数据恢复实例

软件加密与

郑雪雪 编著

人民邮电出版社

计 算 机 技 术 丛 书

软件加密与数据恢复实例

郑雪雪 编著

人 民 邮 电 出 版 社

内 容 提 要

本书主要介绍了以下内容：文件加密和磁盘信息加密的方法及实例；为防止非法拷贝，对磁盘做特殊格式化的实例；针对误删和某些破坏所提供的恢复补救办法。

本书实例丰富，尤其是第六部分的前一个完整加密系统的实例，已用于实际，效果较好。相关章节详细介绍了一些应用软件（例如 PCTOOLS、NORTON、DEBUG 等）的使用方法，实用性很强，适于计算机软件开发人员及各大学院校有关专业师生参考，对各种水平的软件爱好者都会有所裨益。

计算机技术丛书

软件加密与数据恢复实例

◆ 编　　著　郑雪雪

责任编辑　段云洁

◆ 人民邮电出版社出版发行　北京崇文区夕照寺街 14 号

北京顺义向阳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：21.75

字数：542　千字　　1997 年 7 月第 1 版

印数：8 000 册　　1997 年 7 月北京第 1 次印刷

ISBN7-115-06435-0/TP・416

定价：28.00 元

丛 书 前 言

世界上发达国家普遍重视发展以计算机和通信为核心的信息技术、信息产业和信息技术的应用，一些经济发达国家信息产业发展迅速。

当前，我国处于国民经济高速发展时期。与此相伴随，必将有信息技术、信息产业和信息技术应用的高速发展。各行各业将面临信息技术应用研究与发展的大课题以及信息化技术改造的大任务、大工程。

为了适应信息技术应用大众化的趋势，提高应用水平，我们组织编写、出版了这套“计算机技术丛书”。这套丛书以实用化、系列化、大众化为特点，介绍实用计算机技术。

这套丛书采取开放式选题框架，即选题面向我国不断发展着的计算机技术应用的实际需要和国际上的实用新技术，选题不断增添又保持前后有序。

这套丛书中有的著作还拟配合出版软件版本，用软盘形式向读者提供著作中介绍的软件，以使读者方便地使用软件。

我们希望广大读者为这套丛书的出版多提意见和建议。

前　言

当你自己编制一个软件后，想为你的软件加上保护吗？作者在教学与实践中积累了一些可用于实际的对数据、文件、磁盘信息进行加密、解密的算法与程序，在这里作为实例提供给读者，希望能有益于读者。

本书中的实例都在微机上调试通过，且大部分可用于实际。有的略作改动就能有很好的效果，如第一部分中适合于少量数据加密的算法，你可以把它加到你采用的口令或密钥字的算法中去，对口令或密钥字加密。这样你可把口令或密钥字放在程序中，而真正使用的是加密后的口令或密钥字，使得你自己不会忘记而别人又得不到真正的口令或密钥字。同时，你可以在这些实例的基础上，略作些改动或将几个算法组合起来使用，就能成为你个人独特的加密方法。只要大家动手去做，相信通过这些实例一定能衍生出很多更好、更有效的加密、解密算法。

本书第一部分介绍了十几种算法的实例，有的适用于纯数字的，有的适用于任何ASCII字符，其中有一小部分是演示性的，适用于少量信息。本书第二部分的实例着重于对文件的加密，共给出了9种方法的实例，读者可将这部分与第一部分结合起来使用。第三部分是对磁盘信息加密的实例，其中穿插了一些软件工具的使用。第四部分是对磁盘作特殊格式化的实例，使非法拷贝起不了作用。但这部分的实例使用时要谨慎，搞清楚后再用，以免丢失数据或损坏磁盘（不是物理性损坏）。第五部分是对误删或某些破坏提供一些恢复和补救的方法，使尽可能地减少损失。第六部分是两个完整加密系统的实例，都是由几种方法组合起来的。尤其是第一个实例，它可对一个文件进行多种方法、多次加密，很有实用价值。

本书主要介绍了用于软件加密的各种方法所对应的实用程序，关于它们的算法在邮电出版社出版的《数据安全与软件加密技术》中有更深入的讨论，有兴趣的读者可以参阅。这些实用程序以源程序的形式给出，需要编译、链接、产生可执行文件后，方能运行。对于一些有限制性的实例都给出了说明，使用时应予以考虑。

实例中很多程序的明文文件名与密文文件名是通过输入给出的。你在初试时最好采用不同名，以保证原文件不发生变化，而真正使用时，如果允许的话，明文与密文应采用相同的文件名，即用密文覆盖明文，否则也应通过改名，使明文不致为别人获得。

如果你是初次采用加密技巧来对数据、文件加密的话，请注意，一定要将被保护数据和文件作好备份，一方面可用来对解密后的数据和文件进行正确性的检验，另一方面，万一操作错误或忘了password密钥时，不会因没有备份而使原始数据和文件丢失。

本书在编写过程中，始终得到智强电子技术公司在技术上、物质上的大力支持，在此特向智强电子技术公司总经理王道忠副教授及其公司的全体成员表示衷心的感谢。本书中有些实例参考了北工大计算机学院同学谈德豪、张建文、王凡、于海兵、周楠、冀彬彬、贾昱、董亚惠、张瑞冬、徐劲英等的部分毕业设计，在此一并表示感谢。

本书是在教学之余编写的，难免有缺点错误，望读者能批评指正。

作者

目　录

第一章　数据加密的实例

这部分的加密方法可用来加密少量的字符或数字，如密钥、口令等。你可将密钥或口令放在某一文件中，将此文件名作为输入文件，通过加密，使密钥或口令以密文形式存放，也可取加密后的内容作为密钥或口令使用。当然有些加密方法也可直接用来加密文件或数据。

本部分共有32个程序，均在PC机上调试通过。C语言的程序不作说明的话用的是Turbo C 2.0；汇编语言的程序是在Macro Assembler V5.00环境下调试的；Pascal语言的程序是在Turbo Pascal 3.0下生成COM型文件的。Basic语言的程序则应在DOS系统中的QBASIC下运行。

运行操作时，下面带_的与↙(回车)是要求用户输入的，其它是程序或DOS的显示。

1.1　单一换字

此实例(程序1.1)的明文范围为小写字母，对大写字母不作加密处理。

加密时的对应关系为：

明文 a b c d e f g h i j k l m n o p q r s t u v w x y z

密文 l k j h g f d s a q w e r t y u i o p z x c v b n m

解密时的对应关系为：

密文 a b c d e f g h i j k l m n o p q r s t u v w x y z

明文 i x v g l f e d q c b a z y r s j m h n p w k u o t

程序1.1　文件名：P1_1.C 清单如下：

```
#include <stdio.h>
static char origin[]={'l','k','j','h','g','f','d','s','a','q',
      'w','e','r','t','y','u','i','o','p','z','x','c','v','b','n','m'};
static char wrap[]={'i','x','v','g','l','f','e','d','q','c','b',
      'a','z','y','r','s','j','m','h','n','p','w','k','u','o','t'};
char filename[10],enfile[10];

void show()
{
  int i;
  printf("\n1 : NEW\n");
  printf("2 : ENCRYPT\n");
  printf("3 : DECRYPT\n");
  printf("4 : EXIT\n");
```

```
  printf("Please input 1--4\n\n");
}
void new()
{
  FILE *fp;char ch;
  printf("\n                    Create a new file ! \n");
  printf("\nPlease input the file name : \n");
  scanf("%s",filename);
  printf("Please input the file's content : (end of #)\n");
  if((fp=fopen(filename,"w"))==NULL)
    {printf("cannot open file\n");exit(0);}
  ch=getchar();
  while(ch! ='#')
    {
      fputc(ch,fp);
      ch=getchar();
    }
  fclose(fp);
  show();
}

void content()
{
  FILE *fp;char ch;
  printf("Show the file's content :");
  if((fp=fopen(filename,"r"))==NULL)
    {printf("cannot open file\n");exit(0);}
ch=fgetc(fp);
while(ch! =EOF)
    {
      putchar(ch);
      ch=fgetc(fp);
    }
  fclose(fp);
}

void encrypt()
{
  FILE *fp,*fp1;char ch;int n;
  printf("\n Creat a encrypted file! \n");
  printf("\n Please input the file name : \n");
  scanf("%s",filename);
  content();
  printf("\n Please input the encrypted file's name :\n");
```

```
    scanf("%s", enfile);
    printf("Show the encrypted file's content :");
    if((fp1=fopen(enfile,"w"))==NULL)
      {printf("cannot open file\n");exit(0);}
    if((fp=fopen(filename,"r"))==NULL)
      {printf("cannot open file\n");exit(0);}
    ch=fgetc(fp);
    while(ch!=EOF)
      {
        n=ch-97;
        if((n<26)&&(n>=0)) { putchar(origin[n]); fputc(origin[n],fp1); }
          else {fputc(ch,fp1);putchar(ch);}
        ch=fgetc(fp);
      }
fclose(fp);fclose(fp1);
show();
}
void decrypt()
{
  FILE *fp,*fp1;char ch;int n;
  printf("\n                Creat a decrypted file ! \n");
  printf("\n Please input the file name : \n");
  scanf("%s",filename);
  content();
  printf("\n Please input the decrypted file's name : \n");
  scanf("%s",enfile);
  printf("Show the decrypted file's content :\n");
  if((fp1=fopen(enfile,"w"))==NULL)
    {printf("cannot open file\n");exit(0);}
  if((fp=fopen(filename,"r"))==NULL)
    {printf("cannot open file\n");exit(0);}
  ch=fgetc(fp);
  while(ch!=EOF)
    {
      n=ch-97;
      if((n<26)&&(n>=0)) { putchar(wrap[n]); fputc(wrap[n],fp1); }
        else {fputc(ch,fp1);putchar(ch);}
      ch=fgetc(fp);
    }
  fclose(fp);fclose(fp1);
  show();
}

main()
```

```
{
  int j,k=0;
  show();
  do{
      j=getch();
      switch(j)
        {
          case    49 : new();break;
          case    50 : encrypt();break;
          case    51 : decrypt();break;
          case    52 : k=1;break;
          default    ;;
        }
      }while(k==0);
  printf("\n THANKS TO USE THIS PROGRAM !");
  getch();
}
```

不妨设可执行文件 P1_1.EXE 在 B 盘上，运行其可执行文件后，可产生如下的结果：

```
B:\>P1_1↙
1:NEW
2:ENCRYPT
3:DECRYPT
4:EXIT
please input 1——4
  输入 1（不必回车）
            Creat a new file !

Please input the file name :
text.txt↙
Please input the file's content : (end of #)
This is an apple. Do you want to eat? #↙

1:NEW
2:ENCRYPT
3:DECRYPT
4:EXIT
please input 1——4
    输入 2（不必回车）
          Creat a encrypted file !

Please input the file name :
text.txt↙
```

```
Show the file's content :
This is an apple. Do you want to eat?
Please input the encrypted file's name :
jj↙
Show the encrypted file's content :
Tsap ap lt luueg. Dy nyx vltz zy glz?

1:NEW
2:ENCRYPT
3:DECRYPT
4:EXIT
please input 1——4
    输入3 (不必回车)
            Creat a decrypted file !

Please input the file name :
jj↙
Show the file's content :
Tsap ap lt luueg. Dy nyx vltz zy glz?
Please input the decrypted file's name :
kk↙
Show the decrypted file's content :
This is an apple. Do you want to eat?

1:NEW
2:ENCRYPT
3:DECRYPT
4:EXIT
please input 1——4
    输入4 (不必回车)
    THANKS TO USE THIS PROGRAM! ↙
```

接着返回 DOS。

jj 中为加密后的内容,kk 中为解密后的内容。

1.2 字节取反

这是一种很简单的加密算法。因两次字节取反就能还原,故加密与解密能统一在一个程序中。执行奇数次是加密,执行偶数次就是解密。如 01100001 的取反为 10011110,再一次取反又为 01100001。

程序 1.2 是用 C 语言编写的。程序中调用了 DOS 的功能调用,如功能调用 AX=3D00H(对读访问打开文件)、AX=3E00H(关闭文件描述字)及 AH=4202H(LSeek 移动读/写指

针)——从文件尾开始移动指针,当CX:DX中内容为0时,可以确定当前文件的大小。这也是在C语言中使用DOS功能调用的例子。

这里加密的字符数是由文件长度来控制的。

程序1.2(文件名P1_2.C)清单如下:

```
#include<stdio.h>
#include <dos.h>
main()
{
  char c,fname[12],fname2[12];
  FILE *output, *input;
  union REGS in,out;
  struct SREGS segreg;
  long int i,filechar;
  int tmp;
  printf("Please input filename to jia—mi or jie—me:\n");
  scanf("%s",fname);
  printf("output filename:\n");
  scanf("%s",fname2);
  input=fopen(fname,"r");
  output=fopen(fname2,"w");
  if (input==NULL) {printf("Cannot open file or file not exist!! \n");
    exit(1);};
  if(output==NULL) {printf("Cannot open file!! \n");
    exit(1);};

  in.x.ax=0x3d00;
  in.x.dx=FP—OFF(fname);
  segreg.ds=FP_SEG(fname);
  intdosx(&in,&out,&segreg);
   in.x.bx=out.x.ax;
   tmp=in.x.bx;
   in.x.cx=0;
   in.x.dx=0;
   in.x.ax=0x4202;
   intdos(&in,&out);
   filechar=16*out.x.dx+out.x.ax;
   in.x.bx=tmp;
   in.x.ax=0x3e00;
   intdos(&in,&out);

   for(i=0;i<filechar;i++) {
     c=fgetc(input);
     if (c=='\n') i++;
```

```
    /* CR is TWO byte 0aH,0dH;
       But in Turbo C, CR='\n', is ONE char !!!!! */
    c=(c ^ 0xff);
    fputc(c,output);
      }
  if(fclose(input)) printf("File close error\n");
  if(fclose(output)) printf("File close error\n");
}
```

操作运行过程：

```
B:\>P1_2↙
Please input filename to jia-mi or jie-me:
Text.txt↙
output filename:
JJ.C↙
```

JJ.C 中为加密后的文件，加密后返回 DOS。

若需解密，再运行一次：

```
B:\>P1_2↙
Please input filename to jia-mi or jie-me:
JJ.C↙
output filename:
KK.C↙
```

则 KK.C 中的内容与 Text.txt 中内容一样。

如果你想覆盖原文件，在显示 output filename:时键入原文件名即可。

1.3 字符串循环异或法

这也是加密与解密完全相同的一种方法。用输入的 Password(字符串)与明文作循环异或得到密文，再用相同的 Password 与密文字符作循环异或回复明文。

程序 1.3 的实例，输入的 Password 只限为数字，位数不限，在屏幕上不显现，用此实例加密时一定要记住 Password。

程序 1.3(文件名 P1_3.C)清单如下：

```
#include<stdio.h>
main()
{
  int password=0;
  FILE *sfp,*tfp;
```

```
    char sname[10],tname[10];
    int ch;
    printf("\n Input S_file name:\t");
    scanf("%s",sname);
    printf("\n Input T_file name:\t");
    scanf("%s",tname);
    printf("\n Input Password:\t");
    while((ch=getch())! =13)
    {
      if((ch<48)||(ch>57)) continue;
      password * =10;
      password+=ch-48;
    }
    if(! (sfp=fopen(sname,"rb")))
    {
      printf("Cannot open file:%s\n",sname);
      exit(1);
    }
    if(! (tfp=fopen(tname,"wb")))
    {
      printf("Cannot open file:%s\n",tname);
      exit(1);
    }
    while((ch=fgetc(sfp))! =EOF)
    {
      ch=ch^password;
      if(ch==EOF) printf("EOF error!");
      if(fputc(ch,tfp)==EOF) printf("EOF error!");
    }
    fclose(sfp);
    fclose(tfp);
}
```

对其可执行文件 P1_3.EXE 的操作如下。

```
B:\>P1_3↙
Input S-file name:text.txt↙
Input T-file name:tf.c↙
Input password :6897754(不显现)↙
```

进行加密后返回 DOS。tf.c 中为加密后的文件,text.txt 为源文件(明文)。

```
B:\>P1_3↙
Input S-file name:tf.c↙
Input T-file name:sf.c↙
```