一种面向 Android 平台的多媒体数字版权管理系统

王 真 张志勇 常亚楠
(河南科技大学信息工程学院 洛阳 471023)

摘 要 针对移动终端多媒体音视频数字版权保护问题,设计了一种面向 Android 平台的数字版权保护方案。该方案采用 3DES 加密算法,并将数字许可证与移动终端设备硬件绑定,实现了多媒体安全播放和使用控制,以及终端设备间的数字权利分享。实验结果表明,该方案安全性高、加解密速度快,使用控制功能满足了 Android 平台数字内容版权保护的实际需求。

关键词 数字版权管理,移动多媒体,数字权利分享,Android
中图法分类号 TP309 **文献标识码** A

Multimedia DRM System for Android Platforms

WANG Zhen ZHANG Zhi-yong CHANG Ya-nan
(Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China)

Abstract For digital rights management of mobile multimedia in mobile terminals, a Digital Rights Management (DRM) approach for Android platforms was proposed. The solution adopts 3DES encryption algorithm and binds digital license with hardware of the device to implement usage of control and secure display of multimedia contents. Meanwhile, sharing digital rights between devices was supported in this approach. A prototype confirms that the solution has the features of high security and faster encryption speed. Otherwise, functions of usage of control can meet the practical requirements for digital rights management with Android platforms.

Keywords Digital rights management, Mobile multimedia, Digital rights sharing, Android

1 引言

由于数字内容(包括电子书、数字图像、多媒体音视频等)具备易于无损复制、分发等特性,出现了随意批量复制受知识产权法律保护的有价数字内容的产品,及其通过各类通信网络载体进行非授权分发、传播和滥用的侵权行为和现象,给整个经济社会和文化发展造成严重的不良后果和损失。于是 DRM[1](Digital Right Management)应运而生。它由一系列技术、工具、流程和处理方法组成,主要目的就是保护数字作品内容,维护版权所有者和用户的合法权益[2]。

早期的 DRM 提供商对于数字内容和权利的分发采用很强的控制,现在对分发问题普遍向分发的控制灵活性方面考虑。数字权利分享是 DRM 系统中的一个关键技术,但传统的 DRM 系统主要关注权利在版权所有者和普通用户之间的传递,而较少注意权利在普通用户间的分享[3]。事实上,支持权利的分享在 DRM 技术中具有非常重大的意义。数字版权保护系统支持设备间数字权利分享,可以有效地提高用户购买和使用数字内容的积极性,增加用户对版权保护系统的接受程度,同时也可以极大降低用户破解版权保护系统的动机。

2 相关工作

现有的数字内容保护方法通常采用加密技术,将普通数字内容文件(明文)加密成密文,以防止有价值的信息被非法拦截或窃取,从而达到数字内容版权保护的目的[4]。针对数字内容的版权保护问题,在 PC 机上已经有了比较成熟的方案在广泛使用。比如微软基于 Windows Media Player 开发的 Windows Media DRM[5] 和由 Real 公司提出的主要适用于流媒体直播的 Helix DRM 解决方案。

2.1 移动终端 DRM 系统

Bhatt 等人[6] 在智能手机 Motorola E680i 上提出了一个基于 Peer-to-Peer 模型的个人 DRM 系统,保护内容主要是用户个人文件,如拍摄的照片和视频。原生 Android 平台通过部署 OMA DRM 1.0 方案[7] 来实现数字内容和应用程序的保护,由于 OMA DRM 1.0 方案本身的脆弱性,该方案并不能很好地保护设备上的受保护内容。张硕、马兆丰等人[8] 基于 Windows Mobile 系统提出了一种针对 MP3 文件的动态解密播放方案,该方案基于 MP3 文件结构,对 MP3 文件按每一帧进行加密,因此,在文件播放时可以实现对加密文件的动态

到稿日期:2013-07-26 返修日期:2013-11-04 本文受国家自然科学基金(61003234),河南省科技创新人才计划杰出青年基金(134100510006),河南省教育厅科学技术研究重点项目基础研究计划(13A520240)资助。

王 真(1989—),男,硕士生,CCF 学生会员,主要研究方向为数字版权管理技术、移动终端多媒体;张志勇(1975—),男,博士后,副教授,CCF 高级会员,主要研究方向为数字版权管理、可信计算与访问控制,E-mail:z.zhang@ieee.org;常亚楠(1989—),女,硕士生,CCF 学生会员,主要研究方向为数字版权管理技术。

解密播放，而不会在移动终端上产生任何临时文件。

以上系统和方案在几种常见的智能手机系统上部署了DRM系统，但是在用户使用最多的Android智能手机系统上并没有出现可以有效保护音视频内容的DRM系统。本文基于Android平台并结合OMA DRM V2.0[9]标准，提出了以保护音视频为目的的Mobile DRM system方案。

2.2 数字权利分享

在数字内容(权利)共享的实现机制方面，为了便于内容在不同设备上的共享使用，Digital Video Broadcasting联盟首先提出了“授权域(Authorized Domain)”概念，随后OMA DRM方案也在V2.0之后的版本中使用了这一概念，并实现了RI(Rights Issuer)对域的统一管理，包括创建和撤销域、用户设备的加入与退出域等，域内设备之间可以共享内容和数字权利。

目前DRM数字内容共享研究场景侧重于家庭网络域和个人娱乐域(Personal Entertainment Domain)。李平、凌贺飞等提出利用组密钥技术加密数字内容加密密钥(CEK)，实现了数字内容在家庭网络下的内容分发与共享[10]。文献[11]中提出了一种利用代理重加密的方式来实现数字权利分享，数字权利分享时由第三方代理完成CEK的解密和重新加密。文献[12]采取遍历加密的方式，假设注册用户拥有N个设备，利用这些设备的公钥产生N个加密的CEK同时存放于数字权利证书中，实现了数字权利证书在N个设备中的共享。文献[13]采取的是将权利证书中的权利平均分成M等份并对每一份进行唯一标记，合法用户可以分发M等份中的若干份给其他用户。

以上权利分享方案存在的不足为使用场景受限或系统构建复杂。因此，2.1节提出的Mobile DRM system方案中又引入了权利分享模块，采取将数字权利证书与设备硬件信息绑定的方法实现了任意设备间的分享。

3 基于Android平台音视频版权保护方案

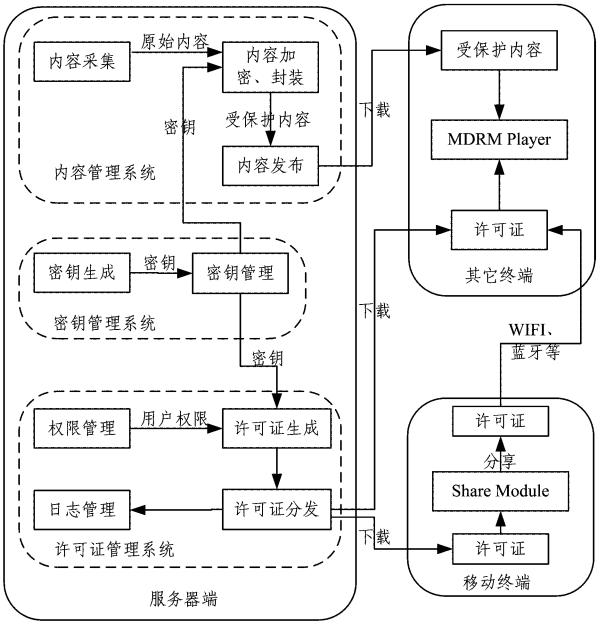

图1 Mobile DRM system系统架构

为了解决移动终端上音视频数字版权保护问题，我们选取目前移动终端市场占有率最高的Android系统为目标平台，并参考OMA DRM v2.0标准，最终实现并部署了我们设计的DRM原型系统Mobile DRM system。该系统不但能够有效保护授权用户对数字内容的合法使用，且能够实现授权用户将自己所拥有的数字权利分享一部分给其他用户或设备。系统架构图如图1所示。

Mobile DRM system系统功能是实现从内容提供商的内容加密封装及发布、许可证的分发，到终端用户的解密及对内容使用控制的全生命周期过程。通过受保护数字内容及其许可证的分离分发与授权使用控制方法，防止内容的滥用和随意分享，达到数字内容的安全使用控制与数字版权保护功能。该模型分为服务器端和移动终端两个部分。对该Mobile DRM system系统而言，主要研究点是Android平台下多媒体音视频内容的解密播放和使用控制，实现一个基于Android平台的多媒体音视频数字版权保护系统。

3.1 数字权利许可证

权利许可证是终端设备上用于控制用户授权播放受保护内容的文件，由服务器端生成并发送。许可证内包含了受保护内容的解密密钥和用户对该数字内容所拥有的权限。本文方案中，许可证直接与设备的唯一识别号码绑定，且许可证中包含有重要数据的MD5值，保证了许可证书在使用过程中不被非法传播和恶意篡改，使内容提供商与授权用户的合法权益得到有效保护。作为原型系统，我们定义用户对受保护内容所拥有的权限是受保护内容可以在用户移动设备上授权播放的次数。数字权利证书采用标准XML语言书写，样例如图2所示。

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<rights>
  <agreement>
    <set>
      <ID>888888888</ID>
    </set>
    <asset>
      <UID>000000000000000</UID>
      <key>60D8A787DF529216F8DEA22022C15594</key>
    </asset>
    <permission>
      <count>55</count>
    </permission>
    <show>
      <MD5>26fedcf200106379fb15469286b8eda0</MD5>
    </show>
  </agreement>
</rights>
```

图2 数字权利证书样例

为了便于理解该权利证书样例，以表1的形式列出证书中的各个元素并分别解释。

表1 权利证书中元素列表

元素名	含义
ID	权利证书的唯一识别号，在权利证书生成时由服务器端许可证管理系统产生
UID	终端设备的唯一识别号，在终端由DRM Agent运算产生，由用户填写保存在服务器端，用户可以拥有并管理多台设备
Key	内容加密密钥CEK加密后的密文，其值为Encrypt(UID,CEK)
Count	数字内容在授权设备上的可播放次数
MD5	数字权利证书中重要数据的MD5值，用于校验权利证书是否被非法篡改，其值为MD5(ID,UID,Key,Count)

3.2 移动终端多媒体使用控制

移动终端(即客户端)通过安装在Android平台上的

MDRM Player 来实现对多媒体音视频的播放功能；受保护内容的鉴别、解密、使用控制功能则由位于系统运行库层的 MDRM Agent 模块处理。图 3 直观地显示了为了在 Android 平台上实现具有 DRM 功能的播放器，我们在 Android 平台上所做的工作。

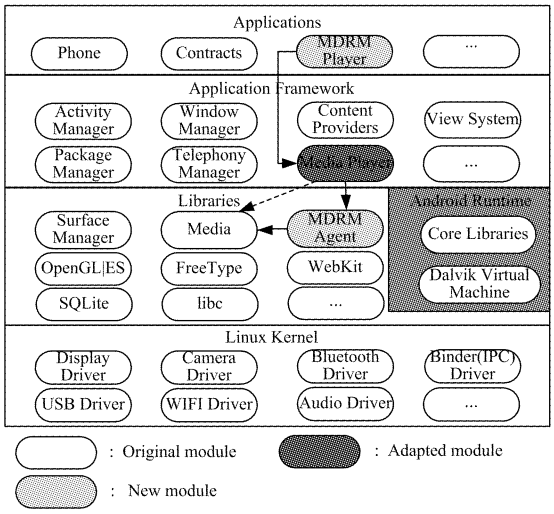

图 3　支持 MDRM 的 Android 系统架构

在 Android 原来的系统中，当上层多媒体应用调用应用程序框架层的 MediaPlayer 类时，MediaPlayer 类直接调用系统运行库的多媒体模块来进行处理。在我们所设计的 Android 平台播放器的系统运行库中添加了 MDRM Agent模块，当用户使用 MDRM Player 播放音视频时，应用程序框架层的 MediaPlayer 类先调用 MDRM Agent 模块，MDRM Agent 模块对 MediaPlayer 类传递过来的参数进行处理后再调用多媒体模块进行操作。详细的参数传递过程如图 4 所示。

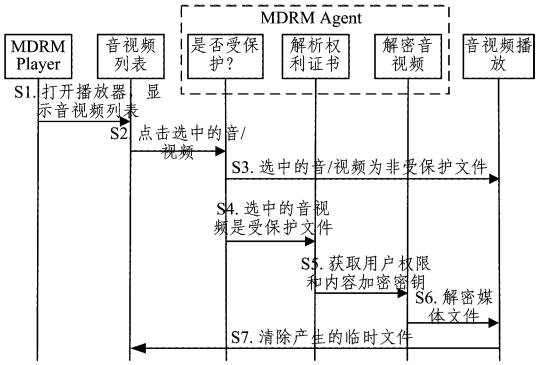

图 4　MDRM Player 处理信息流图

Step S1　用户打开 MDRM Player 应用程序，该应用在开启后会扫描设备的存储空间，找出设备上存储的所有支持格式的音视频文件，扫描完成后将音频与视频分别按照列表的形式呈现给用户。对于一个音视频文件，列表中呈现的信息除了文件名之外，还显示了该文件是否受保护以及用户对于该文件所拥有的权限(播放次数)。

Step S2　用户根据音视频列表显示的信息，选择一个感兴趣的内容进行播放。

Step S3　系统运行库层的 MDRM Agent 模块接到上层框架传递过来的参数后，判断用户所选择的内容是否是受保护的内容。如果不是受保护内容，则直接调用多媒体模块对文件进行解析播放。

Step S4　MDRM Agent 如果判断出用户所选择的内容为受保护内容，则查找设备上该受保护内容的权利证书。

Step S5　找到该受保护内容的权利证书后，MDRM Agent 的证书解析子模块解析对应的权利证书，从中读取出用户对于该受保护内容所拥有的权限和受保护内容的解密密钥。证书解析过程如下：

1)读取证书中的重要数据并对其做 *MD*5 计算，将计算结果与证书中的 *MD*5 元素值进行比较，若一样，进行下一步，否则抛出错误。

2)读取并计算设备的唯一识别号 *UID*，将其与证书中的 *UID* 元素值进行比较，若一样，进行下一步，否则抛出错误。

3)将授权的可播放次数减 1，重新计算重要数据的 *MD*5 值，并将元素 *Count* 和 *MD*5 的值写回到权利证书中。

4)解密元素 *Key* 的值，获得 CEK。

Step S6　MDRM Agent 内容解密子模块根据步骤 S5 解析出的 CEK 解密该媒体文件，并调用多媒体模块播放解密后的临时文件。

Step S7　多媒体模块播放完成之后，清除产生的临时文件并返回步骤 S2。

对于受保护的内容，本方案并不强制用户将其放在设备上的特定区域，也不强行限制受保护内容的来源必须是用户通过自己设备上的浏览器下载得到。用户可以将受保护的内容存储在设备上的任何地方，其来源既可以是用户通过设备下载，也可以是通过 WIFI、USB、蓝牙等传输方式从 PC 机或者其它设备获得，只要能确保受保护内容的完整性没有被破坏即可。但是，为了方便管理设备上受保护内容的权利证书，我们将权利证书的存放位置指定在了设备上的一个固定文件夹。

3.3　设备间的数字权利分享

方案 Mobile DRM system 中，设备对受保护内容所拥有的权限是媒体文件在设备上的可播放次数。因此，在需要时，用户可以选择将设备 A 对受保护内容 C 所拥有的权限分享一部分给设备 B，使设备 B 也能够得到 C 的合法授权。假设设备 A 对内容 C 拥有的可播放次数为 *M*，将要分享给设备 B 的可播放次数为 *N*(*N* 不大于 *M*)，分享过程如图 5 所示。

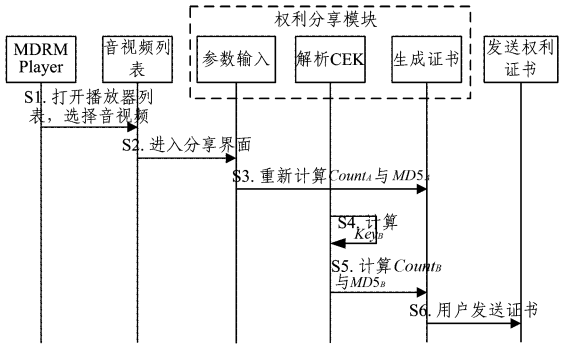

图 5　设备间权利分享信息流图

具体描述如下：

1)选中将要进行权利分享的内容 C，检查设备 A 是否对该内容拥有合法的授权证书，若拥有，进行下一步，否则抛出错误。

2)打开权利分享界面，填写分享次数 *N* 和设备 B 的唯一

识别号 UID_B。

3)元素 $Count_A$ 的值由 M 变为 M 减 N,并重新计算 $MD5_A$ 的值,将它们写回到权利证书中。

4)读取 Key_A 的值,利用 UID_B 和 UID_A 计算 $Key_B = E(UID_B, D(UID_A, Key_A))$。

5)设置 $Count_B$ 的值为 N,利用与设备 B 相关的元素值计算 $MD5_B$,并生成新的数字权利证书到指定的目录。

6)将新生成的权利证书发送到设备 B,分享完成。

将在指定目录下生成的新的权利证书发送到设备 B 后,设备 B 即可拥有内容 C 的合法授权,其授权的播放次数为 N。

4 原型系统与性能分析

系统搭建完成后,测试显示该原型系统可支持 MP3、MP4、3GP 这 3 种格式文件的正常播放;设备导入有效的授权证书后,加密过的以上 3 种格式文件也可正常播放。为了测试说明系统性能,分别对 Mobile DRM system 的设备间权利分享和内容加解密速度这两个方面做了测试与分析。

4.1 权利分享测试

权利分享测试时,由一台操作系统为 Android4.04 的中兴手机作为设备 A,向作为设备 B 的 PC 机上运行的一台模拟器分享内容 C 的授权播放次数。分享时打开设备 A 上的播放器 MDRM Player,在列表中长按将要分享权限的某个媒体文件 C,出现如图 6 所示的信息详情界面,点击“分享”按钮后进入如图 7 所示的权利分享界面。所要输入的两个参数分别为被分享次数和设备 B 的唯一识别号 UID_B,最终在设备 A 的指定目录下生成了一个唯一适用于设备 B 解密播放媒体文件 C 的数字权利证书。将该数字权利证书导入设备 B 的指定目录,设备 B 能够正常解密播放受保护内容 C。

图 6 内容详细信息界面

图 7 权利分享界面

4.2 加解密速度测试

加密打包程序的测试环境是一台 CPU 主频 3.4G、内存 4G 的普通 PC 机;MDRM Player 程序的测试环境是建立在 PC 机上的一台 UBUNTU 系统虚拟机运行 Android 模拟器,其中 UBUNTU 虚拟机内存设置为 1G。分别将 block_size(以字节为单位)的大小设置为 1024000、102400、10240 来对两个程序进行测试。

图 8、图 9 分别显示了加密打包程序和解密程序在不同 block_size 下的测试结果。结果显示,加密打包程序性能表现最好的 block_size 大小是 1024000,平均速度是 38M/s;解密程序性能表现最好的 block_size 大小是 102400,平均速度是 0.906M/s。表明在不同的硬件环境下,同样的加解密程序在速度最快时的 block_size 大小是不一样的,这就要求我们在实际部署时应根据不同情况动态调整 block_size 的大小以达到更快的加解密速度。

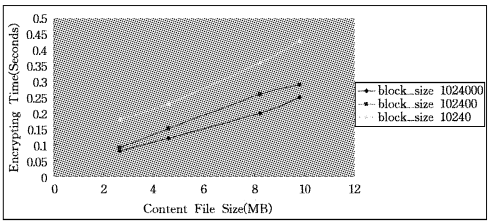

图 8 加密打包程序测试

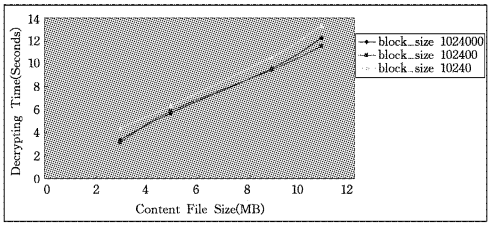

图 9 解密程序测试

结束语 本文针对移动终端上的音视频数字版权保护问题,选取目前移动终端市场占有率最高的 Android 系统为目标平台,分析 Android4.04 源代码和编译规则,参考 OMA DRM v2.0 标准,最终基于 Android 平台实现并部署了我们设计的原型系统。测试结果显示,系统构成之一的 MDRM Player 能够在 Android 平台上按照服务器端设定的规则,在用户权限内对受保护内容按次进行播放控制,同时,设备间能够自由地相互分享数字权利,满足了基本 DRM 需求和用户对于数字权利分享的需求。

参 考 文 献

[1] 张志勇,牛丹梅.数字版权管理中数字权利使用控制研究进展[J].计算机科学,2011,38(4):48-54

[2] 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

[3] Zhang Z Y. Digital rights management ecosystem and its usage controls: A survey [J]. JDCTA: International Journal of Digital Content Technology and its Applications, 2011, 5(3): 255-272

[4] Zhang Z Y. Security, Trust and Risk in Digital Rights Management Ecosystem [M]. 北京:科学出版社,2012

[5] Cristian T, Catalin B. Survey of Mobile Digital Rights Management Platforms [J]. Journal of Mobile, Embedded and Distributed Systems, 2009, 1(1): 32-42

[6] Bhatt S, Sion R, Carbunar B. A Personal Mobile DRM Manager for Smartphones [J]. Computers & Security, 2009, 28(6): 327-340

[7] Chuang Y, Wang C, Lin B. Digital right management and software protection on Android phones[C]// Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st. IEEE, 2010: 1-5

[8] 张硕,马兆丰,卢效峰,等.音乐内容动态加密与许可授权系统设计与实现[J].计算机科学,2011,38(12):43-48

[9] Open Mobile Alliance™, OMA DRM Requirements Candidate Version 2.0, OMA-RD-DRM-V2_0-20040715-C[OL]. http://technical.openmobile alliance.org/Technical/release_program/drm_v2_0.aspx

[10] 李平,卢正鼎,邹复好,等.一个面向家庭网络的数字版权管理系统[J].计算机科学,2009,36(11):116-119

(下转第 142 页)

经过优化后 BSBC 数据隐私技术位合并时间与 AES 加密算法的解密时间如表 8 所列。

表 8　优化后位合并时间与 AES 解密时间(单位 s)

	AES 加密时间	优化后合 2 份时间	优化后合 4 份时间
1M	0.381	0.015	0.018
9.98M	3.974	0.073	0.09
100M	39.955	0.599	0.725
293M	107.019	1.883	2.021
503M	179.189	4.008	4.703
763M	274.266	7.847	9.772

从表 8 中可以看出,优化后的位合并时间较 AES 加密算法的加密时间有了很大的优化,如图 17 所示。图 17 所表示的是 BSBC 相较于传统的 AES 加密算法其加密性能提升了多少倍,BSBC 数据隐私保护技术合并数据的时间是 AES 加密算法加密时间的 35 倍。如果要处理大量数据,BSBC 数据隐私保护技术的位合并方式具有明显优势。

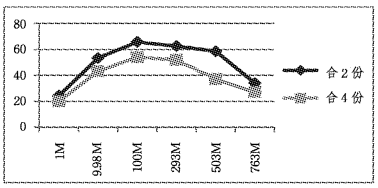

图 17　优化后位合并时间与 AES 解密时间对比提升图

结束语　本文中描述了一种不依赖于原有加密算法的数据隐私保护方式对用户的数据进行隐私保护工作。传统的加密算法对数据进行加密时主要依赖于密钥,一旦密钥丢失,用户的数据就不能得到恢复。而且当用户需要对大量的数据进行隐私保护,或用户数据的版本要经常修改的时候,传统的加密算法会因此产生大量的密钥,如何有效地、高效地对这些密钥进行管理是用户必须考虑的问题。针对以上问题,本文提出利用位拆分方式实现对数据的重新编码,把数据分成多份,再把分离好的数据分别传到云存储平台上,从而实现对用户数据的隐私保护。用户想要查看原数据时,只需从云存储平台分别下载分离后的数据,再通过相应的位合并方式,实现对数据的解码,从而恢复出原数据。

用 C 语言实现了 BSBC 隐私保护技术后,又针对 BSBC 隐私保护技术主要进行移位和与、或操作的特点,在原有 C 语言实现的 BSBC 隐私保护技术的基础上,本文核心代码利用汇编语言进行编程,加快了数据拆分和合并的速度。随后根据计算机指令调度的原理对汇编语言的指令顺序做了调整,从而优化了原有的 BSBC 隐私保护技术。从实验结果中可以看出,BSBC 隐私保护技术相对于传统的加密算法在加密时间和解密时间上有大幅度的提升。

今后,还可以通过多线程的方式进一步提升位拆分、位合并的速度,并把 BSBC 数据隐私保护技术应用到移动终端,为用户移动终端上的数据提供隐私保护。

参 考 文 献

[1] Cloud storage [EB/OL]. http://en.wikipedia.org/wiki/Cloud_stroage,2012-5-10

[2] 傅颖勋,罗圣美,舒继武. 安全云存储系统与关键技术综述[J]. 计算机研究与发展,2013,50(1)

[3] Cloud computing [EB/OL]. http://en.wikipedia.org/wiki/Cloud_computing,2012-5-10

[4] Twinstrara [EB/OL]. http://twinstrara.com,2012-05-10

[5] 侯清铧,武永卫,郑纬民,等. 一种保护云存储平台上用户数据私密性的方法[J]. 计算机研究与发展,2011,48(7)

[6] Amazon simple storage service [EB/OL]. http://aws.amazon.com/s3,2012-05-10

[7] Using Data Encryption [EB/OL]. http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingEncryption.html,2006-3-1

[8] Shraer A,Cachin C,Cidon A,et al. Venus: Verification for untrusted cloud storage [C]// Proc of the 2010 ACM Workshop on Cloud Computing Security Workshop. New York: ACM,2010: 19-30

[9] iCloud: iCloud 安全性与隐私政策概览[EB/OL]. http://support.apple.com/kb/HT4865? viewlocale=zh_CN&locale=zh_CN,2013-2-11

[10] Alani D M M. DES96-Improved DES Security [C]// 2010 7th International Multi-Conference on Systems Signals and Devices (SSD). Amman,2010: 1-4

[11] Shao Jun-xiang, He Zhi-min. High-speed implementation of 3DES encryption algorithm based on FPGA[C]// Modern electronic technology. 2004

[12] Kelsey J,Schneier B,Wagner D. Key Schedule Cryptanalysis of IDEA,G-DES,Gost,Softer and Triple DES[M]. Springer Verlag,1997

[13] NIST Advanced Encryption Standard (AES)[OL]. Development Effort web site http://csrs.nist.gov/encryption/aes/aes-home.htm

[14] Daemen J, Rijmen V. AES Proposal: Rijndael Version 2[EB/OL]. http://www.east.kuleuven.ac.be/~ rijmen/rijndael, 1999-10-05

[15] Rivest R,Shamir A,Aldeman L. A Methoed for Obtaining DigitalSignatures and Public-key Cryptosystems[J]. J. Communications of the ACM,1978,21(2): 120-126

[16] Shimizu Y, Nuno F. Performance Evaluation of Novel DSA Scheme that combines Polling Method with Random Access Method[C]// PIMRC' 06. Helsinki,Finland,Sept. 2006

(上接第 132 页)

[11] Ma G,Pei Q,Wang Y,et al. A General Sharing Model Based on Proxy Re-encryption[C]// 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP). IEEE,2011: 248-251

[12] Feng X,Tang Z,Yu Y Y. An efficient contents sharing method for DRM [C] // Consumer Communications and Networking Conference,2009(CCNC 2009),6th IEEE. IEEE,2009: 1-5

[13] Lee S,Kim J,Hong S J. Redistributing time-based rights between consumer devices for content sharing in DRM system[J]. International Journal of Information Security,2009,8(4): 263-273