关于二叉树的加密算法

喻言

（浙江大学 宁波理工学院，浙江 宁波 315100）

摘 要：本文主要研究了关于二叉树的加密算法，利用二叉树的中序遍历和先序遍历（或后序遍历）可以唯一确定一棵二叉树来进行加密解密，并给出了基本算法，最后对算法的时间空间复杂度进行了一个简单的说明，并说明了其在实际生活中的应用。

关键词：二叉树；加密；解密；先序遍历；中序遍历；后序遍历

中图分类号：TP309.7 **文献标识码**：A **文章编号**：2096-4706（2018）10-0158-03

Encryption Algorithm for Binary Tree

YU Yan

（Ningbo Institute of Technology，Zhejiang University，Ningbo 315100，China）

Abstract：This paper mainly studies the encryption algorithm of binary tree，which can uniquely determine a binary tree for encryption and decryption by using the middle-order traversal and the first-order traversal （ or the second-order traversal ） of the binary tree，and gives the basic algorithm. finally，it gives a simple explanation of the time-space complexity of the algorithm，and illustrates its application in real life.

Keywords：binary tree；encryption；decryption；first-order traversal；middle-order traversal；later-order traversal

0 引 言

在数据结构中，树是一类非常重要的数据结构类型，其在生活中和理论中都有非常重要的应用。其中二叉树是树的一种特殊的结构，当然也是比较基础又比较重要的数据结构。二叉树有许多的性质，可以设计二进制数的前缀编码，哈夫曼树就是一个很典型的例子。关于二叉树的三种遍历：先序遍历，中序遍历，后序遍历；这其中只需已知中序遍历和另外任意一种遍历就可以唯一确定一棵二叉树，而根据先序遍历和后序遍历是无法确定一棵二叉树的。因此，可以利用二叉树的这些性质，实现对信息的加密和解密。因此本文介绍了利用二叉树进行加密与解密的方法，并说明了利用多种遍历实现密钥的多方面保存。

1 树与二叉树

1.1 树的定义

树是 N（N≥0）个结点的有限集合，N=0 时，称为空树，这是一种特殊情况。在任意一颗非空树中满足：

（1）有且仅有一个特定的成为根的结点。

（2）当 N＞1 时，其余结点可分为 m（m＞0）个互不相交的有限集合 T1，T2，…，Tm，其中每一个集合本身又是一棵树，并且称为根节点的子树。

1.2 二叉树的定义

二叉树是另一种树形结构，其特点是每个结点至多只有两棵子树（即二叉树中不存在度大于 2 的结点），并且二叉树的子树有左右之分，其次序不能任意颠倒。

收稿日期：2018-06-28

与树相似，二叉树也以递归的形式定义，二叉树是 n（n≥0）个结点的有限集合：

（1）或者为空二叉树，即 n=0。

（2）或者由一个根结点和两个互不相交的被称为根的左子树与右子树组成。左子树和右子树分别是一棵二叉树。

2 加密算法

2.1 加密过程

在使用二叉树加密算法对信息进行加密时，首先信息的拥有者要构造一棵二叉树树形用于对信息进行加密，加密之后，信息拥有者可以把该树的前序遍历和中序遍历分别交给不同的人，这样就可以实现密钥的多方保存，使加密具有更强的稳定性，不容易被破解。

二叉树加密实际上就是利用二叉树对二进制信息进行编码的过程，定义一个加密二叉树，该二叉树的任意一个结点用 Pi 表示，任意一个结点的左分支用字符“0”表示，右分支则用字符“1”表示，加密二叉树的树形是不确定的。

那么如何进行加密呢？首先，要把明文信息转化为对应的二进制字符串，因此加密的过程就是分解明文的字符串的过程，从根结点开始，按“0”或“1”查找其“左孩子”和“右孩子”，直到到达叶结点或者只有一个“孩子”的树结点，从而就得到了该子串对应的结点内容，把所有得到的结点内容按顺序排列在一起，即可得到密文。而密钥则是加密二叉树的前序遍历，中序遍历或者后序遍历。

如图 1 所示，假设以上为一棵加密二叉树，假如明文信息为 0011010110110100，则对应的密文为 P4P6P7P6P8P9，解密之后的明文为 0011010110110100。

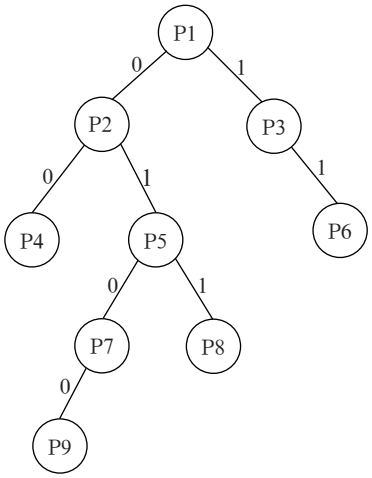

图 1 加密二叉树

由上可知，加密二叉树的树形和算法的安全性有着极为重要的联系，下面就加密二叉树树形进行讨论：

一般来说，在对明文进行加密前，首先要构建一棵完全二叉树，然后对其进行减枝和易位操作，由此即可形成加密二叉树。

对结点 Pi 进行减枝操作是指：将以 Pi 为根的子树插入到一个孩子数小于 2 的结点之下，作为其左孩子或右孩子。

对结点 Pi 与 Pj 进行易位操作是指：交换二叉树中分别以 Pi 与 Pj 两个结点为根的子树的位置。

虽然加密二叉树的树形可以任意建立，但是有些树形是不可以用来加密使用的，例如只存在左分支和右分支的二叉树，这种树形可以加密的信息极为有限，而且编码中含有较多连续的“0”和“1”，容易被破解；完全二叉树也不可以任意建立，因为加密二叉树是根据完全二叉树转变而来的，破解者首先会对完全二叉树进行破解，因此也很容易被破解。

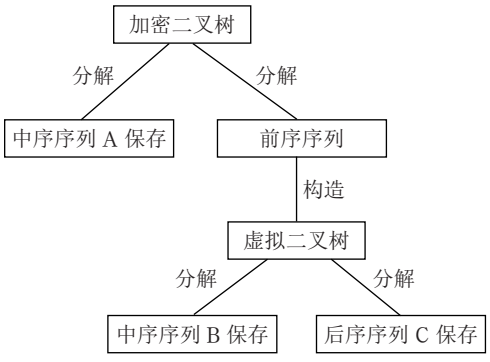

图 2 密钥的多方面保存

信息的拥有者在进行信息保密时，并不是只可以把密钥分给两个人，而是可以根据具体需求分派给多个人，例如要将密钥分给 A、B、C 三人，可以将加密二叉树的中序序列先分给 A，然后利用该树的前序遍历任意构造一棵二叉树，虽然树形可能不同，但是前序遍历序列和加密二叉树是相同的，因此把这棵由前序序列构造出的树称为虚拟二叉树（如图 2 所示），然后可将虚拟二叉树中的前序遍历和后序遍历分给 B 和 C 保存，这样就实现了密钥的三方保存。如果想要再继续将密钥分给更多的人，则可以用后序序列构造虚拟二叉树，将该虚拟二叉树的前序序列和中序序列分给别的人，以此类推，可以把密钥分派给无数的人。

但需要注意的是，在构造虚拟二叉树时，只可以用前序遍历和中序遍历或者中序序列和后序序列来构造，不可以只用中序序列构造虚拟二叉树，否则无法继续分割。将这些密钥保管者保管的密钥信息组合起来的过程就是密钥分解的逆过程，例如将之前 A、B、C 三者保管的密钥取出，先由虚拟二叉树的中序序列和后序序列构造出虚拟二叉树，得出此虚拟二叉树的前序序列，再由此前序序列和 A 保管的中序序列求出原加密二叉树的树形，也就是完整的密钥。

下面就上面的例子来详细说明一下加密过程：

假设明文信息为 0011010110110100：

（1）先取明文第一个数字 0，从根结点开始向左转，走到结点 P2，取第二个数字 0，由于 P2 有左孩子，因此继续向左转，走到结点 P4，因为 P4 没有孩子结点，所以编码到此为止，第一个密文信息为 P4。

（2）取明文第三位数字 1，从根结点开始向右转，走到 P3，取下一位数字 1，由于 P3 有右孩子，因此走到 P6，由于 P6 无孩子结点，因此编码结束，该密文信息为 P6。

（3）取下一位数字 0，从根结点开始向左转，走到 P2，再取下一位数字 1，由于 P2 有右孩子，因此右转到达 P5，取下一位数字 0，由于 P5 有左孩子，因此左转走到 P7，取下一位数字 1，由于 P7 无右孩子结点，因此编码结束，该密文为 P7。

（4）取下一位数字 1，从根结点出发右转，走到 P3，取下一位数字 1，由于 P3 有右孩子，因此走到 P6，由于 P6 无孩子结点，因此编码结束，该密文信息为 P6。

（5）取下一位数字 0，从根结点出发向左转，走到 P2，取下一位数字 1，向右转走到 P5，取下一位数字 1，向右转走到 P8，由于 P8 无孩子结点，因此编码结束，该密文信息为 P8。

（6）取下一位数字 0，从根结点出发走到 P2，取下一位数字 1，向右转走到 P5，取下一位数字向左转走到 P7，取下一位数字 0 向左转走到 P9，编码结束。

因此该密文信息为 P4P6P7P6P8P9，转化成明文信息为 0011010110110100。

2.2 解密过程

要想进行解密，首先要获得各个保管者所保管的密钥，也就是前序遍历、中序遍历和后序遍历的序列，然后不断地往回求对应的遍历序列，最终可以得到原二叉树树形，也就可以轻松得到明文信息了。

那么如何根据密钥来还原二叉树呢？下面就给出回复二叉树的方法：

根据二叉树的前序遍历和中序遍历得到后序遍历：

```
BT* restore(char *pre,char *in,int k)//pre 为先序遍历序列，in 为后序遍历 // 序列，k 为结点个数
{
BT *p;
for(int i=0;i<k;i++)
```

```
{
if(pre[0]==in[i])
{
p=(BT*)malloc(sizeof(BT));
p->s=in[i];
p->L=restore(pre+1,in,i);// 恢复左子树
p->R=restore(pre+i+1,in+i+1,k-(i+1));// 恢复右子树
return p;
}
}
return NULL;
}
```

根据二叉树的后序遍历和中序遍历得到后序遍历：

```
BT* restore(char *ord,char *in,int k)//ord 为后序遍历序列，in 为中序遍 // 历序列，k 为结点个数
{
BT *p;
for(int i=0;i<k;i++)
{
if(ord[k-1]==in[i])
{
p=(BT*)malloc(sizeof(BT));
p->s=in[i];
p->L=restore(ord,in,i);// 恢复左子树
p->R=restore(ord+i,in+i+1,k-(i+1));// 恢复右子树
return p;
}
}
return NULL;
}
```

最终遍历原二叉树即可获得明文，解密也就完成了。

2.3 算法时间复杂度分析

时间复杂度：解密算法的核心在于如何使用中序序列和前序序列（或者后序序列）还原二叉树，求得想要的遍历序列，假设恢复一个结点的时间复杂度为 1，n 为结点个数，则整个算法的时间复杂度为 O（n）。

空间复杂度：假设加密二叉树有 N 个结点，采用 4 字节一结点存储，则其先序遍历和中序遍历需要 8N 个字节；每个二叉树结点包含一个数据域，一个记录指向左孩子结点的指针，一个记录指向右孩子结点的指针，在 C 语言中，一个指针占 4 个字节的存储空间，那么一棵二叉树占用 12N 个字节的存储空间，总共需要 20N 字节的空间。

3 实际应用

3.1 用于密钥的多方保管

如果 A 想要保密一些信息，但是又怕被别人窃取，这时候就可以利用二叉树加密的密钥了，可以把原信息转化为二进制数字，然后构建加密二叉树，将其中序序列交给一个人保管，再用前序序列构造虚拟二叉树，将虚拟二叉树的中序序列和后序序列交给不同的人保管，如此，可以实现密钥的多方保存，使得信息加密更加稳固。

3.2 用于加密和解密

假如 A 想要发给 B 一段信息，又担心信息会被别人窃取，那么就要进行加密，A 可以先生成一棵加密二叉树，然后把中序序列发给 B，并通过秘密渠道发给 B 一个先序序列，这样 B 收到两个序列之后就可以生成原树，进而可以进行解密得到明文。

4 结 论

本文介绍了有关二叉树加密解密过程以及相关应用，二叉树的加密不仅可以实现对信息进行有效加密，而且可以实现密钥的多方保存，使加密具有更强的稳固性，在实际应用方面也具有重要价值。

参考文献：

[1] 秦科 . 二叉树与信息加密 [A]. 重庆计算机学会 .’2004 计算机应用技术交流会议论文集 [C]. 重庆计算机学会，2004：3.

[2] 陈伟，付宇洁，秦科 . 基于二叉树的加密算法 [J]. 实验科学与技术，2006（S1）：81-83+125.

作者简介：喻言（1996.05-），男，汉族，浙江人，本科。研究方向：软件工程。

（上接 157 页）现状与展望 [J]. 中华流行病学杂志，2014，35（6）：617-620.

[4] 郭丽，胡栋，王俊，等 . 生物医学大数据背景下学习生物信息学的学科特点分析 [J]. 高教学刊，2016（19）：48-49.

[5] 宁康，陈挺 . 生物医学大数据的现状与展望 [J]. 科学通报，2015，60（Z1）：534-546.

[6] 刘雷 . 大数据时代的生物医学 [OL].[2014-09-22]. https://blog.csdn.net/aimatfuture/article/details/39480617.

[7] 朱蕊，彭龑 . 医疗大数据的应用 [J]. 中国西部科技，2015，14（5）：95-97.

[8] Levine AG. An explosion of bioinformatics careers [J]. Science，2014.

[9] 姚贺文 . 大数据背景下的生物医学的现状与发展 [J]. 人人健康，2016（22）：296.

[10] 周雪晴，罗亚玲 . 信息化建设中医疗大数据现状 [J]. 中华医学图书情报杂志，2015，24（11）：48-51.

[11] 吕欣，韩晓露 . 大数据安全和隐私保护技术架构研究 [J]. 信息安全研究，2016，2（3）：244-250.

作者简介：李全权（1991-），男，助理馆员，硕士。研究方向：科研信息管理。