WPS 文本文件误加密的处理方法

杨锁昌 陈鹏 (军械工程学院)

一、前 言

金山汉字系统的 WPS 功能强大,操作简单,是目前应用较为广泛的文字处理软件之一。

WPS 编辑系统具有加密功能,用户在处理一些重要文件时,可以自行设置密码。这一功能的存在解除了用户担心泄密的忧虑。但也存在某些问题:

1.使用者遗忘了密码或丢失了密码,那么将无法再使用或编辑该加密文件。

2.使用者在编辑文件时因误操作设置了密码,存盘后系统对文件进行加密处理,在不知密码的情况下也无法再编辑该文件。

3.使用者在前一次使用中并未设置密码,但在第二次编辑时却无故出现"输入密码"提示,而且密码无从得知。

这些情况会给使用者带来极大不便,造成一些不必要的损失。解决上述问题,首先应了解 WPS 文件的结构和加密原理。

二、WPS 文本文件的结构

在 WPS 文本文件的首部有一个 1K 字节的文件头,其中包含有该文件的版本标志、所定义的块的位置、左右边界、恢复删除的数据个数及内容、密码、自定义字号大小及篇眉等等一些文件的隐含设置。在它的第 733–740 连续的 8 个字节中存放着文件的 8 位密码代码值。这 8 个字节的代码值是由 8 个密码字符的 ASCII 码值经过一系列运算得到的。简单的说就是:"高四位与低四位交换,然后逐位取反"而得。例如输入密码时,键入了"A",其 ASCII 码值为"41",即:"01000001",然后四位交换为"00010100",再逐位取反为"11101011",所以密码值为"EB"。

此外在文件头中 472–727 共 255 个字节是恢复删除数据区,正好在密码区的前面,其中存放着前一次删除的内容。如果用户一次删除的内容较多,在文件头的恢复删除数据区中存放不下,就会向下存放,占据了密码区。而在存盘时 WPS 就会根据密码区中的值对文件进行了加密。因此第二次调用时,系统认为用户设置了密码而询问,这就是"无故出密码"的原因。

三、WPS 文本文件的加密原理

WPS 文件在设置密码以后,其加密的过程因版本的不同而不同。

1.对于 WPS2.1 版本的文本文件,加密时从正文开始依次取出代码值与密码代码值相异或,其结果为密文的代码值。例如:正文的第一字节与第一个密码代码相异或,第二个字节与第二个密码代码相异或,……,第九个字节与第一个密码代码相异或,第十个字节与第二个密码代码相异或,如此循环直到文件尾。

2.对于 WPS3.0F 版及 UCDOS 中的 WPS2.2 版本的文本文件,加密的过程要复杂一些,在加密原文时是以 2 个字节为一组处理的。从正文开始前两个字节代码值与前两个密码代码相异或,得到两个字节的密文代码。第三个原文代码与第一个原文代码相加后再与第三个密码代码相异或,为第三个密文代码。第四个原文代码与第二个原文代码相加时有两种情况:若一、三代码相加大于 255,则二、四代码相加后再加 1,否则直接相加。然后再与第四个密码代码相异或,得第四个密文代码,……第九个原文代码与第七个原文代码相加后与第一个密码代码相异或得第九个密文代码。第十个原文代码与第八个原文代码相加时也要由第七、九代码相加是否大于 255 来决定其相加后是否加 1,最后再与第二个密码代码相异或为第十个密文代码值,如此循环直至原文件尾。

需要提醒用户注意的是:不论用户口令密码为几个字节,系统都取密码区的 8 个字节为密码代码对正文进行加密处理。

了解了 WPS 文本文件的加密原理,就可以采用相反的运算进行解密。

四、WPS 文本文件的解密程序

笔者编写了一个用来对 WPS 文本文件进行解密的程序。该程序可以将一个已加密的文本文件解密并转换为一个正文文件,也可以显示出已加密文件的密码。在解密过程中,不仅将正文完整无误地恢复,而且控制符也能够全部保留,不需要解密后再重新编辑。

该程序中包含四个函数,分别是:

1.函数 judge wps file()用来判断 WPS 的版本号并取出密码值,若待转换的文件是 WPS 文本文件,则函数返

回文件的版本标志 1 或 2 或 3(其中 1 为 WPS2.1 版,2 为 UCDOS 中的 WPS2.2 版,3 为 WPS3.0F 版),若不是 WPS 文件则函数返回 1。

2.函数 wps50()用来对 WPS2.1 版本的文本文件进行解密。

3.函数 wps60orucdos()用来对 WPS3.0F 或 WPS2.2 版本的文本文件进行解密。

4.函数 disp_pass()用来显示该加密文件的密码。如果密码为可键入字符,则直接显示;否则显示错误字符(errorchar)。

本程序的使用有两种格式:

1.wpstotxt＜WPS 格式文件名＞＜解密文本文件名＞

该方式将生成一个无密码的正文文件。

2.wpstotxt＜WPS 格式文件名＞

该方式将显示出加密文件的密码。

```
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "dos.h"
#include "dir.h"

void wps60orucdos(FILE *readfp,FILE *writefp,unsigned char *x);
void wps50(FILE *readfp,FILE *writefp,unsigned char *x);
int judge_wps_file(FILE *fpp,unsigned char *pass);
void disp_pass(unsigned char *pass,int len);

main(int argc,char *argv[])
{
FILE *fp1,*fp2;
unsigned char x[8],pass_word[9],c;
int file_mode,done,i;
struct ffblk f;
  for(i=0;i<9;i++) pass_word[i]=0x00;
  if(argc<2){
     printf("\n调用格式: wpstotxt <WPS格式文件名> <生成解密文本文件名>\n");
     printf("\n调用格式: wpstotxt <WPS格式文件名> (将显示密码)\n");
     exit(0);
     }
  done=findfirst(argv[1],&f,0);
  if(done!=0){
     printf("\n没有此文件\n");
     exit(0);
     }
  if(f.ff_fsize<1025L){
     printf("\n不是WPS格式文件或者WPS文件中无内容\n");
     exit(0);
     }
  if((fp1=fopen(argv[1],"rb"))==NULL){
     printf("connot open file");
     exit(0);
     }
  if(argc==3){
     if((fp2=fopen(argv[2],"wb"))==NULL){
        printf("connot creat file");
        exit(0);
        }
     file_mode=judge_wps_file(fp1,x);
     if(file_mode==1) wps50(fp1,fp2,x);
     else if((file_mode==2)||(file_mode==3)) wps60orucdos(fp1,fp2,x);
     else printf("\n不是WPS格式文件\n");
     fclose(fp2);
     }
  else{
     file_mode=judge_wps_file(fp1,x);
     if((file_mode==2)||(file_mode==3)||(file_mode==1)){
        for(i=0;i<8;i++){
           c=x[i];
           if(c==0x00) break;
           pass_word[i]=~(((c<<4)&0xf0)|((x[i]>>4)&0x0f));
           }
        pass_word[i]=0x00;
        disp_pass(pass_word,i);
        }
      else printf("\n不是WPS格式文件\n");
      }
  fclose(fp1);
  return 0;
}

//  WPS2.2,WPS3.0F版文本文件的解密函数
void wps60orucdos(FILE *readfp,FILE *writefp,unsigned char *x)
{
unsigned char a,c,c1,c2,c3;
unsigned int ui;
int d,m=0,count;
   fseek (readfp,1024,0);
   d=0;count=0;
   for(;;){
     a=getc(readfp);
     if(feof(readfp)) break;
     count++;
     if((count>2)&&(((float)count/2.0==count/2))){
        c=(a)^x[m];
        c=c-c2;
        ui=c1+c3;
        if(ui>=0x100) c=(c-1);
        }
     else if((count>2)&&(((float)count/2.0!=count/2))){
        c=(a)^x[m];
        c=(c-c1);
        c3=c1;
        }
     else c=a^x[m];
     m=(m+1)%8;
     putc(c,writefp);
     if(d==0) c1=c;
     else if(d==1) c2=c;
     d=(d+1)%2;
     }
}
//  WPS2.1版文本文件的解密函数
void wps50(FILE *readfp,FILE *writefp,unsigned char *x)
{
unsigned char a,c,c1,c2,c3;
int m=0;
   fseek (readfp,1024,0);
   for(;;){
     a=getc(readfp);
     if(feof(readfp)) break;
     c=(a)^x[m];
     m=(m+1)%8;
     putc(c,writefp);
     }
}
//  判断是否为WPS文件,如是取出密码且返回版本号,否则返回-1.
int judge_wps_file(FILE *fpp,unsigned char *pass)
{
unsigned char a;
int i;
long int n=733;
  a=getc(fpp);
  if(((a==1)||(a==2)||(a==3))&&(getc(fpp)==0xff)){
    for(i=0;i<8;i++){
      fseek(fpp,n+i,0);
      pass[i]=getc(fpp);
      if(feof(fpp)) break;
      }
    return a;
    }
  else return -1;
}

//   显示密码,如果是特殊键则显示其键名,不可键入的码值显示<errorchar >,
//   一般只有无故加密时才会出现<errorchar >密码
void disp_pass(unsigned char *pass,int len)
{
int i;
  printf("\npass word is ");
  for(i=0;i<len;i++){
    if(pass[i]==0x1b) printf("<esc> ");
    else if(pass[i]==0x08)  printf("<backspace> ");
    else if(pass[i]==0x09)  printf("<tab> ");
    else if(pass[i]==0x00)  printf("F1 ");
    else if(pass[i]==0x20)  printf("space ");
    else if((pass[i]>0x20)&&(pass[i]<0x7f)) printf("%c ",pass[i]);
    else printf("<errorchar> ");
    }
}
```