如何为 XENIX 文件加密

湖南省双峰县工商银行 罗 辉

摘要：本文介绍一则加解密 *XENIX* 系统下文件的程序，加解密用同一密钥同一程序，简单实用，保密性强。

在 XENIX 系统的多用户环境下，各个用户都是在各自的注册帐户下行使自己的系统权限，一般说来，互不侵入。但是，掌管超级用户密码的系统管理员或窃了密的其他用户对用户有可能构成侵权的威胁，有时候，用户也有自己一些重要的程序或文档资料不愿意让别人了解和浏览，这就存在一个文件加密问题，为此，我们参照有关 DOS 通用文件加密的思想，移植并优化了一个适用于 XENIX 系统文件加密的 C 语言程序，足可以满足这一需要。

采用位异或加密算法，明文循环地与密钥进行位异或，密钥可以为任意字符任意长度的组合，包括数字、字符、汉字、组合字符甚致控制字符的组合，还可以每次用不同的密钥进行多次加密，当然解密需以加密的逆过程进行。

由于文本文件中的空格符（ASCⅡ码为 20H）和二进制文件中的十六进制数字 00H 等数字一般很多且集中，如果直接与密钥进行位异或，密文将呈现出一定的规律，比如说密钥“luo”与一串空格符（或一串数字 00H）相异或，将出现“LUOLUOLUOLUO…… ……”（或“luoluoluoluo……”）这样一串很容易破译的密文，为避免这种情况造成的可能泄密，本程序对用户的密钥附加上一个密钥变码，然后再异或，这样，打破了密文可能呈现的规律，保证了高保密性。

在 XENIX 环境下，用 vi 命令以 jm・c 的文件名将下面程序敲入并存盘，再用编译命令 cc 编译成可执行文件后，移到二进制文件公用子目录 / bin 下即可使用：

```
#vi jm・c <CR>
#cc-o jm jm・c <CR>
#mv jm / bin <CR>
```

使用方法：jm 源文件名密钥＜新文件名＞。

其中新文件就以源文件名存盘，对同一文件，用同一密钥，厅数次为加密，偶数次为解密，它用二进制方式读写文件，适用于任何形式的文件加密，不管它是可执行的还是文本文本文件。

本程序文件加密过程快而方便，破解却不那么容易，不失为一个好加密工具。

```
#imclude <stdio ・h >
main (argc, argv)
int argc;
char * argv[]
    FILE * fpl , * fp2 ;
    int t , j , k , i = 0
    char c [4 ];
    if(argc < 3 )
       {          printf(“ / 033 [2j”);
          printf( “ / t / t Usage: jm Sfilename Keyvaule
<Dfilename> / n ”);
          exit(0 );
       }
     if((argc = = 3 ) || (strcmp(argv[1 ], argv[3 ]) = = 0 ))
          srtcpy(c , “r +b ”);
          else strcpy(c , “w +b ”);
    if((fpl = fopen (argv[1 ], “r +b ”)) = = NULL)
       {
          printf(“ / 033 [ 2 J”);
          printf(“ / t / t Source File Don’t Exist or Openi / n
”);
          exit(0 );
       }
    if(argc > = 4
       fp2 = fopen (argv[3 ], c );
    j = strlen(argv[1 ], c );
    k = 1
    while ((t = getc(fp1 ))1 = EOF)
       {
          t = t ∧(argv[2 ])[i ]+k );
          putc(t , fp2 );
          i ++;
          k ++;
          if(i = = j )i = 0 ; }
       fclose(fp1 );
    fclose(fp2 );
    printf(“ / t / t / t OK1 / n / n );
    }
```

▲