基于(t,n)门限的数据库水印算法研究

王具龙[1] 陈继红[2]
(南通大学电子信息学院 南通 226019)[1] (南通大学计算机科学与技术学院 南通 226019)[2]

摘 要 在已有水印算法的基础上提出一种基于(t,n)门限的数据库水印算法。算法首先利用改进的Lagrange插值多项式秘密地将版权水印分存,得到分存后的二进制水印信息,然后采用单向Hash函数对数据库中的元组进行标记,根据该标记和候选属性MSB位之间的奇偶关系确定水印嵌入位置,最终引入一种匹配关系将二进制水印信息嵌入到数据库的数值型属性中。实验结果表明,该算法具有很好的安全性和抗攻击能力。

关键词 数据库,门限,Lagrange插值多项式,水印
中图法分类号 TP309.2 **文献标识码** A

Database Watermarking Algorithm Research Based on (t,n) Threshold

WANG Ju-long[1] CHEN Ji-hong[2]
(School of Electronics and Information, Nantong University, Nantong 226019, China)[1]
(School of Computer Science and Technology, Nantong University, Nantong 226019, China)[2]

Abstract A new watermark technology for databases based on (t,n) threshold was presented in this paper. Firstly, (t,n) threshold is used to divide watermark into some shade shares. Secondly, the tuples of relational database are marked by their hash values. The position for watermark embedding is based on the parity deference between mark and Most Significant Bits of the attributes. Finally, the watermark information is embedded into the numerical attributes of the database based on a new matching rule. Experimental results show that this algorithm is safe and effective to prevent various of attacks.

Keywords Database, Threshold, Lagreange interpolation formula, Watermark

1 引言

随着网络技术的快速发展,数据库在当今社会各个领域尤其是在新兴的外包数据库领域[1]中扮演着至关重要的地位。在巨大应用价值的背后,数据库也面临着被攻击、窃取、破坏的危险,因此迫切需要一种机制来验证数据库数据的所有权、保护数据库数据的安全,数据库水印技术在解决这类问题上,具有很好的应用价值[2]。

数据库水印技术是一门新兴技术,目前成熟的算法并不多见。国外见报最早的数据库水印算法分别由IBM Almaden研究中心的R. Agrawal和Purdue大学的R. Sion提出。R. Agrawal提出了基于允许失真范围内数据变形的数据库水印算法[3],该算法的缺点是嵌入的是无意义水印。R. Sion等提出了基于元组排序和子集划分的数据库水印算法[4],该算法水印容量有限,运算开销大,难以实现水印同步嵌入要求。国内,牛夏牧教授等实现了将具有实际意义的字符串嵌入到数据库数值属性的最不重要位上[5],该算法无法抵抗截尾攻击。蒙应杰教授等提出了关系数据库零水印的构造算法[6],该算法不需要修改载体数据,但是水印检测必须依靠第三方检测机构。

为了更好地解决数据库水印的安全性和鲁棒性,本文提出了一种基于(t,n)门限的数据库水印算法。将代表版权信息的字符串或者图像作为水印信息,利用改进的拉格朗日插值多项式构造(t,n)门限方案,实现对水印信息的分存,相比以往的(t,n)门限水印算法而言,减少了分存后总水印的长度,增加了水印的嵌入量,提高了鲁棒性。水印嵌入根据属性标记值与属性MSB(most significant bits)位的奇偶性异同引入一种匹配关系,在不破坏数值型属性有效性的前提下,修改其LSB(least significant bits)位或者次LSB位,达到嵌入水印的目的。由于水印并不是嵌入在属性值的同一个位置,因此增加了水印的隐蔽性,提高了数据库被攻击的难度。本文所用数据库为关系型数据库,以下出现的数据库都是指关系数据库,对此不再赘述。

2 (t,n)门限方案

在密码学研究领域中,密钥的管理问题是衡量加密算法保密程度的重要因素之一。密钥的选取、存储和发放对信息的安全都发挥着至关重要的作用,如何妥善管理密钥成为信息安全领域中的一个重要问题。1979年,Shamir[7]和Blakley[8]分别提出了基于Lagrange和射影几何理论的(t,n)门限秘密共享方案,该方案较好地解决了密钥的管理问题。其基本思想是:在n个参与者组成的集体中共享密钥k,这样由任

到稿日期:2013-06-21 返修日期:2013-08-25 本文受国家自然科学基金(61202006),江苏省高等自然科学研究项目(12KJB 520014)资助。
王具龙(1988—),男,硕士,主要研究方向为数据库安全、数据库水印,E-mail:wjl_20130618@163.com;陈继红(1966—),男,硕士,副教授,主要研究方向为数据库安全、数据库与信息系统。

何 $t(t\leqslant n)$ 个参与者组成的子集都能重构密钥 k，但是小于 t 个参与者组成的子集将无法重构 k。

(t,n) 门限方案较好地解决了密钥的管理难题，将密钥 k 拆分给 n 个人，只要少于 t 人就无法恢复出密钥 k，同样，他们猜不到其他人所持有的部分，原始密钥 k 是十分安全的。需要恢复密钥 k 时，n 人中的任何 t 人都可以重构 k。该方案同样可以通过给某些人更多的部分来使某些人比其他人更重要。事实证明，这是一种既可靠又安全的密钥管理方案。

3 Lagrange 插值多项式

文本采用 Shamir 提出的基于 Lagrange 插值多项式的 (t,n) 门限方案，其基本思想主要如下[9]：假定要共享的密钥为 k，秘密独立随机地选择 $t-1$ 个元素，分别记为 $a_1,a_2,\cdots,a_{t-1}$，构造一个 $t-1$ 次随机多项式：

$$a(x)\equiv(k+\sum_{i=1}^{t-1}a_i x^i)(\mathrm{mod}\ p) \tag{1}$$

其中，p 为大于 k 和 a_i 的素数。每个参与者都得到了这个多项式的一个点 (x_i,y_i)。当有任意 t 个参与者想要重建密钥 k 时，我们可以利用 Lagrange 插值多项式求得密钥 k，即由

$$a(x)\equiv\sum_{i=1}^{t}y_i\prod_{\substack{j=1\\j\neq i}}^{t}\frac{x-x_j}{x_i-x_j}(\mathrm{mod}\ p)$$

可以得出：

$$k\equiv a(0)\equiv\sum_{i=1}^{t}y_i\prod_{\substack{j=1\\j\neq i}}^{t}\frac{-x_j}{x_i-x_j}(\mathrm{mod}\ p) \tag{2}$$

从而可以方便地计算出密钥 k。

4 关系数据库水印算法

本章详细介绍了水印信息的预处理过程、数据库元组的标记算法、候选属性的筛选条件以及水印嵌入算法、水印提取算法，其中水印嵌入算法和水印提取算法是关系数据库水印算法的核心内容。

4.1 水印信息的预处理

由于水印最终的形式都是二进制序列的形式，因此以有意义的字符串、文本或者图像作为水印在原理上都是一样的。为了方便描述，本文选取字符串作为水印。首先获得字符串的 ASCII 码十进制的形式，将该值作为一个大整数来处理，作为要分存的密钥 k。然后根据门限 t、n 的取值，将 k 按顺序分割成 t 部分，分别作为 Lagrange 多项式的 t 个系数(包括常数项)，其分割遵守这样的规则：多项式中次幂越高的项，对应的系数值越小。这样做的好处是最终得到的子水印的值相对来说会比较小，总水印长度大约为原始水印的 2.3 倍。以往门限水印算法[9,10]大多数是将 k 直接作为多项式的常数项，这种方法的弊端是得到的子水印跟原始水印的长度一样，总水印长度为原始水印的 5 倍。相对而言，本文提出的算法减少了预处理后待嵌总水印的长度，增加了嵌入量，提高了水印的鲁棒性。

4.2 标记算法

由于数据库具有行列顺序可以任意改变的特点，为了能够在数据库正常更新或者被恶意攻击之后方便地找到水印的嵌入位置，本文采用哈希函数对其元组进行标记[11]。哈希函数一般具有以下特点：

(1)任意长度的输入，固定长度的输出。

(2)正向计算比较容易，反向计算很难。

(3)很少发生冲突。即很难找到两个不同的输入，得到相同的输出。

以主键 P、密钥 Key 为参数计算每个属性的哈希标记值 $Hash(P,Key)$，在主键不被攻击的情况下，攻击者不知道密钥 Key，很难找到水印的嵌入位置来破坏水印。

4.3 水印嵌入算法

首先对算法中用到的一些参数进行描述，如表 1 所列。

表 1 参数定义

参数符号	定义
P	数据库中的主键
A_j	数据库中属性列
r_i	数据库中的元组
LSB	属性的最不重要位
LSB2	属性次最不重要位
L	二进制水印的长度
x	待嵌水印位的位置
index	属性的哈希标记值
u	候选属性控制参数
W_1	原始水印数组
W_2	提取出的水印数组
Key	哈希中密钥参数

嵌入算法描述如下：

1. 按 4.1 节的方法获得分存后二进制水印数组 W_1；

2. 循环数据库中每一个数值型 tuple；

3. 筛选候选属性 if(r_iA_j>u)；

4. 获得每个元组的哈希标记 index=Hash(P,Key)，以及对应的水印在 W_1 中的位置 x=mod(index,L)；

5. 嵌入水印

```
addmark(riAj,W1,x,index)
{
  if(mod(index,2)==mod(MSB,2))
  {
    switch(W1[x])
    {
      case 1:
        if(mod(LSB,2)==0)
          riAj=riAj+1;
        break;
      case 0:
        if (mod(LSB,2)==1)
          riAj=riAj-1;
    }
  }
    else
    {
      switch(W1[x])
      {
        case 1:
          if (mod(LSB2,2)==0)
            riAj=riAj+10;
          break;
        case 0:
          if (mod(LSB2,2)==1)
            riAj=riAj-10;
      }
  }
}
```

6. 更新数据库完成版权水印的嵌入。

4.4 水印提取算法

提取算法描述如下：

1. 分别定义水印存放数组 W_2，数组 S_0 和 S_1，S_0、S_1 分别存放每个位置上检测到 0、1 的个数；
2. 循环数据库中每一个数值型 tuple；
3. 筛选候选属性 if(r_iA_j>u)；
4. 获得每个元组的哈希标记 index=Hash(P,Key)，以及可能嵌入的水印在 W_1 中的位置 x=mod(index,L)；
5. 检测水印

```
Detectmark(riAj,W2,x,index)
{
  if(mod(index,2)==mod(MSB,2))
    {
    switch(mod(LSB,2))
    {
      case 1:S1[x]=S1[x]+1;
             break;
      case 0:S0[x]=S0[x]+1;
    }
  }
  else
{
    switch(mod(LSB2,2))
    {
      case 1:S1[x]=S1[x]+1;
             break;
      case 0:S0[x]=S0[x]+1;
    }
  }
}
```

6. 采用多数选举法，判断数组 S_0、S_1 相同位置上每个元素的大小，确定每个位置对应的水印，存放到 W_2 中，最终利用 Lagrange 多项式恢复版权水印。

5 实验结果与分析

实验以某自来水公司的数据库为载体，该数据库中共有 15000 个元组，每个元组有 23 个属性，其中月平均用水量、总户号、应缴水费等若干属性为数值型属性。本文以字符串“watermark”作为版权水印，采取(3,5)门限方案来实现数据库水印嵌入、提取算法。首先按照 4.1 节中的方法实现水印分存，得到分存后的二进制数组，然后将水印数组存入文本文件并导入数据库中，最终秘密选取哈希密钥 *Key*、候选属性列、控制参数 u(本文取 u=3)，利用 PL/SQL 语言编程实现水印的嵌入、提取操作。

5.1 实验结果

实验分别采用本文提出的数据库水印算法(简称算法 1)、基于未改进的 Lagrange 插值多项式的数据库水印算法(简称算法 2)对数据库进行了子集选取、子集更改、子集增加攻击，实验结果如下：

子集选取攻击：数据窃取者不使用水印数据库的全部属性和元组，仅仅使用其属性或元组的子集，从而希望擦除水印。选取的数据越少，对水印的破坏越大。图 1 示出了不同选取比例下水印的提取情况。

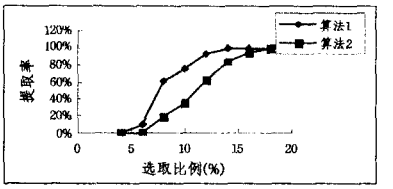

图 1 子集选取实验结果对比图

从图 1 中可以发现，算法 1 对子集选取攻击有较好的抵抗性。当选取比例大于 18%时，两种算法都可以准确提取原始水印，而当选取的比例少于 18%时，算法 1 对子集选取攻击的抵抗性明显要高于算法 2。

子集更改攻击：数据窃取者在窃取来的已经加入水印的数据库中，对部分子集的一些不重要位进行修改，企图破坏水印。对数据库的更改比例越大，对水印的破坏性就越强。图 2 示出了不同更改比例下水印的提取情况。

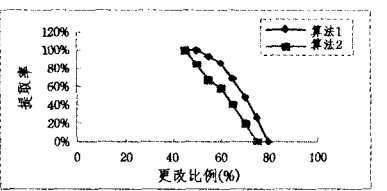

图 2 子集更改实验结果对比图

由图 2 可以看出，当更改比例少于 40%时，两种算法的实验结果相差不大，但是随着更改比例的增加，算法 1 的提取率明显要高于算法 2。更改比例达到 80%时，算法 1 和算法 2 都已经无法准确提取版权水印，但大比例修改数据，将影响原始数据的可用性，攻击者也无利益可图。

子集增加攻击：数据库攻击者在已经加入水印的数据库中增加一部分数据来削弱数据库水印的强度。添加的数据越多，对水印的影响越大，越不利于水印的提取。图 3 示出了不同增加比例下水印的提取情况。

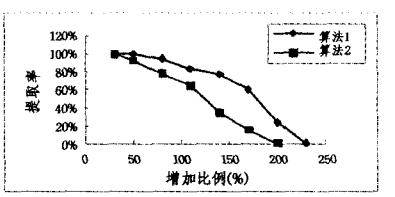

图 3 子集增加实验结果对比图

由图 3 可以发现，算法 1 在抵抗子集增加攻击方面性能较好一些，能容忍的最大增加比例大约为 230%，当增加比例大于 230%时，受到影响的子水印的个数都超过了门限值，导致无法提取原始水印。

5.2 性能分析

将(t,n)门限应用到数据库水印算法中，一方面，由于嵌入的是分存后的子水印，在一定程度上增加了嵌入水印的总长度，对固定容量数据库来说，相对减少了水印嵌入量，这是不利的一面。另一方面，对水印信息的分存处理分散了要嵌入的版权水印，减少了被攻击的风险，即使水印被攻击，只要被破坏的个数不超过 $n-t$，就仍然可以准确恢复原始水印，这是其有利的一面。采用改进的 Lagrange 插值多项式实现数据库水印算法，相比以往门限水印算法，相对减少了分存后嵌入水印的总长度，增加了水印嵌入量，一定范围内，嵌入量越

大，鲁棒性越强。另外，采用哈希标记和非单一位置嵌入，增加了水印的隐蔽性和安全性，可以有效抵抗元组重排以及截尾攻击，增加了水印的健壮性，并且降低了水印遭到破坏的可能性。

结束语 本文采用基于 Lagrange 插值多项式的(t,n)门限方案来实现数据库水印算法。主要工作有：(1)对插值多项式的改进，改进之后增加了嵌入量；(2)对嵌入算法的改进，增加了筛选条件以及水印的嵌入位置。实验结果表明，该算法安全性好，鲁棒性高，水印提取时只需要部分数据就可以恢复版权水印。若引入数字通信中的纠错编码理论或者密码学相关理论，将会进一步提高水印算法的安全性和鲁棒性。由于算法通过修改数据库中部分数值型属性的值来达到嵌入水印的目的，对于精度要求较高或者没有数值型属性的数据库则无法嵌入水印。因此，下一步要研究的内容是数据库零水印算法以及基于非数值型属性的数据库水印算法。

参 考 文 献

[1] 朱勤，陆志明. 基于信息隐藏的外包数据库版权保护系统[J]. 计算机科学，2010，37(1)：163-166

[2] 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

[3] Agrawal R，Kiernan J. Watermarking Relational Databases[C]// Proceedings of the 28th VLDB Conference. Hong Kong，China，2002

[4] Sion R，Atallah M. Sunil Prabhakar Rights Protection for Relational Data[C]//Proceedings of ACM SIGMOD. 2003

[5] 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

[6] 蒙应杰，吴超，是垚，等. 关系数据库零水印的构造算法[J]. 兰州大学学报：自然科学版，2003，4(3)：1-5

[7] Shamir A. How to share a secret [J]. Communications of the ACM，1979，24(11)：612-613

[8] Blakley G R. Safeguarding cryptographic keys[C]// Proceedings of the National Computer Conference，American Federation of Information Processing Societies. 1979，48：313-317

[9] 黄德才，姚瑶，刘瑞阳，等. 基于(t,n)门限的关系型数据库水印算法[J]. 计算机应用与软件，2009，26(7)：81-84

[10] 郑光明，欧阳竟成. 基于拉格朗日插值的关系数据库数字水印[J]. 计算机工程与设计，2008，29(6)：1585-1587

[11] 马瑞敏，陈继红. 基于二维空间元素匹配的数据库水印算法[J]. 计算机应用，2012，32(8)：2268-2270

(上接第 167 页)

结束语 本文根据当前群的统计信息，把信息熵和直方图的统计学技术引入当前群的信息分析，并运用它们去动态调整化学反应的元启发式算法的相关参数。运用实验去验证 PSACRO 算法的优越性，从实验结果来看达到了很好的效果，PSACRO 算法在处理时间和包匹配的时间消耗性能上都优于 PDEPM 算法，更优于 RCCRO 算法。

参 考 文 献

[1] Warkhede P，Suri S，Varghese G. fast packet classification for two-dimensional conflict-free filters[C]// Proceedings of IEEE，INFOCOM，Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Alaska：IEEEE，2001：1434-1443

[2] Gupta P，McKeown N. Algorithms for packet classification [J]. IEEE Network，2001，15(2)：24-32

[3] Srinivasan V，Varghese G，Suri S，et al. Fast and scalable layer four switching[C]//Computer Communication Review. Vancouver：ACM SIGCOMM，1998：191-202

[5] Buddhikot M M，Suri S，Waldvogel M. Space decomposition techniques for fast layer-4 switching[C]//Proc of Conf On Protocols for High speed Networks. Salem：IEEE，1999：25-41

[4] Feldman A，Muthukrishnan S. Tradeoffs for packet classification [C]//Proceedings of INFOCOMM，March. Aviv，Israel：IEEE，2000：1193-1202

[5] Gupta P，McKeown N. Packet classification using hierarchical intelligent cuttings[J]. IEEE Micro，2000，20(1)：34-41

[6] Singh S，Baboescu F，Varghese G，et al. Packet classification using multi-dimensional cutting[C]// Proceedings of The 2003 Conference on Applications，technologies，Architectures，and Protocols for Computer Communications. Karlsruhe：ACM SIGCOMM，2003：213-224

[7] Gupta P，Mckeown N. Packet classification on multiple fields [C]// Proc SIGCOMM，Computer Communication Review. Massachusetts：ACM SIGCOMM，1999：147-160

[8] van Lunteren J，Engbersen A P J. Multi-field packet classification using ternary CAM[J]. Electronics Letters，2002，38 (1)：21-23

[9] 李维，刘斌，郗颖，等. 基于多域并行编码的高速 IPV6 流分类[J]. 电子学报，2007，35(5)：976-981

[10] Sreelaja N K，Pai G A V. Ant colony optimization based approach for efficient packet filtering in firewall[J]. Applied Soft Computing，2010，10(4)：1222-1236

[11] Wang Ze-lin，Wu Zhi-jian，Zhang Bu-zhong. Packet matching algorithm based on improving differential evolution[J]. Wuhan University journal of natural sciences，2012，17(5)：447-453

[12] 王则林，吴志健. IPV6 环境下的高维大规模包匹配算法[J]. 电子学报，2013，41(11)：2181-2186

[13] Brest J，Member. IEEE. Self-Adapting Control Parameters in Differential Evolution：A Comparative Study on Numerical Benchmark Problems[J]. IEEE Transactions on Evoloutionary Computation，2006，10(6)：646-657

[14] Yang Z，He J，Yao X. Making a Difference to Differential Evolution，in Advances in Metaheuristics for Hard Optimization[C]// New York：Springer-Verlag，2007：415-432

[15] Wang C M，Huang Y F. Self-adaptive harmony search algorithm for optimization[J]. Expert Syst Appl，2010，37(4)：2826-2837

[16] Bingul Z，Sekmen A，Zein-Sabatto S. Adaptive genetic algorithms applied to dynamic multi-objective problems[C]// Dagli C H，Buczak A L，Ghosh J，eds. Proc. Artificial Neural Networks Engineering Conf.，Newyork，2000：273-278

[17] Lam A Y S，Li V O K. Chemical-reaction-inspired metaheuristic for optimization[J]. IEEE Transactions on Evolutionary Computation，2010，14(3)：381-339