FastGeo: Efficient Geometric Range Queries on Encrypted Spatial Data

一些缩写和符号表示

GSE: Geometrically Searchable Encryption(空间可搜索加密)

PRF: Pseudo Random Encryption(伪随机加密)

SSW: Shen-Shi-Waters Encryption

FHE: Fully Homomorphic Encryption(全同态加密)

BGN: calculates additions and at most one multiplication on encrypted data.(计算加法和至多一次乘法，这是由D. Boneh, E.-J. Goh, and K. Nissim三个人提出的一个方法，这篇文章中取了三个人名字的首字母命名了这个方法为BGN)

Δ_T^α: 整体的数据空间，α表示维数，T表示每一维的大小。*例：假设每一维大小都相同，并且范围都在[0,T-1]，假设*$\alpha = 2$*，这样每个点都可以被描述为：*$D_i = (d_{i,1}, d_{i,2})$，$d_{i,1}$*和*$d_{i,2}$*分别表示数据点的x分量和y分量，并且*$d_{i,1}$，$d_{i,2} \in [0, T-1]$。

正文

文章的主要贡献：

- 设计了一种支持对加密空间数据进行几何范围查询的方案——FastGeo。
- FastGeo不仅支持对加密空间数据的查询，还支持高效的更新。
- FastGeo的查询速度相较之前的可搜索加密方案提高了至少100倍（非常高效）。

背景介绍

可搜索加密技术是能够使客户端在不用解密的情况下，对远程服务端中的加密数据进行搜索，而且不泄露隐私数据和查询。在以往的可搜索加密方案中，大多数都是聚焦于普通的sql查询，如关键词查询和范围查询。对空间数据进行几何范围查询需要进行先计算再比较的操作，而当前的加密方案没有适用于在密文上进行先计算再比较操作的，因此需要设计一种能够支持先计算在比较操作的加密方案。如何在加密的空间数据上以亚线性的搜索时间执行任意的几何范围查询，而且还支持高效的更新是一个仍待研究的问题。

问题描述

系统模型：客户端和服务端
客户端将空间数据的存储和查询服务交给服务端来进行，但服务端是honest-but-curious，服务端能够提供服务但是会泄露客户端的空间数据和查询，那么客户端就需要在服务端处理之前加密它的空间数据和查询。
因此，需要设计一种能够在加密空间数据上进行高效几何范围查询的可搜索加密方案。

FastGeo方案流程

为了能够执行先计算后比较操作，文中将空间数据和几何范围查询转换成了“等值向量”的形式，并提出一个两级搜索的方案，第一级利用PRF做等值检查，第二级利用SSW判断内积是否为零。

将数据和查询转换成“等值向量”形式

我们以Δ_{10}^2（代表数据是二维的，并且每一维的大小为10，即$x \in [0,9],\ y \in [0,9]$，只考虑整数情况）为例，以字典和链表结合的形式来表示数据集中的点，数据点的x值作为字典的键，数据点的y值以向量的形式作为字典的值，向量的表示方法为：1）向量维数为T值（即数据点的每一维的大小）。2）将向量的y值的位置置为1，其余位置置为0。例如数据集中的点为(1,3), (4,7),(6,2),(6,4),(9,3)。

Dictionary	Link Lists
x=1	-> (0,0,0,1,0,0,0,0,0,0)
x=4	-> (0,0,0,0,0,0,0,1,0,0)
x=6	->(0,0,1,0,0,0,0,0,0,0)->(0,0,0,0,1,0,0,0,0,0)
x=9	-> (0,0,0,1,0,0,0,0,0,0)

对于给定的几何范围查询的表示，将几何范围中的所有数据点都枚举出来，然后用字典和链表的形式进行表示，与数据集中的点表示方法不同，查询中的点的y值表示成向量时是y值的位置置0，其余位置置1。如几何查询范围为三角形，顶点为(5,4), (5,1), (7,1)。

x-subqueries	y-subqueries
x=5	(1,0,0,0,0,1,1,1,1,1)
x=6	(1,0,0,1,1,1,1,1,1,1)
x=7	(1,0,1,1,1,1,1,1,1,1)

用“等值向量”形式进行搜索

给定查询，首先检索字典中的每个x分量的子查询，一旦找到匹配的x值，则继续计算该x值相对应的y分量与y分量的子查询的内积，如果内积为0，那么则表示数据点在几何范围查询中。
例如，上表所示的，x=5和x=7通过等值检查发现都不在数据集的字典中，发现x=6在数据集的字典中，然后用x=6对应的y向量(1,0,0,1,1,1,1,1,1,1)分别与数据集中的x=6对应的(0,0,1,0,0,0,0,0,0,0)和(0,0,0,0,1,0,0,0,0,0)做内积，检查是否有内积等于0的。发现(1,0,0,1,1,1,1,1,1,1)与(0,0,1,0,0,0,0,0,0,0)内积为0，则表明数据点(6,2)在该几何范围查询中。

应用加密原语

在honest-but-curious服务端进行查询时可能会出现这样的情况：用户给出的查询请求为{$\{x_1, v_1\}, \{x_2, v_2\}$}，而服务端将其混淆为{$\{x_1, v_2\}, \{x_2, v_1\}$}。
为防止这种情况的出现，文中又提出了一种加强式的向量形式，其基本思想是将第一级中的值嵌入到第二级的向量中。例如，给定数据点(6,2)，y的加强式向量形式为(0,0,H(6),0,0,0,0,0,0,-1)，对于y子查询$y \in [1, 2]$，其加强式向量形式为(0,1,1,0,0,0,0,0,0,H(6))。很显然原向量的内积为0与加强式向量的内积为0是等价的。

更新

FastGeo能够支持加密数据是因为，FastGeo的加密并没有加密整个数据结构，而是对数据结构中的分量进行加密，因此对加密数据的更新只要遵循数据结构的更新规则即可进行更新。

实验

作者用49870个真实的数据点作为数据集。
实验了设置不同的向量长度对方案执行时间的影响，发现不同的数据集大小下最优的向量长度是不同的。
实验了不同的第一级查询的个数对方案执行时间的影响，发现执行时间与第一级查询个数呈正比关系。
实验了更新时间，发现当将向量长度设置为1000时，更新时间也只有0.977s，若减小向量长度，更新时

间将缩小。
与之前的对于加密空间数据做几何范围查询的方案做了对比。

	Search Time(s)	Complexity	Token Size	Update
GR	1753	linear	0.96KB	No
WLW	1583	logarithimc	20KB	No
FastGeo	13.67	sublinear	132KB	Yes

从表中可以发现，FastGeo相较其他两种方案的复杂度和时间开销都很小，而且只有FastGeo支持更新，但其代价就是Token的Size相较其他两种方案大，这会造成更多的通信开销。

总结

文章提出了FastGeo——高效的两级搜索方案，支持在加密的空间数据集上进行几何范围查询和更新操作。作者通过在真是数据集上的实验结果证明了方案在实际中的有效性。
文章的主要创新点为提出了一种两级搜索的思想，这与以往的搜索思想都不相同，这种搜索思想可以支持更加复杂的操作（例如先计算再比较操作），在未来的问题解决中可能会得到一定的应用。