赖性的二元描述;计算取决于输入还是不计算。统计分析和聚合通常由可信的去分类器处理，使用许多输入并且被视为依赖于无。差异隐私似乎概括了信息流控制的依赖关系到更精细和部分依赖概念的概念。 PINQ为此类背景下的解密提供了正式的工具。

PINQ在很大程度上依赖于差异隐私的先前工作。许多工作已经开发了差异私人计算，并告知PINQ的设计和实现。他们的主要实际缺点在于他们的要求（如同其他工作）分析师和提供者自己建立的实施提供差异隐私。此外，他们没有为分析师提供任何必要的工具来生成新的差异私有算法。展望未来，我们希望将我们的隐私保证建立在可信组件上，而不是根据第一原则重新调整结果，从而避免使算法，证明或实现不正确的可能性。

**1.3贡献**

PINQ的主要贡献是通过不需要隐私专业知识的工具为其用户（包括分析师和提供商）提供差异隐私功能.PINQ因素通常将复杂的隐私推理变为小型，透明且值得信赖的基板;它消除了分析师或提供者对此类推理的责任，并简化了分析从设计到部署的路径。

组装这个平台需要几个重要的步骤，从隐私理论到语言设计，再到实现。从理论上讲，我们重现了几个关于差异隐私属性的重要元结果，并介绍了转换稳定性推理在差异隐私中的应用。语言设计大幅度提取来自LINQ，但需要仔细调整以支持差异隐私：必须调整像Join这样的方法，并且必须引入像Partition这样的新方法以便有效和高效地使用。最后，像PINQ这样的平台的实现具有微妙的问题，以支持灵活和高效部署并限制间接泄漏信息的漏洞利用。

使用PINQ编写的应用程序可以安全地执行合理的隐私专家以前认为危险的许多计算：通过敏感标识符进行分组，加入受保护的表，对二进制数据或文本进行操作，集成多个数据集以及执行时没有完整的程序知识等等。这些重要功能长期以来一直是隐私技术的诅咒，但现在可以提供正式的隐私保障。

这种令人惊讶的灵活性来自差异化隐私。通过从临时和直觉的隐私方法转变，我们不仅提供正式的端到端保证，而且是扩展计算空间的正式基础。而PINQ的要点并非具体为了促进差异隐私，它是我们所知道的唯一隐私定义，支持对先前知识，安全组合和转换逻辑的弹性，以及其他几个属性。如果其他隐私定义确定了PINQ所需的属性，其核心思想应该导致可信赖的平台支持他们的隐私保障。

除了一个有用的平台本身，我们希望PINQ的存在导致一种不同的隐私研究方法。除了使用临时定义，无法访问的证据和随意实施进行和发布研究之外，大量研究可以获得隐私属性通过公共实现的共同原语。虽然PINQ可能无法表达很多研究，虽然PINQ不太可能成为可信赖的隐私平台的最后一个词，但它证明了这种原则性方法在隐私研究中的可行性。

**1.4论文大纲**

本文分为三部分，与1.1节并列。我们从第2节开始，回顾差分隐私的定义，并提供对我们的计算进行端到端分析所必需的支持数学。我们详细介绍了PINQ的设计和实现。 3，以及一些高级功能和安全性挑战。在第4节中，我们开发了一系列针对PINQ编写的应用程序，以展示其易用性和通用性，最终形成了一个用于Web搜索查询的可视化工具。最后，我们在第5节中总结了结束评论和进一步研究的方向。

**2.数学基础**

我们现在为PINQ开发一些支持数学。我们检查我们使用的隐私定义，差异隐私，并开发将程序接口暴露给数据所需的几个属性。具体而言，我们可以支持的数据类型，常见的差异 - 私有聚合，数据集的若干转换如何影响隐私，以及多个分析的隐私保证如何构成。我们的所有结论都是差异隐私的直接后果，而不是额外的假设或实施细节。

**2.1差异隐私**

差异隐私是一个相对较新的隐私定义，建立在[1]的工作基础上，并在[2]中公开阐述。它与大多数先前的定义不同之处在于它不会试图保证防止数据泄露，隐私侵犯或其他不良事件;相反，它保证参与数据集不是他们的原因。

差异隐私的定义要求在几乎相同的输入数据集上执行时，随机计算会产生结果几乎相同的分布。将输入数据集作为任意域上的多个记录集并将⊕用于对称差异：

***定义1.***我们说随机计算M提供差分隐私，如果对于任何两个数据集A和B，以及任何一组可能的输出S⊆RangeM），Pr [M(A)∈S]≤Pr[M(B)∈S]×exp(×|A⊕B|)。当x远小于1时，我们得到exp(x)≈1+ x。差异隐私确保当|A⊕B|相对小于1时，A和B下M的行为基本上无法区分。

定义并不难激励非专家。任何潜在的参与者都可以在两个输入之间进行选择计算M：包含其记录的数据集（A）和删除其记录的等效数据集（B）。他们的隐私问题源于这两种输入可能导致他们明显不同的结果的信念。然而，差异隐私要求任何输出事件在这些记录中发生的可能性几乎与没有这些记录时相同。从任何参与者的角度来看，提供差异隐私的计算几乎就像他们的记录未包含在分析中一样。

举一个具体的例子，考虑大多数网络搜索用户的合理担忧，他们的名字和搜索历史可能出现在纽约时报的头版[12]。对于每个参与者，有一些M的输出集合提示“纽约时报”出版本刊物;我们不一定知道这组S的输出是什么，但我们不需要为保留隐私保证而定义S. 对于所有用户来说，差异隐私确保了《纽约时报》公布其姓名和搜索历史的概率几乎不超过没有将其作为M输入的概率。除非用户告诉其他人，否则这是不可能的。

差别隐私和大多数其他定义之间的一个重要区别是它只限制了事件S概率的变化; 它没有讨论事件本身的概率。事件可能是甚至可能的。但是，如果即使没有这些记录也很可能发生披露，指责机构错误处理参与者数据是不合适的。

**2.1.1数据类型**

差异隐私仅依赖于数据集由记录组成的假设，并且在每个参与者的记录很少时最有意义。它不需要对基础记录的类型做出任何假设。它的隐私保证不是将属性分类为敏感或非属性，也不会扰乱源数据，也不是抑制稀缺或敏感的值。

数据类型的独立性是一个非常自由的属性。我们不必担心为不同的域定制隐私保证，将属性错误分类为不敏感，或忽略不敏感属性的敏感组合。我们可以为非结构化数据（如自由文本和二进制数据）提供有意义的保证，这些数据以前曾困扰过敏感度分类。我们甚至可以支持可变记录，用其内容的时间轴替换每条记录。

此外，通过完全忽略记录的语义，我们可以为它们的任意函数提供保证。这个属性是允许分析师编写自己的临时分析的基础，而不是从解密属性的一组预先筛选的计算中选择。

**2.2聚合：嘈杂的计数**

最简单的差分 - 私有聚合（来自[1]）释放数据集中的记录数量，受到对称指数（拉普拉斯）噪声的干扰，密度函数为p(x) exp(- | x |)，如图2。

将输入数据集从A更改为B可以将真实计数最多改变|A⊕B|。选择拉普拉斯分布是因为它具有将其中心（将真实值移位）转换一个单位的特性，通过最多exp(1)的乘法因子来缩放任何输出的概率。如果噪声首先乘以这成为exp()，导致差异隐私。

***定理1.***机制M(X)=|X|+Laplace()提供差异隐私。

证明。根据拉普拉斯分布的定义，对于任何输入A，在x处的M(A)的概率密度是

Pr [M(A)= x] exp(- × | x-| A ||)。

使用三角不等式| x- | A || ≥|x-|B|| - ||A|-|B ||，并注意到|| A |-| B || ≤|A⊕B|，我们推导出Pr [M(A)= x]≤Pr[M(B)= x]×exp(×|A⊕B|)。通过在S上积分x来实现差异隐私。

拉普拉斯分布在两个方向上具有指数尾部，并且在任一方向上误差超过t/的概率在t中呈指数小。释放的计数很可能接近真实计数。

还有许多其他机制可以提供不同的隐私;关于该主题的每篇论文通常包含几篇。迄今为止，通过基于预期行为的书面数学证明，每个都具有如上所述的隐私。虽然这显然是开发此类计算的重要步骤，但保证只有在证据可访问且实施正确时才具有说服力。

我们的目标是使用少量原始组件创建尽可能多的差异私有计算，其数学属性和实现可以被公开审查并可能被验证。虽然我们不应该排除引入新的原语，但它们应该是设计差异私有算法的特殊方法，而不是默认方法。

**2.3稳定的转换**

相反，它提供对一组固定聚合的访问，创造性使用的潜力有限，我们打算为分析师提供一种编程语言，用于描述新的和无法预料的计算。 PINQ的大部分功能及其主要贡献之一在于为分析师提供一系列丰富的转换，以便在差异私有聚合之前应用于数据集。

我们首先确定了转换的一个通用参数，它允许我们约束此类转换任意序列的隐私影响。

***定义2.***如果对于任何两个输入数据集A和B，，我们说变换T是c-稳定的。

| T(A)⊕T(B)|≤c×|A⊕B| 。

转型稳定性将在PINQ中发挥核心作用。具有有界稳定常数的转换将其输出的差分隐私保证传播回其输入，而稳定性常数则使其减小。

***定理2.***设M提供差异隐私，让T为任意c-稳定变换。复合计算M◦T提供(×c)-差异隐私。

证明。使用差异隐私和c稳定性的定义，我们看到对于任何A和B，

Pr [M（T（A））∈S]

≤Pr[M（T（B））∈S]×exp（×| T（A）⊕T（B）|）

≤Pr[M（T（B））∈S]×exp（×c×|A⊕B|）。

M◦T满足(×c)-差异隐私的定义。

甚至应用于多重变换数据集的差分 - 私有聚合对源数据具有精确的隐私影响。重复应用定理2，将应用的变换的稳定常数与分析的值相结合，产生界限。

备注。在第一个用于对域的任意子集进行计数的差分私有算法中可以看到早期的变换形式。感兴趣的子集可以由分析师指定，并且返回噪声计数。转换意图更正式地分离分析者可以自由地（例如限制数据集）与具有成本的操作（例如，用噪声测量计数）分离。这种灵活性允许我们引入新的转换，并允许分析师根据需要组合转换，反映他们的兴趣和专业知识。

**2.3.1稳定的转换**

我们现在讨论PINQ支持的四种转换，Where，Select，GroupBy和Join，以了解期望的稳定性范围。 PINQ支持的许多其他操作来自LINQ，但这一组很大程度上代表了所面临的问题。

将谓词作为输入并返回满足谓词的数据子集。 Where是1的稳定性，添加或删除源记录最多可以通过该元素的存在来更改结果。

Select执行并应用将每个源记录映射到可能不同类型的新记录的函数。Select通常从关系数据库中提取列，但实质上更为通用。Select的稳定性为1，因为每个源记录只产生一个输出记录。

GroupBy将记录映射到键值的函数，并生成组列表：对于每个观察到的键，映射到该键的记录组。 GroupBy比前两个转换更复杂，因为添加或删除输入记录可以更改输出记录，而不是简单地添加或删除它，从而导致对称差异为2。然而，这是可能发生的最大变化，稳定常数是2。

我们强调GroupBy操作的输出是元素组的受保护列表，而不是受保护的元素组列表;要实现后者，我们必须等到3.5节的特殊Partition运算符。

Join需要两个数据集，每个数据集的键选择函数，并返回其键匹配的所有元素对的列表。不受限制，Join可以将输入记录相乘，这样单个输入记录可以影响任意大量的输出记录，暗示无界稳定。

相反，我们将使用受限形式的Join，其中每个输入数据集首先按其连接键分组，然后使用组键将组列表连接。结果是原始输出的紧凑表示。加入，因为每对组原则上可以扩展到他们的完整笛卡尔积。然而，更重要的是，输出数据包的排列记录，以便我们可以应用稳定性数学。每个输入记录最多参与一对组，并且与GroupBy一样，稳定常数最多为两个。

对Join的这种结构限制限制了可以私下提取的信息，通过坚持每个连接键产生单个记录，例如，无论组有多大，最多只能为NoisyCount贡献一个。但是，没有预先分组隐私边界根本不存在，并且Join将不可用（并且在以前的工作中没有）。此外，强制分组不会干扰许多常见任务，例如使用Join来链接数据集之间的唯一标识符。

备注。 Join介绍了转换可能有多个输入的实际问题，应用分析揭示了有关其两个来源的信息。除非输入来自公共数据，否则这并不复杂。即便如此，对数据集的单一变化共同导致每个变换输入的有界变化，以及其输出的有限变化（即稳定性增加）。

**2.4组成**

任何隐私方法都必须解决构成问题：可以将多个输出结合在一起，即使进行联合分析，仍应提供隐私保障。组合问题是当前隐私保障的许多缺点的基础。例如，Ganta等人[13]表明交叉数据集的独立k-匿名化可能泄漏大量敏感数据。在这里我们回顾两个关于差异隐私的组成属性的先前结果，一个略有改进。

对于具有-differential隐私的一般分析系列，epsilon值增加，提供差异隐私。当我们暴露更多信息时，隐私保证会降低并非不自然;重要的一点是，他们以一种控制得很好的方式这样做，而不是完全崩溃，如[13]中k-匿名和变体所证明的那样。定理3给出了连续组合的界限。

在分析对数据的结构上不相交的子集进行操作的特殊但非常见的情况下，相同的分析序列提供-差异隐私。这种分析序列的一个常见例子是GroupBy-Aggregate分析，确保每条记录最多参与一个聚合。这扩展到更加丰富的算法类，我们将在后面进行扩展。定义4给出了并行组合的界限。

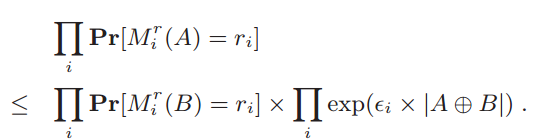
**2.4.1顺序组成**

每个单独提供差异隐私的任何计算序列也按顺序提供差异隐私。重要的是，这不仅在它们独立运行时也是如此，即使后续计算可以包含前面计算的结果也是如此。

在符号上，我们通过前面的结果明确地对每个计算进行索引，允许它们作为这些值的函数任意变化。我们仍然要求每个计算在其输入数据方面满足差异隐私。

***定理3.***让Mi各自提供-differential隐私.Mi(X)的序列提供差异隐私。

证明。对于结果ri∈Range（Mi）的任何序列r，我们为随,...,提供的机制Mi写。来自序列的输出r的概率是Pr [M(A)= r] = Pr [=]。对每个Mir应用差异隐私的定义

。

将第一个产品重构为Pr [M(B)= r]给出差异隐私。

对于希望处理多个查询的任何隐私平台，顺序组合至关重要。对顺序组合不健全的隐私定义，有几个，应该有一些怀疑态度。

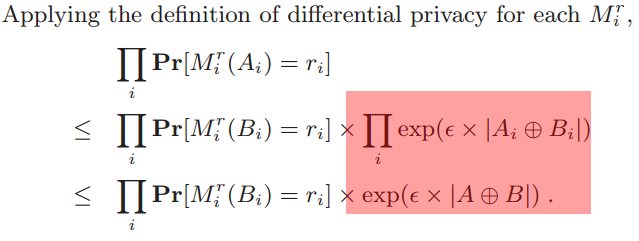
备注。先前已经在[14]中观察到定理3的不可区分性，这里我们的证明是相同的。

**2.4.2平行组合**

虽然一般查询序列累加了隐私成本，但当查询应用于数据的不相交子集时，我们可以改善边界。具体来说，如果输入记录的域被划分为不相交的集合，与实际数据无关，并且对每个部分的输入数据的限制都要进行差异 - 私有分析，则最终的隐私保证仅取决于最差的保证。每个分析，而不是总和。

***定理4***.让Mi各自提供 -差异隐私。让是输入域D的任意不相交子集的序列提供-差异隐私。

证明。对于A和B，让Ai =A∩Di和Bi =B∩Di，并且我们为随,...,提供的机制Mi写。来自序列的输出r的概率是Pr [M(A)= r] = Pr [=]。



对每个Mir应用差异隐私的定义，Y i。重新组装给出了 - 差异隐私的定义。

虽然顺序组合对于任何功能隐私平台都是至关重要的，但并行组合需要从隐私平台中提取良好的性能。现实分析需要在不同子群体上进行聚合和计算。尽管可以将这些操作分析为顺序组合，但隐私保证将随着所分析的子群数量而缩放。利用并行组合，隐私成本是固定的，与总查询的数量无关，因此允许以适度的隐私成本获得相对全面的信息。

备注。先前已经观察到定理4在[1]中将不相关的子群计数添加到噪声中。然而，所使用的隐私定义（不可区分性）在类比定理中引入了两个因子，并且这些因素与推理的每次调用相结合。我们略微改变的隐私定义允许我们任意地应用该定理而不增加界限。

**2.5隐私微积分**

本节的定理支持丰富的隐私计算，允许我们约束由允许的转换和聚合组成的任意查询的任意序列的隐私含义。重要的是，我们可以通过在线方式进行推理。按顺序到达的查询的epsilon值会累积;并行应用的查询要求我们仅跟踪最大值。

这种简单性使我们能够避免给分析人员带来正确或完全描述其查询的数学特征的责任。即使是熟悉数学的研究人员（例如作者），推理过程也可能非常微妙且容易出错。幸运的是，它可以自动化，这是第3节的主题。

**3. PINQ实施**

PINQ是在C＃语言集成查询的基础上构建的.LINQ是.NET框架的最新语言扩展，用于将对数据流的声明性访问（使用与SQL非常相似的语言）集成到任意C＃程序中。对LINQ的中心是IQueryable <T> type，类型为T的记录的通用序列.IQueryable允许转换，例如Where，Select，GroupBy，Join等，在可能的新类型上返回新的IQueryable对象。

PINQ的实现以PINQueryable <T>泛型类型为中心，包含基础IQueryable <T>。这种类型支持与IQueryable相同的方法，但实现确保在调用任何执行之前进行适当的隐私计算。我们强调PINQ不提供执行引擎。相反，PINQueryable将其所有操作呈现为由底层IQueryable执行的LINQ语句，从而允许部署的大量灵活性。

**3.1数据类型和控制流程**

PINQ的中心类型是PINQueryable <T>，T类型对象的受保护列表。类型T可以是任意的。每个PINQueryable由不受保护的数据集（IQueryable）和第二个新数据类型组成， PINQAgent，负责接受或拒绝epsilon的增量.PINQueryable支持聚合和转换.Aggregations测试相关的PINQAgent以确认epsilon的增量在执行之前是可接受的.Transformations导致新的PINQueryable对象具有转换的数据源和新的PINQAgent，包含适合转换的逻辑，用于将epsilon请求转发给其源PINQueryable数据集的代理。

PINQAgent接口有一个方法Alert（epsilon），在执行任何具有相应的epsilon值的差异私有聚合之前调用，以确认访问。对于包裹在原始数据集周围的PINQueryable对象，PINQAgent由数据提供者基于其隐私要求，从头开始或使用多个默认值之一（例如，减少每个分析师的预算）。对于因其他PINQueryable数据集的转换而产生的对象，PINQ构造一个PINQAgent，它使用符合转换的epsilon缩放值查询转换输入的PINQAgent对象。这些查询将以适当的epsilon值递归转发，直到查询完所有源数据。图3概述了该过程。

**3.2差别隐私政策**

数据提供商通过提供实现PINQAgent接口的对象来规定PINQ中的隐私要求，其中包含用于调解对其数据的访问的任意代码。在使用受保护数据执行任何聚合之前，PINQ将调用相关PINQAgent对象的Alert方法以及待处理的隐私递减，epsilon.Each PINQAgent应该以布尔值响应，接受或拒绝请求。

PINQ的主要作用是作为一种隐私执行机制;它的作用不是决定谁应该访问哪种类型的数据，而是在制定这些决策后执行这些决策。我们的意图是提供者应该能够在需要构建PIN IQueryable保护它之前确定任何个人应该对数据集进行多少访问。在提供者识别分析师并确定他们的访问级别之后，他们可以构建一个简单的PINQAgent来强制执行这些限制。

例如，隐私政策可能会针对不同角色要求不同级别的访问权限：外部分析师可能会有固定的预算来提取，而内部分析师可以不受约束地访问。或者，如果分析师请求不太敏感的视图，可能会删除某些列或应用分组，则策略可能会以更宽松的PINQAgent对象的形式允许更精细的细节和更多访问。应用哪些约束的决定可以而且应该是在首次构建PINQueryable之前制作，仅使用PINQAgent来强制执行这些决定。

示例2显示了管理固定隐私预算的代码。

PINQ中还存在其他几种PINQAgent实施，包括执行上述预算，通过引入人工时间延迟来限制信息提取，简单计算和记录隐私损耗。提供商可根据需要自由实施自定义代理，但鼓励在构建代理之前执行尽可能多的隐私推理。

PINQ使用差异隐私作为通用货币为所有参与者提供了明确的语义，但限制了PINQAgent在运行时可用的信息的丰富性。这种限制是故意的.PINQ打算成为一个基于能力的轻量级系统，通过缩小其焦点来简化其实现。此外，一些操作（例如第3.5和3.6节）基于差异隐私是可互换的假设：曾经授权差异私有访问量，可以任意使用访问权限，如分析师认为合适。

**3.3聚合运算符**

PINQ中的每个聚合都将epsilon作为参数，并提供与其直接数据源相关的差异隐私。对于从中导出此数据集的基础数据集，隐私影响可能更糟。在执行之前，每个聚合都会使用此epsilon调用其关联的PINQAgent的Alert方法，仅在最终响应为正时才执行聚合。

根据定理1实现NoisyCount，返回基础数据的精确计数加上拉普拉斯噪声，其大小由分析员指定，如果足够大。示例3描述了NoisyCount的实现。

PINQ包括其他聚合 - 包括NoisySum，NoisyAvg和NoisyMed等 - 每个都采用epsilon和一个函数将每个记录转换为double。为了提供差异隐私，结果值首先被钳位到区间[-1，+ 1}在它们被聚合之前。这对于确保单个记录对聚合的影响有限非常重要，允许相对较小的扰动来提供差异隐私。

这些方法的实现及其隐私保证的证明主要是先前的工作。与NoisyCount一样，NoisySum是通过添加拉普拉斯噪声实现的，并在[1]中讨论过。使用[15]的指数机制实现NoisyMed和NoisyAvg，并以如下所示的概率在 [-1，+ 1]范围内输出值

Pr [NoisyMed（A）= x]αmaxmed（B）= xexp（ - ×|A⊕B| / 2）

Pr [NoisyAvg（A）= x]αmaxavg（B）= xexp（ - ×|A⊕B| / 2）

每个通过对输入A的最少修改来降低x的概率，使得x成为正确的答案。

NoisyAvg的准确度大约为2 /除以数据集中的记录数。 NoisyMed产生的值将输入记录划分为两组，其大小大致相差2 /;它不需要在数值上接近实际中位数。

**3.3.1可扩展性**

PINQ旨在成为一个可扩展的平台，并允许通过子类型进行扩展：其他隐私专家可以对PINQueryable类进行子类型添加自己的差异 - 私有聚合，依靠PINQ的基础设施来确认差异私有访问。聚合应首先查询关联的PINQAgent，如上面的示例3所示，然后可以聚合原始数据。我们以这种方式实现了对指数机制的限制[15]。

与此同时，PINQ的目标是允许分析师使用其工具编写有效的分析，而不是从头开始编写分析并将其作为可信方法加以分析。 PINQueryable仅与其最弱的聚合一样安全，并且应该谨慎添加新功能。

**3.4转换运算符**

PINQ的灵活性源于其转换运算符，每个运算符都会导致新的PINQueryable包含更新的数据源。 LINQ中不会修改数据，而是生成新的查询计划。在接受之前，相关的PINQAgent被连线以将请求转发到参与的源数据集，通过变换的稳定性常数来缩放epsilon。

我们对许多转换的实现主要是构建新的PINQueryable和PINQAgent对象，并使用适当的参数连接。如第3.7节所述，需要注意限制计算。示例4描述了PINQ的GroupBy的实现。大多数转换需要类似的简单隐私逻辑。

Join变换是我们与LINQ的主要偏差，LINQ的实现在应用无限制LINQ Join之前通过join键对每个输入数据集进行预分组。这涉及到一个新的签名，因为reduce方法需要成对的记录组而不是简单地对记录，意味着需要（稍微）重写使用Join的LINQ程序以利用此功能.PINQ承认具有LINQ签名的实现（具有对减少而不是组对减少）

它仅释放每组中第一个匹配对的减少量。我们避免使用这种方法来避免暗示这种方法尊重LINQ语义，并强迫分析师承认离开。

**3.4.1可扩展性**

与聚合一样，PINQ允许通过子类型仔细扩展其转换。有几种具有有界稳定性的自然方法在LINQ中找不到，但仍然很有价值。一个简单的例子是多路连接，其中使用公共密钥连接两个以上的表。如果通过重复使用PINQ的二进制连接来实现，则生成的查询将针对每个应用程序以2的幂进行稳定性缩放，而对于每个数据集，真正的稳定性仅为2。存在若干其他类似操作，并且子类型允许包含它们而不需要更新核心PINQ库。

我们还在DryadLINQ基于群集的LINQ提供程序之上对默认的PINQueryable进行了子类型化，以提高性能，其中数据放置和扩展的意识允许更高效的实现。

**3.5分区操作员**

如第2.3.1节所示，并再次在示例4的类型签名中看到，GroupBy操作按键对输入进行分组，但将组保留为隐私窗帘后面的受保护记录。对于许多分析，我们倾向于将受保护的数据集破碎成多个受保护的集合，沿着某些用户定义的键函数绘制的线条。

定理4告诉我们，结构上不相交的查询仅花费最大的隐私差异，我们希望将此功能公开给分析师。为此，我们引入了分区操作，如GroupBy，但分析师必须明确提供一组候选键。分析师将获得一组PINQueryable对象，每个候选键一个，包含（可能为空）映射到每个相关密钥的记录子集。重要的是PINQ不会泄露实际数据中存在的密钥集，因为这会违反差异隐私。因此，分析人员必须指定感兴趣的密钥，和PINQ一定不能纠正它们。某些子集可能为空，某些记录可能不会反映在任何子集中。

这些新PINQueryable对象的PINQAgent对象都引用源数据的相同源PINQAgent，但是定理4将仅向代理提醒epsilon的最大值的变化。自构建以来，代理共享其累积的epsilon值的向量，并在每次更新时查询此向量以查看最大值是否已增加。如果是这样，他们会转发最大值。如果最大值没有增加，他们接受请求。

在以下两个查询中可以看到GroupQy和PINQ在分区中的使用之间的区别：

Q1。多少邮政编码包含至少10名患者？

Q2。对于每个邮政编码，有多少病人住在那里？

对于Q1，通过ZIP的GroupBy，患者数量的位置和NoisyCount给出了至少10名患者的邮政编码的确切数量的近似答案。对于Q2，通过ZIP进行分区，然后对每个进行NoisyCount part返回每个邮政编码的近似计数。由于测量结果可能很嘈杂，因此查询不一定能为另一个提供良好的估计。然而，两者都是重要的问题，PINQ能够准确地回答问题的提出方式。

分区运算符不仅可以通过聚合，而且可以通过对每个部分进行进一步的差分私有计算来跟踪。它实现了第4节中演示的强大的递归下降编程范例。

**3.6隐私预算**

使用PINQ的分析师不确定是否会接受或拒绝任何请求，并且必须只希望基础PINQAgent接受他们的所有访问请求。相反，我们提供了一种尝试“分配”所请求的隐私预算的方法，请求使用输入预算，如果成功，则将具有此预算的新PINQueryable硬连线到其代理中（如示例2所示）。当代理被销毁时，它会释放任何未使用的预算。

此预算操作在支持子例程和第三方代码时也很有用。虽然分析人员可以将PINQueryable传递给任何子例程，但是他们冒着子例程可能消耗所有剩余隐私预算的风险。通过预先分配具有有限隐私分配的对象，分析师可以确保适当的隔离。

**3.7实施中的安全问题**

尽管稳定性数学，组合属性和差异隐私的定义提供了数学保证，但只有当分析师按照预期使用PINQ（即诚实但好奇的模型）时才会这样做。有许多实现细节必须妥善处理为了防止恶作剧的分析师打开信息泄漏的意外渠道。每个人都支持协商实施经过良好测试的平台，而不是从头开始反复重新实施隐私技术。

以下许多问题都是针对C＃的，但类似的问题可能适用于任何足够丰富的语言。

**3.7.1非功能代码**

LINQ中作为参数的方法，例如Where中的谓词，Select中的函数以及GroupBy和Join中的键选择器都表示为表达式树。这些数据对象编码计算，虽然在很大程度上没有副作用，但可以调用任意C＃代码。还可以间接调用多个函数，例如GetHashCode和Equals，并且可以重载这些函数以通过不安全的通道发出数据。此外，异常和非终止是输入方法报告基础记录的另一种方式。

幸运的是，表达式树在它们直接调用方法的时间和地点相对清楚。我们将方法限制为我们知道[或相信]无副作用的方法，包括大量有用的库代码。由于大多数C＃代码不是无异常的，我们将每个计算都包装在try-catch块中，如果发生异常则返回默认值。间接方法调用更难识别，但可以通过将系统中允许的类型限制为基类型或从它们构建的匿名类型来控制。必须明确调用构造函数，我们只是禁止构造用户定义的类型。

PINQueryable通过将Purify函数应用于所有输入方法来实现此检查（如示例4所示）。我们的实现尝试在语言的相信安全子集中​​重写所有输入方法，除非它失败。该方法可以被覆盖供应商。例如，更深入的代码检查可以为用户定义的代码提供更大的灵活性，但这远远超出了本说明的范围。

**3.7.2特洛伊木马IQueryables**

LINQ的设计目标之一是允许数据提供者在其数据集上提供LINQ操作的自定义实现。这允许在部署方面具有很大的灵活性，并且是PINQ想要保留的东西。但是，恶意分析师可以引入IQueryable，其实现二进制操作（例如Join）（由分析师）实现只读取第二个IQueryable的内容并把它们写出一个不安全的渠道。显然，我们必须避免信任任何IQueryable的实施，或至少委托对受影响的数据提供者的信任责任。

为此，二进制转换需要在两个数据集之间进行“握手”，其中第一个调用第二个数据集，请求使用其未受保护的数据源重新调用初始转换。然后第二个可以评估调用者的类型，以及要调用的方法的句柄。如果建立了信任 - PINQ的默认设置是测试IQueryables的类型是否相同 - 第二个对象使用其原始数据调用该方法，并继续计算。

**4.申请和评估**

在本节中，我们将介绍使用PINQ编写的数据分析。很明显，并非所有分析任务都可以在PINQ中实现（实际上，这就是重点），但我们的目的是让读者相信该集合足够大以至于广泛有用。

我们的主要示例应用程序是基于包含IP信息和查询文本的搜索日志的数据可视化。该应用程序演示了PINQ的许多功能，其他隐私保护数据分析平台基本上没有。这些功能包括直接访问未修改的数据，用户提供的记录记录转换，GroupBy和Join等“敏感”属性，多个独立数据集以及不受限制地集成到更高级别程序中的操作。

PINQ中可以表达许多其他非平凡的分析，包括k均值聚类，感知器分类，列联表测量和关联规则挖掘。大多数是相对直接的，遵循先前的研究[3,4]。

对于我们的实验，我们使用DryadLINQ [16]提供程序。 DryadLINQ是一个研究LINQ提供程序，在Dryad [17]中间件上实现数据并行计算，目前可扩展到至少数千个计算节点。我们的测试数据集大小有限，大约100GB，并没有完全运用DryadLINQ提供商的可扩展性。我们不报告执行时间，因为PINQ的推理是一个微不足道的贡献，而是我们可以私下从数据中提取的信息的数量和性质。

为清楚起见，我们提供了一些示例，就好像数据分析师也是数据提供者一样，负责组装源PINQueryable对象。在实际部署中，此程序集应在单独的受信任基础结构上完成。

**4.1数据分析：第1阶段的第3阶段**

我们从PINQ的简单应用开始，近似搜索指定查询词的不同搜索用户的数量。我们的方法与LINQ一样：我们首先将搜索记录（逗号分隔的字符串）转换为字段具有已知含义的元组（字符串数组），然后使用输入搜索查询将数据限制为记录，然后按提供的IP的地址进行分组获取不同用户，然后计算剩余的记录（字符串数组）。完整程序在例5中重现。

这个相对简单的例子展示了PINQ的几个重要特征。输入数据是文本字符串;我们碰巧先验地知道它们是逗号分隔的，但这些信息在隐私保证中不起作用。过滤是针对anaylst提供的查询术语完成的，可能是频繁的或不常见的，敏感的为了得到一组不同的用户，我们使用记录的IP地址进行分组，显然是高度敏感的信息。

**4.2数据分析：第2阶段的第3阶段**

我们编写的程序给出了单个查询的计数，如果分析师想要额外的计数，他们必须再次运行程序。这会产生额外的隐私成本，并且不适合提取大量查询计数。

相反，我们可以重写以前的程序以使用分区运算符以固定的隐私成本允许任意数量的计数。我们使用相同的键选择函数和一组输入查询字符串来对记录进行分区，而不是使用Where过滤记录。完成后，我们遍历每个查询和相关部分，按IP地址对每个查询中的记录进行分组。为了进一步丰富该示例，我们然后将每个这些数据集分区为每个IP发出查询的次数，然后产生噪声计数。 （参见实施例6）。

因为我们使用分区而不是多个Where调用，所以PINQ可以看到与程序相关的隐私成本仅是每个循环的隐私成本的最大值，与示例5中的成本完全相同。

表1报告了我们输入集上的一些查询的度量。每个报告的测量值是精确计数加上参数10的拉普拉斯噪声，对应于标准偏差10√2。对于某些测量，该误差相对不重要。对于其他测量，它是重要的，但仍然表明原始值非常小。

4.3数据分析：第3阶段3

我们现在将我们的示例程序从简单的报告（一种并非罕见的任务）扩展到更丰富的分析应用程序。我们的目标是可视化各种搜索查询的搜索位置分布。在高层次上，我们将IP地址转换为纬度 - 经度对，通过连接第二个专有数据集，然后将坐标发送到从[18]的工作借来的可视化算法。虽然我们将在高层次上描述可视化算法，但是PINQ提供隐私保证是基本的，而不知道算法计划对数据做什么。

从前面的示例开始，我们通过查询对数据集进行分区并按IP地址对结果进行分组，现在我们演示一个片段，它可以让我们将IP地址转换为纬度 - 经度坐标。我们使用第二个数据集iplatlon，条目是IP地址和相应的纬度 - 经度坐标。我们使用每个中的IP地址作为关键字连接这两个数据集，从而产生一个纬度坐标对代替每组搜索。示例7包含此Join转换的代码，但可能需要一些解释.

LINQ中的Join转换需要四个参数：第二个数据集，第一个键选择器函数，第二个键选择器函数，最后一个要应用的缩减函数记录对。 PINQ中的语法是等效的，但减少函数除外，它减少了一对记录组。我们的减少功能输出第二组中的第一个lat-lon记录，每个IP地址一个。

正如我们所看到的，PINQ支持使用主键自然且相当普遍地使用Join来转换数据集。在这种情况下，我们希望这些组是单例，并且加入适用于非PINQ情况下的预期如果我们使用Join with LINQ语法的实现，我们也会得到预期的行为，在这种情况下，PINQ和LINQ代码在字面上是完全相同的。

备注。我们的第二个数据集提出了关于差异隐私的替代应用的有趣观点。虽然我们执行的操作，将IP地址映射到纬度 - 经度，实质上只是一个复杂的Select，描述映射的数据集是专有的。数据集中的每条记录都需要投入一些精力来制作，业主可能希望从中提取

值。通过PINQ使用这些数据可以防止单个记录的传播，保留数据集的价值，同时仍允许使用。专有数据的这种使用似乎与个人数据的使用有许多相似之处。数据管理是另一种设置，其中人们想要允许访问数据，但是排除了数据本身的传播。

最后，我们的算法获取IP的lat-lon坐标列表，搜索输入搜索查询，并调用Visualization子程序，该子程序使用[18]的算法。在高级别，该子程序对输入数据进行分区（geo-空间坐标）在越来越精细的粒度下，在每个粒度上测量每个区域的噪声计数。从这些计数中，它能够合成一个代表性的数据点分布，大致与计数中观察到的趋势相匹配;密集子区域包含许多代表性数据点，备用区域包含的相对较少。查询“板球”的示例可以在图4中看到。

不完全确定此例程如何或为何如此有效以及可能无法访问[18]的读者与大多数数据提供者的情况大致相同。我们几乎没有直觉知道为什么计算应该保护隐私，也没有任何即将到来的。尽管如此，由于例程只通过PINQueryable提供对数据的访问，即使不了解算法的意图或实现，我们也可以确保差异隐私保证。由于PINQueryable的所有用途都保证差异隐私，数据提供者不需要了解（或曾经知道）分析师计划对数据做些什么，以确保实施差异隐私。

支持隐私算法的“模块化设计”是研究和开发的重要推动因素，消除了对计算的端到端理解的需要。这对于探索性数据分析尤为重要，甚至分析师自己也可能不知道他们需要回答的问题，直到他们开始询问他们。删除整个程序理解的要求也可以实现专有数据分析，分析师可能不想泄露他们打算进行的分析。而执行平台必须明确指示在分析师需要的计算中，数据提供者不需要被告知他们的具体细节。

5。结论

我们提供了“隐私集成查询”（PINQ），这是一个值得信赖的隐私保护数据分析平台.PINQ提供对任意敏感数据的私人访问，无需分析师或提供商的隐私专业知识。界面和行为非常类似于语言集成查询（LINQ）和隐私保证是差异隐私的无条件保证。

PINQ提供了一个为隐私技术和研究建立更正式和透明的基础的机会.PINQ的贡献不仅在于可以编写私有程序，而且只能编写私有程序。由可信组件构建的算法在结构上继承隐私属性，并且不需要专家分析和理解来安全部署。这扩展了敏感数据的能力用户集，增加了可移植性

跨数据集和域的隐私保护算法，并扩大敏感数据分析的范围。

**5.1进一步的研究方向**

差异隐私的保证相当强，但可能以牺牲准确性为代价。其他具有坚实数学基础的较弱定义确实存在，**但值得注意的是近似差分隐私[14]也基于行为边界变化的差分概念作为输入变化函数的隐私**。该定义允许转换和组合逻辑，并且许多PINQ基础结构也可以支持此定义。

PINQ是使用LINQ实现的，并且继承了它的一些功能，包括语言集成，强类型，灵活的执行和优化，以及易于扩展。转换稳定性数学可以很容易地应用于其他数据分析语言，例如SQL。其他数据分析语言存在（例如科学和统计软件包）并理解我们可以在多大程度上为它们设计可信的私有实现是开放的。

我们已经看到了几个用PINQ编写的示例程序，开放性研究是为了考虑PINQ中可以编写哪些其他分析，以及他们如何有效地使用他们的隐私资源。重新研究算法设计，着眼于隐私成本，有可能通知和改进私人数据分析以及PINQ的新迭代。

PINQ的可扩展设计允许其他隐私研究人员利用其基础设施来支持新功能。我们已经看到了一些未包含在PINQ中的转换和聚合示例，这些示例似乎不能与其原语一起重现。扩展支持的操作集是一个自然的研究方向，理想情况下着眼于将更复杂的操作分解为常用的原语，最小化可信基础并最大化重用的可能性。

致谢

作者非常感谢几位合作者的贡献。 Ilya Mironov，Kobbi Nissim和Adam Smith各自对非专家可用的隐私工具和技术表示了极大的兴趣和支持。 Yuan Yu，Dennis Fetterly，Ulfar Erlingsson和Mihai Budiu'极大地帮助了作者对LINQ进行了教育，并为PINQ的设计和实施提供了信息。许多读者和评论者都提供了大大改进论文表达的评论。