基于关系数据库的脆弱性水印算法研究

张立忠,姜 楠,张 洋
ZHANG Li-zhong,JIANG Nan,ZHANG Yang

沈阳化工学院 计算机科学与技术学院,沈阳 110142
School of Computer Science and Technology,Shenyang Institute of Chemical Technology,Shenyang 110142,China
E-mail:zhanglizhong@syict.edu.cn

ZHANG Li-zhong,JIANG Nan,ZHANG Yang.Research of fragile watermarking algorithm based on relational database. Computer Engineering and Applications,2008,44(29):157-160.

Abstract: A fragile watermarking algorithm is proposed to detect malicious modifications of relational databases.In the proposed algorithm,all tuples in relational databases are divided into different groups,and by sorting secretly the tuples in each group, group watermarking matrix made up of attribute watermark and tuple watermark is generated for localizing modifications of relational databases in a group.The watermark is generated dynamically by using one-way hash function and relational data for watermark security and blind extraction.Theoretical analysis and experimental results show that the proposed method can detect tuple insertion,attribute value modification,tuple deletion and attribute variation made to relational database for providing authenticity verification of relational data.

Key words: fragile watermarking;relational database;watermarking matrix;group;modification

摘 要:为了检测对关系数据库的恶意篡改,提出了一种脆弱性数字水印算法。该算法将数据库的元组划分到不同的分组中,在对每个分组内的元组进行秘密排序的基础上,生成由属性水印和元组水印构成的分组水印矩阵,因此可以将对数据库的篡改定位在分组范围内。利用单向哈希函数及关系数据动态生成水印,不但保证了水印信息的安全性,而且也实现了水印的盲检测。理论分析和实验结果表明,该方法能够有效探测攻击者对关系数据库进行元组添加、属性值修改、元组删除和属性变化四类操作,从而为关系数据的真实性认证提供依据。

关键词:脆弱性水印;关系数据库;水印矩阵;分组;篡改
DOI:10.3778/j.issn.1002-8331.2008.29.044 **文章编号**:1002-8331(2008)29-0157-04 **文献标识码**:A **中图分类号**:TP309.2

1 引言

目前,数值型数据得到了广泛的分布与应用。由于数值型数据极易遭到复制和破坏,因此,对数据内容的完整性实施有效保护已引起普遍关注。基于密码学的传统加密技术已成熟地应用于加密状态下数字媒体内容的保护上。但是,加密-解密技术对解密后的数据不能提供进一步的保护。另外,密码学中的数据内容认证方法不仅需另外保存信息认证码,而且认证不容许有一比特的改变,数字水印技术则克服了传统加密方法的缺点。数字水印技术[1]通过将一个秘密编码嵌入到数据中去跟踪数据被非法拷贝和处理的情况。被嵌入的秘密编码,称之为数字水印。由于关系数据库中存在着大量的数值型数据,所以研究关系数据库水印技术具有非常重要的应用前景。

近些年来,研究人员已经意识到数据库水印的重要性,并提出了一些保护关系数据库产权的水印方案。Agrawal 等人[2]提出了一种健壮性数据库水印方案。Li 等人[3]对该方案进行了扩展。Sion 等人[4]对于健壮性数据库水印方案给出了一种不同的实现方法。在国内,绝大多数关系数据库数字水印技术的研究[5-8]都是为版权保护而设计的健壮性水印,仅有少量关于脆弱性数据库水印研究[9]的报道。文献[9]的脆弱性水印算法,是假设数据库的元组划分到各分组以后,每个分组的元组数目是固定值,且并没有提到该算法对属性顺序变化及属性删除的情况具有检测能力。针对这些不足,本文在现有的保护数据库关系完整性(主要是数据的真实性)的脆弱性水印理论和实验的基础上,提出一种脆弱性数据库水印算法,它通过构造水印矩阵来定位攻击者对关系数据库的篡改。

2 脆弱性水印算法

2.1 研究前提

假定关系数据表中所有的属性都是数值型的,并且在数据的使用范围内,关系数据表的属性值能够容忍由于水印嵌入而带来的微小改动。同样是在这个前提下,不允许对关系数据库中的主码做改动。

作者简介:张立忠(1976-),男,讲师,主要研究领域为多媒体信息安全,软件工程;姜楠(1977-),女,讲师,主要研究领域为个性化技术,网络信息安全;张洋(1979-),女,讲师,主要研究领域为数据挖掘,数字水印。

收稿日期:2007-11-27 **修回日期**:2008-02-21

在不失一般性的情况下,设数据库关系表为 $R(P,A_1,A_2,\cdots,A_s)$,其中 P 为主码,$A_1,A_2,\cdots,A_s$ 为 s 个数值型属性,R 由 n 个元组 $r_1,r_2,\cdots,r_n$ 组成,元组 r 的主码值记为 $r.P$,第 i 个元组的第 j 个属性值记为 $r_i.A_j$,分组数目记为 g,水印密钥记为 K,第 i 个分组内的元组数目记为 g_i。

2.2 水印嵌入

在嵌入水印之前,需对关系表做元组划分和排序的预处理,其主要步骤为:

(1)根据分组数目 g,水印密钥 K 与元组主码 $r.P$ 的哈希值将关系表中所有元组划分到不同的分组中。水印密钥 K 及分组数目 g 为数据库所有者私有。

(2)以分组为单位,对分组内的元组基于主码进行排序。

水印嵌入以分组为单位进行,对分组内所有元组嵌入两种水印:属性水印和元组水印,主要步骤包括:

(1)属性水印生成:根据水印密钥 K 和同一属性的所有值(每个属性值排除至少 2 个不重要位)得到属性哈希值,然后从该哈希值的重要位上取 g_i 个二进制位作为属性水印。

(2)属性水印嵌入:将属性值 $r.A_i$ 的最不重要位(即第 0 位)设为相应的属性水印位。

(3)元组水印生成:根据水印密钥 K 和同一元组的所有属性值(每个属性值排除至少 2 个不重要位)得到元组哈希值,然后从该哈希值的重要位上取 s 个二进制位作为元组水印。

(4)元组水印嵌入:将属性值 $r.A_i$ 的次最不重要位(即第 1 位)设为相应的元组水印位。

通过这样的方式,属性水印标识了分组内元组所在的列,元组水印则标识了分组内元组所在的行,从而构成分组的水印矩阵(如图 1 所示),为检测对数据的篡改奠定基础。

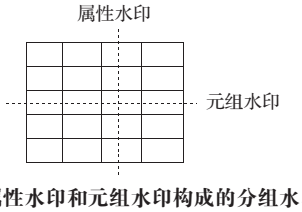

图 1 属性水印和元组水印构成的分组水印矩阵

具体的水印嵌入算法如下:

```
for i=1 to n do//元组的划分
begin
  H_i=Hash(K,r_i.P)
  m=H_i mod g
  group_m=r_i//将第 i 个元组划分到第 m 个分组中
end
for i=1 to g do
begin
  根据主码对第 i 个分组内的元组进行排序
  for y=1 to s do
   begin
  for x=1 to g_i do
   begin
    C_y=C_y+copy(group_i.r_x.A_y,1,length(group_i.r_x.A_y)-2)//排除每个属性值的两个不重要位后构成的字符串
   end
  W_iA_y=ExtractWMBits(Hash(Key,C_y),g_i)//从 C_y 的哈希值中生成 g_i 个属性水印位
  for x=1 to g_i do
   begin
    group_i.r_x.A_y[length(group_i.r_x.A_y)]=W_iA_y[x] //将属性值 group_i.r_x.A_y 的最不重要位(即第 0 位)设为相应的属性水印位
   end
  end
 for x=1 to g_i do
  begin
   for y=1 to s do
    begin
     tuple_str=tuple_str+copy(group_i r_x.A_y,1,length(group_i r_x.A_y)-2)
    end
  W_i T_x=ExtractWMBits(Hash(Key,tuple_str),s)//从 tuple_str 的哈希值中生成 s 个元组水印位
   for y=1 to s do
    begin
     group_i.r_x.A_y[length(group_i.r_x.A_y)-1]=W_i T_x[y]//将属性值 group_i.r_x.A_y 的次最不重要位(即第 1 位)设为相应的元组水印位
    end
   end
 end//水印嵌入算法结束
```

在水印嵌入算法中,ExtractWMBits( )为水印信息提取函数,具体实现算法为:

```
ExtractWMBits(Hash_str,len)
begin
  W=copy(Hash_str,1,len)//len 为水印信息长度
  return W
end
```

本文的 Hash( )为采用 MD5 思想实现的单向哈希函数。

2.3 水印检测

水印检测仍以分组为单位进行,这需要知道关系数据表中的元组是如何分组的。元组的分组思想与水印嵌入算法中的方法相同,但检测者必须知道水印密钥 K 和分组数目 g。水印检测的主要步骤为:

(1)生成属性水印 WA 和元组水印 WT:生成方式与水印嵌入时采用的思想相同。

(2)提取属性水印 WBA 和元组水印 WBT:从每个属性值的最不重要位上提取 1 位构成属性水印,从每个属性值的次最不重要位上提取 1 位构成元组水印。

(3)分别比较 WA 与 WBA、WT 与 WBT,输出检测结果。

具体的水印检测算法如下:

```
for i=1 to g do
begin
  for y=1 to s do
   begin
    for x=1 to g_i do
     begin
      C_y=C_y+copy(group_i.r_x.A_y,1,length(group_i.r_x.A_y)-2)//排除每个属性值的两个不重要位后构成的字符串
      WB_iA_y=WB_iA_y+group_i.r_x.A_y[length(group_i.r_x.A_y)] //从每个属性值的最不重要位上提取 1 位构成属性水印。
     end
    W_iA_y=ExtractWMBits(Hash(Key,C_y),g_i)//从 C_y 的哈希值中
```

生成 g_i 个属性水印位

```
        if W_iA_y=WB_iA_y then
          Result=TRUE
        else Result=FALSE
    end
      for x=1 to g_i do
      begin
        for y=1 to s do
          begin
tuple_str=tuple_str+copy(group_i.r_x.A_y,1,length(group_i.r_x.A_y)-2)
            WB_iT_x=WB_iT_x+group_i.r_x.A_y[length(group_i.r_x.A_y)-1]//从
```

每个属性值的次最不重要位上提取 1 位构成元组水印。

```
          end
        W_iT_x=ExtractWMBits(Hash(Key,tuple_str),s)//从 tuple_str
```

的哈希值中生成 s 个元组水印位

```
        if W_iT_x=WB_iT_x then
          Result=TRUE
        else Result=FALSE
      end
  end //水印检测算法结束
```

3　算法分析

在水印嵌入和检测算法中,元组的划分和排序是非常重要的两个步骤。将元组划分到不同的分组中去,可以将检测结果限制在分组范围内。在通常情况下,元组之间的关系是离散的,通过排序使元组之间保持一个相对稳定的顺序,从而保证水印信息的提取检测与嵌入保持同步。

算法中的单向哈希函数 Hash( )是一个将任意长度的输入字符串转化成固定长度的输出值的单向操作,输入字符串值的任何细微变化都可以在输出值中体现出来,为判断数据是否受到篡改提供依据。

4　算法实现与仿真实验

算法实验环境为 Intel 2.40 GHz CPU,256 MB RAM,40GB HARD DISK, 在 Windows XP Professional 2002 下使用 ADO 连接 SQL Server 2000 数据库,嵌入的水印密钥是“rddw”,分组数目为 8,并设可嵌入水印的最不重要位数为 2。

下面以数据库的一个分组为例(如表 1 所示),给出了当数据库受到属性值改变、元组添加和元组删除三类篡改时的水印检测结果。其中 P_i 为主码值,A_i 为属性名称,i 为正整数,代表序号。

表 1　数据库的某个分组中的元组

	A_1	A_2	A_3	A_4	A_5
P_1	10	15	13	12	14
P_2	140	142	142	146	146
P_3	145	146	151	150	150
P_4	183	182	181	186	184
P_5	205	204	209	208	208

4.1　属性值修改的定位

用 T 和 F 表示水印的检测结果,其中 TT 表示属性水印和元组水印没有发生改变,TF 表示只有元组水印发生了改变,FT 表示只有属性水印发生了改变,FF 表示两种水印均发生了改变。现在假设表 1 中 r_4A_2 的值(即 182)被改变了,通过该分组的水印矩阵(如表 2 所示),可以看出,检测结果均为 F 的第 4 行第 2 列的交叉点即为值改变的属性。对于有多个属性值被改变时也会产生类似的检测结果。

表 2　修改一个属性值的水印检测结果

	1	2	3	4	5
1	TT	FT	TT	TT	TT
2	TT	FT	TT	TT	TT
3	TT	FT	TT	TT	TT
4	TF	FF	TF	TF	TF
5	TT	FT	TT	TT	TT

4.2　元组添加的定位

假定有一新元组 r 被添加数据库中,经过水印检测算法,r 出现在表 1 所示的分组中(如表 3 所示),主码值为 P_3 的为新添加的元组。

表 3　添加一个元组后的某个分组数据

	A_1	A_2	A_3	A_4	A_5
P_1	10	15	13	12	14
P_2	140	142	142	146	146
P_3	20	23	21	19	26
P_4	145	146	151	150	150
P_5	183	182	181	186	184
P_6	205	204	209	208	208

通过实验,可以得到该分组的水印矩阵(如表 4 所示)。

表 4　添加一个元组后的的水印检测结果

	1	2	3	4	5
1	FT	FT	FT	FT	FT
2	FT	FT	FT	FT	FT
3	FF	FF	FF	FF	FF
4	FT	FT	FT	FT	FT
5	FT	FT	FT	FT	FT
6	FT	FT	FT	FT	FT

从表 4 看出,第 3 行水印检测结果均为 F,从而可以确定该行所对应的元组为新添加的。

当添加多个新元组时, 水印矩阵仍能检测到相应的结果,只不过它们可能分别出现在不同分组中。

4.3　元组删除的定位

仍以表 1 所示的分组为例,如果删除主码值为 P_2 的元组,通过实验,可得到如表 5 所示的水印矩阵。

表 5　删除一个元组后的的水印检测结果

	1	2	3	4	5
1	FT	FT	FT	FT	FT
2	FT	FT	FT	FT	FT
3	FT	FT	FT	FT	FT
4	FT	FT	FT	FT	FT

从表 5 所示的结果中很容易看出该分组中所有元组的属性水印都发生了改变,而元组水印未发生改变,这意味着某些元组被删除了。由于在水印检测以前,即便是数据库私有者也不能得知每个分组中的元组数目,所以无法判断每个分组中被删除的元组数目,但是通过该算法,可以将元组删除定位在某一个或某一些分组范围内。

4.4 属性顺序变化及末端属性删除的定位

若攻击者调整了关系数据库中除了主码以外其他属性的顺序,提出的算法仍能对水印进行有效的检测。因为属性的顺序虽然发生了变化,但同一属性的值没有变化,所以属性水印的认证结果都是真值。而对于一个元组而言,虽然它的各个属性值没有变化,但在该算法中,元组的各属性值按顺序构成一个序列用于生成元组水印,所以若调整了属性顺序,至少有一部分元组水印的认证结果不是真值,而实际上,属性的值没有发生变化。若出现这样的情况,数据库认证者可首先恢复数据库属性的原始顺序,然后再进行水印的提取和检测。对于末端属性删除,例如删除表 1 的最后一个属性 A_5,可得到如表 6 所示的分组水印矩阵。

表 6 末端属性删除的水印检测结果

	1	2	3	4
1	TF	TF	TF	TF
2	TF	TF	TF	TF
3	TF	TF	TF	TF
4	TF	TF	TF	TF
5	TF	TF	TF	TF

由表 6 可以看出,属性水印的认证结果均为真,而元组水印认证结果均为假, 说明每个元组都遭受了相同情况的攻击,从而可以肯定数据库的末端属性被删除了。对于有连续的多个末端属性被删除的情况,水印矩阵也产生类似的认证结果。

5 结论

本文以关系数据库为水印载体,提出了一种基于关系数据库的脆弱性水印算法,并采用仿真实验的方法对关系数据库进行了篡改攻击试验。结果表明,该算法的主要特点是:

(1)水印信息能够根据密钥、分组数目及关系数据动态生成,不需要原始数据库,这使得数据库水印能够实现盲检。

(2)算法对元组的添加、属性值的修改、元组的删除三种常规的数据操作具有很好的检测能力。

(3)算法对攻击者改变属性顺序特别是删除末端属性具有灵敏的检测能力。

(4)与现有的大多数算法不同,在该水印方案中,每个分组中元组的数目是由算法决定的,每个分组所包含的元组数目也是不完全一样的,从而表明数据库中的元组数目与每个分组所包含的元组数目没有必然联系。

参考文献:

[1] Tirkel A,Rankin G,van Schyndel R,et al.Electronic watermark[C]//Digital Image Computing,Technology and Applications.Macquarie University,Sidney,1993:666-673.

[2] Agrawal R,Kiernan J.Watermarking relational databases[C]//Proceedings of the 28th VLDB Conference.Hong Kong:University of Science & Technology,2002:155-156.

[3] Li Y,Swarup V,Jajodia S.Constructing a virtual primary key for fingerprinting relational data[C]//Proceedings of the 3rd ACM Workshop on Digital Rights Management.New York,NY,USA:ACM Press,2003:133-141.

[4] Sion R,Atallah M,Prabhaka S.Rights protection for relational data[C]//2003 ACM SIGMOD International Conference.San Diego,California,USA:ACM Press,2003:98-109.

[5] 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.

[6] 黄敏,张浩,曹加恒.一种基于关系数据库的水印技术[J].计算机工程与应用,2005,41(10):153-155.

[7] 张桂芳,孙星明,肖湘蓉,等.基于中国剩余定理的数据库水印技术[J].计算机工程与应用,2006,42(7):135-136.

[8] 彭沛夫,林亚平,张桂芳,等.基于有效位数的数据库数字水印[J].计算机工程与应用,2006,42(11):166-168.

[9] GUO Hui-ping,LI Ying-jiu,LIU An-yi,et al.A fragile watermarking scheme for detecting malicious modifications of database relations[J].Information Sciences,2006,176(10):1350-1378.

(上接 134 页)

[2] Snoeren A C,Balakrishnan H.An end-to-end approach to host mobility[R].MIT Laboratory for Computer Science Cambridge,MA,2000:155-166.

[3] 黄遵国.面向生存能力的应急响应与事故恢复技术研究[D].长沙:国防科技大学,2003.

[4] 魏楚元.安全群组通信中分布式密钥管理协议的研究[D].南宁:广西大学,2005.

[5] Bresson E,Catalano D.Constant round authenticate group key agreement via distributed computation[C]//Lecture Notes in Computer Science:Proc 7th International Workshop on Practice and Theory in Public Key Cryptography PKC'04,2004,2947:115-129.

[6] Bresson E,Chevassut O.Multual authentication and group key agreement for low-power mobile devices[J].Comput,Commun,2004,27:1730-1737.

[7] Boneh D.The decision diffie-Hellman problem[C]//LNCS:Proc 3rd Algorithmic Number Theory Symp,Portland,Oregon,June 21-25,1998:48-63.

(上接 141 页)

表 1 的结果说明通过数学变换 $\boldsymbol{u}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}$ 把原来 n 个特征的信息尽量集中到较少的 $k(k<n)$个新特征中去,然后用这 k 个新特征识别入侵者,可以提高 IDS 的整体性能。表 1 同时也说明用本文所给出的方法可以较好地求出变换 $\boldsymbol{u}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{x}$ 中的最优矩阵 $\boldsymbol{A}$,使得在变换后的 k 维空间中,分类效果评价标准 J 达到最大。

5 结束语

理论和实验都证明用所给出的特征降维方法可以提高 IDS 的整体性能,可以提高 IDS 对入侵者的正确识别率。

参考文献:

[1] Biermann E,Cloete E.A comparison of intrusion detection systems[J].Computers & Security,2006,20(8):341-343.

[2] Richard P.Improving intrusion detection performance using keyword selection and neural net works[J].Computer Networks,2005,15(6):253-254.

[3] Zamboni D.Using internal sensors for computer intrusion detection[D].the Faculty of Purdue University,2001-08.

[4] Zhang YG,LEE WK.Intrusion detection techniques for mobile wireless networks[J].Wireless Networks,2003,18(9):545-556.

[5] Mukherjee B.Network intrusion detection[J].IEEE Networks,2005,23(4):284-285.

[6] Intrusion Detection Attacks Database[R/OL].http://www.ll.mit.edu.