第34卷 第18期 计 算 机 工 程 2008年9月
Vol.34 No.18 Computer Engineering September 2008
·安全技术· 文章编号：1000—3428(2008)18—0191—03 文献标识码：A 中图分类号：TP309

基于水印的数据库自适应信息隐藏算法

黄冬梅[1]，朱仲杰[2]，王玉儿[1,2]
(1. 上海水产大学信息学院，上海 200090；2. 浙江万里学院宁波市DSP重点实验室，宁波 315100)

摘 要：提出一种基于水印的自适应信息隐藏算法用于数据库的知识产权保护。算法根据数据库中原始数据的特点自适应地选择最佳嵌入方案，使水印的嵌入对数据库的影响降到最低，利用纠错编码技术和多数选举原理提高算法的稳健性。实验仿真结果表明，该算法满足不可见和盲检特性，对子集删除、增加、更新等常见攻击具有良好的抵抗能力。

关键词：数据库水印；知识产权保护；鲁棒性；不可见性；自适应算法

Adaptive Information Hiding Algorithm for Database Based on Watermark Technology

HUANG Dong-mei[1], ZHU Zhong-jie[2], WANG Yu-er[1,2]
(1. College of Information Technology, Shanghai Fisheries University, Shanghai 200090;
2. Ningbo Key Lab for DSP, Zhejiang Wanli University, Ningbo 315100)

【Abstract】A watermark-based adaptive information hiding algorithm is proposed for the Intellectual Property Rights(IPR) protection for database. The algorithm adaptively chooses an optimal embedding scheme according to the characteristics of the data to make the degradation due to the watermark embedding to be smallest. And the error-correcting coding technology and the majority voting principle are employed to improve the algorithm's robustness. Experimental results reveal that the algorithm can meet the requirements of invisibility and blind detection as well as good resistance to conventional attacks such as subset deleting, adding, and updating.

【Key words】database watermark; Intellectual Property Rights(IPR) protection; robustness; invisibility; adaptive algorithm

1 概述

随着数据库技术的广泛应用，数据库中数据的窃取、非法拷贝等问题越来越突出[1]。因此，面向数据库的知识产权保护技术越来越受到关注。由于传统的加密方法只能提供数据传输过程中的信息安全，当数据被接收并解密后，其信息就不再受到保护，无法防止盗版[2]，因此人们开始采用信息隐藏或信息伪装技术，将有用或重要的信息隐藏在其他信息中，以始终有效地标识著作权[3]。其中，基于水印的信息隐藏技术是近年来出现的一种可用于数字产品知识产权保护的有效的新技术，成为当前学术界的研究热点[2,4]。

本文基于数字水印技术，提出一种面向数据库的信息隐藏算法。算法引入自适应策略，根据数据库中数据的特点自适应地选择水印嵌入方案，使得水印嵌入对数据的改动达到最小，对数据库使用价值的影响降到最低。同时采用纠错编码技术和多数选举原理以提高水印算法的稳健性。

2 信息隐藏新算法

本文提出的自适应信息隐藏算法主要分为预处理、水印嵌入和水印提取3个步骤。预处理阶段主要是根据数据的特点，选择一种对数据影响最小的最优方案。水印嵌入和提取阶段根据选择的最优方案完成信息的嵌入和提取。

2.1 预处理

设 $W=\{w_k\}\ (0\le k<n-1)$ 表示原始水印图像的0, 1二进制序列，首先给定密钥Key对 W 进行置乱处理以增强算法的稳健性，置乱后的二进制序列用 $W'=\{w_k'\}\ (0\le k<n-1)$ 表示。同时为了保证关系数据库的使用价值不被破坏，引入数值可更改范围的约束限制，仅对满足约束限制的属性值嵌入信息。

设$R(P,A_1,A_2,\cdots,A_v\cdots)$是数据库中某一关系数据表，$P$为主键属性，$A_1,A_2,\cdots,A_v$为$v$个可嵌入水印的数值型属性列(不包括主键)，$R$由$n$个元组$r_1, r_2, \cdots, r_n$组成。每个元组$r$都有1个主键$r.p$和$v$个数值型属性$r.A_1, r.A_2, \cdots, r.A_v$。通过改变这$n\times v$个属性值的最低有效位，即可实现水印信息的嵌入。

为了提高算法的鲁棒性，在对数据库进行水印嵌入之前，利用单向hash函数计算r_i的标记[5]。设$index(r_i)$为r_i的标记号，则$index(r_i)=hash(K, r_i.P)$。水印信息可选择$r_i$的$A_1,A_2,\cdots, A_v$数值型属性中的一个或几个进行嵌入。根据标记号和属性选择方式，算法预定义4种不同的嵌入方案：

方案1 $f1=\text{hash}(\text{index}(r_i),K)\bmod v$，将信息嵌入$r_i.A_{f1}$中；

方案2 $f2=(\text{hash}(\text{index}(r_i), K)+a)\bmod v$，将信息嵌入到$r_i.A_{f2}$中；

方案3 $f3=(\text{hash}(\text{index}(r_i), K)+b)\bmod v$，将信息嵌入到$r_i.A_{f3}$中；

方案4 $f4=(\text{hash}(\text{index}(r_i), K)+c)\bmod v$，将信息嵌入到$r_i.A_{f4}$中。

其中，a, b, c 为互不相同的整数，且均小于 v。定义

基金项目：国家自然科学基金资助项目(60472100)；国家“863”计划基金资助项目(2006AA102239-1)；宁波市自然科学基金资助项目(2006A6100013)

作者简介：黄冬梅(1964－)，女，教授，主研方向：数据库技术；朱仲杰，教授、博士；王玉儿，硕士

收稿日期：2007-10-15 **E-mail：**zhongjiezhu@hotmail.com

$P=\{p_0p_1\}=\{00,01,10,11\}$ 为方案的二进制代码，分别表示 4 种嵌入方案。

设$q_0, q_1, \cdots, q_v$为$A_0, A_1, \cdots, A_v$属性的权重，其大小取决于属性的重要性及信息冗余程度，且满足$q_0+q_1+\cdots+q_v=1$。设$count_0, count_1, \cdots, count_v$为$A_0, A_1, \cdots, A_v$属性的改变量，初值均置为 0。对元组$r_i(1\le i\le n)$待嵌入信息的属性$A_{if}$，若嵌入的信息和$A_{if}$的最低有效位不同，则$A_{if}$的属性改变量$count_{if}=count_{if}+1$。

设 C 表示水印嵌入对 R 产生的影响，计算如下：

$$C=\sum_{j=0}^{v}((count_j / sum(abs(R.A_j)))\times q_j) \tag{1}$$

其中，$sum(abs(R.A_j))$ 为表R中所有元组的A_j属性的绝对值之和，即

$$sum(abs(R.A_j))=\sum_{i=0}^{n}abs(r_i.A_j) \tag{2}$$

设 $C_{方案1}$，$C_{方案2}$，$C_{方案3}$，$C_{方案4}$ 分别表示采用 4 种不同方案嵌入水印对数据表 R 产生的影响，定义 C^* 如下：

$$C^*=\min(C_{方案1}，C_{方案2}，C_{方案3}，C_{方案4}) \tag{3}$$

C^* 表示对 R 改动最小的最优方案的影响值，自适应算法就是根据 C^* 值来选择对 R 的水印嵌入方案，由式(4)给出：

$$p_0'p_1'=\begin{cases}00 & C^*=C_{方案1}\\ 01 & C^*=C_{方案2}\\ 10 & C^*=C_{方案3}\\ 11 & C^*=C_{方案4}\end{cases} \tag{4}$$

其中，$p_0'p_1'$ 表示嵌入方案的代码。对该代码进行重复编码，并作为水印的头信息一起嵌入到数据库中，即实际嵌入表 R 的水印序列为

$$W_R'=\{p_0'p_0'p_1'p_1'\}\cup\{w_k'\} \tag{5}$$

2.2 嵌入算法

某个关系表 R 的水印嵌入流程如下：

(1)利用(7, 4)汉明码对 $W'=\{w_k'\}$ $(0\le k<n-1)$ 进行编码，计算监督码，并将其作为水印提取时的密钥K_1。

(2)计算每个元组r_i的标记号，即$index(r_i)=hash(K, r_i.P)$。

(3)根据标记号模除$(n+4)$的结果对各个元组进行分组，如图 1 所示。对于元组r_i，计算$j=index(r_i)\bmod(n+4)$，分组的结果使得r_i位于S_j分组中。

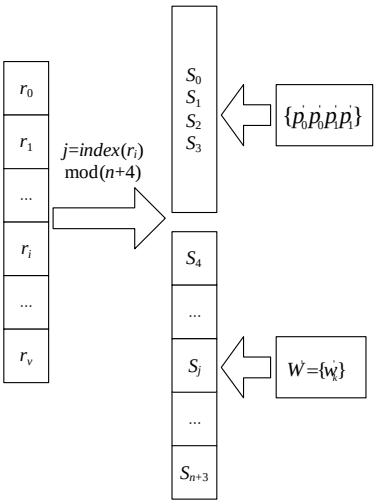

图 1　分组及自适应的水印嵌入

(4)$S_4, S_5, \cdots, S_{n+3}$分别对应 $W'=\{w_k'\}$ $(0\le k<n-1)$ 的各个比特信息，进行预处理，得出最佳方案。

(5)对方案代码重复编码所得的信息 $\{p_0'p_0'p_1'p_1'\}$ 进行(7, 4)汉明码编码，产生的监督码作为提取水印时的密钥K_2。利用事先预定的方案，对S_0, S_1, S_2, S_3嵌入方案代码 $\{p_0'p_0'p_1'p_1'\}$。

(6)根据选出的最佳方案，对分组$S_4, S_5, \cdots, S_{n+3}$嵌入 $W'=\{w_k'\}$ $(0\le k<n-1)$。

各个分组的水印嵌入方法为：对于分组中的每个元组，判断其选择嵌入的属性值是否满足该属性的约束限制。如不满足，则不进行操作，反之则嵌入信息。嵌入时，若水印信息为 0，则将该属性值的最低有效位置 0；反之则将其置 1。

2.3 提取算法

水印提取时，首先提取方案代码，根据方案代码判断嵌入时所采用的方法，再按照该方法提取所嵌入的信息，并利用校验码进行比特纠错。具体步骤如下：

(1)根据每个元组r_i的标记号计算$index(r_i)\bmod(n+4)$，利用所得的余数，对各个元组进行分组，如图 2 所示。

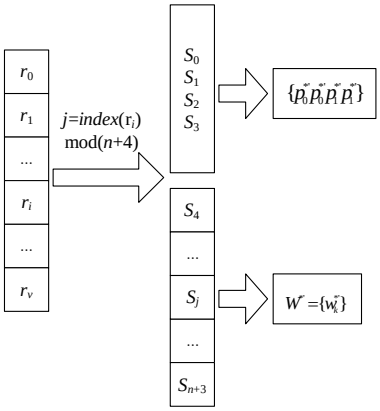

图 2　分组及自适应的水印提取

(2)利用事先定义的方案从S_0, S_1, S_2, S_3提取方案代码 $\{p_0^{*'}p_0^{*'}p_1^{*'}p_1^{*'}\}$，并根据监督码(密钥$K_2$)及多数选举方法对其进行纠错。

(3)根据 $p_0^{*'}p_1^{*'}$ 可确定水印序列的嵌入方案。

(4)根据方案所采用的方法，对$S_4, S_5, \cdots, S_{n+3}$提取对应的水印信息。

(5)利用监督码(密钥K_1)及多数选举原理对所得的水印信息进行校验纠错，得到 $W^{*'}=\{w_k^{*'}\}$ $(0\le k<n-1)$。

(6)根据密钥 Key，恢复原来的二进制序列 $W^*=\{w_k^*\}$ $(0\le k<n-1)$。

在分组中提取水印信号时，对每个元组采用方案提供的方法选择属性，判断该属性值是否满足所要求的约束条件。如不满足，则不操作，如满足，则提取水印信号，即若属性值的最低有效位为 0，则认为嵌入的信息为 0，反之为 1。

3　实验结果及分析

嵌入的二进制序列 $W=\{w_k\}$ 和提取的二进制序列 $W^*=\{w_k^*\}$ 可能并不完全一样。设m_0, m_1分别表示嵌入的水印中比特 0 和 1 的数目，e_0和e_1为相应正确检出的比特 0 和 1 的数目，定义水印正确检出率为

$$WDR=\frac{e_0e_1}{m_0m_1} \tag{6}$$

本算法嵌入的水印信息是一幅 32×32 的二值图像，如图 3 所示。

图 3　水印图像

实验所用的数据库是数据表 *R*1 和 *R*2。每个表选择了 7 080 个元组、4 个可嵌入信息的属性列。表 1 是 *R*1 和 *R*2 的数据情况(平均值和方差)及其属性权重。参数 *a*, *b*, *c* 分别为 1, 2, 3。

表 2 为分别采用方案 1, 2, 3, 4 及自适应选择的方案进行水印嵌入对*R*1 和*R*2 产生的影响*C*及总的影响。由表 2 可得，采用自适应方案对数据库中的*R*1, *R*2 进行水印嵌入后，对 2 个表的总的影响为 $13.563\ 43\times10^{-5}$，而采用任一种固定方案对数据总的影响均大于该值。同理，数据库中数据表越多，自适应方案的优点越明显。

表 1　所用数据表情况及各表的属性权重

表名	属性名	平均值	方差	权重
*R*1	attr1	－283.413 842	724 376.588 733	0.35
	attr2	23 331.517 090	8 212 880.124 203	0.2
	attr3	236 235.532 486	834 488 444.761 27	0.15
	attr4	2 389.195 198	84 276.621 280	0.3
*R*2	attr1	7 675.883 616	171 1031.769 984	0.25
	attr2	－741.103 531	527 879.162 853	0.4
	attr3	20 152.997 600	23 948 163.121 536	0.15
	attr4	－16 467.149 011	58 241 476.407 243	0.2

表 2　不同方案对各个数据表的影响*C*及总的影响　($\times10^{-5}$)

表名	自适应方案	方案 1	方案 2	方案 3	方案 4
*R*1	(方案 1)7.202 24	7.202 24	7.250 86	7.993 69	7.421 87
*R*2	(方案 3)6.361 19	6.500 15	6.646 04	6.361 19	6.687 58
总影响	13.563 43	13.702 39	13.896 90	14.354 88	14.109 45

图 4~图 6 为分别对数据表 *R*1 和 *R*2 进行子集删除、增加、更新的模拟攻击后提取水印的实验结果。

数据表	*R*1			*R*2		
删除比例/(%)	5	40	80	5	40	80
提取水印						
WDR	1	1	86.90%	1	1	89.60%

图 4　对 *R*1 和 *R*2 进行子集删除后所提取的水印信息和 *WDR* 值

数据表	*R*1			*R*2		
添加比例/(%)	5	40	80	5	40	80
提取水印						
WDR	99.88%	96.13%	88.36%	1	1	99.88%

图 5　对 *R*1 和 *R*2 进行子集增加后所提取的水印信息和 *WDR* 值

数据表	*R*1			*R*2		
更新比例/(%)	5	40	80	5	40	80
提取水印						
WDR	1	95.88%	61.33%	99.88%	97.56%	58.97%

图 6　对 *R*1 和 *R*2 进行子集更新后所提取的水印信息和 *WDR* 值

由图 4~图 6 可见，在子集删除、增加、更新的比例小于 40%时，能提取 95%以上的水印信息。随着数据被破坏程度的提高，水印的正确检出率降低，提取出的图片模糊。但是，在数据破坏程度达到 80%时，水印的正确检出率仍大于 55%。因此，算法具有较强的抵抗攻击能力。

4　结束语

本文提出了一种基于水印的数据库自适应信息隐藏算法。水印信息的嵌入对数据库原始数据的影响很小，水印具有良好的不可见性。进行模拟子集删除、增加、更新等攻击后，特别是在攻击的破坏程度不高的情况下，水印仍具有较高的正确检出率，水印算法具有较好的鲁棒性和抗攻击能力，并且满足盲检要求。

参考文献

[1] Agrawal R, Kiernan J. Watermarking Relational Databases[C]// Proceedings of the 28th VLDB Conference. Hong Kong, China: [s. n.], 2002: 105-109.

[2] 张春田, 苏育挺, 管晓康. 多媒体数字水印技术[J]. 通信学报, 2000, 21(9): 46-52.

[3] Johnson N F, Jajodia S. Exploring Steganography: Seeing the Unseen[J]. IEEE Computer, 1998, 31(2): 26-34.

[4] Ruanaidh J J K Ó, Csurka G. Watermarking Methods[C]//Proc. of the 26th International Conference on Computer Graphics and Interactive Techniques. Los Angeles, CA, USA: [s. n.], 1999-08-08.

[5] 张　浩, 黄　敏, 曹加恒. 数据库水印中的标记算法[J]. 计算机应用研究, 2005, 22(5): 42-44.

(上接第 190 页)

参考文献

[1] Shamir A. How to Share a Secret[J]. Communications of the ACM, 1979, 22(11): 612-613.

[2] Blakley G R. Safeguarding Cryptographic Keys[C]//Proceedings of American Federation of Information Processing Societies Conference. New York, USA: [s. n.], 1979.

[3] Ito M, Saito A, Nishizeki T. Secret Sharing Scheme Realizing General Access Structure[C]//Proceedings of the IEEE Global Telecommunications Conference. Tokyo, Japan: IEEE Press, 1987.

[4] Sun H M, Shieh S P. An Efficient Construction of Perfect Secret Sharing Schemes for Graph-based Structures[J]. Journal of Computers and Mathematics with Applications, 1996, 31(7): 129-135.

[5] Sun Hongmin, Shieh S P. Secret Sharing in Graph-based Prohibited Structures[C]//Proceedings of the IEEE International Conference on Computer Communications. Kobe, Japan: IEEE Press, 1997.

[6] 郭渊博, 马建峰, 王亚弟. 一种基于图的攻击结构的高效秘密共享方案[J]. 计算机研究与发展, 2005, 42(5): 877-882.