

基于属性标记的关系数据库水印方法

戴华¹ 秦小麟² 金倩倩² 付雄¹ 季一木¹

(1 南京邮电大学计算机学院,南京,210003; 2 南京航空航天大学计算机科学与技术学院,南京,210016)

摘要:数据库水印是保护数据库版权的一项重要技术。针对现有数据库水印方法无法抵抗有序关系模式变换攻击(ORSM)的问题,提出了一种基于属性标记的关系数据库水印方法。首先给出了有序关系模式变换攻击的定义,然后通过分析现有关系数据库水印算法无法抵抗ORSM攻击的问题,提出一种能够抵抗ORSM攻击的水印算法ALBWM,包括水印嵌入算法(ALBWM-EM)和水印检测算法(ALBWM-DT)。实验结果表明,与传统数据库水印算法相比,ALBWM算法在不影响性能的前提下对ORSM攻击具有更好的鲁棒性。

关键词:数据库水印; 版权保护; 鲁棒性; 有序关系模式变换攻击

中图分类号: TP301

文献标识码: A

文章编号: 1005-2615(2012)04-0580-07

Attributes Labeling Based Relational Database Watermarking Method

Dai Hua¹, Qin Xiaolin², Jin Qianqian², Fu Xiong¹, Ji Yimu¹

(1 College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China;

(2 College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China)

Abstract: Database watermarking is an important technology mainly used for copyright protection for database. Given the fact that current database watermarking methods cannot resist ordered relation schema mutation (ORSM) attack, attributes labeling based relational database watermarking (ALBWM) algorithm is proposed. Firstly, the definition of ORSM attack is given. Then, through analyzing the fragileness of current watermarking methods under ORSM attack, ALBWM is introduced which includes watermark embed algorithm(ALBWM-EM) and watermark detect algorithm(ALBWM-DT). Experimental results indicate that compared with the traditional watermarking algorithms, ALBWM has better robustness against ORSM attack and does not affect performance.

Key words: database watermarking; copyright protection; robustness; ordered relation schema mutation attack

随着网络技术的应用向各领域渗透,数据共享技术迅速发展,数据交换的需求日益增长。越来越多的数据库数据通过网络进行存储和发布,这些数据往往经过若干年的积累,蕴含着丰富的信息,具有巨大的社会价值与经济价值,如网上银行用户资料、生物医学数据、超市商品销售数据等。这些数据

库在通过网络共享的同时也面临窃取、非法拷贝、恶意篡改,并被用于各种商业或非法用途的危险。这就导致原始数据库的版权受到威胁,版权所有人的利益蒙受巨大损失。因此,数据库版权保护成为亟待解决的问题。2002年,Agrawal等首次提出用关系数据库水印进行数据库版权保护的思想^[1-3]:

基金项目:国家高技术研究发展计划(“八六三”计划)(2007AA01Z404)资助项目;国家自然科学基金(60673127, 60973139, 61103195)资助项目;江苏省科技支撑计划(BE2008135, BE2010198, BE2011844, BE2011189)资助项目;国家教育部博士点基金(20103218110017)资助项目;南京邮电大学引进人才科研启动基金(NY211043)资助项目。

收稿日期: 2011-06-04; **修订日期:** 2011-07-22

通讯作者:戴华,男,博士,讲师,1982年1月生, E-mail: daihua@njupt.edu.cn。

将版权信息通过数字水印技术嵌入数据库,利用水印的隐蔽性、安全性、鲁棒性,有效地解决数据库版权保护问题。此后,数据库水印陆续被用于数据库防篡改^[4-6]、指纹追踪^[7-9]、访问控制^[10]等用途,成为数据库安全控制的重要技术之一。

关系数据库水印技术是由图像水印技术发展而来,目前多数数据库水印方法借鉴了空域图像水印的基本思想,在嵌入和检测水印信息时使用相同的位置选择策略。由于图像和数据库中的关系表都可以看成二维数据集,攻击者都可以通过列变换,使水印信息移位,从而实现对水印信息的破坏。然而这种攻击在破坏图像水印的同时也明显改变了原始图像内容,降低了受保护数据可用性,违反了攻击者的意图,因此这种攻击方法对于图像水印而言没有实用价值;对于数据库而言,关系模式中属性列顺序的变化不影响关系模式的语义^[11],例如关系模式 $R(A_1, A_2, A_3, A_4)$ 和 $R'(A_4, A_3, A_2, A_1)$,可见变化属性列顺序对于数据库水印而言是一种水印攻击方法,称之为有序关系模式变换攻击(Ordered relation schema mutation attack, ORSM 攻击)。然而,现有的数据库水印技术^[1-3, 12-18]都没有考虑属性列顺序发生变化的情况,因此对于 ORSM 攻击无能为力。

在现有的关系数据库水印技术中, Agrawal 等主要利用数值型属性允许的误差,依赖密钥和主键在候选元组、属性和最不important 比特位(Least significant bit, LSB)中选择具体的水印嵌入位置,嵌入能够标识版权的水印信息^[1-3],该方法能够验证数据库中有无标识版权的水印,实现水印的盲检测;缺点是依赖主键和属性列顺序,无法抵抗 ORSM 攻击,而且无法嵌入有意义水印。Sion 等提出元组排序和划分集合的思想,通过改变数据分布特性嵌入水印信息^[12],该方法虽然能够抵抗 ORSM 攻击,但是隐蔽性不够好,计算开销大。在 Agrawal 算法的基础上, Li 等通过构造虚拟主键,解决了原来算法对主键依赖的缺点,并将数据库水印技术应用于数字指纹追踪^[13]; Gaurav 等利用差分扩展实现可逆数据库水印^[14]; Burepalli 等采用子集选择方法实现数据库水印^[15], Mahmoud 等提出基于时间戳协议的盲可逆水印方法^[16], 国内牛夏牧等引入了数据分块的思想,实现了有意义水印的嵌入^[17], 姜传贤等提出基于离散小波变换的水印方法^[18], 但 these 方法都无法有效解决针对 ORSM 攻击鲁棒性较弱的问题。

针对现有水印方法的不足, 本文提出了一种基

于属性标记的关系数据库水印算法 ALBWM。首先给出了有序关系模式变换攻击 ORSM 的定义, 并结合实例对现有关系数据库水印算法在 ORSM 攻击下鲁棒性弱的问题进行分析; 然后提出一种能够抵抗 ORSM 攻击的水印算法 ALBWM, 包括水印嵌入算法 ALBWM-EM 和水印检测算法 ALBWM-DT; 最后给出 ALBWM 算法与传统水印算法的性能对比实验以及在 ORSM 攻击下的鲁棒性对比实验。

1 问题描述

传统关系模式的语义与其属性列的相对位置无关, 变换属性列顺序不影响关系模式的语义^[9], 而目前多数关系数据库水印方法都是以传统关系模式语义为基础设计和实现的, 并没有考虑属性列顺序变换对水印的影响。为更加准确方便地描述问题, 首先给出有序关系模式变换攻击的相关定义, 然后给出针对传统 Agrawal 水印算法(简称为 Ag-WM 算法)的攻击实例。

定义 1 有序关系模式: 对关系的描述, 形式化表示为 $\hat{R}(P, \langle U \rangle)$ 。其中, \hat{R} 为关系名, P 为主键, $\langle U \rangle$ 为有序属性序列, 即在有序关系模式中属性的相对位置与有序关系模式的语义相关。

与数据库中的传统关系模式相比, 定义 1 中的有序关系模式对于属性列顺序敏感。以学生课程成绩表(Score)为例, 其有序关系模式记为 $\text{Score}(\text{SID}, \langle \text{OS}, \text{DBMS}, \text{Java} \rangle)$, 其中: SID 为主键; OS, DBMS 和 Java 表示课程属性列; $\langle \text{OS}, \text{DBMS}, \text{Java} \rangle$ 为属性序列。若变换属性课程 OS 和 DBMS 在属性序列中的位置, 得到新的有序关系模式 $\text{Score}'(\text{SID}, \langle \text{DBMS}, \text{OS}, \text{Java} \rangle)$, 则 Score' 与 Score 表示不同的有序关系模式语义。

定义 2 有序关系模式的等价判定: 设 $\hat{R}(P, \langle U \rangle)$, $\hat{R}'(P', \langle U' \rangle)$ 为有序关系模式, 若满足 $P = P' \wedge |\langle U \rangle| = |\langle U' \rangle| \wedge \forall a(a \in \langle U' \rangle) \rightarrow a \in \langle U \rangle$, 则称 \hat{R} 等价于 \hat{R}' , 记为 $\hat{R} \equiv \hat{R}'$ 。其中 $|\langle U \rangle|$ 表示属性序列的秩, 即 $\langle U \rangle$ 中包含属性列的个数。

定义 3 有序关系模式等价变换: 设 $\hat{R}(P, \langle U \rangle)$, $\hat{R}'(P', \langle U' \rangle)$ 为有序关系模式, 若满足 $\hat{R} \equiv \hat{R}' \wedge \sum_{i=1}^{|\langle U' \rangle|} (a_i \oplus a_i') > 1$, 则称 \hat{R} 和 \hat{R}' 可以通过等价变换进行转换。

其中 $a_i \in \langle U \rangle$, $a_i' \in \langle U' \rangle$, \oplus 表示异或操作, 即比较有序关系模式属性序列中相同位置的两个属性是否相同, 若相同则异或结果为 0, 否则为 1。

定义 4 有序关系模式变换攻击(ORSM 攻击):通过有序关系模式等价变换企图破坏数据库水印,使水印检测率降低的攻击方法称为有序关系

模式变换攻击。

同样以 Score 关系表为研究对象给出 ORSM 攻击的实例,如图 1 所示。

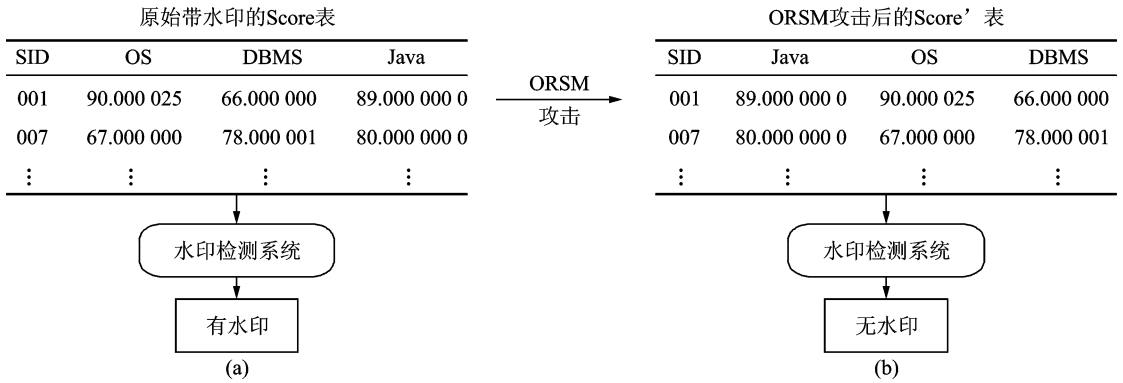


图1 对 Score 表的 ORSM 攻击实例

在图 1(a)中,对 Score 表中利用传统的 AgWM 算法嵌入水印信息,假设在 OS 属性列中的数据“90.000 025”中,最后的“25”即为水印信息。通过 ORSM 攻击,变换 Score 表中的属性列位置,得到图 1(b)中的表 Score', Score' 的有序关系模式变为 Score' (SID, <Java, OS, DBMS>)。攻击前对 Score 进行水印检测能成功检测到水印;在 ORSM 攻击后,由于带水印的属性位置发生变化,使得水印检测系统不能成功地检测到水印。可见,传统的数据库水印算法对于 ORSM 攻击无能为力。

为了消除传统数据库水印算法对有序关系模式的依赖性,提高数据库水印在 ORSM 攻击下的鲁棒性,本文提出了一种基于属性标记的关系数据库水印算法。

2 基于属性标记的数据库水印算法

基于属性标记的水印算法(Attribute labeling based database watermarking, ALBWM)主要包含水印嵌入和水印检测这两部分。首先给出 ALBWM 算法的工作流程,然后再给出具体算法步骤。

2.1 ALBWM 工作流程

ALBWM 算法的核心思想是假设任意关系模式中的属性都拥有唯一的标记,根据属性的标记确定一种标准的有序关系模式,水印的嵌入和检测都在这个标准有序关系模式下进行,其工作流程如图 2 所示。

水印调度单元负责水印嵌入和水印检测任务的调度,水印嵌入模块对输入的关系进行属性标记,获得标准的有序关系模式,然后选择水印位置并且嵌

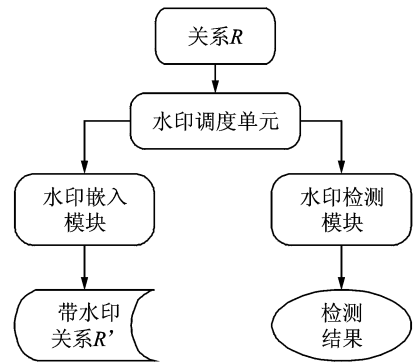


图2 ALBWM 的工作流程

入水印信息。水印检测模块对输入的关系进行属性标记,获得标准的有序关系模式,然后选择水印位置,进行水印信息匹配并判断水印是否存在。

2.2 ALBWM 算法描述

ALBWM 算法利用单向 HASH 函数(如 MD5, SHA 等),并以关系表的属性信息和数据所有者的密钥信息作为种子计算 HASH 值,根据 HASH 值将属性映射到较大的地址空间(MSpace);然后,根据映射的地址对属性进行标记,从而确定候选水印属性序列的标准顺序。在水印嵌入和提取前,针对待处理的有序关系模式,用相同方法对各属性进行标记,从而恢复标准有序关系模式,利用上述处理过程确保在水印嵌入与提取时的属性选取的一致性。该算法主要包括水印的嵌入算法(ALBWM-EM)和检测算法(ALBWM-DT),具体内容如算法 1 和 2 所示。

算法 1. 基于属性标记的水印嵌入算法(ALBWM-EM) 输入:待嵌入水印的关系 R , 候选属性序列 CA_Se

quence, 密钥 K , 属性映射空间 $MSpace$

输出: 嵌入水印的关系 R'

```
(1) 提取  $R$  的有序关系模式  $\hat{R}(P, \langle U \rangle)$ , 获得  $\langle U \rangle$ 
中候选属性序列  $CA\_Sequence$ ;
(2) 初始化  $MSpace$  为空;
(3) FOR  $\forall a(a \in CA\_Sequence)$  // 对候选水印序列中的
属性进行标记
(4)    $a.label = HASH(a, K) \bmod MSpace.len$ ;
(5) END FOR
(6) FOR  $\forall r(r \in R)$ 
(7)   IF  $r$  为待嵌入水印的记录 THEN
(8)      $SelectLocation(r, CA\_Sequence, K, MSpace)$ ; // 调用水印位置选择函数, 获取水印位置信息  $attribute\_index$  和  $bit\_index$ , 见算法 3。
(9)     计算水印信息  $wm\_bit$ ;
(10)    将  $CA\_Sequence[attribute\_index]$  的第  $bit\_index$  位的值替换为  $wm\_bit$ ; // 嵌入水印
(11)   END IF
(12) END FOR
(13) RETURN  $R'$ ;
```

算法 2. 基于属性标记的水印检测算法 (ALBWM-DT)

输入: 待检测水印的关系 R , 候选属性序列 $CA_Sequence$, 密钥 K , 误检率水平 α , 属性映射空间 $MSpace$

输出: $FALSE$ 表示无水印, $TRUE$ 表示有水印

```
(1) 提取  $R$  的有序关系模式  $\hat{R}(P, \langle U \rangle)$ , 获得  $\langle U \rangle$ 
中候选属性序列  $CA\_Sequence$ ;
(2) 初始化  $MSpace$  为空;
(3) FOR  $\forall a(a \in CA\_Sequence)$  // 对候选水印序列中的
每个属性进行标记
(4)    $a.label = HASH(a, K) \bmod MSpace.len$ ;
(5) END FOR
(6) FOR  $\forall r(r \in R)$ 
(7)   IF  $r$  为待检测水印的记录 THEN
(8)      $SelectLocation(r, CA\_Sequence, K, MSpace)$ ; // 调用水印位置选择函数, 见算法 3
(9)     计算水印信息  $wm\_bit$ ;
(10)    IF  $CA\_Sequence[attribute\_index]$  的第  $bit\_index$  位的值与  $wm\_bit$  相同 THEN
(11)       $matchcount++$ ; // 如果水印信息匹配
成功, 匹配计数器加 1
(12)    END IF
(13)       $totalcount++$ ;
(14)    END IF
(15)  END FOR
(16)  $\tau = GetThreshold(totalcount, \alpha)$ ; // 计算水印检测
的匹配阈值
(17) IF  $matchcount \geq \tau$  THEN // 如果实际匹配数
不小于匹配阈值
(18)   RETURN  $TRUE$ ; // 返回有水印
```

```
(19) ELSE // 否则返回无水印
```

```
(20)   RETURN  $FALSE$ ;
```

```
(21) END IF
```

算法 3. 水印位置选择函数 $SelectLocation(r, CA_Sequence, K, MSpace)$

输入: 记录 r , 候选属性序列 $CA_Sequence$, 密钥 K , 属性映射空间 $MSpace$

输出: 水印属性号 $attribute_index$, 水印比特位 bit_index

```
(1)    $map\_index = HASH(r.P, K) \bmod MSpace.len$ ; // 计算映射空间水印嵌入位置的索引
(2)   IF  $\exists a(a.label = map\_index \wedge a \in CA\_Sequence)$  THEN // 若存在候选标记与索引值匹配
(3)     选择  $a$  为水印属性;
(4)   ELSE
(5)     寻找属性标记中与  $map\_index$  最近的属性  $a$  为水印属性;
(6)   END IF
(7)   获得  $a$  在  $CA\_Sequence$  中索引号  $attribute\_index$ ;
(8)   确定水印嵌入的比特位索引号  $bit\_index$ ;
(9)   RETURN  $attribute\_index$  和  $bit\_index$ ;
```

算法 1 和算法 2 的步骤 (1~5) 给出了通过属性到地址空间的映射对属性进行标记的过程, 其中, $HASH()$ 为单向哈希函数, $MSpace.len$ 为映射空间的大小, 这个过程中密钥的参与保证了只有密钥所有者才能得到正确的候选水印属性标记。算法 2 中 $totalcount$, $matchcount$ 分别为水印检测时比较的位数和匹配数, 用 $GetThreshold()$ 函数计算水印检测的匹配阈值, 具体通过伯努利方程 $B(\tau, totalcount, 1/2) < \alpha$ 来获得 τ 。算法 1 和算法 2 中都调用了函数过程 $SelectLocation()$ 选择水印的具体位置 (具体过程见算法 3)。算法 3 中, 步骤 (1~7) 在 $CA_Sequence$ 中选定水印属性, 步骤 (8) 选择具体的水印比特位, 最后返回选定位置的属性号和比特位号, 其中 $r.P$ 为记录 r 的主键。

ALBWM 算法主要利用属性到冗余空间的映射来实现属性的标记, 在映射过程中发生冲突的属性拥有相同的标记。在水印位置选择时, 对相同标记的属性都进行水印的嵌入或检测。这种做法是以降低水印的隐蔽性为代价的。为了减少隐蔽性的损失, 即减少冲突, 应该选取散列效果较好的哈希函数和较大的映射空间, 但这也影响水印分布的均匀程度, 所以要综合考虑数据允许的误差范围和哈希函数的性能来进行映射空间大小的选取, 从而使水印获得较好的鲁棒性。

由上述 ALBWM 算法过程及分析可知,ALBWM 算法在水印选择策略上与 AgWM 类似,水印信息都能均匀地分布于整个关系中,但 ALBWM 算法中增加了属性标记的过程,使水印位置的选择在标准的有序关系模式上进行,从而消除了水印算法对有序关系模式的依赖。因此,同 AgWM 算法相比,对子集选取、属性删除等传统的水印攻击方法,ALBWM 算法同样具有较好的鲁棒性;而对 ORSM 攻击,ALBWM 算法则具有更好的抗攻击性能。

3 实验结果及分析

实验的硬件环境为 Intel Pentium (R) 2.66 GHz 处理器,1.25 GB 主存,40 GB 硬盘;软件环境为 Windows XP 操作系统、SQL Server 2000 数据库系统和 VC++6.0 开发工具。

实验采用 University of California-Irvine KDD Archive 的标准数据集 Forest Cover Type dataset^[19]。该数据集包括 581 012 条记录,每条记录有 61 个属性。为了实验需要,在每条记录前增加一列属性作为主键,单向哈希函数选用 MD5,每次实验都采用 100 组不同的样本数据,结果取 100 次实验结果的平均值。

3.1 水印嵌入和检测过程执行时间对比实验

本实验对 ALBWM 和 AgWM 的水印嵌入和检测过程执行时间进行比较。本组实验中取数据集的前 20 个属性作为候选水印属性,取数据的最后一位作为候选水印比特位,ALBWM 中属性映射空间 MSpace 设置为 103。

实验中分别采用 ALBWM 和 AgWM 在相同条件下进行水印的嵌入和检测,在不同的数据集规模下,对比两种算法在水印嵌入和检测过程中消耗的时间。实验结果如图 3 所示。

由 2.2 节中的算法分析可知,ALBWM 较 AgWM 的额外时间开销主要来自属性标记的过程。从图 3 中可以看出,ALBWM 和 AgWM 的水印嵌入和检测过程消耗的时间相差很小。其中,ALBWM 的水印嵌入过程的执行时间仅比 AgWM 高了 3.61%,水印检测的执行时间也仅高了 3.35%。实验结果表明,ALBWM 中的属性标记过程对水印嵌入和检测性能的影响很小。

3.2 属性映射空间对水印分布影响实验

ALBWM 算法中通过候选水印属性到 MSpace 的映射实现属性的标记,来提高水印在 ORSM 攻击下的鲁棒性,而 MSpace 的不同取值会影响水印

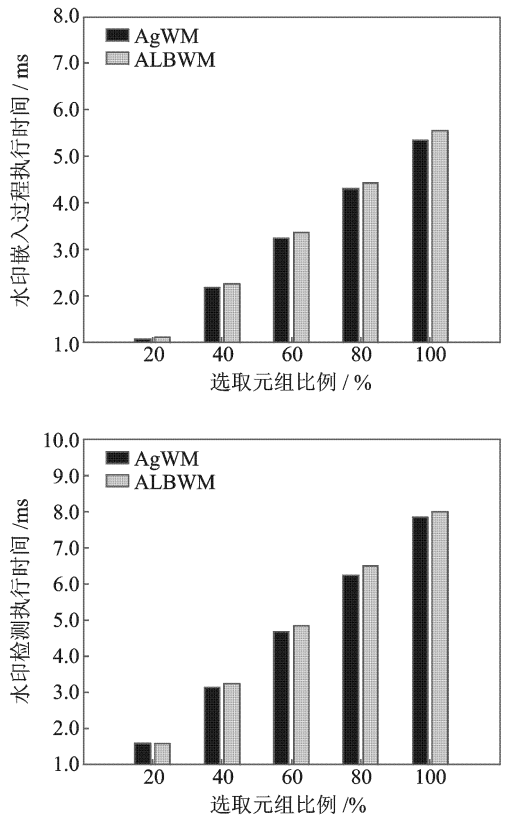


图3 ALBWM 和 AgWM 执行时间比较

的嵌入位置在候选水印属性上的分布。本实验对不同大小的映射空间 MSpace 下 ALBWM 的水印嵌入分布的均匀情况进行比较。本组实验中选择数据集前 20 个属性作为候选水印属性,对数据集的全部元组记录进行水印嵌入。

实验中用 ALBWM 算法选择水印嵌入的位置,在不同映射空间 MSpace 取值下,每个候选水印属性上水印分布均匀性差异度计算方法如式(1)所示。

$$\delta = \left| \frac{x - AVG}{AVG} \right| \quad (1)$$

式中: x 为候选水印属性上实际嵌入的水印次数; AVG 表示水印均匀分布时每个候选水印属性上嵌入的水印次数。最后将 δ 按从小到大的顺序进行排列。实验结果如图 4 所示。

由图 4 可知,不同的 MSpace 对水印嵌入位置分布的均匀程度产生不同的影响,其中,当 MSpace = 103 时,每个候选水印属性上的 δ 都低于 MSpace 为 23 和 997 时的 δ 。这表明,属性映射空间大小取 103 时的水印分布均匀程度明显优于 MSpace 取 23 和 997 时的水印分布均匀度。由此可见,MSpace 的选取与水印分布情况密切相关,MSpace 取值过大或过小都不利于水印的均匀分布,所以对于不同数量

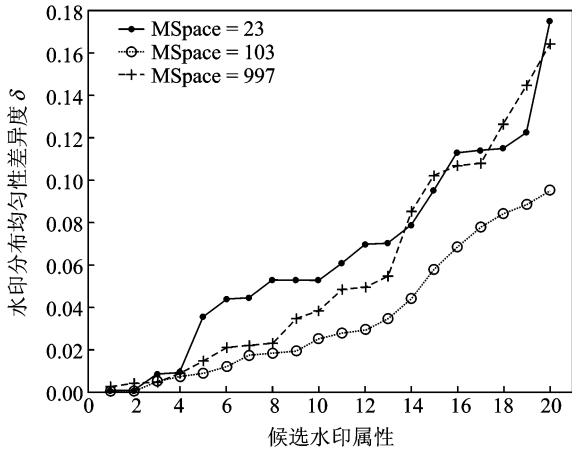


图4 MSspace对ALBWM的水印分布均匀性的影响

的候选水印属性,需要选取多个属性映射空间取值进行比较,选择对水印分布均匀性影响最小的值。

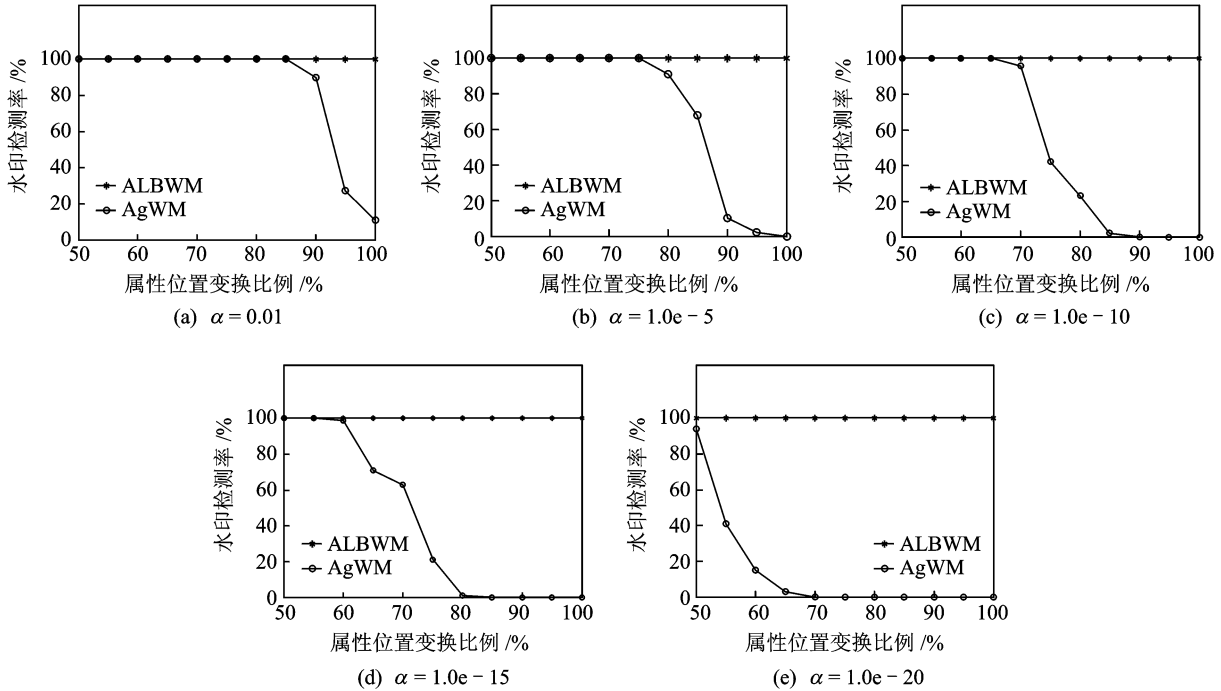


图5 ALBWM和AgWM在ORSM攻击下的鲁棒性对比

测率开始下降。随着 α 的降低,使水印检测率发生下降的属性位置变换比例也逐步减小。当 α 取 $1.0e-20$ 时,水印检测率在属性位置变换比例为50%时开始迅速下降,在变换比例达70%时,水印检测率降至0%。

实验结果表明,ORSM攻击对AgWM有较好的攻击效果,特别是在误检率水平较低的情况下,攻击效果更加显著,而ALBWM的水印检测率不受 α 的影响,在任意强度的ORSM攻击下始终保持100%。

3.3 鲁棒性对比实验

本实验对ALBWM和AgWM分别进行ORSM攻击,并对比两种水印算法的抗攻击性能。本组实验中取数据集的前20个属性作为候选水印属性,取数据的最后一位作为候选水印比特位,选取0.1%的记录进行水印嵌入,映射空间大小为103。

实验中分别采用ALBWM算法和AgWM算法进行水印的嵌入,在不同的误检率水平 α 下,对添加水印的关系进行不同比例的属性位置变换,然后再用两种算法对水印进行检测,并对比水印检测率。实验结果如图5所示。

由图5可知,在不同的误检率水平 α 取值下,随着属性位置变换比例增大,AgWM的水印检测率开始明显偏离100%,并迅速降至或者趋近0。若 α 取0.01时,当属性位置变换比例达到90%,水印检

综合以上分析可知,对于ORSM的攻击,本文提出的ALBWM算法则与传统水印算法相比具有更好的鲁棒性。

4 结束语

数据库水印的鲁棒性是衡量水印算法性能的重要标准之一,也是数据库水印能有效实现数据库版权保护的重要保证。本文在分析传统数据库水印算法的基础上,提出了基于属性标记的数据库水印算法,弥补了现有算法无法抵抗ORSM攻击的缺

点。并通过执行时间对比实验和鲁棒性对比实验,验证了 ALBWM 算法对执行性能影响很小,且在 ORSM 攻击下较现有数据库水印算法具有更强的鲁棒性。由于数据库水印算法通常需要处理大规模数据,通过分析每条记录来选择水印位置并进一步进行水印的嵌入或检测,算法的时间复杂度较高,因此在后续工作中,拟在提高算法效率方面做进一步的研究,使数据库水印技术有更广泛的应用。

参考文献:

- [1] Agrawal R, Kiernan J. Watermarking relational databases [C]//Proc of the 28th International Conference on Very Large Data Bases. Berlin Heidelberg: Springer-Verlag Press, 2002:155-166.
- [2] Agrawal R, Haas P, Kiernan J. A system for watermarking relational databases [C]//Proceeding of the 2003 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2003: 674-675.
- [3] Agrawal R, Haas P, Kiernan J. Watermarking relational data: framework, algorithms and analysis [J]. The VLDB Journal, 2003, 12(2): 157-169.
- [4] Guo Huiping, Li Yingjiu, Liu Anyi, et al. A fragile watermarking scheme for detecting malicious modifications of database relations [J]. Information Sciences, 2006, 176(10): 1350-1378.
- [5] Li Yingjiu, Guo Huiping, Jajodia S. Tamper detection and localization for categorical data using fragile watermarks[C]//Proc of the 4th ACM workshop on Digital Rights Management. New York: ACM Press, 2004: 73-82.
- [6] Prasannakumari V, Balu V. Audio database watermarking for tamper detection[J]. Communications in Computer and Information Science, 2011, 142(3): 446-448.
- [7] Li Yingjiu, Swarup V, Jajodia S. Fingerprinting relational databases: schemes and specialties [J]. IEEE Transactions on Dependable and Secure Computing, 2005, 2(1): 34-45.
- [8] Lafaye J, Gross-Amblard D, Constantin C, et al. Watermill: an optimized fingerprinting system for databases under constraints [J]. IEEE Transactions on Knowledge and Data Engineering, 2008, 20(4): 532-546.
- [9] David G. Query-preserving watermarking of relational databases and XML documents [J]. ACM Transactions on Database Systems (TODS). 2011, 36(1): 3185-3196.
- [10] 郑吉平,秦小麟,崔新春. 基于数字水印的数据库角色访问控制模型[J]. 电子学报, 2006,34(10): 1906-1910.
- [11] Zheng Jiping, Qin Xiaolin, Cui Xinchun. Digital watermark based database model using RBAC [J]. Acta Electronica Sinica. 2006,34(10): 1906-1910.
- [12] 王珊,萨师焯. 数据库系统概论[M]. 4版. 北京:高等教育出版社, 2006: 41-47.
- [13] Wang Shan, Sha Shixuan. An introduction to database systems[M]. 4th edition. Beijing: Higher Education Press, 2006: 41-47.
- [14] Sion R, Atallah M, Prabhakar S. On watermarking numeric sets[C]//First International Workshop on Digital Watermarking. Berlin Heidelberg: Springer-Verlag Press, 2003: 130-146.
- [15] Li Yingjiu, Swarup V, Jajodia S. Constructing a virtual primary key for fingerprinting relational data [C]//Proceeding of the 3rd ACM workshop on Digital Rights Management. New York: ACM Press, 2003: 133-141.
- [16] Gupta G, Pieprzyk J. Reversible and blind database watermarking using difference expansion [J]. International Journal of Digital Crime and Forensics, 2009, 1(2): 42-54.
- [17] Burepalli R, Munaga K. Prasad: Subset selection approach for watermarking relational databases [C] //Proceeding of 2rd International Conference Data Engineering and Management. Berlin Heidelberg: Springer-Verlag Press, 2010:181-188.
- [18] Mahmoud F, Shi-Jinn H, Jui-Lin L, et al. A blind reversible method for watermarking relational databases based on a time-stamping protocol[J]. Expert Systems with Applications, 2012, 39(3): 3185-3196.
- [19] 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(12A): 2050-2053.
- [20] Niu Xiamu, Zhao Liang, Huang Wenjun, et al. Watermarking relational databases for ownership protection [J]. Acta Electronica Sinica, 2003,31(12A): 2050-2053.
- [21] Jiang Chuanxian, Chen Xiaowei, Li Zhi. Watermarking relational databases for ownership protection based on DWT[C]//Proceedings of the Fifth International Conference on Information Assurance and Security. Providence, Rhode Island: IEEE Computer Society, 2009:305-308.
- [22] University of California-Irvine KDD Archive. Forest cover type dataset [EB/OL]. <http://kdd.ics.uci.edu/databases/coverttype/coverttype.html>. 1998-09-04/2011-3-04.