# 7、复杂性和安全性分析

本节介绍了提出的图加密模型Connor的复杂性和安全性分析。

## 复杂性分析（个人理解：图加密模型Connor组成：建立+查询。 各过程复杂度主要影响因素分析：建立算法时间和空间复杂度：O (nμ)，n-顶点数，μ-总草图；

## 查询算法过程：查询令牌生成+云服务端CSD查询。时间复杂度：O(ηdθ)，空间负责度： O ()。 -G每个顶点关联最大值，dθ-我不知道）

Connor主要由建立和查询算法组成，如算法2和3所述。

决定建立算法复杂度的主要因素是对图G生成的2HCLI进行加密 .让μ成为G中所有顶点的总草图,然后时间和空间复杂度都是O (nμ),其中n是G的顶点数。

查询算法包括用户端的a)查询令牌生成过程和b)云服务器端的CSD查询过程。让η为与G中的每个顶点关联的最大值。a)查询令牌生成过程的复杂度主要是由构建成本约束树决定，时间复杂度和空间复杂度都是O ()。b)对于CSD查询过程,获取Ls和Lt、执行成本约束过滤和进行距离计算的时间复杂度分别是O(η)，O(ηdθ)和O(η)；上面三个组件的空间复杂度分别是O(η)，O(η+ 2 dθ)和和O(η)。因此,CSD查询过程总的时间复杂度和空间复杂度分别是O(ηdθ)和O(η+ 2 dθ)。

## B .安全分析

我们现在介绍Connor的安全分析。为了清晰起见，我们首先讨论泄漏函数，然后证明了Connor在CQA2 安全模型下是安全的。

### 1)设置泄漏

我们构造的泄漏函数LSetup揭示了从图G安全2HCLI中可以推导出的信息，包括图n中顶点总数，所有示意图上的最大距离

B = maxu∈vmax {(V, du, V,Cu、V)∈△out(V, du, V,Cu、V)∈△in}du, V,

和的大小。更准确地说，的大小由和中草图实体的总数组成，分别用和表示。因此，泄漏函数LSetup = (n,B, , )。

注意，成对成本的顺序关系和成本与成本约束之间的顺序关系不包括在其中LSetup。因为每个实体的草图，加密后使用一个唯一的整数值进行XOR操作,这使每个实体在草图是无法区分的。

### 2)查询泄漏（个人理解：泄露函数组成：查询模式泄漏、草图模式泄漏、成本模式，分别将其形式化进行分析，模拟器S构造伪加密查询列表，敌手对于输出结果无法区分真假。）

我们构建的泄漏函数LQuery由查询模式泄漏、草图模式泄漏、成本模式泄漏组成。直观地说，查询模式泄漏揭示了以前是否出现过查询，草图模式泄漏揭示了与查询顶点关联的草图、两个不同草图之间的公共顶点以及查询顶点的草图的大小。成本模式泄漏揭示了1)成本之间的顺序关系，2)查询过程中成本与成本约束之间的顺序关系。我们将这些泄漏函数形式化如下。

定义4(查询模式泄漏):设q = (q1, q2，…， qm)是一个非空的查询序列。每个查询qi指定一个元组(ui、vi、θi)。

对于任何两个查询qi和q j,定义Sim(qi,qj) = (ui = u j, vi = v j,θi =θj), i.e., 即对于每个元素qj= (ui、vi、θi)是否成立。 然后，查询模式泄漏函数LQP (q)返回一个m×m(对称)矩阵，其中每个条目(i, j)等于Sim(qi, q j)。注意，LQP (q)不会泄漏查询顶点的标识。

定义5(草图模式泄漏):给定一个图G的安全2 HCLI 和查询q = (u, v,θ),素描泄漏模式函数LSP（，q)被定义为(∑,)。∑是一个列表,其中的每个元素都是与顶点相关联的草图,ϒ是一对(X, Z),其中X = h (v): (v,d,c)∈Iout和Z = h (v): (v d c)∈Iin是多重集合，h:{0,1}λ×{0,1}∗→{0,1}λ是一个特定的伪随机函数。

定义6(成本模式泄漏):成本约束查询中θ基本上可以由一定数量的均匀间隔构成。让dθ成本约束树Tθ(由第六节)的深度。与θ相关联的间隔((i−1)θ/ 2 dθiθ/ 2 dθ],当1≤≤2 dθ。分配每个区间μ列表。,第i个间隔和μi有关。 它存储这个区间的所有成本值。泄露间隔信息形式的Arr数组,其中第i个间隔是μi 元素(即。,Arr[i]=μi)。此外，假设z是被查询顶点的草图中的条目总数。对于每一对cost ci和cj，其大于、相等和小于的顺序关系分别可以表示为1、0和- 1。泄露的订单信息的成本是一个z×z(对称)矩阵∇每个条目(i, j)是1,0,或者−1。

因此,成本模式泄漏函数 Lcp (, q) = (Arr∇)。 因此，LQuery = (LQP(q)，LS P(， q)，LC P(, q))。

泄漏函数是在2HCLI上定义的，而不是在原始图上定义的。事实上，原始图的信息泄漏仅限于被查询的源-目标顶点的最小路径数。它可以定义为一个n×n(对称)矩阵，其中n是图中顶点的个数。其中的每个元素为NULL、0或正整数，表示不确定状态(拓扑结构得到很好保护)、断开连接或两个查询顶点的最小路径数。

对于2HCLI成本值，我们引入一个使用者放大系数φ来扩大明文和密文空间。因此，服务器不能仅从泄漏函数LC P(，q)所显示的订单信息来推断真实的成本值。对于2HCLI中的距离值，我们使用SWHE加密来保护它们的真实值不受服务器的影响。

定理1:如果加密原语g、h、ORE和SWHE是安全的，那么所提议的图形加密方案= (K eyGen、Setup、Query)是(LSetup、LQuery)-在自适应选择查询攻击情形下是安全的。

证明:关键思想是构造一个模拟器S。给定泄漏函数LSetup和LQuery, S构造一个伪加密的2HCLI结构△∗= {I\*out, I\*in}和一个查询\*列表。如果对于所有的PPT敌手A来说，他们无法区分这两款游戏的真假，我们可以说我们的图形加密方案(LSetup,LQuery)是安全的，可以抵抗自适应的选择查询攻击。

### 3) 仿真 （个人理解：对于敌手攻击，先从已有查询结果查询是否存在，有就用以前的生成，没有就根据定理公式生成）

S处理每一个顶点的ui(1≤i≤n)，生成一个基于泄漏函数LSetup在2HCLI中假的I out。S随机选择wi ui n 1 wi =,和样品Ii←{0,1}λ和ηi←{0,1}λ均匀无重复。对于所有的0≤i< wi, S会采取以下步骤来模拟每个素描:S计算lwi = h (li wi)和ηwi = h(ηi wi), h是一个特定的伪随机函数。然后，它通过计算v∗= h(K∗，v||0)来加密ui草图中的每个顶点v，其中K∗是一个伪密匙。它随机生成两个整数d和c，并使用SWHE和ORE方案对2N - d (N = 2B + 1)和c进行加密，得到密文D\*和C\*。

让=ηwi⊕(V∗| | D∗| | C∗)。S将W\*存在Iout\*索引中，I\*out显示在索引外。也就是说，I\*out [lwi] =。类似地，S生成一个伪I∗in，最后得到伪2HCLI∗= {I∗out, I∗in}。

模拟q∗：给定泄漏函数LQuery = (LQP(q)，LS P(△,q)，LCP(q))， S模拟查询令牌如下。S首先检查被查询的顶点S和t是否出现在之前的任何查询中。如果s之前出现过，s就会将S∗out s和T∗out 设置为之前使用的值。否则,它会T∗out =li和S∗out,s =ηi一些以前未使用的li和ηi。然后记得ηi之间的协会,李。查询顶点t，S会采取相同的步骤:设置S∗in,t和T∗in,t,类似地,将t与所选ηi和li联系起来。

模拟一个假成本约束树T∗θ,S首先检查如果θ出现在查询之前任何查询。如果θ出现，此前,S设置T∗θ以前使用的值。否则,构造一个完整的基于二叉树的θ和带有随机生成密钥的ORE模型加密每一个树节点。S返回这个加密树作为T∗θ。

S模拟查询过程，如下所示。给定查询令牌(S∗out,T∗out, s, S∗in,t, T∗in,t, T∗θ), S首先检查先前查询查询结果是否存在。如果是，S返回以前用作查询结果的值。否则，S检查被查询的顶点S(或t)之前是否被查询过。如果查询顶点s在之前的查询中出现，s将Ls\*设置为以前使用过的Lsp(f, q)的∑值。否则，对于新出现的顶点s, s将执行以下步骤:用s生成草图,首先初始化一组Ls∗年代和计数器ω∗= 0。(细节我也不太懂)

然后,它迭代计算T∗out,s,v = h (T∗out,s,w∗)和S∗out,s,v = g (S∗out,w\*),并添加元组(V∗, D\*s,d, C∗s,d)到L∗s,直到I∗out(T∗out, s, d)不存在,在那里( V∗s,D\*s,v, C∗s,v) =I∗[T∗out, s, d]⊕S∗out,s,v。类似地，S获取顶点t的集合 Lt。在获得L\*s和L∗t, S执行成本约束过滤操作基T∗θ得到候选集Y∗。该定理由SWHE的CPA-security得到，也就是说，S基于Y\*执行SWHE计算并返回查询结果。

由于加密原语g、h、ORE和SWHE是安全的，所以伪的2HCLI结构\*和查询序列q与真实的无法区分。因此，对于所有的PPT敌手A来说，他们无法区分这两款游戏是真实的还是理想的。因此,我们有

|Pr(Real ii, A(λ)= 1)−Pr(Ideal,A, S(λ)= 1]|≤negl(λ), 其中negl(λ)是一个可忽略的函数。