

基于改进的 KPPCT 编码的防篡改动态图水印方案

陈艳琳, 杨秋翔, 陈够喜

(中北大学 电子与计算机科学技术学院, 山西 太原 030051)

摘 要: 针对动态图水印中的 KPPCT 编码的数据率低和鲁棒性差的问题, 提出了一种改进的 KPPCT (Improved Radix-k Planted Plane Cubic Tree, IKPPCT) 编码方案. 该方案利用二叉树左右分支的唯一性进行系数编码, 扩大了水印值的表示范围; 并且引入了防篡改的思想, 用以验证水印结构是否被破坏, 加强了保护的功能. 理论分析和实验结果表明, 该改进方案提高了数据率和鲁棒性, 是一个较好的方案.

关键词: 动态图水印; IKPPCT 编码; 防篡改; 数据率; 鲁棒性

中图分类号: TP309

文献标识码: A

文章编号: 1000-7180(2013)11-0035-04

A Tamper-proofing Dynamic Graph Watermarking Scheme Based on Improved KPPCT Encoding

CHEN Yan-lin, YANG Qiu-xiang, CHEN Gou-xi

(School of Electronics and Computer Science and Technology, North University of China, Taiyuan 030051, China)

Abstract: This paper proposed a improved KPPCT (IKPPCT, Improved Radix-k Planted Plane Cubic Tree) encoding scheme towards the problem of the lower data rate and the weak robustness of KPPCT encoding in dynamic graph watermarking. This scheme took advantage of uniqueness of binary tree's left-right branches to encode coefficient and expanded the range of watermarking value. And also it strengthened the ability to protect software by introducing the tamper-proofing to verify whether the watermarking structure was destroyed. Theoretical analysis and experimental results show that this scheme has high data rate and robustness, and is a better scheme.

Key words: dynamic graph watermarking; IKPPCT encoding scheme; tamper-proofing; data rate; robustness

1 引言

随着信息技术和计算机网络的不断发展和广泛应用, 软件盗版问题变得日益严重, 因此软件保护也越来越受到重视. 软件水印是目前最常用的一种软件保护技术之一^[1].

根据水印的嵌入时刻, 可分为静态水印和动态水印. 最具有潜力的就是文献[2-3]提出的著名的 CT 算法, 即动态图水印 (DGW) 算法, 该水印能有效抵抗代码优化和迷惑攻击. 图结构的选取对于动态水印系统来说非常重要, 目前有四种编码方式: Radix-k 编码、排列图编码、父指针树、PPCT^[2-3].

文献[4-5]对编码结构进行了改进, 也取得了一

定的进展, 可是软件单靠水印嵌入来保护是远远不够的. 本文提出了一种改进的动态图软件水印编码方案.

2 动态图水印 (DGW)

根据目前数论中的大数难分解, 产生两个大整数 P 和 Q (P 和 Q 通过某种转换, 可以不是素数), 用 N (P 和 Q 的乘积) 代表软件版权的信息, 即水印值. 将 N 转换成某种图结构嵌入到软件中, 当输入正确的密钥时, 会生成水印图拓扑结构, 版权所有者可以根据检测到的拓扑结构计算出 N . 再者由于大数分解问题是算法中的 NP 问题, 所以即使检测到 N 的值, 也很难求出 P 和 Q , 因此可以证明软件所有者的

版权信息.

2.1 DGW 的嵌入与提取

DGW 的嵌入是指在程序运行时将 N 表示成一种图拓扑结构嵌入到软件中,具体过程描述如下^[3]:

- (1)产生一个代表版权信息的大数 N ;
- (2)将 N 映射到某种图结构;
- (3)把上述图结构转化成水印代码;
- (4)把水印代码嵌入到程序中.

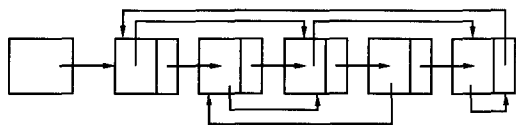
在输入正确的密钥后,程序会生成水印图结构.水印的提取是水印嵌入的逆过程,由于水印嵌入者对水印图结构的相关特性是清楚的,所以可以很容易的分析得到水印值 N ,从而证明其版权^[6].

2.2 现有的 DGW 编码方案

DGW 水印的编码方案主要有基数 K 编码、PPCT 枚举编码、KPPCT 编码等.

2.2.1 基数 K 编码

基数 K 编码^[7-8]是由 1 个头节点和 $K-1$ 个节点的循环链表组成.除了头节点以外,每个节点都包含两个指针,左指针和右指针,节点的右指针指向下一个节点,左指针用来编码系数,如果为空,系数为 0;指向自身,系数为 1,依次类推.图 1 所示为基数 6 的编码.基数 K 编码所能表示的水印值范围为 $0 \sim k^{k-1} - 1$.该方案编码率高,但是结构简单,抗攻击性较差.



$$61 \times 73 = 3 \times 6^4 + 2 \times 6^3 + 3 \times 6^2 + 4 \times 6^1 + 1 \times 6^0$$

图 1 基数 k 编码

2.2.2 PPCT 枚举编码

PPCT 枚举编码^[7-8]的实质是在二叉树的性质上增加了一个右指针指向根节点的生成节点(Original Node).二叉树的所有节点和生成节点组成一个循环链表.

图 2 所示为有 4 个叶节点的 PPCT 枚举编码.有 $2n$ 个节点(n 个叶节点)的 PPCT 拓扑结构图,可以编码的水印值为 $0 \sim \frac{1}{n}C_{2n-1}^{n-1} - 1$.而在该集合内的 PPCT 大小定义规则在文献^[7-8]中有详细的定义.

PPCT 编码结构稳定,抗攻击性好,但是编码效率不高.

2.2.3 KPPCT 编码方案

KPPCT 编码方案整合了基数 K 编码和 PPCT

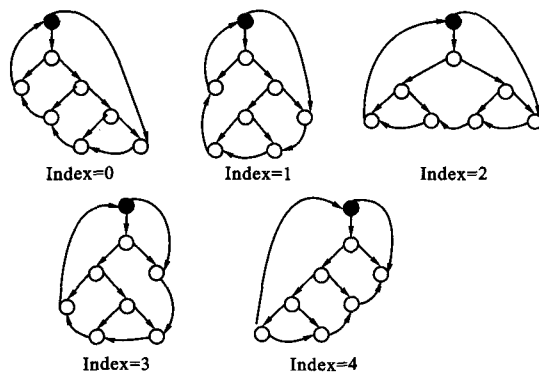
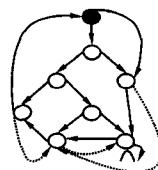


图 2 有 4 个叶节点的 PPCT 枚举树

编码的优势,编码率较高而且稳定.该结构是在 PPCT 图结构的基础上,把叶节点的右指针看作是基数 K 编码中的左指针,用来表示系数.具有 m 个叶节点的 KPPCT 结构可以表示的水印值范围为 $0 \sim (m+1)^m - 1$.图 3 所示为 KPPCT 编码的一个例子.



$$603 = 3 \times 5^3 + 0 \times 5^2 + 4 \times 5^1 + 4 \times 5^0$$

图 3 KPPCT 编码

3 改进的 KPPCT 编码方案

3.1 IKPPCT 编码的思想

本编码方案在 KPPCT 的基础上,利用二叉树左右分支的唯一性进行系数编码,扩大了水印值的表示范围.在该编码结构中,生成节点左指针指向的叶节点的幂为 0,该叶节点左指针指向的下一个节点的幂为 1,以此类推,最左边的节点的幂为叶节点数目-1.编码方案的基本思想:

(1)本编码方案是用 java 语言实现的,每个节点定义为一个类对象,具体的属性信息字段如表 1 所示.其中,nodeID 为节点编号;leftPointer 为该节点所指向的左孩子的节点编号;rightPointer 为该节点所指向的右孩子的节点编号;flag 标识是否为叶节点,若是叶节点,则标明是第几个叶节点,从最右边开始;radix 是叶节点右指针参与编码的系数值;path[]是在 KPPCT 编码结构的基础上新增的一个属性域.

(2)从生成节点(Original Node)的右指针开始,遍历所有的叶节点,分支按照左指针 1 和右指针 0 的规则进行编码,如图 4 所示.

(3)节点新增加的属性域 path[],用以存放从生成节点到该叶节点的分支路径信息。

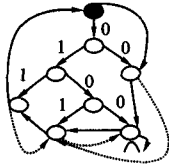
(4)生成节点的 flag 域中存放原始水印值。

(5)提取水印时,先执行防篡改代码:①访问生成节点中的 flag 域中的原始水印 N_1 ;②将 N_1 和计算出的水印值 N_2 进行比较,如果相等,说明水印完整,否则,水印遭到破坏。

表 1 节点的属性字段

nodeID	leftPointer	rightPointer	flag	radix	path[]
--------	-------------	--------------	------	-------	--------

m 个叶节点的 IKPPCT 编码所能表示的水印值范围为 $0 \sim \sum_{i=0}^{m-1} (a_i + b_i)(m+1)^i$, 其中 a_i 和 KPPCT 编码中的 a_i 一样,而 b_i 等于叶节点的 Data1 中的值, b_i 的范围为 $0 \sim 2^{m-1} - 1$ 。



$$1123 = (3+0) \times 5^4 + (0+4) \times 5^3 + (4+5) \times 5^2 + (4+3) \times 5^1$$

图 4 IKPPCT 图结构编码

3.2 IKPPCT 水印的嵌入和提取

水印的嵌入是一个动态过程,如果输入的嵌入水印口令正确,那么程序生成水印结构。提取水印时,同样需要输入正确的水印提取口令。

以图 4 为例,水印嵌入的程序如下:

```
public class MainIKPPCTEncoding{
    public static void main(String[] args){
        ArrayList<Node> arr;
        Scanner s1 = new Scanner(System.in);
        String pwd1 = s1.nextLine();
        //验证嵌入水印口令是否正确
        if (pwd1.equals("embedding")){
            System.out.println("请输入一个水印值:");
            Scanner s2 = new Scanner(System.in);
            double N1 = s2.nextDouble();
            Embed e = new Embed(N1);
            //embedding 是水印嵌入方法
            arr = e.embedding();
            System.out.println("水印嵌入完成!");
            //此处是水印识别的过程
            Scanner s3 = new Scanner(System.in);
            System.out.println("提取水印值口令:");
            String pwd2 = s3.nextLine();
            //验证提取口令是否正确
            if (pwd2.equals("recognizing")){
```

```
                Recognize r = new Recognize(arr);
                //recognizing 是水印提取方法
                double N2 = r.recognizing();
                if (N1 != N2){
                    水印被破坏;
                    程序进入故障模式;
                }
                System.out.println("水印值 = " + N2);
            }
        }
        else
            System.out.println("识别口令有误");
        }
        else
            执行正常的功能
    }
}
```

提取水印值时需要给提取方法传递一个水印结构,本方案采用的是 ArrayList<Node> 类型,上述代码中传递的是嵌入完成之后返回的链表 arr。因为水印的结构特性水印嵌入者是很清楚的,所以根据链表能够识别出水印值来。

3.3 防篡改思想

把嵌入的水印图结构和程序功能结合起来,能够实现防篡改,起到双重保护的目的。提取水印时,要求输入提取口令,口令正确才执行防篡改代码。本文的防篡改思想是,把保存的原始水印值与计算出的水印值进行比较,如果相等,则水印结构完整,否则,程序进入故障模式。

```
//N1 为保存的原始水印, N2 为计算出的水印值
if (N1 != N2){
    水印被破坏;
    程序进入故障模式;
}
```

4 性能分析

4.1 数据嵌入率

如果能以最少的节点表示最大的自然数,那么这种水印结构就拥有最大的数据嵌入率。假设有 m 个叶节点参与编码,根据数据嵌入率的定义,可以得到以下四种水印图结构的数据嵌入率:

$$DR_{Radix-k} = \log_2(m+1)^m / 2m + 1 \approx \log_2 m / 2, m = k - 1$$

$$DR_{PPCT} = \log_2 \left(\frac{1}{m} C_{2m-2}^{m-1} \right) / 4m \approx 0.5, m \text{ 达到一定程度}$$

$$DR_{KPPCT} = \log_2(m+1)^m / 4m \approx \log_2 m / 4$$

$$DR_{IKPPCT} = \log_2 \{ (m + 1)^m + [2^{m-1} - 1 + (2^m/2m + 1)](m + 1)^{m-1} \} / 4m = \log_2 m / 4 + \log_2 [2^{m+1} + m + (2^m/2m + 1)] / 4m$$
 m 足够大时, $\approx \log_2 m / 4 + 1/4$

从上述式子可以看出,IKPPCT 编码结构要比 KPPCT 编码结构的编码率高,这样可以表示的水印值范围也就大很多. 当要编码的水印值很大时,较少的节点就可以编码此水印值,所以节约了空间.

4.2 鲁棒性

鲁棒性是指水印面临攻击者攻击时的抵抗能力和自我修复能力. 该编码方案的鲁棒性体现在:

(1)动态图水印是在程序运行时动态生成的,不容易定位分析,而且本方案在实施时,运用了大量的口令判断,用以保护水印的嵌入与提取.

(2)IKPPCT 编码方案继承了 KPPCT 编码方案的结构稳定、抗攻击能力强的优点.

(3)IKPPCT 编码方案引入了防篡改的思想,提取水印时先验证水印结构有没有被破坏掉,如果被破坏了,那么程序进入故障模式,这样可以阻止攻击者进一步攻击.

通过水印值分析和隐蔽性分析表明,本文提出的改进方案虽然在数据率上低于 Radix-k 编码,但是其继承了二叉树结构稳定的优点,所以抗攻击能力要明显好于 Radix-k 编码. 其次,本方案所带来的空间过载和时间过载也是在可以接受的范围内. 综合数据率、隐蔽性和鲁棒性,本文提出的改进方案的整体性能是最优的. 通过定性分析可以得出几种编码结构的整体性能,如表 2 所示.

表 2 四种编码方案的整体性能

方案	数据率	隐蔽性	鲁棒性	整体性能
Radix-k	$\log_2 m/2, m=k-1$	好	不好	一般
PPCT	$\log_2 \left(\frac{1}{m} C_{2m-2}^{m-1} \right) / 4m$	好	好	一般
KPPCT	$\log_2 m/4$	好	好	较好
本方案	$\log_2 m/4 + 1/4$	好	好	好

5 结束语

事实上任何一种软件水印技术如果在充足的时间和技術支撑条件下都是可攻破的. 本编码方案在

KPPCT 编码方案的基础上引入了左右分支唯一性的思想,扩大了水印值的编码范围. 并且把程序控制流和水印提取结合起来,体现了防篡改思想,提高了水印的鲁棒性.

本方案虽然扩大了水印值的表示范围,也体现了防篡改思想. 但并没有提高水印结构的隐蔽性,还有在把防篡改代码和程序控制流结合起来的方式上比较单一,所以提高隐蔽性和选择合适的结合点是本编码方案下一步要研究的问题.

参考文献:

[1] 张立和,杨义先,钮心忻,等. 软件水印综述[J]. 软件学报,2003,14(2):268-277.

[2] Collberg C, Thomborson C. Software watermarking: models and dynamic embeddings[C]//Proc of ACM Sigplan Sigact symposium on Principle of Programming Language. Association for Computing Machiney Press, 1999:311-324.

[3] Collberg C, Thomborson C, Townsend G M. Dynamic graph-based software watermarking[R]. Technical Report TR-04-08, Department of computer science University of Arizon & Auckland,2004.

[4] 虞涛,杨杰,胡君萍. 基于排列图和 PPCT 混合编码的软件水印技术的研究与实现[D]. 武汉:武汉理工大学,2006.

[5] 刘建蓉,秦拯,彭程. 改进的动态图水印技术编码方案[J]. 计算机应用研究,2011,28(2):720-723.

[6] Palsberg J, Krishnaswami, Kown M, et al. Experience with software watermarking[C]//Proc of the 16th Annual Computer Application Conference. Washington DC: IEEE Computer Society,2000:308-316.

[7] 陈海兵. 一种软件水印算法研究[D]. 武汉:华中科技大学,2006.

[8] 田奇. 动态图水印算法的研究与应用[D]. 吉林:吉林大学,2008.

作者简介:

陈艳琳 女,(1989-),硕士研究生. 研究方向为软件水印.

杨秋翔 男,(1969-),硕士,教授. 研究方向为网络信息安全和软件保护.

陈够喜 男,(1966-),博士,副教授. 研究方向为信息隐藏和隐写.