关于二叉树的加密算法

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

小组成员：白烨淞、马震宇、余波、黄奔、欧阳毓柯

Encryption Algorithm for Binary Tree

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

**引 言**

在数据结构中，二叉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。关于二叉树的三种遍历：先序遍历、中序遍历和后序遍历；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．实现过程**

**1.1 二叉树的定义**

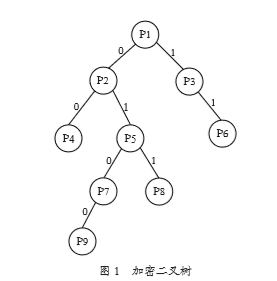
二叉树是 n（n≥0）个结点的有限集合：

（1）或者为空二叉树，即 n=0。

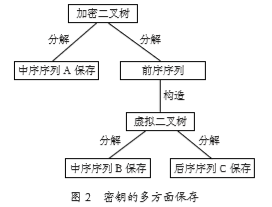
（2）或者由一个根结点和两个互不相交的被称为根的 左子树与右子树组成。左子树和右子树分别是一棵二叉树。

**1.2 加密算法**

首先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，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。如图 1 所示，假设以下为一棵加密二叉树，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。



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。 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 进行剪枝操作是指：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。



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。

**1.3 解密算法**

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。

根据二叉树的前序遍历和中序遍历得到后序遍历的算法实现：

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．算法时间复杂度分析**

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．结果及应用分析**

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。过程如下图2，图3。

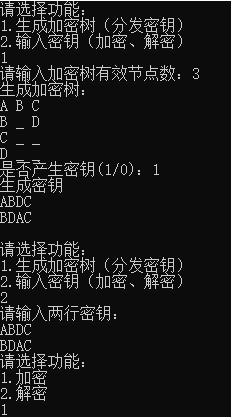
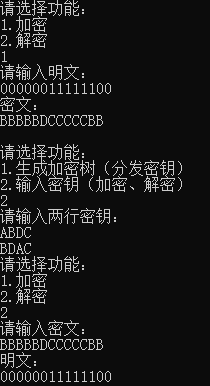
 

图2加密树及密匙生成 图3 加密及解密过程

**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。