音乐内容动态加密与许可授权系统设计与实现

张 硕 马兆丰 芦效峰 杨义先 钮心忻
(北京邮电大学网络与信息攻防技术教育部重点实验室 北京 100876)
(北京国泰信安科技有限公司 北京 100086)

摘 要 在数字音乐业务蓬勃发展的同时,版权侵权案件屡禁不止。以MP3为主的网络音乐可被随意下载和传播,因此其版权迫切需要得到保护。根据现有的DRM保护技术与方案,设计了一种应用于Windows Mobile终端系统的MP3文件DRM保护方案。该方案不需要分析MP3编解码过程,仅分析MP3文件结构,结合AES加密算法对文件进行结构化加密。用户通过AES解密、许可证策略、数字签名、CA证书等DRM技术,进行实时解密播放。实验验证了该方案具有较高的安全性与实时性特点,加密速度快,可以很好地应用于移动终端音乐版权保护中,达到移动终端数字版权保护的目的。

关键词 数字版权管理,MP3,许可证书,数据加密

Design and Implementation of Music Content Dynamic Encryption and License Authorization System

ZHANG Shuo MA Zhao-feng LU Xiao-feng YANG Yi-xian NIU Xin-xin
(Key Laboratory of Network and Information Attack & Defense Technology of MOE, Beijing University of Posts and Telecommunications, Beijing 100876, China)
(Beijing National Security Science and Technology Co. Ltd, Beijing 100086, China)

Abstract Copyright infringement cases often occur while the digital music business becomes booming. The MP3-based online music is downloaded and disseminated freely, so the copyright of MP3 must be protected urgently. According to the existing DRM protection technologies and solutions, a MP3 audio file DRM protection scheme based on Windows mobile system was designed. The solution doesn't require analysis of the process of MP3 encoding and decoding, only by analyzing the structure of MP3 files, and combining the technology of AES encryption algorithm, encrypts the file in struct. When the content is packaged, it should be uploaded to the service platform and downloaded by the user. The user uses the DRM technology of AES decryption, license strategy, digital signature, CA certificate to finish real-time decryption play. The experiment shows that the scheme has the features of high security and real-time, and encryption is fast, which can be well applied to mobile terminal music copyright protection, achieving the purpose of the digital copyright protection.

Keywords Digital rights management, MP3, License, Data encryption

1 引言

近年来,我国网络音乐市场发展迅速,音乐产品通过互联网、移动通信网等各种有线或者无线方式的传播,形成了数字化的音乐产品制作、传播和消费模式。随着3G时代的到来、手持设备高速接入的实现以及其功能的多样化,人们对数字内容消费的需求大大增加,如在手机上听音乐、观看影视短片等,这给数字作品的创作、传播、消费提供了广阔的前景。而智能手持设备向娱乐方面的侧移,更是刺激了数字内容的发展,拓宽了数字内容的市场,作为实现数字内容价值的权利管理也就成了业界关注的焦点。MP3音乐作为网络音乐中音频媒体文件的代表,采纳了优秀的压缩编码技术,能够以较高的压缩率实现接近CD的音质。然而,MP3音乐的随意下载和传播,威胁着有关MP3音乐公司的生存和音乐创作者的积极性,因此MP3的版权迫切需要得到保护。移动数字版权管理[7,8](Digital Rights Management Mobile, DRM Mobile)的出现,恰好满足了这方面的需求。

数字版权管理[5](DRM)是保护多媒体内容免受未经授权的播放和复制的有效办法,它为内容提供者保护音乐作品免受非法复制和使用提供了一种手段。

到稿日期:2011-01-03 返修日期:2011-03-22 本文受国家973项目(2007CB311203),国家自然科学基金(60803157,90812001),国家242项目(2009A105),国家标准制定计划(20080200-T-339),国家质检公益性科研专项(10-126)资助。

张 硕(1986—),男,硕士生,主要研究方向为移动数字版权管理等,E-mail:zhangshuosp@126.com;马兆丰(1974—),男,博士,讲师,主要研究方向为数字版权管理、数字内容安全、计算机网络安全;芦效峰(1976—),男,博士,讲师,主要研究方向为多媒体编码技术、多媒体水印算法、数字版权管理、网络攻防技术;杨义先(1961—),男,博士,教授,主要研究方向为密码学、计算机网络与信息安全;钮心忻(1963—),女,博士,教授,主要研究方向为数字水印、信息隐藏、隐写分析。

密码技术[4]是信息安全技术领域的主要传统技术之一，现有的数字内容的保护多采用加密的方法来完成。以数据加密和防拷贝为核心的 DRM 技术基本上以密码学理论为基础，将文件加密成密文的密钥系统或公钥系统，通过不断增加密钥的长度来提高加密、解密系统的密级。只有授权用户才能得到解密的密钥，而且密钥是与用户的硬件信息绑定的。加密技术加上硬件绑定技术，防止了非法拷贝，能有效地达到版权保护的目的。

目前，针对多媒体信息的加密方法的研究逐渐增多，其中音频加密是一个研究的热点。Andres Torrubia[14]提出通过参数控制加密数据占原始数据的百分比来产生不同音质的音频密文，但是该方案将音频的编码过程和加密过程分割开来，编码和加密过程需要两次操作音频码流，从而导致这种方法效率不高。Antonio Servetti[15]通过选择加密 MP3 编码过程中 MDCT 系数的高频部分，起到加密原始音频数据、产生采样音频的作用。但是为了解密恢复原始音频，必须在音频的起始处增加额外的文件头，这种做法增加了加密的复杂度，也改变了 MP3 文件的结构。葛龙[10]等人提出了一种与 MP3 音频编码过程相结合的快速音频选择加密方案，即通过统计分析获得对编/解码过程敏感程度高的参数，将其加密。但是这种做法只针对部分数据进行加密，加密的数据量相对较少，尽管具有较高的加密效率，数据的安全性却无法得到保障。纪猛[11]等人提出了一种基于分层的 MP3 加密保护模型，即通过结合加密频谱区间和加密对象尺寸，进行分层的音频加密。但是该方案在加密时，需要对码字层、亚区层、颗粒层进行变换，同时要改变边信息参数顺序，加密过程相对复杂且安全性不高。

以上几种加密方式，是从编解码角度进行分析，采取部分加密方式加密文件，无法提供较高的安全性，同时不支持 DRM 的实时解密播放功能，无法有效地应用到 DRM 保护中。本文根据已有的 DRM 保护方案，结合 OMA DRM 2.0[12]，采取结构化数据加密方式，提出了一种基于移动终端 Windows Mobile 系统上的 MP3 音频文件 DRM 保护方案。本方案通过分析 MP3 每一帧的结构，结合 MP3 帧结构特点，对文件进行结构化加密。这种加密方式具有较高的安全性，不需要考虑 MP3 编码过程，易于理解和运用，加密速度快，同时可以在播放时实现实时解密播放，从而达到保护 MP3 版权的目的。

2 基于 MP3 动态加密与许可授权的版权保护方案

本文提出的 MP3 版权保护新方案，主要包括以下几个功能模块：内容加密打包系统、密钥管理系统、安全引擎、许可证管理与分发系统、DRM 播放器等功能单元。系统功能架构如图 1 所示。

内容加密打包系统使用密钥管理系统提供的内容加密密钥，对文件内容进行加密与打包，然后将受 DRM 保护的内容传送到业务平台，供用户下载使用。

密钥管理系统负责管理 DRM 系统中使用的各种密钥，并维护 MP3 文件与内容主控密钥的映射关系。

安全引擎系统提供各种加解密算法，并进行内容主控密钥加密、数字签名、身份验证等加解密运算。

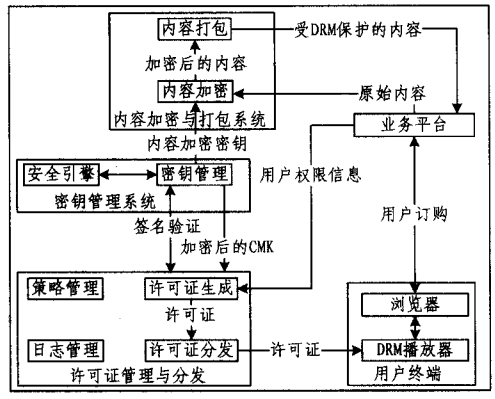

图 1 系统功能架构图

许可证管理与分发系统负责向用户提供许可证的生成、下载和管理等服务，包括许可证生成、许可证分发、许可证策略管理、许可证日志管理等模块。

2.1 内容加密与打包系统

2.1.1 内容的加密打包

MP3[1]文件是由帧(frame)构成的，帧是 MP3 文件最小的组成单位。MP3 文件大体分为 3 部分：TAG_V2(ID3V2)，Frame，TAG_V1(ID3V1)。ID3V1 记录在 MP3 文件的末尾，ID3V2 记录在 MP3 文件的首部。Frame 是 MP3 中最重要的结构，个数由文件大小和帧长决定。

Step 1 计算 ID3V2 长度。ID3V2 由标签头和标签帧组成。

(1)标签头

程序定义一个文件指针 FILE * fp，使用二进制读写方式，fp=fopen(filename,"r+b")，打开 MP3 文件，通过 fseek(fp,0L,SEEK_SET)，使文件指针指向 MP3 文件起始位置。在文件的首部顺序记录 10 个字节的 ID3V2.3 的头部。数据结构如下：

```
struct TagHeader
{
  char Header[3]; /*必须为"ID3",否则认为标签不存在*/
  char Ver; /*版本号 ID3V2.3 就记录 3*/
  char Revision; /*副版本号此版本记录为 0*/
  char Flag; /*存放标志的字节*/
  char Size[4]; /*标签大小,所有的标签帧的大小,不包括 10 字节标签头*/
};
```

如图 2 标记所示，文件首部 10 个字节，49 44 33 03 00 00 00 00 1F 76 组成标签头。其中 49 44 33("ID3")代表标签存在。03 代表 ID3V2 第三版，即 ID3V2.3。00 00 1F 76 4 个字节记录了 ID3V2 标签的长度，通过式(1)，可以计算出标签的长度。

$$Total\_size=(Size[0]\&0x7F)*0x200000+(Size[1]\&0x7F)*0x400+(Size[2]\&0x7F)*0x80+(Size[3]\&0x7F) \quad (1)$$

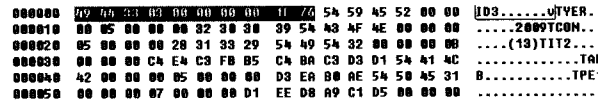

图 2 文件首部 10 字节

(2)标签帧

每个标签帧都由一个 10 字节的帧头和至少一个字节的不固定长度的内容组成。它们也是顺序存放在文件中，标签头和其他的标签帧也没有特殊的字符分隔。得到一个完整的帧的内容只有从帧头中得到内容大小后才能读出，读取时要注意大小，不要将其他帧的内容或帧头读入。帧头的定义如下：

```
struct TagframeHeader
{
char FrameID[4]; /*用四个字符标识一个帧，说明其内容*/
char Size[4]; /*帧内容的大小，不包括帧头，不得小于1*/
char Flags[2]; /*存放标志，只定义了6位*/
};
```

如图 2 所示，标签头后的 10 字节，54 59 45 52 00 00 00 05 00 00 组成一个标签帧帧头。其中 54 59 45 52("TYER")代表帧标识，00 00 00 05 4 个字节记录了标签帧的大小(不包括帧头)，通过式(2)，可以计算出标签帧长。

$$FSize=Size[0]*0x100000000+Size[1]*0x10000+Size[2]*0x100+Size[3] \quad (2)$$

Step 2　MP3 数据帧内容加密。

(1)MP3 帧格式

每个 MP3 数据帧里都有一个帧头，长度是 4BYTE(32bit)，帧头后面可能有两个字节的 CRC 校验，这两个字节是否存在，决定于 FRAMEHEADER 信息的第 16bit，为 0 则帧头后面无校验，为 1 则有校验。校验值长度为 2 个字节，紧跟在 FRAMEHEADER 后面，接着就是帧的实体数据，格式见表 1。

表 1　MP3 的 FRAME 格式

帧头	附加信息	声音数据
4 BYTE	32 BYTE	长度由帧头计算

与标签头一样，每一帧数据帧帧头也存储了本帧的一些重要信息。数据帧头长 4 字节，对于固定位率的 MP3 文件，所有帧的帧头格式一样，其数据结构如下：

```
struct FrameHeader
{
  unsigned int syn; /*同步信息*/
  unsigned int version; /*版本*/
  unsigned int layer; /*层*/
  unsigned int protection; /*CRC校验*/
  unsigned int bitrate; /*位率*/
  unsigned int frequency; /*频率*/
  unsigned int padding; /*帧长调节*/
  unsigned int private; /*保留字*/
  unsigned int mode; /*声道模式*/
  unsigned int mode extension; /*模式*/
  unsigned int copyright; /*版权*/
  unsigned int original; /*原版标志*/
  unsigned int emphasis; /*强调模式*/
} HEADER;
```

帧头 4 字节使用说明见表 2。

表 2　MP3 帧头说明

名称	长度(位)		说明
同步信息	11	第1、2字节	所有位均为 1，第 1 字节恒为 FF
版本	2		00-MPEG 2.5，01-未定义，10-MPEG 2，11-MPEG 1
层	2		00-未定义，01-Layer 3，10-Layer 2，11-Layer 1
CRC 校验	1		0-校验 1-不校验
位率	4	第3字节	取样率，单位是 kbps
频率	2		采样频率，对于 MPEG-1：00-44.1kHz，01-48kHz，10-32kHz，11-未定义
帧长调节	1		调整文件头长度，0-无需调整，1-调整
保留字	1		没有使用
声道模式	2	第4字节	声道，00-立体声，01-Joint Stereo，10-双声道，11-单声道
扩充模式	2		当声道模式为 01 时才使用
版权	1		文件合法性，0-不合法 1-合法
原版标志	1		是否原版，0-非原版，1-原版
强调方式	2		用于声音经降噪压缩后再补偿的分类。00-未定义，01-50/15ms，10-保留 11-CCITT J.17

以帧头 FF FB B0 04 为例，转换成二进制为 11111111 11111011 10110000 00000100。参见表 2，可以得出以下信息。MpegVersion＝MPEG 1，Layer＝3，Bitrate＝192kbps，SamplingRate＝44.1kHz，PaddingBit＝0。通过这些信息，能够计算出一帧帧长。MP3 帧长取决于位率和频率，计算公式为

$$FrameSize=(((MpegVersion == MPEG1\ ?\ 144:72)*Bitrate)/SamplingRate)+PaddingBit \quad (3)$$

式中，MpegVersion 为 MPEG 版本。若 MpegVersion 为 MPEG1，其值为 144，否则其值为 72。Bitrate 为该帧的比特率，SamplingRate 为采样率，PaddingBit 为填充位。

(2)MP3 数据内容加密

帧头后面是可变长度的附加信息，对于标准的 MP3 文件来说，其长度是 32 字节，紧随其后的是压缩的声音数据，可以通过文件指针，跳过 32 字节的附加信息，直接指向数据信息。数据加密采用 AES(128 位)对称加密算法，具有高加密强度、低运算开销的优势。为了使加密密钥更安全，内容加密和打包系统随机生成辅助因子 AK(16 字节)，同时向密钥管理系统申请内容主控密钥 CMK。内容加密和打包系统在获取 CMK 之后，调用安全引擎模块，使用 AES 算法通过 CMK(16 字节)对 AK 进行加密，生成内容加密密钥 CEK(16 字节)。内容加密和打包系统使用 CEK，对每一帧的数据内容进行 AES 加密。若数据长度不够 16 字节，不进行加密。加密程序主要代码如下：

```
BlockSize=FrameSize-4-32; /*计算一帧中，数据内容大小*/
while(BlockSize > 0){
if(BlockSize <256) {/*把数据块分为256字节一组*/
fread(en_array,1,BlockSize,fp); /*读取BlockSize数据，放入数组中*/
fseek(fp,0L,SEEK_CUR); /*定位文件指针位置*/
Encrypt1(cek,en_array,BlockSize); /*调用加密函数，对数据加密*/
fwrite(en_array,1,BlockSize,fp1); /*在新文件中写入加密后的数据*/
fseek(fp1,0L,SEEK_CUR); /*定位文件指针位置*/
Clean(en_array); /*清理数组*/
BlockSize=0;} /*此帧结束*/
```

```
else{
fread(en_array,1,256,fp);/*读取256字节数据,放入数组中*/
fseek(fp,0L,SEEK_CUR);/*定位文件指针位置*/
Encrypt(cek,en_array);/*调用加密函数,对数据加密*/
fwrite(en_array,1,256,fp1);/*在新文件中写入加密后的数据*/
fseek(fp1,0L,SEEK_CUR);/*定位文件指针位置*/
Clean(en_array);/*清理数组*/
BlockSize=BlockSize-256;
}}
```

程序通过式(3),计算一帧数据内容 BlockSize 的大小。当 BlockSize 长度大于 0 时,执行循环体中的加密函数。程序将 BlockSize 按每 256 字节分为一个组处理,当剩余数据内容大于等于 256 字节时,每次读入 256 字节数据(一种处理方式),通过调用 Encrypt()加密函数进行加密;当剩余数据内容小于 256 字节时,一次性读入剩余内容,进行加密。

2.1.2 内容的打包

为了使播放器能够自动识别当前播放文件是否受版权保护,MP3 文件完成加密后,需要对文件做一些额外的处理。本文采取引入媒体头的方式,媒体头记录包括辅助因子 AK、媒体内容 ID、许可证服务器地址、加/解密算法信息等。播放器播放文件前,先识别媒体头,就能区分当前播放文件是否受版权保护。媒体头数据结构定义如下:

```
struct CPSECDRMMediaHeaders
{
  char Tag[4];/*记录为"DRM3",表示受DRM保护的MP3文件*/
  char ContentID[16];/*内容ID*/
  char RightsIssuerURL[128];/*许可证服务器URL*/
  char KeyFactor[16];/*辅助因子AK*/
  int EncryptionMethod;/*内容加密算法标识*/
    ..................
}* PCPSECDRMMediaHeaders
```

本文设计时,把媒体头存储在 MP3 文件开始位置。这样设计的好处在于,一方面能够记录文件打包时的信息,另一方面播放器播放文件前,能够自行识别媒体头,选择相应的播放流程。打包完成后,系统需要对加密后的文件内容进行 Hash 计算,将输出的 Hash 值发送给密钥管理系统。许可证服务器生成许可证时将相应的 Hash 值打包到许可证中,以便播放器在解析许可证时验证音乐内容的完整性。整个 MP3 加密过程如图 3 所示。

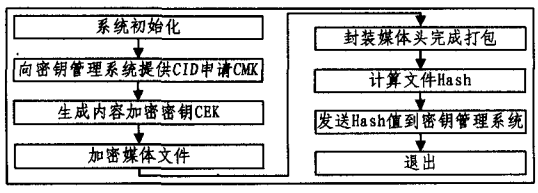

图 3 内容打包子系统工作流程图

2.1.3 加密校验

解密加密后的音乐文件,计算解密后数据的 Hash 值,与原始文件的 Hash 值做比较,这样可以校验原始文件是否被正确加密。若两个 Hash 值不相等,则重新进行加密步骤。

2.2 密钥管理

密钥管理系统的功能包括内容主控密钥的生成和管理、用户签名信息的验证、许可证签名、加密后内容 Hash 值的存储、管理和密钥分发、存储等。

2.2.1 内容主控密钥生成

在系统初始化阶段,密钥管理系统调用安全引擎系统生成服务系统公私钥,并将用户公钥发送到 CA 系统请求 CA 系统签发公钥证书。

在接收到内容加密系统的内容密钥生成请求后,调用安全引擎生成内容主控密钥,并返回给内容加密系统。

2.2.2 签名与验证

签名功能用于实现许可证权利信息的签名,验证功能用于服务系统与客户端信息交互过程中用户身份的验证。密钥管理服务器对许可证权利信息签名时使用服务系统的私钥,验证用户身份时使用用户的公钥。

2.2.3 密钥分发

内容主控密钥使用用户公钥进行加密,封装在许可证中分发到用户手中。

2.2.4 密钥存储

对内容主控密钥、系统端私钥进行加密存储,并采用相应的措施保证密钥的完整性。

2.3 许可证的管理与分发

许可证管理与分发系统负责向用户提供许可证的生成、下载和管理等服务,包括许可证生成、许可证分发、许可证策略管理等模块。许可证生成模块接收用户下载许可证时上传的参数信息,根据权利信息为用户创建许可证,同时向密钥管理系统申请相应的内容主控密钥,以密文形式封装到许可证中;许可证分发模块实现许可证的分发和下载功能;许可证策略管理模块实现许可证策略的管理功能。

2.3.1 许可证策略

许可证策略是基于数字权限描述语言实现的。根据用户需要可以按照许可证策略灵活定制各种许可证模板。一般应用软件对用户的操作限制包括:时间限制(许可证的生效日期和失效日期)、次数限制、功能限制、许可证与用户使用的硬件环境绑定等[7]。考虑到以上情况,本文提出几种比较实用的许可证:(1)试用许可证。提供给用户一定时间或次数的使用;(2)功能许可证。授权用户可以使用应用程序中相应的功能模块;(3)单机用户许可证。授权的用户只能在特定单一的机器上使用,不能转发。

2.3.2 许可证生成

许可证服务器负责根据从业务平台接收的订购信息,包括用户标识、内容标识以及权限信息等,生成许可证提取号,并返回给业务平台,再发送给用户。

许可证服务器收到许可证下载客户端的许可证提取请求,包含许可证提取号、设备绑定信息、用户公钥证书及用户签名信息,将用户公钥证书及用户签名信息发送到密钥管理服务器请求验证用户身份。验证通过后,根据许可证提取号查找到内容标识,然后将内容标识发送给密钥管理服务器。获得密钥管理服务器返回的、经过加密的 CMK 和内容 Hash 值后,基于许可证权利描述语言生成待签名的许可证,再将其发送给密钥管理服务器请求签名。在得到密钥管理服务器返回的签名信息后,将签名封装在许可证中,生成最终的许可证文件[10,14]。许可证生成流程如图 4 所示。

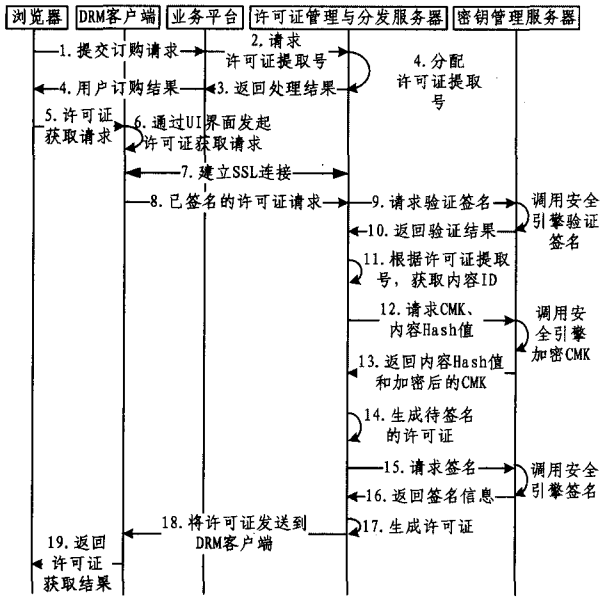

图 4　许可证生成流程

2.3.3　许可证分发

许可证分发模块接收和响应许可证下载客户端的许可证下载请求，并将相应的许可证分发到下载客户端。

2.4　MP3 内容解密与播放

为了使播放器能够支持播放受 DRM 保护的 MP3 文件，播放器需要加入文件媒体头识别、许可证查找、许可证解析、媒体解密播放等多个功能[2,3]。

2.4.1　播放控制器的数据处理流程

(1)用户使用播放器选择 MP3 文件播放，播放器通过文件媒体头自动识别当前播放文件是否受 DRM 保护。若文件不受 DRM 保护，则执行正常播放流程，否则转向步骤(2)。

(2)播放器搜索本地许可证，若发现当前播放文件的许可证存在，则转向步骤(3)。否则提示用户，播放许可证不存在，需要用户点击链接提示，登录业务平台获取许可证提取号，进行许可证下载。播放器停止播放功能，返回初始界面。

(3)解析许可证，包括文件播放次数、许可证有效期、加密内容主控密钥 CMK'、文件打包时的内容 Hash 值等。利用数字签名技术，结合许可证用户公钥证书，验证许可证完的整性，保证许可证未被篡改，合法可用。若许可证不合法，则自动删除非法许可证，提示用户重新下载许可证。

(4)播放器从许可证证书中读取用户私钥，解密许可证中的 CMK'，恢复原始的内容主控密钥 CMK，然后和媒体头中记录的辅助因子 AK 生成内容解密密钥 CEK，即 CEK＝AES(AK,CMK)。

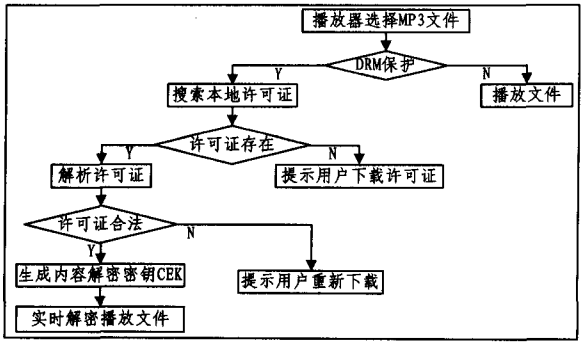

图 5　播放控制器的数据处理流程图

(5)播放器使用 CEK，完成对加密数据的实时解密与播放。

播放控制器的数据处理流程如图 5 所示。

2.4.2　MP3 解密播放过程

(1)验证内容 Hash 值

播放器首先对选中播放的 MP3 音乐文件进行 Hash 计算，计算结果与许可证中存储的 Hash 值进行比对。若 Hash 值相等，则进行下一步，否则提示用户该音乐文件已经被非法修改，不允许播放。

(2)识别媒体头

播放器首先读取文件开始的 10 字节，存入 IDtag，判断是否为 DRM 保护文件。若 IDtag 数组前 4 个元素为字符“DRM3”，则当前播放文件受 DRM 保护，设置标记变量 Signal＝1；若 IDtag 数组前 3 个元素为字符“ID3”，则当前播放文件不受 DRM 保护，设置标记变量 Signal＝0。

(3)生成内容解密密钥 CEK

播放器从许可证证书中读取用户私钥，结合列表文件 data. xml，解密许可证中的 CMK'，恢复原始的内容主控密钥 CMK，然后和媒体头中记录的辅助因子 AK 生成内容解密密钥 CEK。代码如下：

```
int nRetl = CPSECLICENSE_GetCMK("1", cmk, &cmk,"\\My Documents\\Licence\\data. xml",
"\\My Documents\\ Licence\\server. cer");
CPSECCRYPTO_AESDecrypt_Bytes(cmk,key,cek);
```

(4)解密恢复原始数据，实时播放

根据解码器的解码流程，在播放器播放数据内容前进行解密，恢复 * stream->main_data 中的原始数据。解密函数原型如下：

```
Void CPSECCRYPTO_AESDecrypt(char cek[16], char * * start, char * * end,int frame_lenth)
```

char 型数组 cek 存储内容解密密钥；指针 start 指向解密数据的起始位置；指针 end 指向解密数据的结束位置；变量 frame_lenth 记录解密数据的长度。

3　实验结论与分析

音乐加密过程中，本文使用 PC 机(配置 Pentium(R) Dual-Core 2.6 GHz CPU，1G 内存)分别对不同大小的 MP3 音乐进行结构化加密，实验结果见表 3。对不同大小的 MP3 音乐进行重编码，实验结果见表 4。进行实验结论分析，得出以下结论。

表 3　MP3 结构化加密结果

内容大小(MB)	时长(s)	位速(kbps)	加密时间(s)	加密速度(MB/s)
6.02	262	192	3	2.01
3.98	260	128	2	1.99
6.29	229	256	3	2.09
6.47	283	192	3	2.15
5.40	192	192	3	1.8
11.9	313	320	6	1.98

表 4　MP3 重编码结果

内容大小(MB)	时长(s)	位速(kbps)	编码时间(s)	编码速度(MB/s)
6.02	262	192	66	0.09
3.98	260	128	68	0.06
6.29	229	256	41	0.15
6.47	283	192	49	0.13
5.40	192	192	49	0.11
11.9	313	320	83	0.14

(1)加密方式。本文采用基于结构化的加密方式,通过分析 MP3 每一帧的结构,结合 MP3 帧结构特点,对文件进行结构化加密。这种加密方式不需要考虑 MP3 编解码过程,易于理解和运用。而基于编解码过程的加密方式,如葛龙[10]、纪猛[11]等人提出的算法,需要熟悉整个 MP3 编解码流程。

(2)加密效率。如表 3 所列,本文提出的结构化加密算法的加密速度基本符合线性关系,平均速度为 2MB/s,加密效率比较理想。如表 4 所列,对 MP3 文件仅进行重编码过程,不考虑加密,平均编码速度仅为 0.1MB/s。若再加入加密过程,平均加密速度会低于 0.1MB/s,效率很低。如果只加密数据总量的 10%[10],不加密的数据仍需要经过解码再编码过程,所用时间仍然会大于表格中给出的时间,无法达到本文算法的加密速度 2MB/s,且没有较高的安全性。

(3)可感知性。文献[10,11,14,15]等提出的基于编解码的加密方式,破坏了音频文件的可感知性,但是由于加密数据相对较少(1%～10%),人耳依然能够分辨出加密的音乐内容。而本文提出的加密方式,能够完全破坏可感知性,以致于人耳也无法分辨出加密的音乐内容,能够防止音乐内容的泄露,达到了音乐保护的目的。

(4)感知安全性。图 6 是一个加密前的音频片段,图 7 是同一位置上加密后的音频片段。每个图中,上半部分绿色波形代表左声道,下半部分红色波形代表右声道。通过分析波形振幅可以发现,使用加密技术,能够改变音频数据的幅值,使得播放解码时每一帧中加密码字无法通过解码变换,还原成原始码字,原始音频码流遭到破坏,可感知性大幅降低。播放时人耳只能听到噪声,达到了加密音频信息的目的。

图 6　未加密的一段波形

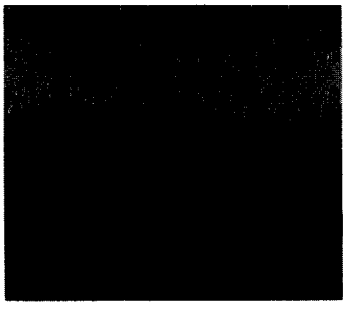

图 7　加密后的一段波形

音乐解密播放过程中,本文采用模拟器仿真和真机测试两种方式。

(1)模拟器仿真:使用 Windows Mobile 6.0 SDK,通过模拟器播放加密音乐。播放过程中,CPU 平均占用率为 35%,可支持快进、回退等实时播放功能,无音质损坏、长时间缓冲等现象出现,播放效果比较理想。

(2)真机测试:真机使用多普达 T3333 智能手机,Windows Mobile 6.5 操作系统、高通 MSM7200A 528MHz 处理器,解密播放过程中可支持快进、回退等实时播放功能,无音质损坏、卡壳、死机等现象出现,播放效果很理想。

通过实验分析,可以得出本文提出的 MP3 结构化加密方式具有较高的安全性与实时性特点,加密速度快,可以很好地应用于移动终端音乐版权保护中,支持实时播放,达到保护移动终端数字版权的目的。

结束语　本文采用数据加密方式,设计了一种基于移动终端 Windows Mobile 系统的 MP3 音频文件 DRM 保护方案。采用 AES 加密算法对 MP3 文件进行结构化加密,既能有效防止未经授权的传播行为,又能支持实时解密播放;采用许可证策略限定 MP3 音乐的播放次数、权限、有效期等,能够有效防止未经授权的播放行为;采用数字签名技术验证许可证的完整性,能够有效防止许可证被篡改、损坏等行为;采用 CA 证书形式存放用户私钥,能够更好地保护密钥的安全。实验结果表明,运用数据加密技术能够实现 MP3 音频文件的 DRM 保护,达到保护数字版权的目的。

参 考 文 献

[1] 汪勇,熊前兴. MP3 文件格式解析[J]. 计算机应用于软件,2004,21(12):126-128

[2] 褚伟杰,商宇. 支持数字版权保护的流媒体播放控制器的设计与实现[J]. 福建电脑,2008(03):160-161

[3] 孔德萍,苑红晓,常立立. 基于 WRM 的流媒体数字版权管理(DRM)系统实现[J]. 信息技术与信息化,2007(01):53-55

[4] 杨义先. 信息安全新技术[M]. 北京:北京邮电大学出版社,2002

[5] 范科峰,莫玮,曹山,等. 数字版权管理技术及应用研究进展[J]. 电子学报,2007,35(6):1139-1147

[6] 吴海华,邢桂芬. 基于多种许可证的数字版权管理技术[J]. 计算机工程与设计,2006,27(19):3679-3681

[7] 李吟虹,毛敏. 面向移动增值业务的数字版权管理方案[J]. 信息技术,2007(10):157-160

[8] 袁琦,闵栋. 移动 DRM 技术的发展和应用分析[J]. 电信网技术,2009:12:28-31

[9] 马兆丰,范科峰,陈铭,等. 支持时空约束的可信数字版权管理安全许可协议[J]. 通信学报,2008,29(10):153-164

[10] 葛龙,廉士国,孙金生,等. 基于参数敏感性的 MP3 音频选择加密方案[J]. 计算机应用研究,2006(10):122-127

[11] 纪猛,张萌. 一种基于分层的 MP3 加密保护模型[J]. 电子器件,2003,26(1):71-74

[12] Open Mobile Alliance™. DRM Architecture Draft Version 2.0.1 [S]. Feb. 2008

[13] 马兆丰,冯博琴. 基于动态许可证的信任版权安全认证协议[J]. 软件学报,2004,15(1):131-140

[14] Torrubia A, Mora F. Perceptual Cryptography on MPEG Layer IlI Bit. streanls[J]. IEEE Transactions on Consumer Electronics, 2002,48(4):1046-1050

[15] Servetti A, Testa C, Marti J C D. Frequency. selective Partial Encryption of Compressed Audio[C]//ICASSP. 2003:668-671