第35卷 第23期 计 算 机 工 程 2009年12月
Vol.35 No.23 Computer Engineering December 2009
·安全技术· 文章编号：1000—3428(2009)23—0161—03 文献标识码：A 中图分类号：TP301.6

基于信息隐藏的关系数据库数字水印算法

王志伟，孔祥维
(大连理工大学电子与信息工程学院，大连 116024)

摘 要：设计一种无需修改关系数据的属性值即可嵌入水印的关系数据库数字水印算法。通过使用信息隐藏技术，把对关系数据的修改转换到对信息隐藏载体的修改，使水印的嵌入过程能有效地避免对原始数据的破坏，保持原始数据的真实性和使用价值。实验证明该算法具有较好的鲁棒性。

关键词：关系数据库；数字水印；信息隐藏

Digital Watermark Algorithm for Relational Database Based on Information Hiding

WANG Zhi-wei, KONG Xiang-wei
(School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116024)

【Abstract】This paper designs a digital watermark algorithm for relational database, which need not modify the attribute values of the data. Through information hiding technology, it transforms modifying relational data to modifying information hiding carrier, so that the original data can avoid to be damaged during watermark embedding, and reality and value of the data are maintained. Experiments show that the algorithm is robust.

【Key words】relational database; digital watermark; information hiding

1 概述

在国外，文献[1-2]先后提出将数字水印技术应用到关系数据库中。我国对数据库水印的研究大约从2003年开始，文献[3]提出了一种可以在关系数据库中嵌入有实际意义字符串的数据库水印算法，很多研究人员[4-5]在从事关系数据库数字水印的研究工作。文献[6]就将最优化算法引入关系数据库的数字水印技术中。现有的关系数据库数字水印算法基本都是基于一种认知，即关系数据中的一些数值属性可以进行少量修改而不影响数据的使用价值。各种新算法的设计目标是对原始数据的修改尽可能地少，同时能够抵抗一些常见的攻击行为。这样做有以下2点不足：(1)由于修改数据而破坏了数据的真实性和使用价值；(2)限制了算法的适用范围。例如，如果关系数据都是整数型数据，那么要想通过修改属性值来嵌入水印就比较困难。因此，本文借助信息隐藏技术提出一种无需修改关系数据库中的数据即可嵌入水印的算法。

2 基于信息隐藏技术的关系数据库数字水印算法

2.1 算法模型及描述

本文水印算法的实质是将对关系数据的修改转换到对信息隐藏载体图像的修改，水印嵌入完毕后会得到一幅隐密图像。嵌入算法的流程如图1所示。具体步骤如下：

Step1 将关系数据分段。关系数据由 n 个元组组成，需要把它分成 m 段，最终每一个元组都将被唯一地分配到这 m 段中的某一段中。

Step2 使用提取函数提取0, 1比特流。提取函数定义为：$f(\boldsymbol{x})=\{\boldsymbol{x}\mid R^n\rightarrow R^1\}$。关系数据分成 m 段之后，每一段都会包含一定数目的元组。对于第 i 段的所有元组，将准备嵌入水印的属性值表示成向量 $\boldsymbol{X}_i$，通过提取函数 $f(\boldsymbol{X}_i)$，可以计算得到一个数值。所以，由这 m 个数据段可以计算得到 m 个数值 $f(\boldsymbol{X}_i)$。然后依据这 m 个 $f(\boldsymbol{X}_i)$的某种关系(例如前后的大小关系)，将其用0, 1比特表示出，最后可以得到 m-1维的0, 1比特向量，称为提取0, 1比特流。

Step3 异或运算。水印可以表示成长度为 L 的0, 1比特流。由Step2已经得到长度为 m-1($m>>L$)的提取0, 1比特流。水印通过循环表示也可以组成长度为 m-1的0, 1比特流，称为水印0, 1比特流。然后将水印0, 1比特流与提取0, 1比特流进行异或运算，得到一个隐密的长度为 m-1的0, 1比特流，称为隐密0, 1比特流。

Step4 信息隐藏。将Step3得到的隐密0, 1比特流作为秘密信息，通过一种基于图像的信息隐藏算法将其嵌入载体图像中，最后得到一个隐密图像。水印的嵌入过程完成。

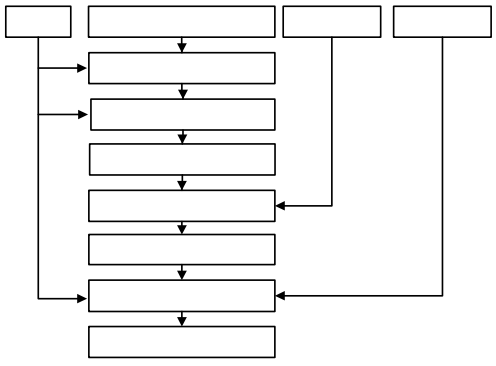


图1 数字水印的嵌入算法流程

水印提取算法的流程如图2所示，具体步骤如下：

作者简介：王志伟(1983－)，男，硕士研究生，主研方向：信息隐藏，数字水印；孔祥维，教授、博士、博士生导师

收稿日期：2009-03-30 **E-mail：**dlutwzhiwei@gmail.com

Step1 将待检测关系数据分段。同嵌入算法的 Step1。

Step2 使用提取函数提取 0,1 比特流。同嵌入算法的 Step2。

Step3 从隐密图像中提取隐密比特流。通过信息隐藏的提取算法，可以从隐密图像中提取出隐密 0, 1 比特流。

Step4 提取水印比特流。将 Step2 中得到的提取比特流和 Step3 中得到的隐密 0,1 比特流进行异或运算，得到水印 0,1 比特流。由于水印是重复嵌入的，因此再根据最大判决就可以得到提取出的水印。

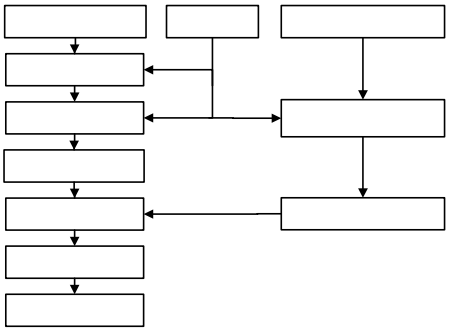


图 2　数字水印的提取算法流程

2.2　数据分段的算法

关系数据集定义为 $D(P, A_1, A_2, \cdots, A_v)$，其中，$P$ 是元组的主键；$A_1, A_2, \cdots, A_v$ 是 v 个候选的可以嵌入水印的属性；$|D|$ 表示 D 中元组的个数；D 中所有元组最后将被分成 m 段，即 D 会被分成 m 个不重叠的段$\{S_0, S_1, \cdots, S_{m-1}\}$，每一段平均包含$|D|/m$ 个元组，段间不重叠意味着对于任意的 2 段 S_i 和 S_j $(i \neq j)$，$S_i \cap S_j = \varnothing$。

本算法中的元组分段是通过哈希函数实现的。对于任意的一个元组 $r \in D$，将其主键和密钥作为一个哈希函数的输入，得到一个信息鉴别码(MAC)，再对 MAC 进行模 m 操作，得到分段序号 i，则该元组 r 被分到第 i 段中：

$$partition(r) = H(Ks \,\|\, H(r.P \,\|\, Ks)) \bmod m$$

其中，$H()$是哈希函数；||表示连接操作，$r.P$是元组的主键；m 是分段的个数。

由于哈希函数的输出是均匀分布的，因此平均每一段可以分到$|D|/m$ 个元组，而且经过这样的处理，攻击者无法判别某一个元组会被分到哪一段。

2.3　提取函数的算法

提取函数定义为：$f(\boldsymbol{x}) = \{\boldsymbol{x} \mid R^n \to R^1\}$。对于第 i 段数据，$\boldsymbol{X}_i$ 就是该段所有元组的属性值的向量，最后输出一个数值 $y_i = f(\boldsymbol{X}_i)$。

提取函数应具有抵抗攻击的能力。例如在该分段中插入或者删除一些元组，或者对该分段中元组的属性值进行微小修改，对于结果的输出不会产生太大影响，可以抵抗攻击者的攻击行为。

对于第 i 段的关系数据，算法中设计的提取函数过程可以描述如下：首先抽取元组，依据类似于数据分段的算法，对于任意一个元组 $r \in S_i$，将其主键和密钥作为一个哈希函数的输入，得到一个 MAC，再对 MAC 进行模 l 操作(l 是一个经验值，实验中选为 10)，然后把所有模值为 1 和 2 的元组作为选中的元组，其属性值就可以组成 2 个向量，记为 $\boldsymbol{F}_1$ 和 $\boldsymbol{F}_2$。之后对 $\boldsymbol{F}_1$ 和 $\boldsymbol{F}_2$ 分别求均值，记为 $MeanF_1$ 和 $MeanF_2$，再利用它们组成 2 个坐标点$(1, MeanF_1)$, $(2, MeanF_2)$。由这 2 个点做两点插值可以得到一条直线，把这条直线在纵轴上的交点作为输出的函数值 $f(\boldsymbol{X}_i)$。

2.4　基于图像的信息隐藏算法

随着信息隐藏技术的迅速发展，基于图像的信息隐藏算法越来越成熟、安全，选择的余地很大。图像格式有 JPEG, BMP, Gif 等许多，针对不同的图像格式，又有多种信息隐藏的算法。

3　实验结果及鲁棒性分析

对关系数据库的数字水印攻击方式一般包括 3 种：删除攻击，插入攻击，篡改攻击。本文针对这 3 种攻击方式分别进行实验。实验过程中，用文献[6]的水印算法作为比较。

实验数据是由 Matlab 随机产生的，包括均匀分布和高斯分布 2 组，属性值服从均匀分布的数据范围是[200, 400]，属性值为高斯分布的服从分布 $r.A \sim N(0,100)$。关系数据集中包括 15 000 个元组，所有元组被分成 200 段，水印采用的是 12 个 0, 1 比特。算法中的哈希函数选用 MD5 算法，信息隐藏算法选用 JPEG 图像的 F5 算法。

3.1　删除攻击

在删除攻击中，假设攻击者在关系数据集中随机选择 α 个元组进行删除。

实验结果如图 3 所示，可以看出，对于服从均匀分布和高斯分布的数据，当原始数据分别被删除 55%和 35%时，能够准确地提取出完整的水印。在文献[6]的算法中，当原始数据被删除 85%时仍能完整地提取水印，所以，本算法在抵抗删除攻击方面差一些。

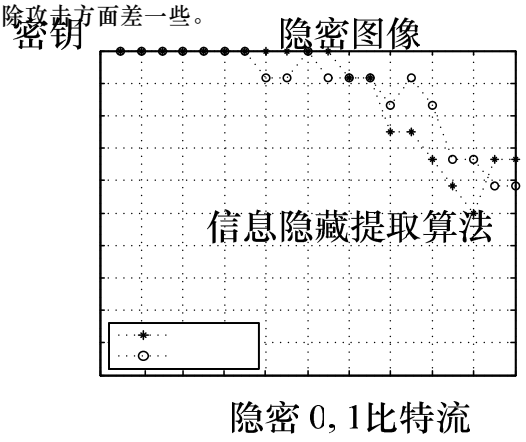
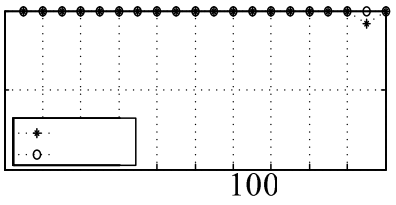


图 3　删除攻击实验结果

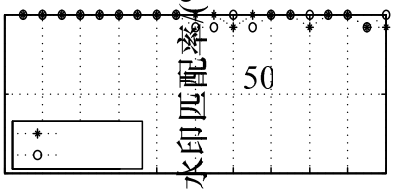
3.2　插入攻击

插入攻击指攻击者在关系数据集中新增 α 个元组。实验采用 2 种插入数据的方式：(1)在原有数据集中复制 α 个元组作为新的数据加入数据集。(2)通过随机产生 α 个元组的新数据放入数据集中，新数据的分布服从原始数据的分布。

实验结果如图 4 所示。图 4(a)显示的是第(1)种插入攻击方式。从中可以看出，无论原始数据是均匀分布还是高斯分布，当新插入的数据个数达到原始数据个数的 90%以上时，水印几乎都能完整地提取出来，说明算法对于抵抗插入攻击具有很好的效果。在抵抗这种攻击的能力上，文献[6]的算法与本算法性能相似。图 4(b)显示的是第(2)种插入攻击方式。对于服从均匀分布的原始数据，在插入数据个数达到 55%时，还能够完整地提取出水印；当原始数据服从高斯分布时，在插入数据的个数达到 45%时，能够完整地提取出水印。在抵抗这种插入攻击的能力方面，文献[6]的算法在插入的数据达到 85%时仍能有效地提取水印，所以，本算法在这方面的性能略差一些。



(a)复制原始数据的插入攻击



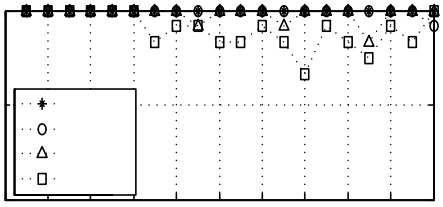
(b)随机生成数据的插入攻击

图 4　插入攻击实验结果

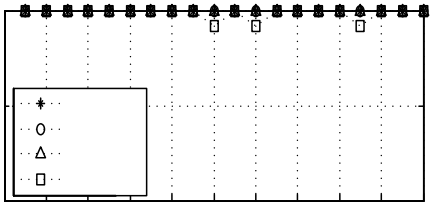
3.3　篡改攻击

篡改攻击中假设攻击者对元组的属性值做了少量修改。实验中使用 2 种篡改方式：固定(α,β)篡改(记为 *Fix*(α,β))以及随机(α,β)篡改(记为 *Random*(α,β))。*Fix*(α,β)篡改方式是在数据集中随机地选择 α 个元组进行修改，其中，α/2 个元组的属性值乘以(1+β)；另外 α/2 个元组的属性值乘以(1-β)；而 β 的取值固定。*Random*(α,β)篡改方式是在数据集中随机选择 α 个元组进行修改，其中，α/2 个元组的属性值乘以(1+ζ)；另外 α/2 个元组的属性值乘以(1-ζ)，ζ服从均匀分布，取值范围是[0, β]。

图 5 是算法抵抗 *Fix*(α,β)攻击的实验结果。图 5(a)是原始数据服从均匀分布时的结果，从中可以看出，当属性值的修改范围 β 达到 15%时，几乎不能破坏水印；当 β 达到 30%、修改的数据个数 α 达到 40%时都不能破坏水印。对于高斯分布的原始数据，抵抗攻击性能更好，当 β 达到 50%时都不能破坏水印，而这种修改一定会对原始数据造成巨大破坏，一般情况下不再具有使用价值。因此，本算法在抵抗 *Fix*(α,β)攻击方面有较好的性能。



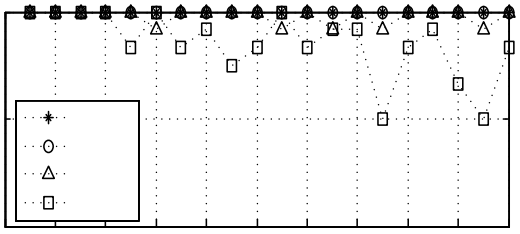
(a)原始数据为均匀分布的篡改攻击*Fix*(α,β)



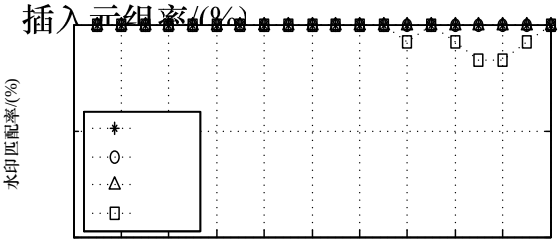
(b)原始数据为高斯分布的篡改攻击*Fix*(α,β)

图 5　*Fix*(α,β)篡改攻击实验结果

图 6 是算法抵抗 *Random*(α,β)攻击的实验结果。图 6(a)是当原始数据服从均匀分布时的结果，从中可以看出，当 β 达到 30%时本算法都能完整地提取出水印。对于高斯分布的原始数据，抗篡改攻击性能更好，甚至当 β 值达到 100%时都不能破坏水印。因此，本算法能够有效地抵抗 *Random*(α,β)攻击。



(a)原始数据为均匀分布时的篡改攻击*Random*(α,β)



(b)原始数据为高斯分布时的篡改攻击*Random*(α,β)

图 6　*Random*(α,β)篡改攻击实验结果

在抵抗篡改攻击方面，本算法与文献[6]的算法性能基本相当，有时甚至更占优势。

4　结束语

本文设计了一种基于信息隐藏技术的关系数据库数字水印算法。其特点在于：认为关系数据库水印技术嵌入方式需要修改关系数据的传统思想，通过使用信息隐藏技术可以不修改原始数据而同样嵌入水印。经过实验分析，本算法具有较好的鲁棒性。此外，设计的算法模型不仅适用于数值属性的关系数据，同样适用于非数值属性的关系数据，这可以作为下一步的研究方向。

参考文献

[1] Agrawal R, Kiernan J. Watermarking Relational Databases[C]//Proc. of the 28th VLDB Conference. Hong Kong, China: [s. n.], 2002: 155-168.

[2] Sion R, Atallah M, Prabhabkar S. Rights Protection for Relational Data[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1509-1525.

[3] 牛夏牧, 赵　亮, 黄文军, 等. 利用数字水印技术实现数据库的版权保护[J]. 电子学报, 2003, 31(12): 2050-2053.

[4] 张　勇, 赵东宁, 李德毅. 关系数据库数字水印技术[J]. 计算机工程与应用, 2003, 39(25): 193-195.

[5] 刘伟群, 刘云如, 易叶青. 基于 ICA 数据库水印嵌入算法的研究[J]. 科学技术与工程, 2006, 6(5): 628-631.

[6] Shehab M, Bertino E, Ghafoor A. Watermarking Relational Databases Using Optimization-based Techniques[J]. IEEE Trans. on Knowledge and Data Engineering, 2008, 20(1): 116-129.

编辑　张　帆