# 第八章性能评估

本节通过对真实的数据集进行实验，从而对图形加密方案进行评估。

## A.步骤

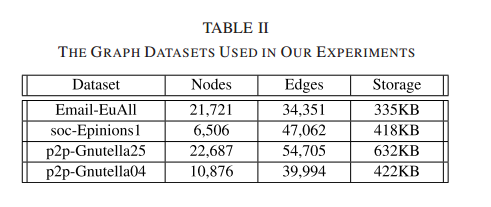
### 1.试验

我们实现了在 [2] 中介绍的构建 2HCLI 的方法。实现中的 ORE 和 SWHE 分别遵循 [15] 和 [16] 中描述的方法，此外，我们采用GMP库进行大整数运算。我们设置了安全参数 λ= 128 并且所有基本密码基元依赖OpenSSL库做支撑。我们实验中的所有算法都是用c++实现的。

硬件配置：Intel Xeon处理器，2.6 GHz, 8gb内存。

### 2.Graph 集合

我们实验中使用的数据集见 表2。所有这些数据集都是公开的来自斯坦福快照网站e3，并按照指导建模图表。对于socket - epinions1和Email-EuAll数据集，我们在有限的计算资源下随机选择它们的子集来构造可用索引。因为这些图是未加权的，我们为每条边生成一个长度和一个权重，其值服从均匀分布在1到100之间。采用成本准则作为约束条件。



### 3.方法比较

由于这是第一项解决加密图上CSD查询问题的工作，所以我们将我们的这个方法与未加密图上的方法进行了比较。我们按照 [2] 中引入的最先进的纯文本图方法实现了这种方法。唯一的区别是我们在原始图上构造2HCLI，而不是覆盖图。因此，我们的这个方法的实现具有更高的查询效率，但与此同时，索引构造的复杂性会更高。

### 4.查询集

我们针对每个数据集随机生成 200 个查询。每个查询中的源 s 和目标 t 也是随机选择的。另外，每一对 (s, t) 的成本约束 θ 的设置如下所述：我们用cmin表示从s 到t的所有路径的最小成本的下界,用cmax表示从s到t的最短路径的的最小花费的上界。**如果成本约束 θ< cmin,那么将会没有有效的查询结果;如果成本约束 θ> cmax,那么最短的距离（shortest distance）将会是一直是有效的查询结果**，为了减轻 θ 对性能的影响,我们为每个查询随机选择 50 个 θ 值,这些值得区间为[cmin, cmax]。

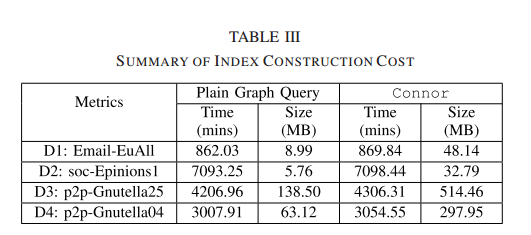
**另一个重要的参数是 α ,这决定了****近似保证 α-CSD查询。由于 α 对于所有查询是一个恒定常量,所以我们把它看作一个系统参数而不是特定查询的一部分。为了实现查询精度和系统效率之间的平衡,我们为所有查询设置了近似系数 α= 1.5**

## B.安全2HCLI和查询令牌的评估

### 1.索引大小和构造时间

图的索引构造是一次性的并且是离线计算的，这个过程包括两个步骤:

一、是构建无格式**2 hcli**,也就是与原始无格式**CSD**的索引构造相同,另一个是加密无格式**2 hcli**（这是本文的重点）,因此, ,我们考虑将第一步的输出作为未加密图的索引。索引大小和构建时间如表三所示。需要注意的是，不同数据集的索引大小和构建时间有很大的差异，这主要是由于图的拓扑的差异造成的。**与原来最短距离查询中任意两个顶点之间只有一条最短路径不同，在CSD查询问题中，任意两个顶点之间通常存在多个约束最短路径。从直观上看，密集图可能比稀疏图具有更高的索引构建成本。**



通常情况下，**每个加密索引的大小大约比对应的普通索引大6倍**。最重要的是索引的构建时间的比较：加密图略高于未加密图。因此，提高加密图上索引构建效率的关键是加快构建该图的无格式 2HCLI的过程。我们把这些尝试留到以后再做。

### 2.查询令牌生成

查询令牌的构造独立于特定的图形，我们现在分析查询令牌的大小和生成时间。查询令牌主要由5个元素,也就是： Sout,s, Tout,s,  
Sin,t , Tin,t , 和 Tθ. 前4个元素的长度都是16字节。**因为每个ORE密文的大小是16字节,一个深度为 dθ 成本树 Tθ 的大小为**

**字节。因此,查询令牌的总大小是。由于 dθ 是一个相对较小的值，所以查询令牌的大小通常是小于1 KB。不同dθ的查询令牌生成时间如表4所示**。虽然查询令牌生成时间在 dθ 的影响下显著增加，但是一般情况下时间花费适中(例如,当dθ小于等于6)。

