К. т. н. М. Д. СКУБИЛИН,
д. т. н. Д. А. СЕЧЕНОВ, О. Б. СПИРИДОНОВ

Россия, г. Таганрог, Гос. радиотехнический ун-т

Дата поступления в редакцию
13.04 1999 г.
Оппонент *д. ф.-м. н. Б. К. ОСТИСТЫЙ*

СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ В КАНАЛАХ КОММУНИКАЦИИ

Предлагается способ камуфляжа информации с целью защиты информации от несанкционированного доступа.

Method of the information camouflage is offered to protect the information from unauthorized access.

Известные способы защиты информации от доступа неопределенного круга лиц отличаются значительными временными затратами на шифрование текста [1], а еще большими — на его расшифровку [2], потерей части информации [3] и неидентичностью технических средств кодирования и декодирования информации [4, 5].

Не заботясь о помехозащищенности информации в канале связи, упростить процесс ее шифрования и дешифрования представляется возможным программными средствами [6].

Если информационное сообщение (текст, файл) распылить (рассредоточить — диверсифицировать) беспорядочно, то возврат к исходному тексту (его репликация) тем более затруднен, чем больше объем исходного информационного сообщения. Исходя из этого допустимо, не усложняя процесса восстановления (репликации), осуществлять преобразования, при которых исходный файл информации разбивается на блоки переменной длины и в каждом блоке осуществляется варьируемый сдвиг по кольцу ASCII-кода каждого символа блока. Закодированный (зашифрованный) таким образом файл можно оперативно декодировать (расшифровать) путем обратного сдвига символов блоков файла.

Программная реализация диверсификации и/или репликации информации на языке программирования Borland C [6] предполагает наличие конфиденциальной информации, например в файле «proba.txt», и запускающего модуля — в файле «kod.exe». При этом осуществляется ввод с командной строки KOD proba.txt KiRi, KiLi, ..., «Enter» или KOD proba.txt UiRj, UiLj, ..., «Enter» (для кодирования и декодирования, соответственно), где K — кодировать, U — декодировать, i (i=$\overline{1,m}$) — число символов в данном блоке, R — сдвиг вправо, L — сдвиг влево, j (j=1,n) —число позиций сдвига символов в данном блоке.

Программа диверсификации/репликации информации:

```
#include <stdio.h>
#include <io.h>
#include <dos.h>
#include <sys\stat.h>
#include <errno.h>
#include <fcntl.h>
main(argc, argv)
int argc; char *argv[];
{FILE *ff; int ch; int i,j;
unsigned char *ptrsymb; unsigned char symb[1024];
unsigned char str1, str2; unsigned long int size;
int sizeblock=0; int hk; int handle; int smesh=0;
ptrsymb=&symb[0];
if(argc!=4)
printf("Использование программы :\n" "C:>KOD
<имя файла> <операция> <направление>\ "\t
Операции :\n\t\tK : кодировать"\n\t\tU :
раскодировать\n""\t Направление :" "\n\t\tR :
вправо""\n\t\tL : влево."); exit(0); }
if(*argv[2]=='K')goto zk; if(*argv[2]=='U')goto uk;
else exit(0);
zk: if((ff=fopen(argv[1],"r+"))==NULL)
{printf("\nНе найден файл %s",argv[1]); exit(1);}
printf("\nКодируется файл :%s", argv[1]);
handle=open(argv[1], O_CREAT);
size=filelength(handle); close(handle);
povtor: printf("\nМаксимальный размер блока в
байтах :");
printf("%d", size); printf("\nВведите размер блока
в байтах :");
scanf("%d",&sizeblock);
if(size<sizeblock)
{printf("\nРазмер блока введен не корректно!");
goto povtor;}
printf("\nВведите ключ :"); scanf("%d",&hk);
fseek(ff,smesh,0); fread(ptrsymb,1,sizeblock,ff);
if(*argv[3]=='L')goto zk1;
for(j=0;j<hk;j++)
{str1=symb[0]&1; symb[0]=symb[0]>>1;
for(i=1;i<sizeblock;i++)
{str2=symb[i]&1; symb[i]=symb[i]>>1;
symb[i]=symb[i]|(str1<<7); str1=str2; }
symb[0]=symb[0]|(str1<<7); }
fseek(ff,smesh,0); fwrite(ptrsymb,1,sizeblock,ff);
```

```
smesh=smesh+sizeblock; size=size−sizeblock;
if (size!=0) goto povtor; fclose(ff);
printf("\nФайл %s закодирован.", argv[1]); exit(0);
zkl: for(j=0;j<hk;j++)
{ str1=symb[0]&0x80; symb[0]=symb[0]<<1;
for(i=1;i<sizeblock;i++)
{ str2=symb[i]&0x80; symb[i]=symb[i]<<1;
symb[i−1]=symb[i−1]|(str2>>7); }
symb[sizeblock−1]=symb[sizeblock−1]|(str1>>7); }
fseek(ff,smesh,0); fwrite(ptrsymb,1,sizeblock,ff);
smesh=smesh+sizeblock; size=size−sizeblock;
if (size!=0) goto povtor; fclose(ff);
printf("\nФайл %s закодирован.", argv[1]); exit(0);
uk:if((ff=fopen(argv[1],"r+"))==NULL)
{ printf("\nНе найден файл %s",argv[1]); exit(1); }
handle=open(argv[1], O_CREAT);
size=filelength(handle);
close(handle);
printf("\nРаскодировка файла :%s", argv[1]);
povtor2: printf("\nВведите размер блока в байтах :");
scanf("%d",&sizeblock); printf("\nВведите ключ :");
scanf("%d",&hk); fseek(ff,smesh,0);
fread(ptrsymb,1,sizeblock,ff);
if(*argv[3]=='L') goto ukr;
for(j=0;j<hk;j++)
{ str1=symb[0]&0x80; symb[0]=symb[0]<<1;
for(i=1;i<sizeblock;i++)
{ str2=symb[i]&0x80; symb[i]=symb[i]<<1;
symb[i−1]=symb[i−1]|(str2>>7);}
symb[sizeblock−1]=symb[sizeblock−1]|(str1>>7); }
fseek(ff,smesh,0); fwrite(ptrsymb,1,sizeblock,ff);
smesh=smesh+sizeblock; size=size−sizeblock;
if (size!=0) goto povtor2; fclose(ff);
printf("\nФайл %s раскодирован.", argv[1]); exit(0);
ukr: for(j=0;j<hk;j++)
{ str1=symb[0]&1; symb[0]=symb[0]>>1;
for(i=1;i<sizeblock;i++)
{ str2=symb[i]&1; symb[i]=symb[i]>>1;
symb[i]=symb[i]|(str1<<7); str1=str2;}
symb[0]=symb[0]|(str1<<7); }
fseek(ff,smesh,0);
fwrite(ptrsymb,1,sizeblock,ff);
smesh=smesh+sizeblock;
size=size−sizeblock;
if (size!=0) goto povtor2; fclose(ff);
printf("\nФайл %s раскодирован.", argv[1]);
exit(0);}
```

Пример реализации диверсификации и репликации информации

Файл на передающей стороне (исходное информационное сообщение):

СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Файл на принимающей стороне (в канале связи) для несанкционированного пользователя при различных вариантах кодирования:

1)2F>:F:□ B-□ f»J|B»6R:BH@KDD□G1□FTLLF+ KDGF-□¦P□--¦□q- □CT2RqЁ;

2)21Ёt1¦i□<□¦DФ-D6R:B2¦□Z»B:HB¦BLLF+ KDGF-HGA@F¦□□q □$tMФЬ|;

……;

i)□ °ш□ ш□ Ё□ f□ Sd□□ CT□ P□ T□□□ГИ# ad'#bев#гдЙ□ ш(□++шq-□ CT2RqЁ□;

…… .

Файл на принимающей стороне для санкционированного пользователя:

СПОСОБ ЗАЩИТЫ ИНФОРМАЦИИ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА

Число вариантов данного файла для несанкционированного пользователя превышает 12000 и растет с ростом объема файла, что представляется трудноразрешимой задачей за разумное время.

***

Использование приведенной выше программы обеспечивает идентичность технических средств на передающей и принимающей сторонах канала связи и оперативную репликацию информации санкционированным пользователем.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Эйдельмант И. Б. Кодированный издательский оригинал для автоматического набора.— М. : Машиностроение, 1965.

2. Берлин А. С. Системы программирования набора.— М. : Сов. радио, 1971.

3. Сяо Д., Керр Д., Мэдник С. Защита ЭВМ.— М. : Мир, 1982.

4. Пат. 2022346 России, G06F 13 / 00, G09C 1/00. Устройство защиты информации / В. А. Скрипко, В. М. Шувариков.— 1994.— Бюл. № 20.

5. Пат. 2099890 России, H04L 9/00. Способ шифрования двоичной информации и устройство для осуществления способа "Албер" / Б. В. Березин.— 1997.— Бюл. № 35.

6. Трой Д. Программирование на языке Си для персонального компьютера IBM PC.— М. : Радио и связь, 1991.

Одобрено к печати Ученым советом Одесского государственного политехнического университета

Редакция: *П. Ф. Маев, Л. М. Лейдерман, Е. А. Тихонова.*
Техническая редакция: *Е. И. Корецкая.* Компьютерное обеспечение: *А. П. Соломяный.*
Подписано к печати 05.07.99 г. Формат 60×84 1/8. Печать офсетная. Печ. л. 6,0. Уч.-изд. л. 7,0. Тираж 700 экз.
Оригинал-макет изготовлен в ДП «Нептун-Технология» (65028, г. Одесса, ул. Б. Хмельницкого, 59).
Отпечатано в типографии издательства «Астропринт» (65100, г. Одесса, ул. Преображенская, 24, к. 13).