软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(12):3837−3852 [doi: 10.13328/j.cnki.jos.005303]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种保序加密域数据库认证水印算法*

项世军[1,2], 何嘉勇[1,2]

[1](暨南大学 信息科学技术学院/网络空间安全学院,广东 广州 510632)

[2](信息安全国家重点实验室(中国科学院 信息工程研究所),北京 100093)

通讯作者: 项世军, E-mail: Shijun_Xiang@qq.com

摘 要: 加密域水印技术适用于云环境下的隐私保护(加密)和数据安全认证(加水印).通过结合保序加密、离散余弦变换、密码哈希和数字水印技术,提出了加密域数据库认证水印算法.首先对数据进行保序加密,以达到对敏感数据内容的隐私保护;对加密后的数据进行分组和离散余弦变换处理,然后将交流系数的哈希(Hashing)值作为认证信息嵌入到直流系数中来认证数据的完整性;可通过比对交流系数的哈希值和从直流系数中提取的水印信息,来判断加密数据是否受到篡改.水印嵌入设计很好地结合了保序加密的特性,使得对加密数据的水印嵌入不会影响到明文数据的正确恢复,利用密钥对加水印的加密数据库直接解密可得到原数据库.实验结果表明:所提出的算法不仅能够用于保护数据库中的内容隐私,而且能检测出不同程度的篡改和有效认证数据库数据的完整性.

关键词: 保序加密方案;数据库;水印;完整性认证;离散余弦变换

中图法分类号: TP309

中文引用格式: 项世军,何嘉勇.一种保序加密域数据库认证水印算法.软件学报,2018,29(12):3837−3852. http://www.jos.org.cn/1000-9825/5303.htm

英文引用格式: Xiang SJ, He JY. Database authentication watermarking algorithm in order preserving encrypted domain. Ruan Jian Xue Bao/Journal of Software, 2018,29(12):3837−3852 (in Chinese). http://www.jos.org.cn/1000-9825/5303.htm

Database Authentication Watermarking Algorithm in Order Preserving Encrypted Domain

XIANG Shi-Jun[1,2], HE Jia-Yong[1,2]

[1](College of Information Science and Technology/College of Cyber Security, Jinan University, Guangzhou 510632, China)

[2](State Key Laboratory of Information Security (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

Abstract: Digital watermarking in encrypted domain is a potential technology for privacy protection (with encryption) and integrity authentication (with watermarking) in cloud computing environments. Based on order-preserving encryption scheme (OPES), discrete cosine transformation (DCT), cryptography hash and watermarking technologies, this paper proposes a new database authentication watermarking algorithm in encrypted domain. Firstly, data in a database are encrypted with OPES for privacy protection. Then, the encrypted data are divided into groups for DCT operations. The watermark bits generated by hashing AC coefficients are embedded into DC coefficients for authenticating the encrypted data. The receiver can determine whether the data have been tampered by matching the hash value of AC coefficients and the extracted watermark bits from DC coefficients. The watermark embedding process in encrypted domain is lossless to plaintext data by exploring order-preserving property of OPES. In the receiver, an illegal user can recover the original database by directly decrypting the watermarked ciphertext data. Experimental results have shown that the algorithm can efficiently detect different tampering operations while protecting data content privacy with the encryption.

Key words: order preserving encryption scheme; database; watermarking; integrity authentication; discrete cosine transformation

* 基金项目: 国家自然科学基金(61272414, 61772234); 信息安全国家重点实验室开放课题(2016-MS-07)

Foundation item: National Natural Science Foundation of China (61272414, 61772234); Open Research Fund from the State Key Laboratory of Information Security (2016-MS-07)

收稿时间: 2016-11-30; 修改时间: 2017-03-22; 采用时间: 2017-05-17

加密域信息隐藏技术很好地结合了加密和信息隐藏的优点,可以在不暴露数据隐私的情况下对数据进行信息隐藏处理,适用于网络空间安全下的云计算处理,是信息安全领域的一个研究热点[1,2].

随着计算机技术的快速发展以及云计算时代的到来,人们越来越多地把敏感数据(如涉及个人隐私、具有重大商业价值等)以数据库形式上传到云端进行存储和发布,此时需要考虑数据隐私及数据安全等问题[3,4].为了避免聘请专业管理人员的昂贵费用,数据库服务(database as a service,简称 DaaS,可以提供与本地数据库一样的数据管理服务)[5]应运而生,数据拥有者将自身的数据库访问、维护、管理等任务委托给专门的第三方管理,用户利用不同的访问特权可以获取所需要的数据.

在这种新兴的数据库服务模式中,将数据库直接交给非可信的数据库服务提供者,有数据隐私暴露和数据被非法篡改的安全问题,因此有必要对数据库进行加密并利用数据库水印等技术来保护数据安全.传统上,数据库安全主要采用用户身份验证和用户权限管理[6],在访问过程中保证数据库的完整性、一致性以及数据的整体质量[7].在明文域,可以利用认证水印技术来认证数据的完整性,如,可通过半脆弱水印来认证图像的完整性[8,9].近年来,越来越多的黑客事件表明:访问控制防线一旦被破解,数据就失去了保护,此时,敏感数据将直接被攻击者随意非法使用.针对这个问题,一个有效手段是对数据库进行加密,这样在访问控制失效时,数据可以得到加密算法的保护.

数据加密后如何管理和操作,是数据库加密研究的一个重要方面.目前,数据库加密的主要技术有随机数求和、多项式函数、存储桶、秘密同态、子密钥和加密智能卡等[10].这些方法运用于不同的场景,但都存在一个弊端——加密后的数据库不能进行对比操作.数据库的对比操作涉及到索引技术[11],如果加密后索引失效,那么数据库的操作性就会降低.因此,文献[10]率先提出了一种基于数值型数据的保序加密(order- preserving encryption,简称 OPE)方案,能够保证密文顺序和对应明文顺序的一致,在不解密情况下,允许对加密数据进行比较操作.这样,等式和范围查询(MAX,MIN,COUNT,GROUPBY,ORDERBY)等关系运算都可以直接应用于加密数据.文中,Agrawal 等人并没有给出算法正式的安全性证明.为了确保攻击者即使得到全部密文也得不到除顺序以外的其他有用信息,Boldyreva 等人提出了一种利用随机保序函数和超几何分布设计的可证安全的 OPE 算法[12],该算法适用于云计算下的数据隐私保护,并且取得了很好的效果[13,14].接着,Boldyreva 等人进一步分析了 OPE 算法的安全性,提出了一种可提高任何 OPE 安全性的 Modular OPE 方案[15],该方案下,加密后的密文不再严格保序,但仍允许范围查询——模查询.基于保序加密的安全分析还有许多研究[16−18],很好地解决了保序加密的安全性问题.

数据库与一般多媒体数据有所不同,它具有低冗余、低敏感、无序、更新频繁等特点,嵌入时需要进行特殊的设计,需要考虑两个问题:① 数据冗余小,数据库不同字段的属性要求不同,难以找到大量可嵌水印空间;② 进行完整性认证时,要求水印算法有很强的敏感性.目前,针对明文数据库有几个水印算法.Agrawal 等人[19]在允许一定的误差范围内对数值型数据的 LSB(least significant bits)进行水印替换操作,该算法易于实现,但鲁棒性较差,抗攻击能力较弱.随后,Sion 等人[20]通过修改连续序列的数据分布特征来实现水印嵌入的算法.该算法具有较好的鲁棒性,但只适用于具有相同分布、数值相差不大的数据.由于数据库中不同字段的取值范围各不相同,因而限制了水印的嵌入容量.接着,Zhou 等人[21]提出一种基于混沌的 DCT 域关系数据库水印算法,利用主键哈希值选出有效的元组作为载体,并对其候选属性进行 DCT 变换,然后把水印嵌入到中频系数中.在 DCT 域中嵌入可以将水印能量分散,弥补了空域水印算法隐蔽性不足的问题.上述算法都有一个共同之处:在不影响数据库使用的情况下,允许对数据做一些稍微的改动.由于数据库中数据冗余非常小,需要设计比较复杂的寻找方案来确定嵌入水印的候选数据.此外,当前的数据库水印技术主要针对于明文数据在云环境下有数据隐私可能暴露的安全问题.

考虑到云环境下数据隐私保护和数据安全的需要,本文提出了一种保序加密域数据库认证水印算法.该算法首先对数据库某一属性列数据进行保序加密,加密数据分组后进行 DCT 变换,然后,每组随机选取一个交流系数并用哈希算法生成认证水印信息;利用 QIM(quantization index modulation)算法将水印嵌入到直流系数中,然后进行 IDCT 变换,得到含水印信息的加密数据库;数据库的完整性认证可通过比对交流系数的哈希值和直

流系数中的水印信息来完成.本文所提出的算法很好地结合了保序加密的特点:(1) 利用了保序概率加密引入的冗余,使得加密数据库里的每个数值都可用于水印嵌入,没有明文数据库难以寻找嵌入位置的问题;(2) 水印嵌入步长根据密文可修改范围进行设计,确保了加密数据的水印嵌入不会影响到明文数据的正确恢复,即,对加水印后的加密数据库直接解密可以得到原数据库.实验结果表明,本文所提出的加密域水印算法不仅可以用于保护数据库数据的隐私,而且能检测出不同程度的篡改,适用于云环境下的数据库数据隐私保护和完整性认证.

1 保序概率加密算法[12]

Boldyreva 等人在文献[12]中提出了一种利用随机保序函数和超几何分布设计的可证安全的 OPE 算法.其根据语义安全的概念定义了 OPE 算法的理想安全状态,并给出了 OPE 算法的安全性证明,即:攻击者即使得到全部密文,除了密文的顺序以外,就再也得不到其他任何有用的信息.保序加密框架(OPES)能够保证密文顺序和对应明文顺序的一致性,在不解密情况下,它允许查询操作直接应用在加密数据上.

下面是对保序概率加密算法[12]的介绍和相应说明.

1.1 OPES定义

若 E 是一个保序加密函数,p_1 和 p_2 是两个明文值,并且有密文 $c_1=E(p_1)$,$c_2=E(p_2)$,则有:if $(p_1<p_2)$ then $(c_1<c_2)$.

1.2 随机保序函数和超几何分布的关系

- 超几何分布

在一个抽球模型中,假若有 N 个球,其中黑球 M 个,则白球 $N-M$ 个,每次不放回地随机抽取一个球.用随机变量 X 表示在抽取的 y 个球中黑球的个数,那么 $X=x$ 的概率为

$$P_{HGD}(x;N,M,y)=\frac{\binom{y}{x}\cdot\binom{N-y}{M-x}}{\binom{N}{M}} \tag{1}$$

- 随机保序函数

假设有明文域$[M]$和密文域$[N]$,$N>M$,从密文域$[N]$中随机抽取 M 个不同的整数组成有序集合 S,构造了一个保序函数 f,把第 i 个明文 $m\in[M]$映射到第 i 个密文 $c\in S$.那么,一个保序函数 f 对应唯一的组合 C_N^M .为更好地理解该过程,给出明文和密文映射关系的例子,如图 1 所示.

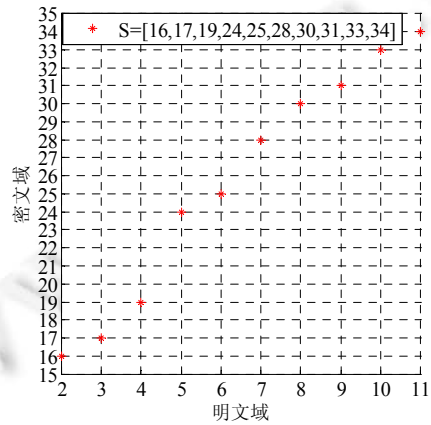


Fig.1 Mapping relation between plaintext and ciphertext

图 1 明文和密文映射关系

对于 $M,N\in\mathbb{N}$,任何 $x,x+1\in[M],y\in[N]$,有:

$$\Pr\left[f(x)\leqslant y<f(x+1):f\xleftarrow{\$}OPF_{[M],[N]}\right]=\frac{\binom{y}{x}\cdot\binom{N-y}{M-x}}{\binom{N}{M}} \tag{2}$$

其中,$OPF_{[M],[N]}$为包含所有保序函数 f 的集合, $f\xleftarrow{\$}OPF_{[M],[N]}$ 表示从集合中均匀随机抽取一个保序函数 f.

根据公式(1)、公式(2)可知,我们可以把构造随机保序函数($f([M])\xrightarrow{\text{映射}}[N]$)看做一个抽球实验(有 N 个球,其中有黑球 M 个).

1.3 保序概率加密

文献[12]中,所用的保序概率加密算法具有可证明的安全性,其加密过程和解密过程如下所示.

加密过程: *Enc*(*key_table*,*D*,*R*,*m*).
输入:*key_table*:密钥表,*D*:明文域,*R*:密文域,*m*:明文;
输出:*key_table*:密钥表,*D*:明文域,*R*:密文域,*c*:密文.

```
1.   M←max(D)−min(D)+1; N←max(R)−min(R)+1
2.   d←min(D)−1; r←min(R)−1
3.   y←⌈N/2⌉
4.   IF M=1
5.       RETURN c←$ {r+1,…,r+N−1}   //随机抽取 c
6.   IF key_table(:,1).iscontain(r+y)   //是否含该抽样点
7.       index←key_table(:,1).find(r+y)   //获取点位置
8.       x←key_table(index,2)−d
9.   ELSE
10.      x←$ HGD(M,N,y)   //超几何分布抽样
11.      key_table←[key_table;r+y,d+x]   //保存抽样点
12.  IF m≤d+x
13.      D←{d+1,…,d+x}
14.      R←{r+1,…,r+y}
15.  ELSE
16.      D←{d+x+1,…,d+M}
17.      R←{r+y+1,…,r+N}
18.  RETURN Enc(key_table,D,R,m)
```

解密过程: *Dec*(*key_table*,*D*,*R*,*c*).
输入:*key_table*:密钥表,*D*:明文域,*R*:密文域,*c*:密文;
输出:*key_table*:密钥表,*D*:明文域,*R*:密文域,*m*:明文.

```
1.   M←max(D)−min(D)+1; N←max(R)−min(R)+1
2.   d←min(D)1; r←min(R)1
3.   y←⌈N/2⌉
4.   IF M=1
5.       RETURN m←min(D)
6.   IF key_table(:,1).iscontain(r+y)
7.       index←key_table(:,1).find(r+y)
8.       x←key_table(index,2)−d
9.   ELSE
10.      x←$ HGD(M,N,y)
11.      key_table←[key_table;r+y,d+x]
12.  IF c≤r+y
13.      D←{d+1,…,d+x}
14.      R←{r+1,…,r+y}
15.  ELSE
16.      D←{d+x+1,…,d+M}
17.      R←{r+y+1,…,r+N}
18.  RETURN Dec(key_table,D,R,c)
```

上述算法中,密钥表(*key_table*)是一个保存抽样点的 $n\times2$ 矩阵,密钥表初始是空的,随着加密明文越多而增加.在整个加密过程中共用该密钥表,其作用类似于使用同一个随机保序函数 f,使得每次加密都沿着该函数进行.解密过程中,密钥表只作查询使用,并不会有所增加.

与文献[12]中描述的算法相比,第 5 行~第 11 行进行了重新描述,以便更好地理解本文的认证水印算法,见第 2.3 节图 4 及其描述说明.

1.4 各OPE算法比较

为逐步解决 OPE 的安全性问题,越来越多的研究(如文献[10,12,15])都提出了新的 OPE 方案.为比较 3 种 OPE 算法的安全性,做出以下实验:用该 3 种算法分别加密相同的明文数据.加密后的数据如图 2 所示.

由图 2 可知:文献[10]几乎成递增的线性变化,文献[12]是递增的非线性变化,安全性比文献[10]高,这两种算法的密文都是严格保序的.文献[15]中的 MOPE 算法是非保序的,不泄露密文的大小关系,可通过模查询方式进行范围查询,安全性更好.虽然 MOPE 算法的安全性更高,但该算法由于不保序,不适用于本文所提出的认证水印算法.文献[15]中的另一种方案保序,算法的安全性类似于文献[12].故而,本文采用了文献[12]中所提出的可证安全的 OPE 算法,用于加密后的数据隐藏.

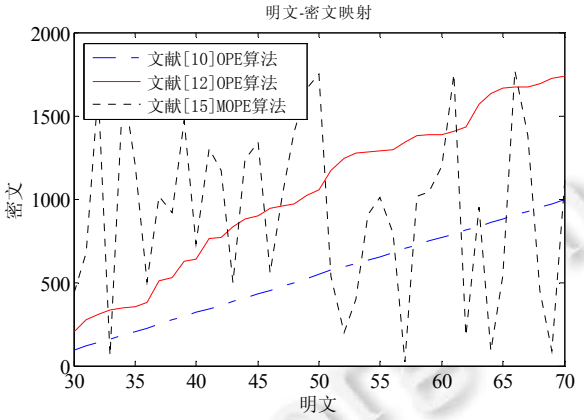


Fig.2 Comparison result of three OPE algorithms

图 2 3 种 OPE 算法的比较结果

2 保序加密域数据库认证水印算法

本文提出的数据库认证水印算法的框架如图 3 所示,数据库所有者首先对数据库某一属性列数据进行保序加密,然后将加密数据库传递给信息隐藏者,信息隐藏者对加密数据进行分组和 DCT 变换,选择交流系数并用哈希算法生成水印信息,然后利用 QIM 算法将水印信息嵌入到直流系数中,进行相应 IDCT 变换后,得到含水印信息的加密数据库;接收者接收到含水印信息的加密数据库后,可通过类似于嵌入过程生成哈希值和提取的水印信息进行对比,以验证加密数据库的完整性.对于拥有加密密钥的接收者,可直接对加密数据库进行解密,得到原数据库.可以看出:所提出的框架虽然在加密域中嵌入了水印信息,但相对于明文是无损的,好处是水印信息可以用来认证加密数据的完整性,同时不会对明文内容造成任何影响.

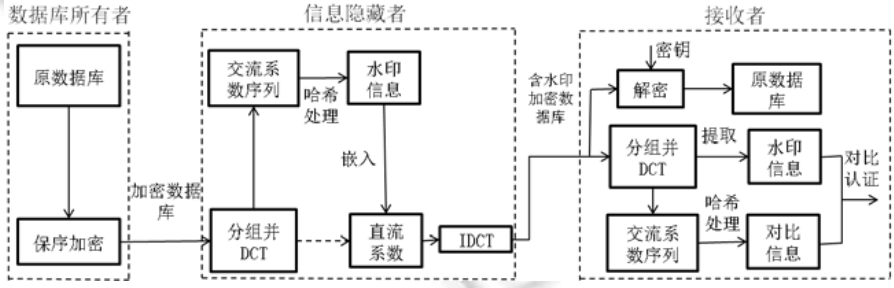


Fig.3 Sketch of the proposed database authentication watermarking algorithm

图 3 所提出的数据库认证水印算法框架

2.1 数据库加密

数据库所有者按照第 1 节 OPE 系统中描述的保序概率加密算法,对原数据库数据进行加密,得到加密数据库,并将加密数据库传递给信息隐藏者,将密钥表(*key_table*)传递给接收者.

2.2 生成水印信息

1) 数据分组

设 I 为加密数据库某属性列数据,长度为 L,对数据进行分组,每组长度为 l,划分成了 $n=\left\lfloor \frac{L}{l} \right\rfloor$ 组数据.

2) 生成交流系数序列

对每组数据进行 DCT 变换,生成的数据组表示为

$$Y_i=\{D_{i,1},A_{i,1},A_{i,2},\ldots,A_{i,(l-1)}\},i\in[1,n] \tag{3}$$

其中,$D_{i,1}$ 表示第 i 组数据的第 1 个直流系数,$A_{i,j}$ 表示第 i 组数据的第 j 个交流系数.获取每一组的第 j 个交流系数组成一个序列,表示为

$$S_j=\{A_{1,j},A_{2,j},\ldots,A_{n,j}\},j\in[1,l-1] \tag{4}$$

3) 生成水印信息

用哈希算法处理所生成的交流系数序列 S_j,可表示为

$$w_j=hash(S_j),j\in[1,l-1] \tag{5}$$

通过 *hash* 函数,把交流系数序列 S_j 转化为 128 比特二值序列 w_j.若只处理一个交流系数序列,则水印信息为

$$W=w_j,j\in[1,l-1] \tag{6}$$

此时,水印信息长度为 128.当随机选择两个序列进行处理时,则水印信息为

$$W=\{w_j,w_k\},j,k\in[1,l-1] \tag{7}$$

此时,水印信息长度为 256.由于在每一组的直流系数中嵌入 1 个比特,所以水印信息的长度 W_l 必须满足 $128\leqslant W_l\leqslant n$,这里,$n$ 是数据分组的数目.

2.3 水印信息嵌入

1) OPE 后水印嵌入的影响

明文数据经过 OPE 后带来了冗余,利用冗余,我们希望水印在加密域的嵌入不会对明文造成影响.为此,我们对密文数据进行逐渐增大的修改操作,观察修改幅度对明文正确恢复的影响.表 1 为明文和密文的对照表以及修改测试的结果.

Table 1 Recovery results of plaintext after different tampering

表 1 不同篡改下明文恢复情况

明文	密文	恢复明文	密文+1	恢复明文	密文+2	恢复明文	密文+3	恢复明文
27	7	27	8	27	9	27	10	28
28	24	28	25	28	26	28	27	29
29	35	29	36	29	37	29	38	29
30	70	30	71	30	72	30	73	31
				…				
227	7 656	227	7 657	227	7 658	228	7 659	228
228	7 674	228	7 675	228	7 676	228	7 677	228
230	7 755	230	7 756	230	7 757	230	7 758	230
231	7 816	231	7 817	231	7 818	231	7 819	231
234	7 939	234	7 940	234	7 941	234	7 942	234

结果表明:密文在一定范围内进行改变,解密后仍可正确恢复出原明文.随着改变程度的增大,正确恢复明文的可能性就越小.表 1 中虚框内的数据,当密文改变到一定幅度时,明文不能全部正确恢复.不同明文对应的密文可修改的幅度不同,测试表明:密文经过一定篡改,解密后仍可恢复出原明文.原因是保序加密的保序特性,使得数据带来了冗余.

2) 确定可改变的幅度

上述分析可知:密文改变幅度不超过某一阈值 T 时,解密后仍可恢复出原明文.经过 OPE 算法[12]加密后,密文域 R 会被划分成 *length*(D)(明文数量)个不等且不相交的子区域,这里称为桶,密文就是从每个桶中随机抽取一个得到.当一个密文篡改后的值在其他桶区间时,就会错误解密成其他明文.桶划分示意图如图 4 所示.

图 4 中,有相邻的桶 i(区间为$[c_{li},c_{hi}]$)和桶(i+1)(区间为$[c_{l(i+1)},c_{h(i+1)}]$),桶间距为 $c_{l(i+1)}-c_{hi}=1$.明文 m_i 对应的密文 $c_i\in[c_{li},c_{hi}]$,明文 m_{i+1} 对应的密文 $c_{i+1}\in[c_{l(i+1)},c_{h(i+1)}]$.位于桶区间内的密文都可以正确解密,密文篡改后的值落在其他桶区间时就会错误解密,所以篡改的程度应小于 1.特殊的有,当 $c_{i+1}=c_{l(i+1)}$时,该密文经过篡改后可能位于区间$(c_{hi},c_{l(i+1)})$内,但仍能成功解密.由所用 OPE 算法的解密过程第 12 行~第 17 行可知,两相邻桶是以点 c_{hi} 划分的,即,第 12 行的 $r+y=c_{hi}$,因此待解密值 $c\in(c_{hi},c_{l(i+1)})$总能被解密成桶(i+1)对应的明文.而密文 $c_i=c_{hi}$ 经篡改后可

能会错误解密,为适应本文认证水印算法,对加密过程第 5 行进行了修改,即,抽样空间不包括最后一个数据,此时,$c_i \in [c_{li}, c_{hi}-1]$,$c_{i+1} \in [c_{l(i+1)}, c_{h(i+1)}-1]$,该修改不影响原加密算法安全性.那么此时阈值 T=1,可篡改范围为(−1,1).

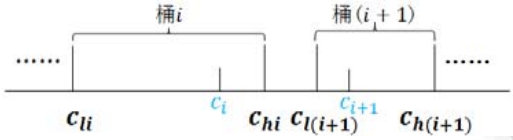


Fig.4 Diagram of bucket partitioning

图 4 桶划分示意图

3) 本文认证水印算法的理论证明

本文水印嵌入算法利用了 DCT 变换的正交特性——改变直流系数不会影响交流系数,当水印嵌入到直流系数时,重新 DCT 分解后的交流系数不会发生任何改变.为了认证加密数据的完整性,水印信息由交流系数经过哈希得到,进而嵌入到直流系数中.在检测端,类似于水印嵌入过程进行 DCT 变换得到交流系数和直流系数,然后通过对比交流系数的哈希值和从直流系数中提取的水印信息来判断数据是否完整.当含水印的加密数据发生改变时,交流系数及其哈希发生改变,将不能匹配从直流系数中提取的水印信息,从而判定发生篡改;当含水印的加密数据没有改变时,从直流系数中提取的水印信息将匹配交流系数的哈希值.为了更好地说明 DCT 下直流和交流系数的正交关系,我们进行了下面的理论推导.

设 $f(x)$为一维离散函数,x=0,1,2,…,N−1,可理解为第 2.2 节中数据分组中的一组,先在不作修改的情况下进行 DCT 和 IDCT,根据文献[22]中公式(1)、公式(2),可得公式(8)~公式(10),对 $f(x)$进行 DCT 变换得直流系数:

$$F(0)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f(x),u=0 \tag{8}$$

得交流系数:

$$F(u)=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{u\pi}{2N}(2x+1)\right],u=1,2,...,N-1 \tag{9}$$

对 $F(u)$进行 IDCT 变换得:

$$f(x)=\frac{1}{\sqrt{N}}F(0)+\sqrt{\frac{2}{N}}\sum_{u=1}^{N-1}F(u)\cos\left[\frac{u\pi}{2N}(2x+1)\right],x=0,...,N-1 \tag{10}$$

现在开始修改直流系数嵌入水印,令 $F(0)=F(0)+C$,再进行 IDCT 变换有:

$$f^*(x)=\frac{1}{\sqrt{N}}(F(0)+C)+\sqrt{\frac{2}{N}}\sum_{u=1}^{N-1}F(u)\cos\left[\frac{u\pi}{2N}(2x+1)\right]=f(x)+\frac{C}{\sqrt{N}},x=0,1,...,N-1 \tag{11}$$

对 $f^*(x)$进行 DCT 变换,得直流系数为

$$F^*(0)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f^*(x)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}\left(f(x)+\frac{C}{\sqrt{N}}\right)=\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}f(x)+C=F(0)+C,u=0 \tag{12}$$

可得 $F^*(0)=F(0)+C$.可见,水印嵌入的确改变了直流系数.可得相应的交流系数为

$$\left.\begin{aligned}F^*(u)&=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f^*(x)\cos\left[\frac{u\pi}{2N}(2x+1)\right]\\&=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}\left(f(x)+\frac{C}{\sqrt{N}}\right)\cos\left[\frac{u\pi}{2N}(2x+1)\right]\\&=\sqrt{\frac{2}{N}}\sum_{x=0}^{N-1}f(x)\cos\left[\frac{u\pi}{2N}(2x+1)\right]+\frac{\sqrt{2}C}{N}\sum_{x=0}^{N-1}\cos\left[\frac{u\pi}{2N}(2x+1)\right]\\&=F(u)+\frac{\sqrt{2}C}{N}\sum_{x=0}^{N-1}\cos\left[\frac{u\pi}{2N}(2x+1)\right],u=1,2,...,N-1\end{aligned}\right\} \tag{13}$$

设:

$$g(u)=\sum_{x=0}^{N-1}\cos\left[\frac{u\pi}{2N}(2x+1)\right],u=1,...,N-1 \tag{14}$$

则展开有:

$$g(u)=\cos\frac{u\pi}{2N}+\cos\frac{3u\pi}{2N}+...+\cos\frac{(2N-1)u\pi}{2N} \tag{15}$$

由和差化积公式:

$$\sin a-\sin b=2\cos\frac{a+b}{2}\sin\frac{a-b}{2} \tag{16}$$

得:

$$\sin 2nx-\sin(2n-2)x=2\cos(2n-1)x\sin x\Rightarrow\cos(2n-1)x=\frac{\sin 2nx-\sin(2n-2)x}{2\sin x} \tag{17}$$

所以,

$$\cos x+\cos 3x+\cos 5x+...+\cos(2n-1)x=\frac{\sin 2x}{2\sin x}+\frac{\sin 4x-\sin 2x}{2\sin x}+...+\frac{\sin 2nx-\sin(2n-2)x}{2\sin x}=\frac{\sin 2nx}{2\sin x} \tag{18}$$

同理可证:

$$g(u)=\frac{\sin\left(2N\cdot\frac{u\pi}{2N}\right)}{2\sin\frac{u\pi}{2N}}=\frac{\sin u\pi}{2\sin\frac{u\pi}{2N}}=0,u=1,...,N-1 \tag{19}$$

即,$F^*(u)=F(u)$.

上面的理论分析表明:改变直流系数不会影响交流系数,可通过将交流系数的哈希值嵌入到直流系数来认证加密数据的完整性.

4) 水印嵌入

用交流系数序列生成水印信息 W 后,利用量化索引调制 QIM 算法[23]把水印信息嵌入到加密数据库中,这是通过嵌入到加密数据的直流系数 $D_{i,1}$ 中实现——按照水印信息,利用不同的量化器将载体数据调制到不同的索引区间,提取水印时则根据数据所属的索引区间来提取水印信息.

嵌入的基本过程是:根据二值水印 $b\in\{0,1\}$,选择相应的量化器 $Q_\Delta(\cdot)$量化载体数据 x.这种量化过程表示为

$$Q_\Delta(x)=\begin{cases}\left\lfloor\frac{x}{\Delta}\right\rfloor\Delta+\frac{1}{4}\Delta, & b=0\\ \left\lfloor\frac{x}{\Delta}\right\rfloor\Delta+\frac{3}{4}\Delta, & b=1\end{cases} \tag{20}$$

其中,Δ为量化步长,符号⌊·⌋表示向下取整.本文方法是将水印信息 W 嵌入到直流系数 $D_{i,1}$ 中.当将数据分成 n 组进行 DCT 时,有 n 个直流系数,设水印 W 的长度为 W_l,且 $W_l\leqslant n$ 时,将水印比特嵌入到前 W_l 个直流系数中(也可以随机选取).水印嵌入函数表示为

$$Dw_i=Q_\Delta(D_{i,1}),i=[1,2,\ldots,W_l] \tag{21}$$

其中,Dw_i 为嵌入水印信息的直流系数.嵌入完成后,对每组进行 IDCT 变换即可得到含水印信息的加密数据库.

在量化水印算法中,量化步长的取值是一个关键.本文中,通过对加密数据进行 DCT 后,在直流系数中嵌入交流系数的哈希值来认证加密数据的完整性,防止非法篡改.为了确保含水印的加密数据在解密后仍为原明文数据,利用保序概率加密的特性,水印嵌入时量化步长的取值要满足密文可改变范围不超过(−1,1).由公式(11)可以知道,含水印的密文值 $f^*(x)$与密文值 $f(x)$的差为 $\frac{C}{\sqrt{N}}$,其中,C 为直流系数在水印嵌入后的变化值,N 为分组长度 l.为避免加密数据的水印嵌入失真超过(−1,1)这一范围,必须满足下面的表达:

$$-1<\frac{C}{\sqrt{l}}<1\Rightarrow\left|\frac{C}{\sqrt{l}}\right|<1 \tag{22}$$

由公式(20)可知:水印嵌入过程中,直流系数的最大失真为 $|C_{\max}|=\frac{3}{4}\Delta$,则有 $\frac{\frac{3}{4}\Delta}{\sqrt{l}}<1$,进而可得量化步长$\Delta$取值范围为

$$0<\Delta<\frac{4\sqrt{l}}{3} \tag{23}$$

公式(23)表明:当量化步长一定范围内取值时,对含水印的加密数据直接解密可得到原始明文数据.这说明了水印嵌入对数据库明文是一个无损过程,有助于认证加密数据库的同时不影响数据库的正常使用.

2.4 数据库完整性认证

接收者接收到含水印的加密数据库,首先要对数据库的完整性进行认证,确保在传输过程中没有被攻击者插入、篡改、伪造等,这可以通过对比交流系数的哈希和直流系数中嵌入的水印信息来完成.

1) 交流系数的哈希

哈希的生成与水印嵌入过程一致,主要分成 3 步.

- 数据分组:对长度为 L 的含水印加密数据库进行分组,每组长度为 l;
- 进行 DCT:对每组数据进行 DCT 变换,获取每组的交流系数组成交流系数序列;
- 生成哈希:用哈希算法处理交流系数序列,生成哈希值作为对比信息.

2) 水印信息的提取

首先,从每一组的直流系数 $D_{i,1}$ 中提取 1 个水印比特,利用量化步长Δ计算该直流系数 $D_{i,1}$ 所属的索引区间:

$$index(i)=D_{i,1}-\left\lfloor\frac{D_{i,1}}{\Delta}\right\rfloor\Delta, i\in[1,n] \tag{24}$$

进而根据索引区间提取出 1 比特水印信息:

$$b_i=\begin{cases}1, & \text{if } index(i)>\dfrac{\Delta}{2}\\ 0, & \text{else}\end{cases}, i\in[1,n] \tag{25}$$

重复以上过程,直到提取出全部水印信息.

3) 完整性认证

将提取的水印信息与交流系数的哈希值作对比:若一致,则加密数据完整;否则,数据已被非法篡改.导致误判的攻击主要为修改密文保留水印信息,让水印认证通过但数据库数据实则已被篡改.要达到上面的攻击非常困难,原因是水印算法结合了 DCT 变换的全局和正交特性,通过嵌入交流系数的哈希值到直流系数中来进行认证,这对篡改具有很强的敏感性.由于水印信息足够长(实验中为 1 024 比特),漏警概率非常小.

2.5 数据库解密

接收者接收到加密数据库后,可以根据解密过程用密钥表(*key_table*)进行解密,密钥表只作查询使用,不会增加表内容.

3 字符数据的处理及密文数据库关系运算

3.1 字符数据的处理

目前,大多数 OPE 方案都是针对数值型数据而设计的,但也有部分研究是关于字符数据的保序加密,如 Li 等人[24]在文献[10]基础上,提出了一种针对字符数据的保序加密方法 OPES+.OPES+继承了 OPES 的基本思想,相对于OPES 方法,OPES+只增加了一步,即:在建模前进行类型转换,把字符型数据转换成数值型,英文字符转换成其 ASCll 值,汉字转换成区位码值.在本算法框架中,当遭遇字符型数据时,在加密前应进行相应的类型转换.

3.2 密文数据库的关系运算

经过 OPE 的数据库并不能实现所有密文域上的数据库关系运算,例如,SUM,AVG 等运算需要解密后才能实现.但由于保持顺序的关系,仍可以在不解密情况下进行等式和范围查询(如 MAX,MIN,COUNT,GROUPBY,ORDERBY)等关系运算.图 5 显示了在加密数据库下用户查询处理的过程,查询 SQL 语句经过翻译层时,对数值型数据和字符型数据分别进行 OPES 和 OPES+,数据库系统返回符合请求的加密数据,在翻译层中进行解密并返回给用户.

Fig.5 Process of query processing in encrypted database

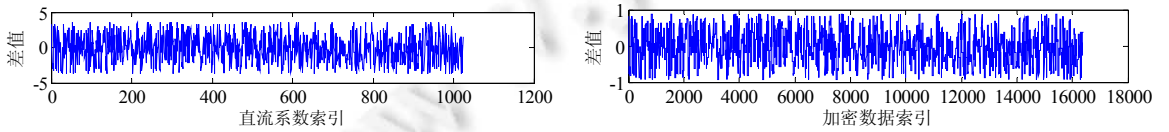
图 5 加密数据库查询处理

4 实验结果及分析

本文算法使用版本为 1.8.0 的 JVM 运行在配置为 3.50GHz Intel i5 处理器,4GB 运行内存的 Windows 7 系统上,通过 JDBC 方式连接到 MySQL 5.6 数据库,实现数据的增、删、改、查操作.首先建立一个数据库,并建立一张含有几个属性列的表,实验选取图像大小为 128×128 的 Lena 灰度图像的像素值作为实验数据.

4.1 水印生成及嵌入

将 128×128 个像素值数据写入数据库某个数值型属性列,进行保序加密得到加密数据.信息隐藏者对加密数据进行分组,每组 16 个数据,共 1 024 组,对每组进行 DCT 变换后获取其后 8 个交流系数,得到 8 个由 1 024 个交流系数组成的交流系数序列,每个序列进行哈希分别得到 128 比特的二值信息,然后对这些信息进行合并,得到 1 024 比特的水印信息.利用 QIM 算法将水印信息分别嵌入到 1 024 个直流系数中,根据公式(23)可知,量化步长Δ取值范围为(0,16/3),这里取Δ=5,然后将含水印的密文更新到数据库.图 6(a)显示了将 1 024 比特的水印信息分别嵌入到 1 024 个直流系数前后的差值,范围在(−4,4)之间;图 6(b)显示加密数据库嵌入水印前后的差值,水印嵌入失真刚好控制在(−1,1)范围内,显示了前面理论分析的正确性.



(a) 直流系数嵌水印前后的差值 (b) 加密数据库数据嵌水印前后的差值

Fig.6 Watermark distortion of DC coefficients and encrypted data

图 6 直流系数和加密数据的水印失真

4.2 完整性认证及篡改对数据库恢复影响

在接收端,接收者可对加密数据库进行完整性认证,也可根据密钥直接对数据库进行解密,得到原明文数据库.为了测试篡改对数据库恢复的影响,表 2 显示了随机选取 5 个含水印的密文值进行了不同程度的篡改,实验结果如图 7 所示.

Table 2 Tampering partial data

表 2 部分数据篡改

索引号	含水印密文值	篡改程度	篡改后含水印密文值
578	3 946.187 5	+0.1	3 946.287 5
1 575	3 912.500 0	−5	3 907.500 0
2 291	835.062 5	+50	885.062 5
10 119	697.562 5	−200	497.562 5
13 906	5 913.562 5	+350	6 263.562 5

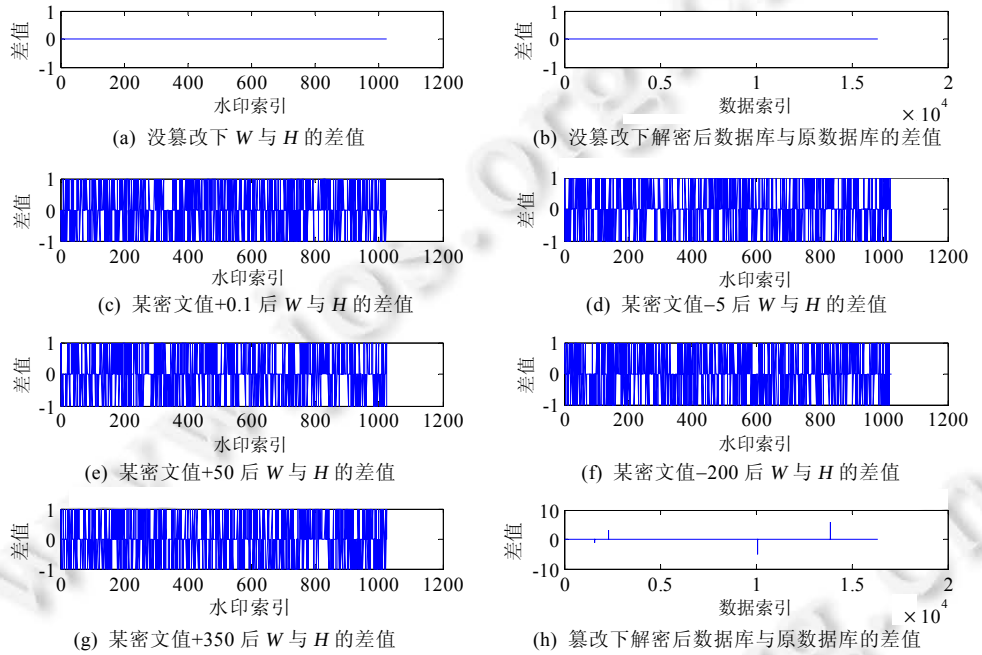


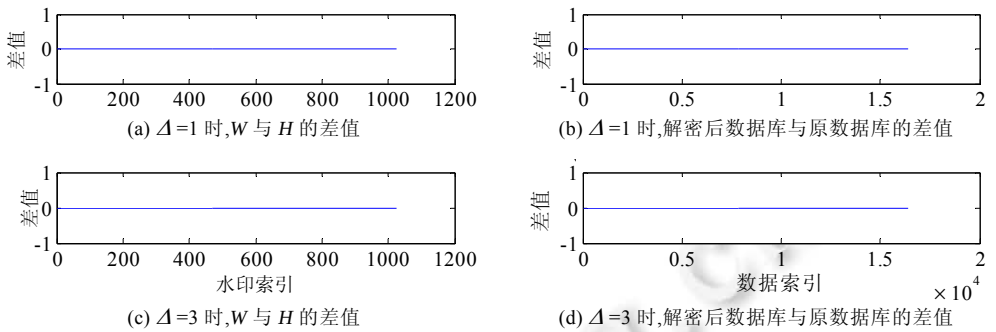
Fig.7 Tamper detection experiment: by comparing the watermark information in DC coefficients and the hash value generated by AC coefficients to determine whether database has been tampered with

图 7 篡改检测实验:通过比对直流系数中的水印信息(*W*)和交流系数的哈希值(*H*)来判断是否被篡改

图 7(a)显示在没有篡改的情况下,含水印加密数据库完整性认证情况,差值均为 0,可以看出,提取出来的水印信息与生成的对比信息完全相同.图 7(b)表示在不篡改情况下,含水印加密数据库的明文恢复情况,差值也均为 0,即,解密后的数据库与原数据库完全相同,说明对加密数据进行水印嵌入不影响明文数据的恢复.图 7(c)~图 7(g)分别表示对表 2 中的一个含水印密文进行不同程度的篡改操作后的认证情况.可以看出:即使是很小幅度的篡改,直流系数中提取出来的水印信息与交流系数生成的哈希也是完全不同的,这充分显示了本文加密域认证水印算法的有效性.图 7(h)表示在表 2 所有密文进行篡改的情况下,对含水印加密数据库进行解密后得到的明文情况,可以看出有 4 处错误(除了+0.1 的篡改),这说明了认证水印对于数据库完整性保护的必要性.通过在加密域引入认证水印,可以保护敏感数据的安全,防止非法篡改.

4.3 不同步长选取对认证水印算法及明文恢复的影响

公式(23)表明,量化步长Δ允许在一定范围内取值.下面通过实验分析不同步长的选取对认证水印算法及明文数据库恢复的影响.随机选取了 2 个不同的量化步长进行实验,分别为Δ=1 和Δ=3.实验结果如图 8 所示.图 8(a)~图 8(d)分别表示了选取不同量化步长下,含水印加密数据库完整性认证情况和明文恢复情况,差值均为 0.结果表明:在公式(23)的约束下,量化步长Δ的选取不会影响认证水印算法以及明文数据库的恢复.



(a) Δ=1 时,W 与 H 的差值　(b) Δ=1 时,解密后数据库与原数据库的差值

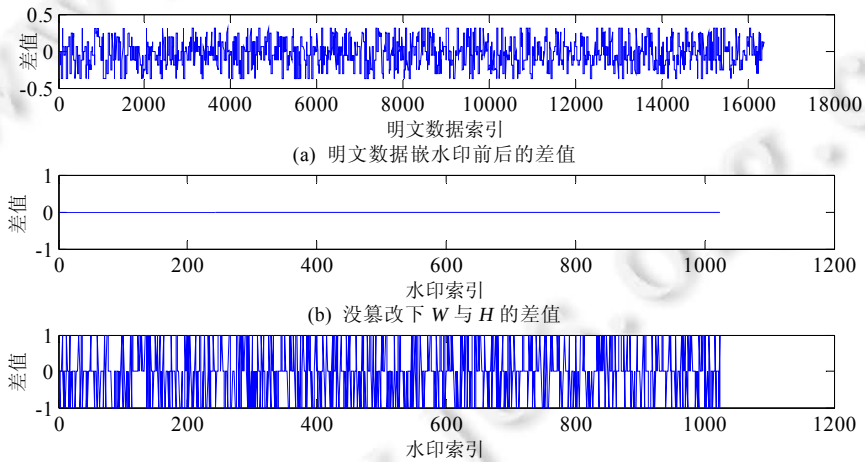
(c) Δ=3 时,W 与 H 的差值　(d) Δ=3 时,解密后数据库与原数据库的差值

Fig.8　Influence of different Δ on authentication watermark algorithm and database recovery

图 8　不同步长选取对认证水印算法及明文恢复的影响

4.4　认证水印算法对明文数据库的可用性测试

下面测试本文认证水印算法对于明文数据库的可用性,明文数据同样是 128×128 的像素值.为了保持明文顺序不变,可篡改范围设为(–0.5,0.5),由公式(23)计算量化步长为Δ=2.测试结果如图 9 所示.图 9(a)显示明文数据库嵌入水印前后的差值,水印嵌入失真刚好控制在(–0.5,0.5)范围内,即,水印的嵌入没有改变明文数据的顺序,印证了公式(23)的正确性;图 9(b)显示在没有篡改的情况下,含水印明文数据库完整性认证情况,可以看出,提取出来的水印信息与生成的对比信息完全相同;图 9(c)表示随机选取一个含水印明文,并进行+5 篡改操作后的认证情况,可以看出,明文数据库下本文认证算法仍然有效.



(a) 明文数据嵌水印前后的差值

(b) 没篡改下 W 与 H 的差值

(c) 随机取一个含水印明文进行+5 篡改后,W 与 H 的差值

Fig.9　Usability testing of authentication watermarking algorithm in original database

图 9　认证水印算法对明文数据库的可用性测试

从实际意义上来说,在明文数据库上直接嵌入水印并不明智,因为明文数据已经遭到了破坏,尤其对于准确性要求较高的数据库.通过保序概率加密和其引入的冗余,本文达到了保护隐私、嵌入认证信息和进行关系运算等目的.

4.5　时间开销

下面分析在资源受限的用户端下,本文的保序概率加密算法和认证水印算法的时间开销(计算时间开销和存储时间开销).为了测试方便,本文实验的数据库表只含有两个属性列(自增的索引列和待加密数据列),每一行数据代表一个元组.

图 10 显示了文献[10,12]算法(本文所用算法)在不同元组数下,加/解密所需的时间开销.可以看出:随着处理元组数量越多,加密和解密所需的时间开销也随之增加.可以看出:本文所用的保序概率加密算法在时间开销上比文献[10]要大一些,但在安全性上比文献[10]要好得多.

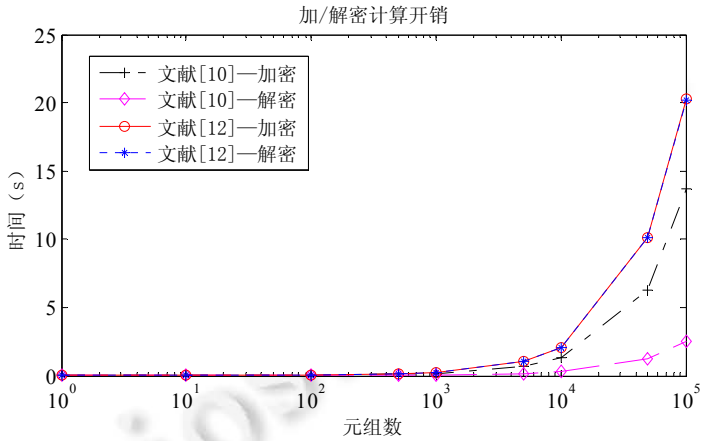


Fig.10 Time required for encrypting/decrypting different number of tuples

图 10 加/解密不同元组数所需时间

图 11 显示了一次性插入或读取不同数量的元组所需要的时间开销,可以看出:随着插入和读取的元组数量增加,所需要的时间开销也随之增加,其中:插入元组时时间开销增加非常快,而读取元组时则很缓慢.这是因为插入操作会改动原有数据库,而查询仅仅是返回所需数据.

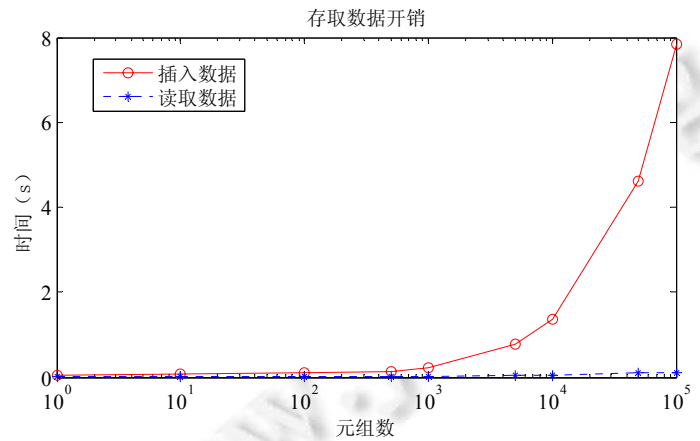


Fig.11 Time needed to one-time insert/read different number of tuples

图 11 一次性插入或读取不同元组数所需的时间

图 12 显示了在不同元组数下,本文认证水印算法的水印嵌入过程和水印提取过程所需要的时间开销.水印嵌入的时间由水印嵌入算法运行时间和含水印数据更新到数据库的时间组成.结合图 11 可知,水印嵌入的时间主要取决于后者所用的时间.水印提取不涉及数据库更改,其时间开销由水印提取算法运行时间和数据读取时间组成,随着元组数的增加,时间开销增加缓慢.

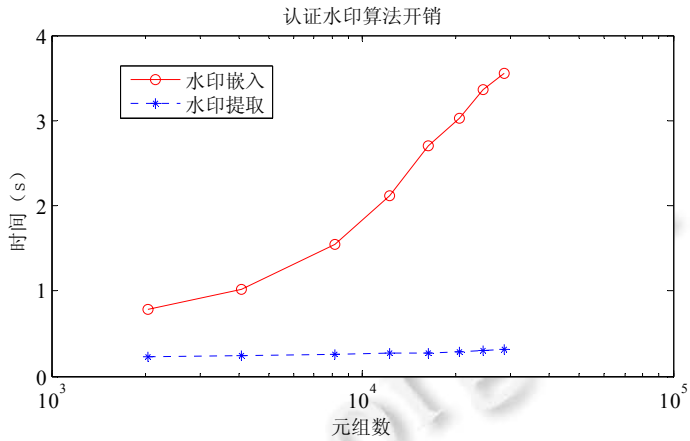


Fig.12 Time overhead of authentication watermarking algorithm in different number of tuples

图 12 不同元组数的认证水印算法时间开销

5 结 论

本文实现了一种新的保序加密域数据库认证水印算法.数据库所有者首先对数据库某一属性列的敏感数据进行加密,以保护其内容隐私;信息隐藏者对加密数据进行水印嵌入处理,防止数据被非法篡改,水印嵌入不影响用户对数据的查询和使用;接收者可通过验证数据库的完整性来确保是否被篡改,对于拥有加密密钥的接收者,可直接对加密数据库进行解密,得到原数据库.本文算法解决了以下几点问题并获得了很好的效果.

1) 通常,明文数据库中数据冗余非常小,进行水印保护时难以找到用于嵌入的冗余空间.经过保序加密的数据库数据不仅保护了内容隐私,而且所有数据都可以进行水印处理,很好地解决了数据库的保护问题;
2) 水印嵌入过程很好地结合了保序加密的特点,对加密数据的水印嵌入相对于明文数据是无损的,对含水印的密文数据库直接解密可得到原数据库.好处是:水印对加密数据提供了认证保护,但不会影响数据库的使用;
3) 水印算法很好地利用了 DCT 的全局和正交特性,将交流系数的哈希值嵌入直流系数,达到了对数据库的完整性认证,能够识别不同程度的篡改,对加密数据的完整性提供了很好的保护.

本文算法适合于网络空间安全大背景下的云数据库服务,很好地结合了加密和信息隐藏技术,为数据内容的隐私保护和数据安全提供了一种有效的潜在技术手段.

References:

[1] Zhang XP, Long J, Wang ZC, Cheng H. Lossless and reversible data hiding in encrypted images with public key cryptography. IEEE Trans. on Circuits and Systems for Video Technology, 2016,26(9):1622−1631. [doi: 10.1109/TCSVT.2015.2433194]

[2] Xiang SJ, Luo XR. Reversible data hiding in encrypted image based on homomorphic public key cryptosystem. Ruan Jian Xue Bao/ Journal of Software, 2016,27(6):1592−1601 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5007.htm [doi: 10.13328/j.cnki.jos.005007]

[3] Huang LS, Tian MM, Huang H. Preserving privacy in big data: A survey from the cryptographic perspective. Ruan Jian Xue Bao/ Journal of Software, 2015,26(4):945−959 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4794.htm [doi: 10.13328/j.cnki.jos.004794]

[4] Marisa P, Andry A, Budi R. Big-Data security management issue. In: Proc. of the 2nd Int'l Conf. on Information and Communication Technology (ICoICT). Bandung, 2014. 59−63. [doi: 10.1109/ICoICT.2014.6914040]

[5] Tian XX, Wang XL, Gao M, Zhou AY. Database as a service—Security and privacy preserving. Ruan Jian Xue Bao/Journal of Software, 2010,21(5):991−1006 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3746.htm [doi: 10.3724/SP.J.1001.2010.03746]

[6] Mario G. New challenges in teaching database security. In: Proc. of the 3rd Annual Conf. on Information Security Curriculum Development. Kennesaw, 2006. 64−67. [doi: 10.1145/1231047.1231060]

[7] Murray MC. Database security: What students need to know. Journal of Information Technology Education, 2010,9:61−77.

[8] Du L, Cao XC, Zhang W, Zhang XP, Liu N, Wei JG. Semi-Fragile watermarking for image authentication based on compressive sensing. Science China Information Sciences, 2016,59(5):1−3. [doi: 10.1007/s11432-016-5542-8]

[9] Schmitz R, Li SJ, Grecos C, Zhang XP. Content-Fragile commutative watermarking-encryption based on pixel entropy. Springer Int'l Publishing, 2015,9386:474−485. [doi: 10.1007/978-3-319-25903-1_41]

[10] Agrawal A, Kiernan J, Srikant R, Xu YR. Order preserving encryption for numeric data. In: Proc. of the 2004 ACM SIGMOD Int'l Conf. on Management of Data. New York, 2004. 563−574. [doi: 10.1145/1007568.1007632]

[11] Ma YZ, Meng XF. Research on indexing for cloud data management. Ruan Jian Xue Bao/Journal of Software, 2015,26(1):145−166 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4688.htm [doi: 10.13328/j.cnki.jos.004688]

[12] Boldyreva A, Chenette N, Lee Y, O'Neill A. Order-Preserving symmetric encryption. In: Proc. of the 28th Annual Int'l Conf. on Advances in Cryptology. Berlin, 2009. 224−241. [doi: 10.1007/978-3-642-01001-9_13]

[13] Wang C, Cao N, Li J, Ren K, Lou W. Secure ranked keyword search over encrypted cloud data. In: Proc. of the 30th IEEE Int'l Conf. on Distributed Computing Systems. Genova, 2010. 253−262. [doi: 10.1109/ICDCS.2010.34]

[14] Tang Q. Privacy preserving mapping schemes supporting comparison. In: Proc. of the 2010 ACM Workshop on Cloud Computing Security. New York, 2010. 53−58. [doi: 10.1145/1866835.1866846]

[15] Boldyreva A, Chenette N, O'Neill A. Order-Preserving encryption revisited: Improved security analysis and alternative solutions. In: Proc. of the 31st Annual Int'l Conf. on Advances in Cryptology. Santa Barbara, 2011. 578−595. [doi: 10.1007/978-3-642-22792-9_33]

[16] Wang C, Cao N, Ren K, Lou WJ. Enabling secure and efficient ranked keyword search over outsourced cloud data. IEEE Trans. on Parallel and Distributed Systems, 2012,23(8):1467−1479. [doi: 10.1109/TPDS.2011.282]

[17] Li K, Zhang WM, Yang C, Yu NH. Security analysis on one-to-many order preserving encryption-based cloud data search. IEEE Trans. on Information Forensics and Security, 2015,10(9):1918−1926. [doi: 10.1109/TIFS.2015.2435697]

[18] Popa RA, Li FH, Zeldovich N. An ideal-security protocol for order-preserving encoding. In: Proc. of the 2013 IEEE Symp. on Security and Privacy. Berkeley, 2013. 463−477. [doi: 10.1109/SP.2013.38]

[19] Agrawal R, Kiernan J. Watermarking relational databases. In: Proc. of the 28th Int'l Conf. on Very Large Databases. Hong Kong, 2002. 155−166. [doi: 10.1016/B978-155860869-6/50022-6]

[20] Sion R, Atallah M, Prabhakar S. Rights protection for relational data. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(12):1509−1525. [doi: 10.1109/TKDE.2004.94]

[21] Zhou F, Zhao HX. Relational database watermarking algorithm based on chaos and DCT. Application Research of Computers, 2012, 29(2):786−788 (in Chinese with English abstract). [doi: 10.3969/j.issn.1001-3695.2012.02.104]

[22] Huang C, Zhu YL. Fast algorithm for arbitrary length discrete cosine transform. In: Proc. of the 5th Int'l Conf. on Natural Computation. Tianjin, 2009. 390−393. [doi: 10.1109/ICNC.2009.640]

[23] Chen B, Wornell GW. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. IEEE Trans. on Information Theory, 2001,47(4):1423−1443. [doi: 10.1109/18.923725]

[24] Li YX, Liu GH. Order preserving encryption method for character data in relational databases. Radio Engineering, 2006,36(4):1−3 (in Chinese with English abstract).

附中文参考文献:

[2] 项世军,罗欣荣.基于同态公钥加密系统的图像可逆信息隐藏算法.软件学报,2016,27(6):1592−1601. http://www.jos.org.cn/1000-9825/5007.htm [doi: 10.13328/j.cnki.jos.005007]

[3] 黄刘生,田苗苗,黄河.大数据隐私保护密码技术研究综述.软件学报,2015,26(4):945−959. http://www.jos.org.cn/1000-9825/4794.htm [doi: 10.13328/j.cnki.jos.004794]

[5] 田秀霞,王晓玲,高明,周傲英.数据库服务——安全与隐私保护.软件学报,2010,21(5):991−1006. http://www.jos.org.cn/1000-9825/3746.htm [doi: 10.3724/S P.J.1001.2010.03746]

[11] 马友忠,孟小峰.云数据管理索引技术研究.软件学报,2015,26(1):145−166. http://www.jos.org.cn/1000-9825/4688.htm [doi: 10.13328/j.cnki.jos.004688]

[21] 周飞,赵怀勋.基于混沌的 DCT 域关系数据库水印算法.计算机应用研究,2012,29(2):786−788. [doi: 10.3969/j.issn.1001-3695.2012.02.104]

[24] 李亚秀,刘国华.关系数据库中字符数据的保序加密方法.无线电工程,2006,36(4):1−3.

项世军(1974－),男,贵州普定人,博士,教授,CCF 高级会员,主要研究领域为信息隐藏,加密域信号处理.

何嘉勇(1992－),男,硕士,主要研究领域为多媒体信息安全,加密域信号处理.