数值型敏感属性的近邻泄露保护方法研究 *

陈伟鹤, 屈洪雪, 邱道龙
(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘 要: 针对在发布数值型敏感属性数据时,因同一分组中个体的敏感属性值之间过小的差异而导致攻击者可以较高的概率以及较小的误差推导出目标个体的敏感信息,从而出现近邻泄露问题,提出了一种有效的防止近邻泄露的模型:(ε_p, l)-anonymity。该模型根据不同的敏感属性值区间设置不同的阈值 $\varepsilon_i (1 \leqslant i \leqslant p)$ 控制敏感属性值之间的相似度,并采用有损链接的方法对隐私数据进行保护。实验结果表明,该方法可以明显减少近邻泄露,提高信息可用性,增强数据发布的安全性。

关键词: 数据发布; 数值型; 有损连接; 可用性; 近邻泄露; (ε_p, l)-anonymity
中图分类号: TP311 **文献标志码**: A **文章编号**: 1001-3695(2012)02-0650-05
doi:10.3969/j.issn.1001-3695.2012.02.066

Research on preservation of proximity privacy of numerical sensitive data

CHEN Wei-he, QU Hong-xue, QIU Dao-long
(*School of Computer Science & Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu* 212013, *China*)

Abstract: Proximity breach occurs when an adversary may have high confidence to infer the victim's value fall in a short interval in publishing numerical sensitive data. In view of such proximity breach in data publishing when sensitive values were numerical, this paper proposed a model called (ε_p, l)-anonymity to prevent proximity breach on the idea of lossy join. To control similarity of sensitive values in different ranges, this model set different threshold values $\varepsilon_i (1 \leqslant i \leqslant p)$. The results of experiments suggest that the model is able to reduce proximity breach apparently and enforces security of data publishing with a better utility.

Key words: data publishing; numerical; lossy join; better utility; proximity breach; (ε_p, l)-anonymity

0 引言

研究表明,通过邮编、性别、出生日期等属性的链接操作,超过87%的美国公民的身份可以被唯一标志[1]。因此,数据发布过程中的隐私泄露问题成为了新的研究热点。多种数据隐私保护技术应运而生,其中最具代表性的有 k-匿名技术[2]和 l-多样性技术[3]。

K-匿名和 l-多样性技术以及其改进算法[4~6]在一定程度上阻止了隐私泄露。但是目前出现的另一种隐私泄露问题——近邻攻击[7]对个体的敏感信息构成了新的威胁。该类攻击主要发生在敏感属性值为数值型的数据中。当敏感属性值之间的差异程度很小时,攻击者可以较高的概率将个体的敏感属性值限定在较小的范围内,即使攻击者不能准确推导出目标个体的精确信息,但这也在很大程度上泄露了目标个体的隐私信息。尽管传统的 k-匿名[2]和 l-多样性[3]技术在解决链接攻击问题上取得了一定的成果,但不能很好地解决近邻攻击。所谓链接攻击,是指攻击者可以通过准标识符属性的链接将目标个体的身份推导出来。例如,某公司要将其员工的工资记录(如表1所示)用于研究,这些员工的工资记录称为原始数据。其中,工资(salary)为敏感属性,年龄(age)和邮编(zipcode)是准标识符属性(QI)。因为在链接攻击中,通过准标识符属性的链接可以将员工的身份推导出来。泛化[1,8]是最常用的阻止链接攻击的方法。该方法将所有的原始数据划分到若干分组里(准标识符组 QI-group),每个分组中的准标识符属性值具有相同的形式。表2显示的就是表1中的原始数据经过泛化后得到的发布数据。通过分组号(group-ID)可以显示泛化产生了7个分组。从表2可以看出,如果攻击者知道 Andy 在已发布的数据表中的第一个分组里,而 Andy 所在的分组中有三种敏感属性值,分别为1000、1010、1020。尽管攻击者不能准确地得出 Andy 的工资,但攻击者可以很容易地推导出 Andy 的工资在1010元上下,而且上下浮动很小。尽管上述分组也满足 l-多样性,但由于敏感属性值之间的差异太小,导致攻击者以较高的概率推导出目标个体的敏感信息落在某一较小的区间内。

1 相关工作

在数值型敏感属性的隐私保护数据发布中,即使分组中的敏感属性不相同,但当敏感属性值相近时也会造成隐私泄露。如表2所示,若攻击者推断出目标个体的年龄在17到24岁之间,而且存在于已发布的数据表中,尽管攻击者不能准确地推

收稿日期: 2011-08-10; **修回日期**: 2011-09-13 **基金项目**: 江苏省教育厅自然科学基金资助项目(09KJB520003);江苏大学高级人才启动基金资助项目(07JDG031)

作者简介:陈伟鹤(1974-),男,江苏常州人,副教授,博士,主要研究方向为数据库安全、模型检测;屈洪雪(1985-),女,硕士研究生,主要研究方向为隐私保护(quruixue945@163.com);邱道龙(1988-),男,硕士研究生,主要研究方向为信息安全.

导出该目标个体的具体工资,但却能很容易地推导出该人的工资在1000元左右。因此,可以说 l-多样性[3]是最早解决近邻泄露问题[7]的方法。只不过该算法解决的是近邻泄露问题的一个特例,即同一分组中的敏感属性值之间的差异度为0或者说相似度为1,但该算法没有考虑差异度较小的情况。Zhang等人[9]采用的方法是:同一分组中至少有 k 个不同的敏感属性值,且保证分组中敏感属性值之间的最大值和最小值之间的差值至少为 e。显然,该算法仍会导致近邻泄露问题。LeFevre等人[10]则提出变量控制的思想:设定一个阈值 t,要求分组里面的敏感数值的差异性至少为 t。但是,无论这个阈值 t 取多少,分组仍然可能遭受近邻攻击。例如,如果一个分组中含有 $|G|$ 个元组,若 $|G-1|$ 个元组的敏感属性值都为 v,其余的敏感属性值与 v 有充分大的差异性,那么 $|G-1|$ 个元组仍然有概率为 $(G-1)/G$ 的近邻泄露风险。另外,Li等人[11]也作了相应的研究,提出了 t-closeness 算法:设整个数据表 T 的分布为 f,分组中敏感属性值的分布为 f_G,要求分组中敏感属性值的分布与整个数据表 T 的敏感属性值分布不超过阈值 t,即 $\mathrm{EMD}^{[12]}(f, f_G)<t$。该算法同样存在局限性。首先,EMD在测量近邻泄露风险时是不可靠的,并且不适用于阻止数值型敏感属性的链接攻击;其次,由于原始数据表的分布未必是安全的,导致即使分组中敏感属性值的分布与整个数据表中的敏感属性值分布相同,也未必能够很好地进行隐私保护。以上文献虽然主要是针对同一分组中敏感属性值出现的频度进行的研究,却为近邻泄露问题的提出和解决奠定了基础。2008年,香港中文大学的LI Jie-xing等人在文献[4]中正式提出了近邻泄露这一概念,并且针对近邻泄露的问题提出了 (ε,m)-anonymity 算法。该算法解决的是数值型敏感属性的近邻泄露问题,主要思想为:假设分组 G 中的任意一个元组,若其敏感属性值为 x,那么 G 中最多有 $1/m$ 的敏感属性值和 x 相似,这个相似度由阈值 ε 来控制。该文献采用的匿名原则是 Mondrian algorithm[13],但该匿名算法仍然是基于对准标志符属性的泛化进行的隐私保护,信息损失率较高,并且算法中的阈值 ε 是固定不变的,这样仍然可能会导致近邻泄露问题。例如,将阈值 ε 设定为500,则由表3可以看出,数据表中的三个分组都满足敏感属性值之间的差距 ε 至少为500,但并不能说明任何一个分组都不会产生近邻泄露。例如分组1中的敏感属性值的取值范围在区间[1000,3000]内,此时取 $\varepsilon=500$ 来控制敏感属性之间的相似度是可以接受的,但对于分组2,该分组中的敏感属性值之间尽管满足差异度至少为500的条件,但由于敏感属性值本身的取值较大,$\varepsilon=500$ 显然已经不能很好地控制该分组中的敏感数值之间的差异度。由此可以看出,分组2中的敏感属性值集中在9000上下,且上下浮动尽管超过阈值500,但相对于敏感属性值本身的大小,阈值仍然很小,以至于不能很好地控制分组中敏感属性值之间的相似度。所以,将控制相似度的阈值 ε 设置为定值是不符合客观实际情况的。此外,文献[14,15]也对近邻泄露问题进行了研究,但仅仅是在理论上进行了尝试,并没有提出完整的算法,也没有实验数据的支持。

以上研究在解决数值型敏感属性数据发布中出现的近邻攻击问题上都存在一定的局限性,不能很好地防止近邻泄露。所以,本文在前人的基础上,基于有损链接技术对隐私数据进行保护,并根据不同敏感属性值区间的敏感度设定了不同的阈值 ε 以控制敏感属性值之间的相似度。

表1　原始数据

tuple-ID	age	zip	salary
t_1	17	12k	1000
t_2	23	14k	1010
t_3	24	16k	1020
t_4	19	14k	2000
t_5	20	15k	5000
t_6	23	19k	3000
t_7	31	20k	2300
t_8	33	21k	5700
t_9	35	23k	12000
t_{10}	27	25k	6300
t_{11}	28	27k	7000
t_{12}	30	29k	8500
t_{13}	32	30k	7200
t_{14}	36	36k	13000
t_{15}	34	38k	20000
t_{16}	43	37k	6000

表2　泛化后的数据

group-ID	age	zip	salary
1	[17,24]	[12k,16k]	1000
1	[17,24]	[12k,16k]	1010
1	[17,24]	[12k,16k]	1020
2	[19,23]	[14k,19k]	2000
2	[19,23]	[14k,19k]	5000
2	[19,23]	[14k,19k]	3000
3	[31,35]	[20k,23k]	2300
3	[31,35]	[20k,23k]	5700
3	[31,35]	[20k,23k]	12000
4	[27,30]	[25k,29k]	6300
4	[27,30]	[25k,29k]	7000
4	[27,30]	[25k,29k]	8500
5	[32,36]	[30k,38k]	7200
5	[32,36]	[30k,38k]	13000
5	[32,36]	[30k,38k]	20000
6	[40,45]	[35k,39k]	6000
6	[40,45]	[35k,39k]	90000
6	[40,45]	[35k,39k]	10000
6	[40,45]	[35k,39k]	8900
7	[45,49]	[37k,43k]	39500
7	[45,49]	[37k,43k]	49500
7	[45,49]	[37k,43k]	15000

表3　满足算法 (ε,m)-anonymity 的数据发布表

group-ID	age	zip	salary
1	[19,23]	[15k,17k]	1000
1	[19,23]	[15k,17k]	2000
1	[19,23]	[15k,17k]	2500
2	[27,30]	[19k,21k]	10000
2	[27,30]	[19k,21k]	10500
2	[27,30]	[19k,21k]	9000
3	[32,36]	[29k,32k]	5000
3	[32,36]	[29k,32k]	1000
3	[32,36]	[29k,32k]	3500

2　(ε_p,l)-anonymity 模型及算法

下面对数值型敏感属性数据发布中的近邻泄露问题进行形式化定义。

假设 T 为原始数据表,T 包含 d 个准标识符(QI)属性 A_1^{qi},$A_2^{qi},\cdots,A_d^{qi}$,并且敏感属性为 A^s。每个准标识符属性既可以是数值型的,也可以是分类型的,但由于本文主要解决的是数值型敏感属性的近邻泄露问题,所以敏感属性必须是数值型的。对于每个元组 $t\in T$,本文用 $t[i]\ (1\leqslant i\leqslant d)$ 表示元组 t 的 A_i^{qi} 值,$t[d+1]$ 表示元组 t 的敏感属性值 A^s。

2.1　基本概念

定义1　准标识符属性QI。如果 T 的一组非敏感且非ID

的属性能够与外部数据表链接起来标志个体,称之为准标识符属性。

定义 2　准标识符组 QI-group。给定一个数据集 T, QI-group 是 T 的子集且,T 中的每一元组只能属于其中的一个子集,且 $\cup_{i=1}^{m} G_i = T$, 对任意一对 i 和 j,若 $i \neq j$,则 $G_i \cap G_j = \varnothing$。

定义 3　近邻攻击[7]。若在发布数值型敏感属性的数据表中,分组中的敏感属性值之间的差异度过小,会导致攻击者以较高的概率推断出目标个体的敏感信息落在一个较小的区间内,致使目标个体的敏感信息泄露。

定义 4　l-多样性划分[16]。设 v 表示分组 $G_j(1 \leqslant j \leqslant m)$ 中出现最频繁的敏感属性值,t 表示敏感属性值为 v 的元组,$c_j(v)$ 表示分组 $G_j(1 \leqslant j \leqslant m)$ 中元组 t 的数目,若 $c_j(v)/|G_j| \leqslant 1/l$,其中 $|G_j|$ 表示分组 G_j 中的元组数目,则这个具有 m 个分组的划分是 l-多样性的。

Xiao 等人[16]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。下面给出 Anatomy 的定义。

定义 5　Anatomy[16]。对于一个给定的具有 m 个 QI-group 的 l-多样性划分,Anatomy 将产生如下的一个准属性表(QIT)和一个敏感属性表(ST),准属性表有如下模式:

$$(A_1^{qi}, A_2^{qi}, \cdots, A_d^{qi}, \text{group-ID})$$

对于每个 $QI_j(1 \leqslant j \leqslant m)$ 和每个属于 QI_j 的元组 t 来说,QIT 元组形式为 $(t[1], t[2], \cdots, t[d], j)$。敏感属性表 ST 则有如下模式:

$$(\text{group-ID}, A^s)$$

对于每一个 $QI_j(1 \leqslant j \leqslant m)$ 和该 QI-group 中的每个不同的敏感属性值 v,敏感属性表 ST 中的记录形式为 $(j, v, c_j(v))$。其中,$c_j(v)$ 指的是 QI-group 中敏感属性值为 v 的元组数。

在一个 QI-group 中,对于敏感属性值 x、y,若有 $|x-y| < \varepsilon$,则敏感属性值 x、y 相似[4]。其中阈值 ε 控制敏感属性值之间的相似程度,并决定分组 QI-group 中是否存在近邻泄露问题。但算法 (ε, m)-anonymity 将阈值 ε 设为定值显然在实际问题中不可行,因此给出定义 6。

定义 6　阈值 ε 的设定。将敏感属性值按升序排序,根据处于不同数值范围的敏感属性值需要设定不同 ε 的原则将所有的元组划分为 p 组,则每组对应的 ε 值为 $\varepsilon_i(1 \leqslant i \leqslant p)$。

定义 7　近邻泄露风险。假设 t 是数据表 T 中的一个元组,并且 G 是泛化表中的一个分组,则元组 t 的近邻泄露风险表示为 $P(t)$,它的值为 $x/|G|$。其中,x 是与元组 t 的敏感属性值相差小于 ε 的元组的数量,$|G|$ 表示分组 G 中元组的数目。对于给定的不同的阈值 $\varepsilon_i(1 \leqslant i \leqslant p)$ 和一个整数 $l(l \geqslant 1)$,若准标识符属性表 QIT 和敏感属性表 ST 满足 (ε_p, l)-anonymity,则对于任意 $t \in T, P(t) \leqslant 1/l$。

2.2　算法

(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,并且该算法仍然是通过传统的泛化方式保护隐私数据,导致较高的信息损失。本文根据实际情况设置了多个不同的阈值 ε 来控制处于不同数值范围的敏感属性值之间的相似度,并通过将原始关系的准标识符属性和敏感属性以两个不同的关系发布,即将数据表划分为独立的 QIT 和 ST 表进行发布,利用它们之间的有损链接来保护隐私数据的安全。由于在发布 QIT 表中的数据时采用的是原始数据表中的精确信息,所以大大提高了信息的可用性。而对于 QI-group 的获取,则要经过对所有元组进行两次划分和提取的过程,从算法的执行过程可知该算法的时间复杂度为 $O(n\log n)$。本文算法的具体描述如下:

输入:数据集 T,参数 l。

输出:准标识符属性表 QIT,敏感属性表 ST。

//a)~e)为划分算法

a)根据敏感属性值 A^s 的大小,对表 T 中的元组进行升序排序。

b)根据 A^s 的取值范围,设置 p 个敏感值区间 $S_i(1 \leqslant i \leqslant p)$。

c)根据 $S_i(1 \leqslant i \leqslant p)$ 设置对应的 p 个 $\varepsilon_i(1 \leqslant i \leqslant p)$,用于控制处于 $S_i(1 \leqslant i \leqslant p)$ 内 A^s 之间的相似度。

d)将 A^s 中处于 $S_i(1 \leqslant i \leqslant p)$ 的元组划分到到子表 $T_i(1 \leqslant i \leqslant p)$ 中。

e)对 $T_i(1 \leqslant i \leqslant p)$ 再划分,具体划分过程如下:

(a)以对应的 $\varepsilon_i(1 \leqslant i \leqslant p)$ 为间隔,将对应的 $S_i(1 \leqslant i \leqslant p)$ 划分为若干小区间 $S_j^*(1 \leqslant j \leqslant n, n \geqslant p)$;

(b)将表 $T_i(1 \leqslant i \leqslant p)$ 中 A^s 处于 $S_j^*(1 \leqslant j \leqslant n, n \geqslant p)$ 中的元组放入桶 $B_j(1 \leqslant j \leqslant n, n \geqslant p)$ 中。

//f)~k)为提取算法

f)根据桶内元组的数目,对 $B_j(1 \leqslant j \leqslant n, n \geqslant p)$ 进行降序排列。

g)从排好序的桶 $B_j(1 \leqslant j \leqslant n, n \geqslant p)$ 中选出前 l 个桶。

h)从 l 个桶中各取一个元组,验证是否可以组成满足条件的分组。具体方法如下:

输入:l 个桶中的元组。

输出:满足模型 (ε_p, l)-anonymity 条件的 QI-group。

(a)l 个桶中的元组按敏感属性值的大小进行降序排序;

(b)从每个桶中各取一条记录(从第一条记录开始);

(c)若记录均来自不同的子表 $T_i(1 \leqslant i \leqslant p)$,则执行(e),否则执行(d);

(d)判断来自同一子表 $T_i(1 \leqslant i \leqslant p)$ 的元组的敏感属性之间的差值,执行如下:

```
if(取出的记录属于同一子表 Ti(1≤i≤p))
{
        while(i<=k&&j<=m)/*k 和 m 为相应桶的记录数*/
        {
        if(|a[i]-b[j]|<εp)/*a[i]和 b[j]分别表示桶 a 的第 i 条记录和桶 b 的第 j 条记录*/
        i++;
        else
        输出 QI-group;
        }
}
```

(e)输出 QI-group;

i)若上述 l 个桶中存在某些桶仍剩余元组,则将这些桶根据桶内元组的数目按降序插入到剩余的 $n-l$ 个桶中,最后使所有的桶按桶内的元组数目仍呈降序排列。

j)若剩余桶的数目大于 l 个,循环执行 g)~i),直至含有元组的数目的桶不足 l 个。

k)将剩余元组按哈希函数插入到已经存在的分组中。

l)初始化一个空的准标识符属性表 QIT 和敏感属性表 ST,将所有分组以 QIT 和 ST 的形式输出,算法结束。

以表 1 中的原始数据为例,对算法的执行过程进行说明,其中参数 $l=3$,即分组中的元组数目为 3。阈值 ε 的取值根据数据表中敏感属性值的数值范围设定为三个,分别为 $\varepsilon_1 = 500$, $\varepsilon_2 = 1000$, $\varepsilon_3 = 5000$。其中,$\varepsilon_1 = 500$ 用来控制[1000, 5000)内敏感属性值的相似度,$\varepsilon_2 = 1000$ 用来控制[5000, 10000)内敏感属性值的相似度,$\varepsilon_3 = 5000$ 用来控制[10000, 50000)内敏感属性值的相似度。算法首先根据元组的敏感属

性值进行升序排序,然后根据阈值控制的相应敏感属性值,将元组划分到三个子表中,如表 4 ~6 所示。接下来对每个子表进行第二次划分,将区间[1000,5000)、[5000,10000)以及[10000,50000)以相应的阈值 ε 为间隔进行划分,划分结果为[1000,1500),[1500,2000),…,[4500,5000);[5000,6000),[6000,7000),…,[9000,10000);[10000,15000),[15000,20000),…,[45000,50000)。将敏感属性值处于上述同一小区间内的元组划分到同一桶中,并按照桶中的元组数目对桶进行降序排列,结果为$\{t_1,t_2,t_3\},\{t_{18},t_9,t_{14}\},\{t_4,t_7\},\{t_5,t_8\},\{t_{16},t_{10}\},\{t_{11},t_{13}\},\{t_{12},t_{19}\},\{t_6\},\{t_{22}\},\{t_{15}\},\{t_{20}\},\{t_{21}\}$。由于 $l=3$,所以首先选择出前三个桶,并从中分别取出一个元组,比较它们之间的敏感属性值,可知 t_1 与 t_{14} 均来自子表 1,需要判断它们的敏感属性值是否至少为 ε_1,结果满足条件;而 t_1、t_{14} 和 t_{18} 来自不同的子表,所以它们之间的差异度很大,因此这三个元组可以组成一个分组。当前两个分组获取后,第三个桶 $\{t_4,t_7\}$ 中已经没有元组。此时,将剩有元组的桶 $\{t_3\}$ 和 $\{t_{14}\}$ 按照桶中元组的数目插入到剩余的桶中,循环执行上述过程。在获取分组 4 时,由于桶 $\{t_{10}\}$ 中元组 t_{10} 与桶 $\{t_8\}$ 和 $\{t_{13}\}$ 中的元组之间的敏感属性值之间的差值均不满足相应的阈值 ε_2,因此,将桶 $\{t_{10}\}$ 按照桶中的元组数目插入到剩余的桶中,并将桶 $\{t_{11},t_{13}\}$ 中的元组与桶 $\{t_8\}$ 和 $\{t_{13}\}$ 中的元组进行比较。最后,若有元组剩余,则采用哈希函数将其插入到其他分组中,元组 t_{10} 最终插入到分组一中。将最终获得的分组输入到准标志符属性表 QIT 和敏感属性表 ST 中。根据上述算法得到的数据发布表如表 7 所示。

表 4　第一次划分后得到的子表 1

age	zip	salary
17	12k	1000
23	14k	1010
24	16k	1020
18	14k	2000
31	20k	2300
23	19k	3000

表 5　第一次划分后得到的子表 2

age	zip	salary
20	15k	5000
33	21k	5700
43	37k	6000
27	25k	6300
28	27k	7000
32	30k	7200
30	29k	8500
41	39k	8900

表 6　第一次划分后得到的子表 3

age	zip	salary
45	37k	10000
35	23k	12000
36	36k	13000
49	43k	15000
34	38k	20000
45	37k	39500

表 7　满足模型(ε_p,l)-anonymity 的数据发布表

age	zip	group-ID	group-ID	salary
17	12k	1	1	1000
45	37k	1	1	10000
18	14k	1	1	2000
47	40k	1	1	49500
23	14k	2	2	1010
35	23k	2	2	12000
31	20k	2	2	2300
20	15k	3	3	5000
43	37k	3	3	6000
28	27k	3	3	7000
33	21k	4	4	5700
32	30k	4	4	7200
30	29k	4	4	8500
41	39k	5	5	8900
27	23k	5	5	6300
18	14k	5	5	2000

3　算法分析

该算法将所有的元组根据敏感属性值的大小进行了升序排序,然后根据处于不同数值范围的敏感属性值需要设定不同 ε 的原则,将所有的元组划分到 p 个子表中。将这 p 个子表的元组根据敏感属性值以对应的 ε 为间隔划分到若干桶中。按桶内的元组数对划分后的若干个桶进行降序排序,取出前 l 个桶,便于获得更多的 QI-group。从 l 个桶内各取一个元组,每次只能从 l 个桶的第一条元组开始提取,由于每个桶内的元组都是根据敏感属性值的大小升序排序的,因此这样可以使 QI-group 内的差异度更大。对于取出的 l 个元组,若它们来自不同的子表,则它们的敏感属性之间的差值的差异度将很大,因此这 l 个元组可以组成满足发布条件的分组。若这 l 个元组存在一些元组来自相同的桶 $B_j(1\leqslant j\leqslant n,n\geqslant p)$,则需要判断这些元组的敏感属性之间的差异度是否至少为 $\varepsilon_i(1\leqslant i\leqslant p)$,若满足该条件,则 QI-group 成立,否则继续验证下一条元组。所以,通过本算法获得的分组,分组中元组的敏感属性值之间的差异度都由相应的 $\varepsilon_i(1\leqslant i\leqslant p)$ 保证,在很大程度上阻止了近邻攻击,减少了近邻泄露。

由于算法(ε_p,l)-anonymity 在发布数据时,将准标识符属性表(QIT)和敏感属性表(ST)以两个不同的关系发布,利用它们之间的有损链接来保护隐私数据的安全,由于在发布准标识符属性表时采用的是原始数据表中的精确数据,这样可以有效地提高信息的可用性。

4　实验结果及分析

本章将通过实验分析算法(ε_p,l)-anonymity 的性能,并将其与算法(ε,m)-anonymity 的性能进行比较。实验的目的主要是分析数据的可用性以及算法的执行效率。实验的硬件环境为 Intel Pentium IV 2.8 GHz CPU,512 MB 内存,操作系统为 Microsoft Windows XP,算法使用 Java 编程实现。采用实际数据集 SAL 进行实验测试,数据集来自 http://ipums.org/。对原始

数据集滤掉不完整记录及非数值属性后,提取 30 K(1 K = 1000)条记录。该数据集是美国部分人口的个人信息,选取出其中的七个整型属性:年龄、性别、种族、国家、出生地、职业和工资。其中工资为敏感属性,其余为准标识符属性。

4.1 信息可用性分析

在分析发布数据的信息可用性时,可以通过集合查询语句来分析。查询语句为

Select count (*) From SAL

Where $A_1 \in b_1$, $A_2 \in b_2 \cdots A_w \in b_w$

其中:$A_1, A_2, \cdots, A_{w-1}$ 表示元组的准标志符属性,A_w 表示敏感属性,$b_i (1 \leqslant i \leqslant b)$ 表示属性 A_i 所在的随机区间。

为了衡量信息可用性的高低,引入平均相对误差(average relative error, ARErr)[10]。平均相对误差 ARErr = (act-est)/est。其中,act 表示查询原始数据表得到的精确结果,est 表示查询匿名后的数据表得到的结果。平均相对误差越低说明信息可用性越高;否则,信息可用性越差。

图 1 给出了数据表中的属性个数对信息可用性的影响。从图中可以看出,随着属性个数的增多,算法 (ε, m)-anonymity 的平均相对误差明显增大,这是因为随着属性个数的增多,算法 (ε, m)-anonymity 在泛化过程中导致分组中的泛化区间过度增大,信息可用性随之下降。算法 (ε_p, l)-anonymity 由于在发布 QIT 表时保留了原始数据表中的精确信息,因此平均相对误差几乎不受属性个数的影响,信息可用性明显优于 (ε, m)-anonymity。

图 2 显示了数据集的大小对发布数据的信息可用性的影响。从图中可以看出,(ε, m)-anonymity 随着数据集的增大,平均相对误差有微小的下降趋势。这是因为采用泛化的匿名方式,数据集的增大可以产生更大的搜索范围,反过来可以降低误差。本文采用的算法随着数据集的增大,平均相对误差有有微小变化,可知信息的可用性稳定,而且采用本文算法出现的平均相对误差总是小于采用算法 (ε, m)-anonymity 出现的平均相对误差。

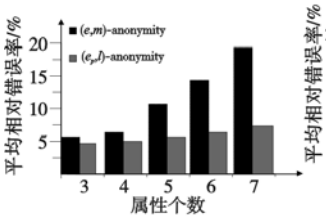

图1 属性个数对信息可用性的影响

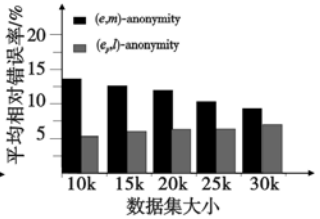

图2 数据集的大小对信息可用性的影响

图 3 显示的是分组中的元组数目(m 或者 l)对发布数据的信息可用性的影响。从图中可以看出,尽管两种算法的相对误差都会随着 m 或者 l 值的增大而增大,但算法 (ε_p, l)-anonymity 出现的平均相对误差总是低于采用算法 (ε, m)-anonymity 出现的平均相对误差,而且算法 (ε, m)-anonymity 中相对误差增长幅度显著,而本文算法增长相对缓慢。出现这种现象的原因是随着分组中元组数目的增大,对于算法 (ε, m)-anonymity 而言,要获取满足其采用的 Mordrian 泛化原则的分组越来越困难。而本文算法由于将分组的敏感属性和准标识符属性分开发布,主要根据敏感属性之间的差距确定分组是否成立,因此分组中元组数目的增大只会在获取满足条件的分组时使元组间比较的次数增大,而对发布数据的信息可用性影响较小。

4.2 执行时间分析

图 4 给出了两种算法在执行时间上的比较。采用算法 (ε, m)-anonymity。执行时间随着分组中的元组数 m 值的增大而逐渐变小,这是因为随着 m 值的增大,可以满足 (ε, m)-anonymity 所采用的 Mordrian[12] 泛化算法的元组越来越少,该算法就会提前终止。这也是导致该算法随着 m 值的增大,信息可用性显著降低的原因。这也和图 3 显示的实验结果吻合。所以,尽管随着 m 值的增大,执行时间减少,但却导致信息可用性的显著降低。本文算法的执行时间随着分组中元组数 l 值的增大而增大,这是因为随着分组中的元组数目增多,算法中敏感属性值之间需要比较的次数随之增大。通过图 3 的实验结果可知,随着分组中元组数目的增大,信息可用性保持相对稳定。

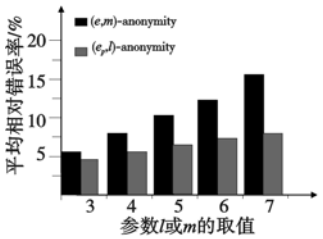

图3 分组中的元组数目对信息可用性的影响

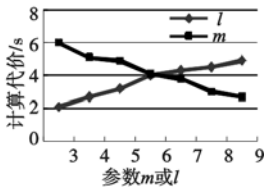

图4 分组中的元组数目对执行时间的影响

5 结束语

现有解决近邻泄露问题的研究没有考虑到随着数值型敏感属性值的变化,用来控制敏感属性值之间相似度的阈值 ε 也应当是变化的。为了更好地处理敏感属性值之间的相似度问题,更好地解决敏感属性值之间的近邻问题,本文根据不同的敏感值区间设置了不同的阈值 ε 以控制相似度,并基于有损链接技术,将数据表分为准标识符属性表 QIT 和敏感属性表 ST 进行发布,这样可以更多地去关注敏感属性值的分布。而且,将整个数据表 T 分为两个表进行发布,可以使 QIT 和 ST 中的数据不发生改变。实验表明,本文提出的算法在保证良好可用性的基础上能够更好地防止近邻攻击。今后的工作将考虑针对多维数值型敏感属性进行近邻泄露保护的方法研究。

参考文献:

[1] SWEENEY L. k-anonymity: a model for protecting privacy[J]. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, 2002, 10(5): 557-570.

[2] SWEENEY L. Achieving k-anonymity privacy protection using generalization and suppression[J]. International Journal on Uncertainly, Fuzziness and Knowledge-based Systems, 2002, 10(5): 571-588.

[3] MACHANAVAJJHALA A, GEHRKE J, KIFER D, *et al*. l-diversity: privacy beyond k-anonymity[C]//Proc of the 22nd International Conference on Data Engineering. 2006: 24-35.

[4] AGGARWAL C C. On k-anonymity and the curse of dimensionality [C]//Proc of Very Large Data Bases Conference. 2005: 901-909.

[5] BAYARDO R, AGRAWAL R. Data privacy through optimal k-anonymization[C]//Proc of International Conference on Data Engineering. 2005: 217-228.

(下转第 657 页)

原始图像的 RS 分析估计值,安全阈值为 0.1 时本文算法能够很好地抵抗 RS 分析,较 LSB 算法有很大的优势,满足统计不可见性。

表 2　嵌入随机比特流后的 PSNR 值

图像	PSNR			
	PVD	改进的 PVD[7]	2-LSB	本方案
Lena	47.787 3	49.263 5	47.945 3	50.706 4
girl	47.368 7	48.915 6	48.203 0	51.428 6
couple	45.162 8	47.438 7	48.186 3	51.854 9

图1　Lena原始图像

图2　Lena载密图像

图3　girl原始图像

图4　girl载密图像

图5　couple原始图像

图6　couple载密图像

表 3　RS 分析的嵌入率估计值

图像	嵌入率估计值				
	正常图像	嵌入率为 1 的 LSB	嵌入率为 0.7 的 LSB	嵌入率为 0.3 的 LSB	本方案
Lena	−0.291 7	0.995 3	0.677 8	0.168 9	−0.111 7
girl	−0.005 1	0.992 8	0.774 5	0.217 8	0.034 1
couple	−0.036 9	1.006	0.757 8	0.281 7	0.039 6

3　结束语

本算法结合人眼视觉特性,借鉴了 PVD 算法的思想,通过对两个子算法进行一定的改进,在保证嵌入率的前提下,能抵抗RS分析,具有良好的视觉不可见性和统计不可见性。今后的研究重点是在保证不可见性的条件下提高算法嵌入效率和嵌入率。

参考文献:

[1] 王朔中,张新鹏,张开文. 数字密写和密写分析[M]. 北京:清华大学出版社,2005:20-83.

[2] ZHANG Xin-peng, WANG Shuo-zhong, ZHANG Kai-wen. Steganography with least histogram abnormality[C]//Lecture Notes in Computer Science, vol 2776. Berlin: Springer-Verlag, 2003:401-412.

[3] 刘粉林,刘九芬,罗向阳,等. 数字图像隐写分析[M]. 北京:机械工业出版社,2010:58.

[4] WU Da-chun, TSAI W H. A steganographic method for images by pixel-value differencing[J]. Pattern Recognition Letters, 2003, 24(9-10):1613-1626.

[5] ZHANG Xin-peng, WANG Shuo-zhong. Vulnerability of pixel-value differencing steganography to historgram[J]. Pattern Recognition Letters, 2004, 25(3):331-339.

[6] WU H C, WU N I, TSAI C S, *et al*. Image steganographic scheme based on pixel-value differencing and LSB replacement methods[J]. IEE Proceedings Vision, Image and Signal Processing, 2005, 152(5):611-615.

[7] YANG C H, WENG C Y. A steganographic method for digital images by multipixel differencing[C]//Proc of IEEE International Computer Symposium. 2006:831-836.

[8] ZHANG Xin-peng, WANG Shuo-zhong. Efficient steganographic embedding by exploiting modification direction[J]. IEEE Communications Letters, 2006, 10(11):781-783.

[9] JUNG K H, YOO K Y. Improved exploiting modification direction method by modulus operation[J]. International Journal of Signal Processing, Image Processing and Pattern, 2009, 2(1):79-87.

[10] YANG C H, WENG Chi-yao, WANG S J, *et al*. Varied PVD + LSB evading detection programs to spatial domain in data embedding systems[J]. Journal of Systems and Software, 2010, 83(10):1635-1643.

(上接第 654 页)

[6] LeFEVRE K, DeWITT D J, RAMAKRISHNAN R. Incognito: efficient full-domain *k*-anonymity[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2005:49-60.

[7] LI Jie-xing, TAO Yu-fei, XIAO Xiao-kui. Preservation of proximity privacy in publishing numerical sensitive data[C]//Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM, 2008:437-486.

[8] SAMARATI P. Protecting respondents' identities in microdata release[J]. IEEE Trans on Knowledge and Data Engineering, 2001, 13(6):1010-1027.

[9] ZHANG Qing, KOUDAS N, SRIVASTAVA D, *et al*. Aggregate query answering on anonymized tables[C]//Proc of the 23rd IEEE International Conference on Data Engineering. 2007:116-125.

[10] LeFEVRE K, DeWITT D J, RAMAKRISHNAN R. Workload-aware anonymization[C]//Proc of ACM Knowledge Discovery and Data Mining. 2006:277-286.

[11] LI Ning-hui, LI Tian-cheng, VENKATASUBRAMANIAN S. *t*-closeness: privacy beyond *k*-anonymity and *l*-diversity[C]//Proc of of the 23rd IEEE International Conference on Data Engineering. 2007:106-115.

[12] RUBNER Y, TOMASI C, GUIBAS L J. The earth mover's distance as a metric for image retrieval[J]. International Journal of Computer Vision, 2000, 40(2):99-121.

[13] Le FEVRE K, De WITT D J, RAMAKRISHNAN R. Mondrian multidimensional *k*-anonymity[C]//Proc of the 22nd International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2006:277-286.

[14] WANG Ting, MENG S, BAMBA B, *et al*. A general proximity privacy principle[C]//Proc of IEEE International Conference on Data Engineering. Washington DC: IEEE Computer Society, 2009:1279-1282.

[15] WANG Ting, LIU Ling. XColor: protecting general proximity privacy[C]//Proc of the 26th IEEE International Conference on Data Engineering. 2010:960-963.

[16] XIAO Xiao-kui, TAO Yu-fei. Anatomy: simple and effective privacy preservation[C]//Proc of the 32nd International Conference on Very Large Data Bases. 2006:139-150.