УДК 681.3.06

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ МУЛЬТИМЕДИА ФАЙЛОВ СКРЫТНО ХРАНИТЬ И ПЕРЕДАВАТЬ ИНФОРМАЦИЮ

АҚПАРАТТЫ ЖІБЕРУ ЖӘНЕ МУЛЬТИМЕДИА ФАЙЛДАРЫН ЖАСЫРЫН САҚТАУ МҮМКІНДІКТЕРІН ЗЕРТТЕУ

RESEARCH OF POSSIBILITIES OF MULTIMEDIA FILES SECRETLY STORE AND TRANSMIT INFORMATION

Б.А. КАЛИЕВ
B.A. KALIYEV

(Алматинский технологический университет)
(Алматы технологиялық университеті)
(Almaty Technological University)
E-mail: bakhyt7@yahoo.com

Исследуются неизвестные ранее свойства мультимедиа файлов, дающие новые возможности при проектировании информационных систем и систем информационной безопасности. Байты информационного файла встраиваются в байты мультимедиа файлов. Также биты, составляющие байты информационного файла или сообщения, встраиваются в байты мультимедиа файлов. Качество воспроизведения мультимедиа файлов остается хорошее. Сокрытый информационный файл восстановлен из мультимедиа файла. Технология позволяет скрытно передавать и хранить информационные сообщения и файлы.

Бұрыннан белгісіз ақпараттық қауіпсіздік жүйесінің және ақпараттық жүйелерді жобалаудың жаңа мүмкіндіктерін беретін мультимедиалық файлдардың қасиеттерін зерттеу. Ақпараттық файлдың байттары файлдың мультимедиасының байттарына енеді. Сондай-ақ, биттер, ақпараттық файлдың немесе хабарламаның жасаушы байттары, файлдың мультимедиасының байттарына енеді. Файлдың мультимедиасының бейнелену сапасы жоғары деңгейде болады. Жасырынды ақпараттық файл мультимедиа файлынан қалпына келтірілген. Бұл технология ақпараттық хаттамалармен файлдарды жасырынды тарату және сақтауға мүмкіндік туғызады.

Studied and previously unknown properties of multimedia files, giving new possibilities for the design of information systems and information security systems..The bytes of the information file embedded in bytes of multimedia files. Also, the bits that make up the bytes of the information file or message is embedded in bytes of multimedia files. Quality playback of multimedia files remains good. Hidden information file are restored from a media file. Technology allows you to secretly transfer and store information messages and files.

Ключевые слова: сокрытие информации, побитовое встраивание, канал проникновения, файл носитель, защита информации.

Негізгі сөздер: ақпаратты жасыру, жеке биттер арқылы кірістіру, ену арнасы, тасушы файл, ақпараттың қорғанысы.

Key words: concealment of information, bitwise inlining, channel penetration, file media, protection of information.

Введение

Мультимедиа файлы способны хранить и переносить сокрытую в них информацию. Автор данной статьи разработал несколько программ, которые позволяют встраивать информацию в байты мультимедиа файлов, а также восстанавливать информацию, сокрытую в мультимедиа файлах. При воспроизведении качество мультимедиа файлов ухудшилось лишь незначительно. Способов обнаружить сокрытую информацию в данных файлах не существует. Можно выполнить сокрытие важной текстовой информации, файла с конфиденциальной информацией или компонентов программного кода.

“Предназначение криптографии – защитить или сохранить в тайне необходимую информацию. Криптография дает средства для защиты информации, поэтому она является частью деятельности по обеспечению безопасности информации” [1].

Объекты и методы исследований

Объектом исследования является информация и физический ее носитель - файл. Методом исследования является технология программного встраивание информационных битов в байты файла носителя. Для программной реализации метода выбрана система “Borland C++ Builder 6” [2].

Исследуем сначала вопрос о сокрытии информационного сообщения в файле мультимедиа.

Для вычислительного эксперимента выберем строку сообщения “Hello world!” или “Привет миру!”. Файлом носителем выберем файл формата bmp или mp3.

Информационное сообщение обрабатывается побайтно. Каждый байт информационного сообщения разлагается побитно. Затем, выполнено программное встраивание в байты файла носителя битов информационной строки. В частности, проведено встраивание в четвертый бит байта файла носителя. В файле носителе первые 3000 и последние 3000 байтов оставим без модификации. Программа вычисляет интервал между модифицируемыми байтами в файле носителе, данный интервал может составить сотни и тысячи байтов. Модифициро-ванный файл носитель 1.bmp или 1.mp3 был воспроизведен. Отметим, что качество мульти-медиа файлов ухудшилось незначительно. Далее, выполнено программное восстановление информационной строки из файла носителя. Информация восстановлена без искажения.

Информационное сообщение удалось скрыть в файле мультимедиа.

Далее, исследуем вопрос о сокрытии информационного файла в mp3 альбоме.

Для вычислительного эксперимента выберем файл 1.xls, содержащий таблицу Microsoft Excel. Файлами носителями выберем mp3 альбом из 9 треков. Имена файлов носителей - Track01.mp3, … , Track09.mp3.

Будем выполнять встраивание в байты файлов носителей биты, составляющие байты исходного файла 1.xls.

В частности, выполнено встраивание в четвертый бит байта файла носителя. В файле носителе первые 3000 и последние 3000 байтов оставляются без модификации. Программа вычисляет интервал между модифицируемыми байтами в файле носителе, как и раньше, данный интервал может составить сотни и тысячи байтов. Модифицированные файлы носители mp3 были воспроизведены. Качество звучания mp3 альбома ухудшилось незначительно. С помощью программы выполнено восстановление исходного файла из файлов альбома mp3. Восстановленный файл дает исходное содержимое – таблицу Microsoft Excel.

Итак, удалось выполнить сокрытие информационного файла в файлах альбома mp3. Затем, автор выполнил сокрытие и последующее восстановление файла динамической библиотеки компоновки DLL размером около 49 килобайт.

Наконец, рассмотрим вопрос о возможности сокрытия файла в файле носителе, при котором будем вставлять уже байты информационного файла в байты файла носителя.

Для вычислительного эксперимента, также выберем файл 1.xls. В качестве файла носителя выберем файл формата bmp или mp3.

В случае, если сокрытие выполняется для информационного сообщения, то файлом носи-телем можно взять также файл формата jpg.

Выполнено программное встраивание байтов исходного файла в файл носитель. В файле носителе первые 3000 и последние 3000 байтов оставляются без модификации. Программа вычисляет интервал между модифицируемыми байтами в файле носителе, как и раньше, данный интервал может составить сотни и тысячи байтов. Модифицированный файл носитель 1.bmp, 1.mp3 или 1.jpg был воспроизведен. Качество воспроизведения также вполне удовлетворительное. Затем выполнено программное восстановление исходного файла из файла носителя.

Программа успешно выполнила сокрытие информационного файла в файле носителе.

Метод, при котором байты информационного файла вставляются в байты мультимедиа файла, наиболее просто реализуем, однако, наиболее уязвим при анализе на взлом.

Приведем исходный код функции сокрытия файла в файле носителе.

Листинг

```
// функция сокрытия файла пользователя в несущем файле void f(void)
{
int iFileHandle1; // дескриптор файла для сокрытия информации
int iFileHandle2; // дескриптор информа-ционного файла
int iFileLength1; // размер файла1 (несущего) в байтах
int iFileLength2; // размер файла2 (скрываемого) в байтах

// буфер для чтения байтов из файла1
char *pszBuffer1;
// буфер для чтения байтов из файла2
char *pszBuffer2;

int i,j=3000;
// в файле1 первые 3000 и последние 3000 байтов не трогаем
int k; // интервал в байтах для вставки информационных байтов в файл1

// открываем файл, получим дескриптор открываемого файла
// файл1 открываем для чтения и записи (несущие файлы: 1.bmp, 1.mp3)
// файл2 открываем для чтения (скрываемый файл 1.xls)
iFileHandle1=FileOpen("1.bmp",fmOpen ReadWrite);
iFileHandle2=FileOpen("1.xls",fmOpen Read);

// установить указатель на конец файла1
iFileLength1 = FileSeek(iFileHandle1,0,2);
// возвращает размер файла1 в байтах

pszBuffer1 = new char [iFileLength1+1];
// создать динамически новый символьный массив размера iFileLength1+1

// установим указатель на 0 байт в файле1
FileSeek(iFileHandle1,0,0);

iFileLength2 = FileSeek(iFileHandle2,0,2);
// возвращает размер файла2 в байтах

pszBuffer2 = new char [iFileLength2+1];
// создать динамически новый символьный массив размера iFileLength2+1

// установим указатель на 0 байт в файле2
FileSeek(iFileHandle2,0,0);

k=(iFileLength1 - 6000)/iFileLength2;
// интервал между модифицируемыми байтами в файле1
// в несущем файле

FileRead(iFileHandle1,pszBuffer1, iFileLength1);
FileRead(iFileHandle2,pszBuffer2, iFileLength2);
for (i=0; i<iFileLength2; i++)
{
  pszBuffer1[j+k*i]=pszBuffer2[i];
}

// установим указатель на 0 байт в файле1
FileSeek(iFileHandle1,0,0);

// перезаписать файл1
FileWrite(iFileHandle1,pszBuffer1, iFileLength1);

FileClose(iFileHandle1); // закрыть дескриптор файла1
FileClose(iFileHandle2); // закрыть дескриптор файла2

// удалим динамически созданные символьные массивы
delete [] pszBuffer1;
delete [] pszBuffer2;

MessageDlg("Файл сокрыт: ",mtConfirmation, TMsgDlgButtons() << mbOK,0);
```

Далее, идет описание кода для восстановления файла пользователя сокрытого в несущем файле.

```
// открываем файл, получим дескриптор открываемого файла
// файл1 открываем для чтения (несущие файлы: 1.bmp, 1.mp3)
iFileHandle1=FileOpen("1.bmp",fmOpen Read);

// получим дескриптор восстанавливаемого файла 2.xls
if (FileExists("2.xls"));
```

```
    DeleteFile("2.xls");
  iFileHandle2=FileCreate("2.xls");

  // установим указатель на 0 байт в файле1
  FileSeek(iFileHandle1,0,0);

  pszBuffer1 = new char [iFileLength1+1];
  // создать динамически новый символьный массив размера iFileLength1+1
  pszBuffer2 = new char [iFileLength2+1];
  // создать динамически новый символьный массив размера iFileLength2+1

  FileRead(iFileHandle1,pszBuffer1, iFileLength1);

  for (i=0; i<iFileLength2; i++)
  {
    pszBuffer2[i]=pszBuffer1[j+k*i];
  }
  // установим указатель на 0 байт в файле2
  FileSeek(iFileHandle2,0,0);

  // перезаписать файл2
  FileWrite(iFileHandle2,pszBuffer2, iFileLength2);

  FileClose(iFileHandle1); // закрыть дескриптор файла1
  FileClose(iFileHandle2); // закрыть дескриптор файла2

  // удалим динамически созданные символьные массивы
  delete [] pszBuffer1;
  delete [] pszBuffer2;

  MessageDlg("Файл восстановлен: ",mtConfirmation, TMsgDlgButtons() << mbOK,0);
  }
  //---------------------------------------------------
```

Результаты и их обсуждение

Вычислительный эксперимент показал, что побитовое сокрытие байтов информационного сообщения или исходного файла, а также, побайтовое сокрытие байтов исходного файла в файлах мультимедиа сохраняет их хорошее качество при воспроизведении. Последующее восстановление сокрытой информации не приводит к ее искажению.

Разработанная автором технология позволяет прятать важную конфиденциальную информацию или компоненты программного кода в файлах мультимедиа, которые широко распространены в сети Internet.

Заключение, выводы

Технология сокрытия информации путем побитового или побайтового встраивания ее в файлы носители mp3 и bmp формата дает новые возможности для реализации систем защиты информации. Кроме того, администраторам информационной безопасности теперь необходимо учитывать новый канал проникновения в информационную систему.

СПИСОК ЛИТЕРАТУРЫ

1. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В.. Основы криптографии. – М.: “Гелиос АРВ”, 2002. – 480 с.

2. Архангельский А.Я. Программирование в C++ Builder 6. – М.: “Издательство Бином”, 2003. – 1152 с.