K-Q:支持海量查询的隐私泄露检测算法

林永妍，宋　玲，陈玉婵
（广西大学 计算机与电子信息学院，南宁 530004）

摘　要：针对推理检测需要的所有历史查询结果的数据规模较大这一问题，K-Q 算法结合 K-匿名模型在历史查询结果的存储规模上进行了优化，通过推理攻击模拟算法在线检测恶意查询。在真实数据集上的实验证明了 K-Q 算法可以自适应于查询规模的增长，在准确率和内存消耗上都明显优于已有的直接基于相关元组合并优化的 T-D 算法。

关键词：K-匿名；数据共享平台；隐私泄露检测；推理攻击
中图分类号：TP309　　**文献标志码**：A　　**文章编号**：1001-3695(2013)12-3767-04
doi:10.3969/j.issn.1001-3695.2013.12.064

K-Q:algorithm for privacy disclosure detection supporting large number of queries

LIN Yong-yan, SONG Ling, CHEN Yu-chan
(*School of Computer, Electronics & Information, Guangxi University, Nanning* 530004, *China*)

Abstract: A key problem remained that the data set required to detect inference attack cannot all fit in memory, K-Q algorithm optimized the real data storage for each history query based on K-anonymization model, it detected the illegal query online through simulating the real inference attack. Experiments on real data demonstrate that K-Q algorithm can scale on query size, and perform on detect accuracy and memory consumption is better than the existed T-D algorithm which directly merge related tuples and also assure the privacy control's granularity.

Key words: K-anonymity; data sharing platform; privacy disclosure; inference attack

0　引言

信息时代的不断发展，网络上的空间数据多种多样，用户可以很随意地查询这些数据，获取自己感兴趣的信息。在相对开放的数据共享平台中，大多数用户操作体现为对后台数据的查询，因此大量的数据查询是可以预见的，数据的查询结果往往是公开的。对于比较简单的恶意查询，例如直接查询一些被数据拥有者界定为隐私数据的部分或全部数据字段时，对返回数据进行一定的泛化和抑制即可防止隐私泄露[1]；但一些潜在的恶意用户往往通过一些高级隐私窃取手段如共谋欺诈和链接攻击等来获取海量数据中潜在的隐私数据。

目前在数据隐私保护研究中，Samarati 和 Sweeney 提出的 K-匿名模型被证明是最简单也是最为有效的防隐私泄露技术。通常 K-匿名要求对于任何一个投影到这些属性上的元组，必须同时存在至少 $K-1$ 条其他记录满足在这个属性集上完全相同于该元组，这 K 条记录组成一个等价组[2]，从而使个体隐藏在 K 条数据之中，而无法确定 K 条数据中的具体哪一条记录是该个体对应的记录，这样对自由访问型的数据[3]仍能达到隐私保护的目的。现有的基于 K-匿名模型的算法虽然可以通过对单次查询返回数据进行严苛的匿名和泛化[4]来避免任何可能的推理攻击，但会存在两个很严重的问题：a）数据质量无法保证，在 K-匿名模型中，K 值越大，隐私保护程度越高，但数据可用性越差；b）推理检测计算的效率问题，由于 K-匿名已被证明为 NP-hard[5]，计算效率较低，因此难以适应于大规模的查询和数据的生成环境。为了保证数据质量和查询响应的及时性，T-D 算法摈弃了匿名模型，为了防止多次查询可能存在的链接攻击，它基于推理攻击的原理对当前查询获得的数据和全部的历史查询所获得的数据进行链接计算，来分析和评估当前查询是否存在隐私泄露的可能。类似的方法[6,7]往往需要对历史查询及其输出进行存储，以确保新的查询到来时能进行全局性质的数据级推理，但存在的问题是会耗费较大的存储空间，即使使用了一些优化机制如数据压缩、索引等仍然会耗费较多的系统资源，无法保证查询响应的快速性。而且这类方法对隐私安全程度的控制粒度不够，保护级别比较单一，而 K-匿名模型可以通过调整 K 值大小来控制隐私保护的粒度和级别。在大规模数据查询中，如何更好地利用历史查询结果来检测当前查询是否存在隐私泄露，如何保证隐私泄露检测的及时性和返回数据的质量是解决海量数据和查询环境下的防推理攻击问题的重点。

针对上述存在的问题，本文提出基于 K-匿名模型的查询概要结构 KData，通过其维护历史查询及其返回数据的摘要信息，既满足推理分析的数据需要，又能避免获取全部历史数据的代价；使用图状结构 KGraph 来维护历史查询及其返回数据的概要信息，同时维护历史查询之间的依赖关系，有利于新增查询的链接分析；在此基础上，提出一种基于 K-匿名的隐私泄

收稿日期：2013-02-28；**修回日期**：2013-04-12

作者简介：林永妍(1985-)，女(壮族)，硕士，主要研究方向为高性能计算与网络系统(119729508@qq.com)；宋玲(1967-)，女，教授，主要研究方向为网络信息安全；陈玉婵(1987-)，女，硕士，主要研究方向为计算机应用.

露检测算法(K-Q)来判断当前查询和历史查询的随机组合所带来的隐私泄露威胁的可能性。

1 相关工作

K-匿名模型被 Sweeney 提出以来,大量的隐私保护工作都是基于这一模型进行扩展和改进,对于这方面的研究也取得了一定的成果。2001 年 Sweeney 提出了 Datafly 算法,对不满足 K-匿名要求的属性进行泛化,直到所有的属性满足 K-匿名要求才停止泛化过程,Datafly 算法能够防范潜在的链接攻击威胁。文献[8]提出了一种新的匿名化原则,该原则要求在发布的数据满足 K-匿名化原则的同时($k>1, p\leqslant k$),还要求同一等价类中的记录至少出现 p 个不同的敏感属性值。该匿名化原则在某些数据集上可能会带来很大的信息可用性损失,也不足以抵抗敏感属性值的偏斜性攻击和相似性攻击。Xiao 等人提出一种新的匿名方法 Anatomy。该方法将一个数据表分成两个数据表进行发布,其中一个数据表包含准标志符属性和分组号,另一个数据表包含敏感属性值和分组号,两个数据表通过分组号联系在一起。(ε, m)-anonymity 算法中,控制相似度的阈值 ε 是固定的,所以不能很好地防止近邻攻击。为解决同质性攻击和背景知识攻击所带来的隐私泄露,Machanavajjhala 等人[9]提出了 L-多样性(L-diversity)原则,它要求准标志符属性行所对应的敏感属性集必须包含 L 个不同的元素,它能保证通过链接攻击泄露概率不超过 $1/L$。基于 K-匿名模型的经典算法在一定程度上有效地解决了数据共享的安全性,但是仍然无法防止恶意用户利用多次查询结果进行链接分析推导出一些涉及个人隐私的信息。为了克服利用查询结果产生的隐私泄露问题,文献[10]提出了一种基于相关元组合并优化的推理检测算法(T-D)。它通过不同的查询返回结果查找相关元组来检测敏感信息是否泄露,一旦检测出当前查询存在隐私泄露,则立即拒绝当前查询。算法中如果两个元组相互关联,那么将被合并成一个元组,从而大大压缩了查询历史记录。随着查询数量增加,该算法几乎可以推断出所有数据的原始关系,算法检测出隐私泄露的准确率也相应提高。但是该算法面对大量查询和复杂数据时需要消耗大量的系统资源,且需要手动限制查询数量才能保证算法的有效性。

2 问题描述

基于推理攻击的敏感信息泄露可表述为

$$\mathrm{Diff}(D,SD)=P(D\cup SD)-(P(D)\cup P(SD)) \tag{1}$$

其中:D 为存储于数据库中的原始关系数据,其中包含大量用户隐私数据;SD 为大量用户通过大规模检索后得到的共享数据;P 函数接收数据集输入,输出为符合隐私安全要求的数据集。式(1)表明当用户通过查询得到经过隐私处理的用户数据 D 时,在与已有的共享数据 SD 进行联合分析 $P(D\cup SD)$ 后,就能得出既不在 D 中又不在 SD 中的隐含数据 Diff(D, SD),而这份数据很可能包含隐私数据。本文提出的 K-Q 算法就是实现大规模数据查询下的 Diff 计算,同时保证较高的检测准确率和性能。下面给出了隐私泄露的例子。表 1 给出了一部分用户的个人信息数据,其 schema 为 person(name, birth, race, gender, job, salary, ZIP, disease),敏感字段为 disease,标志字段为 name。若〈name, disease〉在返回数据集中是唯一的则产生隐私泄露。初始 SD 为空,接受两个用户查询 Q_1(〈salary, disease〉)后 $P(SD)$ 存储为{表 2}。新的查询 Q_2(〈name, salary, ZIP〉, NULL)到达后,D 为 Person 的投影 $\Psi_{\text{name, salary, ZIP}}$person, $P(D)$ 如表 3 所示。根据式(1),$P(D\cup SD)$ 的输出中包含了〈Joy, short breath〉〈Ivy, chest pain〉这些隐私信息,造成了隐私泄露。

表 1 个人信息 Person 表

name	birth	job	salary	ZIP	disease
Joy	1987	teacher	5000	13140	short breath
Chloe	1987	doctor	8000	13242	AIDS
Jean	1986	doctor	8000	13560	obesity
Ivy	1986	doctor	5800	13560	chest pain

表 2 $P(D_{Q_1})$

salary	disease
5000	short breath
8000	AIDS
8000	obesity
5800	chest pain

表 3 $P(D_{Q_2})$

name	salary	ZIP
Joy	5000	13140
Ivy	5800	13560

在生产环境中,SD 会由大量查询的返回数据集组成,针对新查询的隐私检测会在数据集上进行大量的 join 计算,而数据集 SD 由于规模较大往往是存储在磁盘上的,计算开销较大。本文通过对 SD 的精简,使得检测计算可以适应于内存规模,同时又能保证隐私泄露检测的正确性和可行性。

3 K-Q 算法

3.1 基于 K-匿名模型的推理检测

推理检测相关的术语如下:

定义 1 隐私表(private table, PT),遵循关系型数据模型,其属性集合为 $A=\{A_1, A_2, \cdots, A_m \mid |A|=m\geqslant 1\}$。$A$[PT]表示为 PT 的属性集合,$t[A]$ 表示为遵循属性集合为 A 的记录集合。PT 主要由三类属性构成,即标志符属性(IA)、一般属性(GA)和敏感属性(SA)。

定义 2 join 算子(∞)。其为二元算子,输入左值与右值均为 PT,表现为 $\mathrm{PT}_1\infty\mathrm{PT}_2$。当 $t[A[\mathrm{PT}_1]\cap A[\mathrm{PT}_2]]$ 不为空时,$\mathrm{PT}_1\infty\mathrm{PT}_2$ 输出不为空。

定义 3 unique 算子(uniq)。其为一元算子,表现形式为 uniq(PT)。基本输入为 PT,同时参照两个计算参数 IA 集合 IAS 和 SA 集合 SAS。计算结果为 $t[\langle \mathrm{IAS}, \mathrm{SAS}\rangle]/t[A[\mathrm{PT}]]$,即标志符属性和敏感属性所标志的投影记录在整个 PT 中的分布比例情况,输出结果可以看做属性集合是〈IAS, SAS, K〉的 PT。其中 K 为 $t[\langle \mathrm{IAS}, \mathrm{SAS}\rangle]$ 某个具体记录值的计数值。

根据定义 3,当 uniq(PT) 的输出结果中的某条记录的元组 K 值为 1 时,即代表发生了隐私泄露,正如问题描述中的例子所示,uniq($P(\mathrm{DQ}_1)\infty P(\mathrm{DQ}_2)$)出现了两个值为 1 的情况。因此,根据以上定义,推理检测模型如式(2)所示:

$$\mathrm{LEAK_STATUS}=\begin{cases}\mathrm{TRUE} & \sigma_{k=1}(\mathrm{uniq}(\mathrm{PT}_0\infty(\mathrm{PT}_i\infty\mathrm{PT}_{i+1}\infty\cdots\infty\mathrm{PT}_j)))\neq\mathrm{NULL}\\ \mathrm{FALSE} & \sigma_{k=1}(\mathrm{uniq}(\mathrm{PT}_0\infty(\mathrm{PT}_i\infty\mathrm{PT}_{i+1}\infty\cdots\infty\mathrm{PT}_j)))=\mathrm{NULL}\end{cases} \tag{2}$$

其中:PT_0 代表待检测数据集,$\{\mathrm{PT}_i, \mathrm{PT}_{i+1}, \cdots, \mathrm{PT}_j\}$ 代表与 PT_0 进行 join 计算的历史数据集。当对其 uniq 输出进行 select 过滤时,若 $K=1$,则代表发生隐私泄露;若 $K\neq 1$,则代表没有隐私泄露。

T-D 算法所依据的泄露检测模型为式(2),其存在的问题是,当历史查询规模较大时,会产生很多冗余的 join 计算。判断 LEAK_STATUS 是否 TRUE 的依据是 uniq 结果中的 K 值,只有 $K=1$ 的情况是有意义的,因此对于 join 计算来说,$K\neq 1$ 的记录即使参与 join 计算,最后的 K 值仍然不为 1,对这些记录的 join 是冗余的。此外,式(2)所限定的隐私保护级别只有

$K=1$ 和 $K\neq1$ 两种情况,在实际情况下 K 处于较低的值域时,即使不等于 1 仍能产生隐私泄露。因此,本文提出了结合 K-匿名的检测模型,如式(3)所示:

$$\text{LEAK_STATUS} = \begin{cases} \text{TRUE} & \sigma_{k\leq\lambda}(\text{uniq}(\text{KPT}_0 \infty (\text{KPT}_i \infty\ \text{KPT}_{i+1} \infty\ \cdots \infty\ \text{KPT}_j))) \neq \text{NULL} \\ \text{FALSE} & \sigma_{k>1}(\text{uniq}(\text{KPT}_0 \infty (\text{KPT}_i \infty\ \text{KPT}_{i+1} \infty\ \cdots \infty\ \text{KPT}_j))) = \text{NULL} \end{cases} \tag{3}$$

其中:λ 值为隐私保护阈值;KPT 根据查询返回 PT 生成;属性集合增加了 K 值代表相应记录在整个 PT 中的计数;$t[A[\text{KPT}]]$ 只保留了 $t[A[\text{PT}]]$ 中满足 K 值小于 λ 的记录。

根据式(3),当 uniq 返回的 PT 中存在 K 值小于 λ 值时即为产生隐私泄露,隐私保护系统可以根据实际情况来调整 λ 值以达到控制保护粒度的目的。λ 值越大,保护级别越高,延续了 K-匿名模型的特性。其正确性和有效性证明如下:

证明　当 $\lambda=1$ 时,即转换为式(2),因此该检测模型的正确性可以保证。根据式(2),在隐私泄露检测计算中大量的负载来自于 join 计算,其复杂度为 $O(m\times n)$,m 和 n 为两个 PT 的记录条数,整体 join 复杂度为 $O(m\times n\times(J-I))$。Uniq 的复杂度为 $O(p_2)$,p 为输入 PT 的记录条数,因此隐私泄露检测的复杂度为 $O(m_2\times n_2\times(J-I))$,可以简化为 $O(N_4\times(J-I))$。当数据规模增长到一定情况时,$J-I$ 可以视做常数,复杂度主要与参与 join 计算的 PT 规模 N 有关,而式(3)所提出的检测模型在融入 K-匿名的思想后,只保留输入 PT 中会影响最终 LEAK_STATUS 的记录,减小了 N 值,因此其性能肯定优于依据式(2)所提出的 T-D 算法。

3.2　核心数据结构

根据式(3)所提出的结合 K-匿名的推理检测模型所设计的核心数据结构如下:

定义 4　KData,主要用于建模式(3)中的 KPT,其结构为 $\{R,T,V,K\}$。其中:R 代表数据集的元数据,T 代表 R 上的属性组合,V 代表符合 T 的数据实例,K 代表 V 在数据集中的出现次数。KData 主要对应 complete 和 reduced 两种状态。前者对应于输入查询所返回的原始数据集,K 值为数据实际情况;而后者是对处于 complete 状态的 KData 进行压缩,过滤掉 K 值超过预定义的保护上限 λ 的记录。

KData 的计算主要通过对数据集 R 上的组合计算而得到,每一个 R 的子集都对应着多条 KData 元组。以表 1 为例,R = person,假设查询为 SELECT * FROM person,其 KData 为 KD_person,其所有元组中当 T 为〈birth〉时的元组有{person,〈birth〉,1987,5},{person,〈birth〉,1986,3}。

定义 5　KGraph,主要用于建模 KData 之间的连接关系,即历史查询之间的语义关系,体现为连接各个 KData 节点的图,主要包括节点集合 $\text{KN}=\{\text{KN}_1,\text{KN}_2,\cdots,\text{KN}_m\}$ $(m=|\text{KN}|)$ 和边集合 $\text{KE}=\{\text{KE}_1,\text{KE}_2,\cdots,\text{KE}_n\}$ $(n=|\text{KE}|)$,KE_i 连接两个 KN_j。若两个 KData 的属性集合有交集,则构成一条边 KE_i。

KGraph 主要有四种状态,即空状态 EMPTY、历史状态 INACTIVE、激活状态 ACTIVE 和更新状态 UPDATE,主要对应 KGraph 在演进过程中所体现的不同形态。

1)EMPTY　KGraph 的初始化状态为空,即没有对应于历史查询的 KData 节点。

2)INACTIVE　只包含 reduced KData 节点,每个节点对应一次历史查询的返回数据。

3)ACTIVE　比 INACTIVE 状态多了一个 complete KData 节点,该节点保留了所有的记录。当 KGraph 处于当前状态时,会激活推理检测机制。

4)UPDATE　当推理检测计算完毕后,ACTIVE 状态会变更为 UPDATE 状态。它主要包括两个操作,即用户响应和 KData 更新。用户响应是指根据推理检测结果对输入 KData 中的返回数据集进行匿名和泛化后再返回给用户;KData 更新指将 complete KData 更新为 reduced 状态。

图 1 给出了例 1 所对应的 KGraph 状态转换。

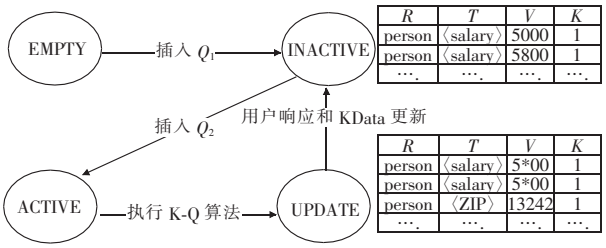

图 1　插入 Q_1 和 Q_2 后的 KGraph 状态转换

3.3　K-Q 算法实现

基于式(3)所提出的 K-匿名推理检测模型而实现的 K-Q 算法核心步骤如下:

a)根据新的输入查询生成 complete KData 节点插入到已有的 KGraph 中,并在 KGraph 中搜索可与之进行 join 计算的 reduced KData 节点,生成新的边。KGraph 由 INACTIVE 变为 ACTIVE。

b)根据系统设定的敏感词集合 $\text{SW}=\{W_1,W_2,\cdots,W_n\}$ 在 KGraph 上搜索属性集合中与 SW 有交集的 KData 节点,得到输出 $\text{SW-WN}=\{(w_1,\text{WN}_1),(w_2,\text{WN}_2),\cdots,(w_n,\text{WN}_n)\}$。

c)任意选取 SW 的两两字段 W_x、W_y 进行计算,分别选取 SW-WN[W_x]和 SW-WN[W_y]任意两个节点 N_1 和 N_2,构建其最短路径 SP,然后在 SP 上所有 KData 节点进行逐一 join 计算,最后再进行 unique 分析。若 uniq(SP) $<\lambda$,则证明隐私泄露发生,跳到 e)。

d)重复步骤 b),直至遍历完所有单词组合。

e)更新 complete KData 为 reduced 状态。KGraph 由 ACTIVE 变为 UPDATE。

其伪代码如下:

```
输入:INACTIVE KGraph,查询 query,敏感词 SW,K 值上限 λ。
输出:UPDATE KGraph。
a)根据 Query 生成 complete KData 更新到 KGraph 中,KGraph 状态变为 ACTIVE
b)在 KGraph 中根据 SW 搜索对应的 SW-WN
c)foreach word pair WP in SW
d)   SP = KGraph. disjikaPath(WP)
  //根据单词对搜索最短距离 SP
e)   if uniq(join(SP)) == TRUE
f)   return//报告产生隐私泄露,更新 complete KData
g)更新 complete KData
  //即使没有产生隐私泄露,也要更新以对用户进行响应
```

由于 K-Q 算法完全基于式(3)所提出的 K-匿名推理检测模型,因此其正确性和有效性已在证明 1 中予以论证。实质上算法中的一些搜索操作如步骤 b)中的 SW-WN 和步骤 c)中的 SP 都可以进行预处理以提高 K-Q 算法的实时性能,当新的查询节点与已有 SP 中的节点进行 join 计算时,直接将其插入到 SP 即可,因为参与 join 的节点的先后顺序并不会影响 join 的

最终结果。步骤 e)中的匿名和泛化操作不需要耗费较大开销，K-Q 算法可追踪产生隐私泄露的 KData 节点的具体记录，因此只需对该记录进行匿名和泛化就可以避免推理攻击的威胁，同时也保证了数据质量，做到数据可用性和隐私保护的平衡。

4 实验与结果分析

对本文提出的 K-Q 算法进行了实验验证，主要考察算法的性能，如算法的准确率、对内存的消耗、K 值对算法的影响等。实验采用 UCI 机器学习数据库中的 adult 数据集来验证算法的性能，这个数据集在数据隐私保护领域被广泛使用，数据集的每个元组都描述了一个美国人的个人信息。实验运行环境：Intel® Core™ i3 CPU M330 2.13 GB，2.00 GB RAM Windows 7，MyEclipse 6.5。

4.1 准确率

本实验是将 K-Q 算法与文献[10]提出的基于相关元组合并优化的推理检测算法(T-D)，在相同查询语句下，对准确率进行对比。K-Q 和 T-D 算法的相同点是两种算法都是利用历史查询结果来分析当前查询是否存在隐私泄露；不同点在于，前者利用 K-匿名思想和查询图对查询结果进行分析，后者则是利用相关元组和表结构。查询次数和准确率如图 2 所示，可知，随着查询次数的增多，两种算法对查询结果的链接与匹配越来越接近原始数据表，因此准确率都相应提高，但 K-Q 算法明显优于 T-D 算法。

4.2 内存消耗

4.2.1 K-Q、F-Q 和 T-D 算法的对比

为了体现 K-Q 算法对内存消耗的优化，实验中用到了 K-Q 算法的一个简化版本，称为 F-Q，它没有采用 K-Q 的基于 K-匿名的内存优化策略，只是简单地将每次查询的返回结果全部存储下来再进行链接分析。查询次数与内存消耗如图 3 所示。图中横轴为查询次数，纵轴为内存消耗的 logarithm 值，底为 10。每一次查询都会增大历史查询记录的规模，从而增加了内存负载。由图 3 可看出，K-Q 算法在相同条件下对内存的消耗远远少于 T-D 和 F-Q 算法。这是因为 K-Q 算法中的查询图只存储节点的相关信息和满足 K-匿名分析的信息，大大降低了内存消耗；T-D 算法利用表结构存储元组的相关信息，需要把查询返回结果的绝大部分信息存储下来进行元组匹配，它在一定程度上压缩了查询历史记录，不过依然占据了大量内存，但仍优于 F-Q 算法。

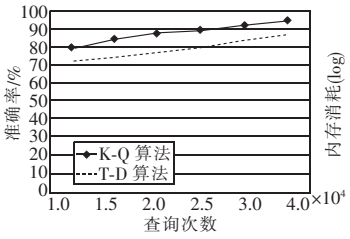

图 2 查询次数与准确率

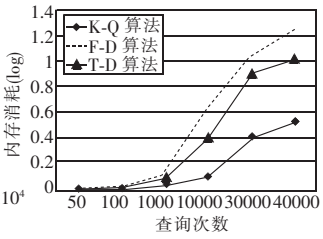

图 3 查询次数与内存消耗

4.2.2 K 值与内存消耗(图 4)

K-Q 算法利用 K-匿名思想过滤掉一些满足 K-匿名要求的属性组合，查询图中的节点只存储满足 K-匿名分析要求的信息，大大减少了内存消耗。本实验对 K-Q 算法的 K 值性能进行验证，在查询数量和查询条件相同的情况下，变化 K 约束条件，查看 K-Q 算法对内存的消耗情况。由图 4 可知，随着 K 值的不断增大，K-Q 算法的内存消耗也随着增大。因为 K 值越大，不满足 K-匿名要求的属性组合越多，每一个查询节点所存储的信息就越多，相应地，K-Q 算法对内存消耗也就越多。

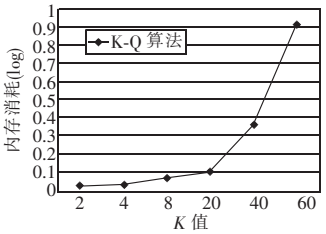

图 4 K 值与内存消耗

5 结束语

本文主要针对开放的数据共享平台如互联网、社交网络等大规模查询下是否存在隐私泄露问题，提出了一种基于 K-匿名的隐私泄露检测算法(K-Q 算法)来判断当前查询是否存在隐私泄露的可能。为了使算法更优，使用图状结构来存储每一次的查询返回结果，以方便下一次的查询链接分析需求，并且结合 K-匿名模型的优势，节省了大量不影响最终推理检测结果的历史冗余数据，间接减小了参与推理检测的数据输入规模。同时提供了多粒度可量化的隐私保护控制机制，提高了系统的可控性。实验证明了该算法的可行性和高效性。

接下来的工作是考虑对该算法中一旦查询结果有产生推理攻击或合谋攻击的可能性时，泛化对查询结果中造成合谋攻击可能性的字段值，采用何种泛化技术使数据损失度尽量减小，进一步提高 K-Q 算法检测出隐私泄露的准确率以及优化 K-Q 算法的效率。

参考文献：

[1] 焉凯，何贤芒. 基于局部聚类的数据匿名化算法[J]. 计算机应用研究，2012，29(1)：148-151.

[2] LI Tian-cheng，LI Ning-hui. On the tradeoff between privacy and utility in data publishing[C]//Proc of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York：ACM Press，2009：517-526.

[3] LI Tian-cheng，LI Ning-hui，ZHANG Jian. Modeling and integrating background knowledge in data anonymization[C]//Proc of the 25th IEEE International Conference on Data Engineering. Washington DC：IEEE Computer Society，2009：6-17.

[4] 王平水，王建东. 匿名化隐私保护技术研究进展[J]. 计算机科学，2010，27(6)：2016-2019.

[5] 王平水，马钦娟. 隐私保护 K-匿名算法研究[J]. 计算机工程与应用，2011，47(28)：117-200.

[6] YIP R，LEVITT K. Data level inference detection in database systems[C]//Proc of the 11th IEEE Computer Security Foundations Workshop. 1998：179-189.

[7] BRODSKY A，FARKAS C，JOJODIA S. Secure databases：constraints，inference channels，and monitoring disclosures[J]. IEEE Trans on Knowledge and Data Engineering，2000，12(6)：900-919.

[8] TRUTA T M，VINAY B. Privacy protection：P-sensitine K-anonymity property[C]//Proc of the 22nd International Conference on Data Engineering Workshops. Washington DC：IEEE Computer Society，2006：94-98.

[9] MACHANAVAJJHALA A，GEHRKE J，KIFER D，*et al*. L-diversity：privacy beyond K-anonymity[J]. ACM Trans on Knowledge Discover from Data，2007，1(1)：3-16.

[10] CUI Bin-ge，LIU Da-xin. An inference detection algorithm based on related tuples P-sensitive K-anonymit property[C]//Proc of the 22nd International Conference on Data Mining. 2005：1011-1017.