(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101057288 B
(45)授权公告日 2010.12.22

(21)申请号 200580038250.3

(22)申请日 2005.10.25

(30)优先权数据
04026516.7 2004.11.09 EP

(85)PCT申请进入国家阶段日
2007.05.09

(86)PCT申请的申请数据
PCT/EP2005/055501 2005.10.25

(87)PCT申请的公布数据
WO2006/051037 EN 2006.05.18

(73)专利权人 汤姆森许可贸易公司
地址 法国布洛涅－比郎库尔

(72)发明人 迪尔克·冈多尔夫
约布斯特·赫雷恩特鲁普
拉尔夫·奥斯特曼 卡斯滕·赫佩尔
乌韦·扬森 哈特穆特·彼得斯
安德烈·谢温措 马尔科·温特

(74)专利代理机构 中科专利商标代理有限责任公司 11021
代理人 罗松梅

(51)Int.Cl.
G11B 20/00(2006.01)
G06F 21/00(2006.01)

(56)对比文件
US 2002194337 A1,2002.12.19, 全文.
US 5644782 A,1997.07.01, 说明书第1页2栏48行－第2页4栏35行,权利要求1-11,图2-4.

审查员 曾雪莲

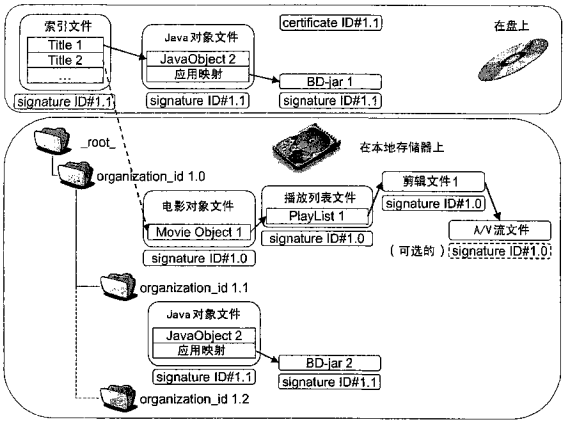
权利要求书 2 页 说明书 7 页 附图 13 页

(54)发明名称

把内容绑定到可移动存储器上的方法和装置

(57)摘要

播放器上的本地存储器提供了向光盘内容增加其它修改和最近补充的能力。这种技术上可行的可能性所出现的问题是:对属于光盘和补充数据的版权进行保护。本发明描述了一种技术,用于通过创建虚拟文件系统(VFS)而获得能够处理这个问题的安全架构,其中通过把基于公共标识符的光盘数据和本地存储数据进行合并而创建虚拟文件系统(VFS)。



CN 101057288 B

1. 一种利用来自第一可移动只读存储介质的第一数据和来自第二存储介质的第二数据来产生虚拟文件系统的方法，其中所述第一存储介质包括多个标题，所述方法包括：

- 从第一存储介质确定第一标识符 (organization_ID)；

- 从所述多个标题中对标题进行选择，并且在选择标题时执行以下步骤：

- 确定在第二存储介质上的根据第一标识符 (organization_ID) 的第一目录中，与所选标题和第一标识符 (organization_ID) 相关的一个或多个文件可用和被授权；以及

- 汇编虚拟文件系统，所述虚拟文件系统包括来自第一存储介质的文件且还包括来自第二存储介质的仅与所选标题和第一标识符 (organization_ID) 相关的所述文件，

其中确定文件被授权包括以下步骤：

- 从第一或第二存储介质中检索文件的签名；

- 利用非对称加密方法的公共密钥来解密所检索的签名；以及

- 使用解密的签名来验证该文件被授权。

2. 根据权利要求 1 所述的方法，还包括从第一存储介质中检索第二标识符 (disc_ID) 的步骤，其中确定在所述第二存储介质上与所选标题和第一标识符 (organization_ID) 相关的一个或多个文件可用包括：确定在第二存储介质上可用的一个或多个所述文件或者位于根据第一标识符 (organization_ID) 的第一目录中、或者根据第二标识符 (disc_ID) 的第二目录中，所述第二目录位于所述第一目录中。

3. 根据权利要求 2 所述的方法，还包括步骤：从第一或第二存储介质中检索证书，并且对所述证书进行解密，从而获得公共密钥。

4. 根据权利要求 3 所述的方法，还包括步骤：将新文件和相应签名文件存储在第二存储介质的第一目录下，其中所述签名文件包括新文件的哈希值并且指向所述证书。

5. 根据权利要求 1 或 2 所述的方法，其中所述确定文件被授权的步骤并非针对一个或多个引用的 AV 流文件执行的，并且所述虚拟文件系统还包括针对其并不执行所述确定步骤的一个或多个引用的 AV 流文件。

6. 根据权利要求 1 或 2 所述的方法，其中第一只读存储介质是可移动介质，并且仅当所述可移动第一存储介质可访问时才维持所述虚拟文件系统，否则，去除所述虚拟文件系统。

7. 根据权利要求 6 所述的方法，其中在存在虚拟文件系统时，所述第二目录是写保护的。

8. 一种利用来自第一可移动只读存储介质的第一数据和来自第二存储介质的第二数据来产生虚拟文件系统的设备，其中所述第一存储介质包括多个标题，所述设备包括：

- 从第一只读存储介质确定第一标识符 (organization_ID) 的装置；

- 从所述多个标题中对标题进行选择的装置，

- 在选择标题时，确定在第二存储介质上的根据第一标识符 (organization_ID) 的第一目录中，与所选标题和第一标识符 (organization_ID) 相关的一个或多个文件可用和被授权的装置；以及

- 汇编虚拟文件系统的装置，所述虚拟文件系统包括来自第一存储介质的文件和来自第二存储介质的仅与所选标题和第一标识符 (organization_ID) 相关的所述文件，

其中所述确定文件被授权的装置包括：

- 从第一或第二存储介质中检索文件的签名的装置；

- 利用非对称加密方法的公共密钥来解密所述签名的装置；以及

- 使用解密的签名来验证该文件被授权的装置。

9. 根据权利要求 8 所述的设备，还包括：从第一存储介质中检索第二标识符 (disc_ID) 的装置，其中所述确定在所述第二存储介质上与所选标题和第一标识符 (organization_ID) 相关的一个或多个文件可用的装置包括：确定在第二存储介质上可用的一个或多个所述文件或者位于根据第一标识符 (organization_ID) 的第一目录中、或者根据第二标识符 (disc_ID) 的第二目录中的装置，所述第二目录位于所述第一目录中。

10. 根据权利要求 9 所述的设备，还包括：从第一或第二存储介质中检索证书的装置，以及对所述证书进行解密从而获得公共密钥的装置。

11. 根据权利要求 10 所述的设备，还包括：将新文件和相应签名文件存储在第二存储介质的第一目录下的装置，其中所述签名文件包括新文件的哈希值并且指向所述证书。

12. 根据权利要求 8 或 9 所述的设备，其中所述确定文件被授权的装置并不应用于位于第二存储介质上且与所选标题和第一标识符 (organization_ID) 相关的一个或多个 AV 流文件，并且所述虚拟文件系统还包括针对其并不应用所述确定装置的一个或多个 AV 流文件。

13. 根据权利要求 8 或 9 所述的设备，其中第一只读存储介质是可移动介质，并且仅当所述可移动只读存储介质可访问时才维持所述虚拟文件系统，否则，去除所述虚拟文件系统。

14. 根据权利要求 13 所述的设备，其中在存在虚拟文件系统时，所述第二目录是写保护的。

把内容绑定到可移动存储器上的方法和装置

技术领域

[0001] 本发明涉及用于把本地存储介质上的内容绑定到可移动 (removable) 存储介质上的内容的方法和装置。具体地，所述可移动存储介质是只读光盘。

背景技术

[0002] 未来的光盘格式将会需要播放器上的本地存储器。这是因为例如只读光盘上所承载的内容很快会变得过时，而且内容生产工作室想要增加其产品的吸引力。播放器上的本地存储器提供了向光盘内容增加其它修改和最新补充的能力，同时不会增加光盘的成本，这对于可写光盘来说是必需的。当与因特网相连时，播放器能够为本地存储器下载补充，或替换只读光盘中的内容。例如，播放器能够使用新的电影片尾来替代过时的电影片尾，或者通过其它音频或字幕轨道对光盘内容进行补充。当相关的光盘位于播放器中时，可以下载可执行代码 - 例如绑定在具体光盘上的游戏 - 并在播放器上运行。

[0003] 这种技术上可行的可能性所出现的问题是 :对属于光盘和补充数据的版权进行保护。期望以这样的方式来保护版权，即当有关的光盘插入重放仪器中时，以独占的方式使用下载的补充数据。对源于内容提供者的补充数据的任何使用以及独立于专用光盘的使用应当在内容所有者 - 即内容提供者 - 的控制下进行。内容提供者希望确保他的所有内容 - 无论是来自光盘还是已经存储在本地存储器上 - 都不会被非法使用、偶然情况下不需光盘而使用或与其它内容一同使用。

[0004] 通常把来自本地存储器和来自光盘存储器的数据安装到播放器内的虚拟文件系统 (VFS) 中。这个步骤仅执行一次，即当插入光盘并启动第一应用时执行。任何其它的应用启动都基于所述虚拟文件系统。这是一种危险因素，因为即使当数据访问受到限制时，内容对于外来应用来说会变得可见。

发明内容

[0005] 本发明描述了一种用于确保安全架构的技术，它能够处理所述的普遍情景。有利地，所述技术还包括如下情况，即来自不同作者的应用位于单一的光盘 - 例如 DVD、Blu-ray 光盘等之上。

[0006] 本发明的要点是 :通过基于公共标识符而把光盘数据和本地存储数据进行合并，从而创建虚拟文件系统 (VFS)。在这里，这个标识符被称作“organization_ID”。它用于认证内容。在启动任意的应用 - 例如标题 (title) 重放 - 之前，动态地执行 VFS 的创建。因此，VFS 的产生可以在光盘呈现期间发生变化，因为所包括的本地存储数据和光盘数据根据标题内的关系而发生变化。organization_ID 的位置和处理被内嵌在安全架构中，在原理上是针对特定应用的。

[0007] 播放器的本地存储器可以包含来自不同内容提供者的数据。根据本发明，这些数据具有不同的 organization_ID，所述 organization_ID 表示这些数据各自的内容提供者。在更为普遍的应用中，光盘还可以包含来自不同内容提供者的标题，因而可以由不同

的 organization_ID 来认证。在启动重放之前，根据在重放仪器上重放的标题而立即产生 VFS。这样，排除了任何非受控的数据组合，与数据是否来自本地存储器或来自光盘－或一般意义上的可移动存储设备－无关。本地存储器上的更新透明度以及临时数据的动态使用得以保护。

[0008] 下文中公开了使用所述方法的装置。

[0009] 下文描述和附图中公开了本发明的优选实施例。

附图说明

[0010] 参考附图来描述本发明的典型实施例，其中：

[0011] 图 1 是签名的产生和验证；

[0012] 图 2 是由可信的第三方证书权威机构来产生证书；

[0013] 图 3 是针对公共密钥的证书验证；

[0014] 图 4 是安全架构内的 organization_ID 的用法；

[0015] 图 5 是应用于不同文件的证书，这些文件根据其在虚拟文件系统中的映射而携带有两种不同的 organization_ID；

[0016] 图 6 是应用于不同文件的证书，这些文件根据其在文件系统中的映射而驻留在可移动光盘和本地存储器上；

[0017] 图 7 是对本地存储器上包括产生的目录组织的组织进行细分；

[0018] 图 8 是所涉及方之间的角色和职责；

[0019] 图 9 是下载的角色和职责；

[0020] 图 10 是虚拟文件系统构造的角色和职责；

[0021] 图 11 是本地删除操作的角色和职责；

[0022] 图 12 是何时构造虚拟文件系统；

[0023] 图 13 是选择绑定单元的理由；

[0024] 图 14 是怎样检查虚拟文件系统的完整性；

[0025] 图 15 是典型的文件结构；

[0026] 图 16 是对来自多个提供者的内容进行绑定；以及

[0027] 图 17 是把共享的片尾绑定到光盘。

具体实施方式

[0028] 本发明基于安全架构的使用，所述安全架构对匹配文件标识的 organization_ID 进行合并，所述匹配文件可以分布在例如第一本地存储介质－例如硬盘驱动器－和第二可移动介质上。在下文中，假定第二可移动存储介质是只读光盘。然而它可以是任意的可移动存储介质，例如快闪存储器等。当把光盘插入播放器中时，虚拟文件系统 (VFS) 得以汇编 (compiled)。基于所执行的标题和其 organization_ID 来选择将要汇编的文件。在下文中，术语“organization_ID”用于解释一般概念。在其它实施例中可以定义像 disc_ID 或 application_ID 的其它子结构，其中光盘中的一部分可以使用粒度 (granularity) 更小的 organization_ID。

[0029] 典型的安全架构基于使用非对称加密、签名和证书的基本原理。图 1 示出了使用

签名以为接收方确保消息或数据集的可靠性。消息发送方（“Alice”）拥有私人密钥，她使用这个私人密钥对从消息内容中自动产生的哈希值进行加密。已加密的哈希值被称作“签名”，并把签名与所要保护的消息一同发送。如果接收方（“Bob”）可以从所述消息中产生与已接收的签名相等的哈希值，则他仅能够打开所述消息。为此，接收方使用用于从已接收的消息中产生哈希值的预定义算法，并使用用于对签名进行解密的发送方公共密钥。由于公共密钥与发送方的私人密钥不同，所以她可以公开地分发公共密钥。公共密钥适于对使用相应的私人密钥而加密的数据进行解密。知晓发送方公共密钥的接收方能够对签名－即接收到的已加密哈希值－进行验证。仅当接收方产生的哈希值和已解密的签名相等时，接收方才能够确信消息在其间没有被修改。

[0030] 为了避免把公共密钥向四处发送－它是危险因素并且是密钥管理开销－而使用证书技术。如图 2 所示，可信的第三方机构（证书权威机构，CA）对客户端（“Alice”）的公共密钥进行签名。这个步骤通过产生公共密钥的哈希值、利用 CA 私人密钥对所产生的哈希值进行加密以及把加密后的哈希值送回客户端而完成。这个签名与客户端的公共密钥一同形成了所使用的证书。根据本发明，把其它信息－具体为像例如 organization_ID 的标识符－增加并包括在证书中。

[0031] 如果接收方（“Bob”）具有 CA 公共密钥，则他可以验证发送方的公共密钥是否可靠。图 3 示出了针对 Alice 的密钥的证书验证。通过使用可信的第三方公共密钥对证书进行解密，接收方可以验证已接收的公共密钥－即使它属于先前未知的发送方－的可靠性。仅当从已接收的公共密钥中再次产生的哈希值与已解密的证书相匹配时，才会相信从发送方接收到的公共密钥。

[0032] 如图 4 所示，本发明可以基于一般的安全架构而使用。针对光盘和本地存储器上的数据，产生了合并有证书并携带有 organization_ID 的签名。签名证书对签名的可靠性和 organization_ID 的可靠性进行验证。本发明包括本地存储器为持久性的，即所存储的数据在掉电后不会丢失且存储到本地存储器的所有数据得以保持，无论它们是来自光盘、通过因特网下载或来自其它来源。

[0033] 图 5 示出了根据本发明把这种安全架构用于光盘文件系统，其中应用了对受 organization_ID 控制的 VFS 进行装配。证书的 certificate_ID#1、#2 携带有各自的 organization_ID，而被牢固地绑定到实际数据内容的签名文件指向其证书。通过所谓的“摘要代码”－即通过把已定义的算法应用于数据内容而产生的代码－而实现签名文件与数据内容之间的语义绑定。可以通过文件结构或清单文件来实现签名与数据之间的物理绑定。图 5 所示的文件例如是位于光盘上的所有文件，即光盘承载有由两个不同的内容提供者所创作的标题，他们具有不同的证书和不同的签名标识符 -signature_ID#1、#2。

[0034] 在开始呈现例如标题 1 之前，本发明对虚拟文件系统 VFS 进行汇编。汇编规则由证书 certificate_ID#1、#2 来制定，其中所述 certificate_ID#1、#2 被分配给属于光盘上数据的签名。在这个示例中，假定附加到标题 Title_1 的签名 signature_ID#1 由 organization_ID#1 来认证。因此根据本发明，在开始呈现 Title_1 之前，仅可以对由 organization_ID#1 所认证的数据进行装配。同样，在开始呈现例如由 organization_ID#2 所保护的 Title_2 之前，仅可以对由 organization_ID#2 所认证的数据进行装配。不同的 organization_ID 可以属于不同的内容创作者或所有者，即内容的作者、工作室、提供者等。

[0035] 如图 5 所示，上文提到的标题可以包括例如电影对象文件（电影对象）或 Java 对象文件（Jave 对象）或其它类型的文件。电影对象描述了用于音频和 / 或视频 (AV) 呈现的流文件，而 Java 对象描述了可以在 Java 虚拟机 (JVM) 上运行的 Java 可执行程序。Java 对象数据被存储在 Java 归档文件 BD-jar1、BD-jar2 中。两种对象类型都可以具有认证的签名文件。

[0036] 一种可能的例外是对 AV 流文件的处理，因为 AV 流文件不必具有签名文件。这是因为流文件较大，而较大文件的摘要代码（即哈希值）计算需要较长的时间。对于 VFS 的装配，对所签名的 AV 流文件的处理方式与上文针对其它文件的处理方式相同。但是装配过程还会包括那些完全没有被签名的 AV 流文件，即仅包括那些利用正确的 organization_ID 进行认证或完全没有签名的流文件。对 AV 流文件的这种处理不会破坏安全链，因为对 AV 流文件的任何访问仅能够借助电影对象、播放列表和剪辑文件来进行。这些文件通常是被签名的且直接或间接地得到认证的。

[0037] 在本发明的另一个实施例中，把 organization_ID 应用于完整的光盘。在这种情况下，在开始呈现光盘之前仅执行一次 VFS 的产生。当考虑到本地存储器时，与现有技术领域中的系统的相对差别变得明显。根据本发明，属于光盘的数据和存储在本地存储介质上的数据携带有签名和包括 organization_ID 的证书。对于 AV 流文件来说，上述例外保持有效。本地存储器上所有未受认证的数据仅能够分开运行，即从这些数据中产生的虚拟文件系统不包括任意受认证的或不同地受到认证的光盘内容。因此，内容提供者可以确信他的内容是安全的，而且光盘上的数据材料不会被滥用。

[0038] 如图 6 所示，在本发明的另一个实施例中，本地存储器使用 organization_ID。被加载到播放器的光盘包含索引文件，其第一标题 Title_1 引用 Java 对象文件。Java 对象文件指向 BD-jar 应用。光盘上的所有文件都由相同的标识符 organization_ID#1- 例如与制作光盘内容的内容提供者相关 - 来签名和认证。对于 AV 流文件，上述例外规则保持有效。本地存储器 - 例如 HDD- 包含来自两个不同的内容提供者的文件。这些文件由取决于内容提供者的不同签名来签名。例如 Java 对象文件和 BD-jar 应用由 organization_ID#1 来签名和认证，而电影对象文件、播放列表文件、剪辑文件和 AV 流文件由 organization_ID#2 来签名和认证。

[0039] 在插入光盘后，根据位于光盘上的证书来装配虚拟文件系统。在这个示例中，光盘上仅有一个证书，即 organization_ID#1，因而根据本发明仅把同样利用匹配标识符 organization_ID#1 进行认证的文件包括在 VFS 中。在 VFS 的生存周期内，所有其它的文件保持不可见和不可访问。由于在这种情况下本地存储器包含 Java 对象文件 - 即其签名 signature_ID#1 中具有相同名称和相同标识符 (organization_ID#1) 的 Java 对象文件 - 的更新，这个更新替代了光盘上的 Java 对象文件，并且运行来自 HDD 的 Java 应用 BD-jar2 而不是来自光盘的原始 Java 应用。

[0040] 如这个示例所示，通过禁止任何非受控的数据组合 - 然而可能是来自本地存储器的数据，使用证书中的 organization_ID- 提供了虚拟文件系统的产生规则 - 明显提高了本地存储器的安全性。在本地存储器上产生数据的应用必须对所产生的文件进行签名，而且签名必须指向携带有适合的 organization_ID 的光盘证书。否则所述应用将不能在本地存储器上再次找到新产生的数据。如果所产生的数据保持在高速缓存中，则在销毁 VFS 之前

该数据对于 VFS 保持可用，所述 VFS 的销毁在例如如下时候发生：把光盘从播放器中取出、关闭播放器、选择了来自光盘的另一个标题或发生其它事件。对 VFS 进行下一次汇编时，所述绑定仅包括来自本地存储器并由正确的 organization_ID 所认证的数据。

[0041] 这导致了如下效果：与具体应用有关且被相应签名的数据可能不会由外来应用－例如来自其它内容提供者－所使用。

[0042] 本发明的另一方面是把 organization_ID 细分为组织结构。这可以通过向证书增加其它标识符而完成，例如 disc_ID、application_ID 等。另一种可能是使用图 4 中的“组织名称”内的“文本”字段。这种细分的优点是，内容提供者－例如迪斯尼工作室－可以控制对其用于不同应用的通用文件和其它数据的访问，特别是应当避免这些文件和数据也可以由来自其它内容提供者的应用所使用的时候。为了实现这个目的，所有已授权的应用必须可以使用这些数据，同时避免来自其它（未授权）应用的访问。

[0043] 然而如上所述，把相应的证书应用于这些文件需要预先知晓随后的使用。随后的验证－例如通过增加新的证书－是不可能的，因为它需要对本地存储器上的相应目录进行写入访问。但如果应用具有另一个证书，则这些目录在随后的 VFS 中是不可见的。作为一种解决方案，提出了预先跨越保护伞并在稍后进行决策－允许哪个应用在这个保护伞下通过。通过定义了根目录的 organization_ID 的主要部分来跨越保护伞。

[0044] 图 7 示出了这种情况。所有的细分都位于根目录 _root_ 中的子目录中。当产生 VFS 时，包括了组织目录内的所有文件和目录以及自有的细分目录。邻目录是不可见的，仅有根目录内的所有数据和文件可见。例如图 7 中的电影对象 Movie_Object_1 可以包含最近的电影片尾。在所示的存储位置处－目录 organization_ID#1.0 中，可以把它绑定到与来自相同工作室的光盘有关的所有 VFS，因而可以从所有这些光盘启动最近的电影广告。由例如 organization_ID#1.2 所认证的光盘还可以包括来自由 organization_ID#1.0 所认证的本地存储器的文件，因为这些文件存储在相同的根目录下。可以把它们存储在不同的子目录下，所述子目录可以与例如来自相同“组织”的不同光盘或不同标题有关。如上所述，这个组织通常可以是任意的证书权威机构。

[0045] 本发明的一方面是：使用标识符或 organization_ID 来控制虚拟文件系统 VFS 的装配。

[0046] 本发明的一方面是：把由标识符或 organization_ID 来控制的虚拟文件系统装配应用于（本地）存储介质－例如 HDD，其中对装配过程进行控制的各个有效的标识符是从可移动的存储介质中得到的。

[0047] 本发明的一方面是：把受标识符控制的虚拟文件系统装配应用于光盘，其中各个有效的标识符是从所述光盘上当前出现的标题中得到的。

[0048] 本发明的一方面是：把标识符或 organization_ID 存储在证书内，所述证书被定义在安全系统架构内。

[0049] 本发明的一方面是：把本地存储介质上的主目录树－其中主目录树由 organization_ID 来认证－细分为包括多个子目录的组织结构，并允许相同主目录中的所有其它子目录对子目录进行访问。

[0050] 本发明的一方面是：对电影对象和 / 或 Java 对象进行签名和认证。

[0051] 本发明的一方面是：为存储在本地存储介质上的新创建的文件创建签名，其中所

述签名指向位于可移动存储介质－具体为光盘－上的证书和标识符。

[0052] 可能存在这样一种规则，即存储在本地光盘上的所有 AV 流文件必须由具体的 organization_ID 来签名和认证。这确保了不能对本地存储介质上的流文件进行未授权的操作。这一点在产生与任意现存剪辑文件相匹配的表面 AV 流文件时是可能的。可以采用其它安全机制以确保这一点，例如对 AV 流文件进行加密。还可能存在这样一种规则一例如要求所有的 AV 流文件由相同的密钥进行加密，其中所述密钥位于可移动的光盘上，而且所有未加密的 AV 流文件不能由仪器来表现。

[0053] 图 8 示出了用户、内容提供者和播放器制造商之间的从属性。他们应当对用于产生虚拟文件系统 VFS 的一组规则达成一致意见。

[0054] 图 9 示出了针对例如通过因特网下载附加内容所考虑的方面。内容作者可以发起这种下载，例如通过把通知消息放在专用因特网站点上。在预定事件发生时－例如当用户把与附加内容有关的光盘插入他的播放器时、或当他启动具体相关的应用时、以及当允许用户对播放器进行网络访问且该播放器支持这种功能时，可以按照上文所述而下载、验证和存储附加内容。

[0055] 图 10 示出了所涉及方如何对虚拟文件系统的构造产生影响。用户可以设置首选项，播放器制造商可以提供所需的软件和／或硬件功能－例如用于在存储卡上存储附加内容的存储卡驱动，而内容作者可以针对他发售的某些或全部光盘而使用所达成的 VFS 规则。

[0056] 图 11 示出了当删除已下载的内容时所考虑到的各个方面。例如可以对播放器编程，从而如果被授权的内容作者（具有各自的标识符）发出相应的信号，则所述播放器自动地删除已存储的属于一个或多个绑定单元的内容。这一点可以在例如电影片尾过期或类似事件发生时使用。

[0057] 在图 12 中讨论何时构造虚拟文件系统这个问题。

[0058] 当在插入光盘的时候构造 VFS 时，从播放器实施方式这一点来看，这是最容易的解决方案，但这种方案在光盘插入期间不能产生或下载数据。

[0059] 第二种可能是在插入光盘时或在选择新的标题时构造 VFS。从用户的观点来看，这是良好的解决方案，因为当标题发生改变时会出现中断，在这个中断期间内可以产生新的 VFS 而不会打扰用户。

[0060] 第三种可能是总在重放开始时构造 VFS。然而在应当把电影对象或 Java 对象包括在绑定中时，这需要与仅包括播放列表、剪辑信息和 MPEG2-TS（传输流）文件时不同的机制。这些机制处于不同的逻辑级别，而在重放开始时，新的电影对象或 Java 对象可能导致关于哪些文件有效的混乱。

[0061] 当可编程平台－例如 Java 虚拟机 JVM－可用时，能够应用第四种可能，即当出现应用程序接口（API）调用时动态地产生 VFS。

[0062] 构造 VFS 被称作“绑定”。图 13 讨论了什么是良好的绑定单元，即绑定对于哪个单元或实体应当有效：绑定可以涉及：光盘、来自具体光盘的标题、涉及来自具体光盘的标题的播放列表或动态绑定。优选的绑定单元是按照标题的绑定，因为 VFS 架构是基于标题的。

[0063] 如上所述，必须检查 VFS 的完整性。这一点在图 14 中讨论。完整性检查优选地在每次构造 VFS 之前完成。目标是把内容与例如不同工作室分开，并且防止使用被破坏的或

无效的数据。

[0064] 图 15 示出了示例文件结构。完整特征索引文件包含对多个标题的引用，这些标题是电影对象表和 / 或 Java 对象。每一个 Java 对象包括应用管理表 AMT。电影对象表和 AMT 分别指向电影对象或 Java 归档文件，它们可以被包括在根据本发明的 VFS 中。电影对象可以是例如视频场景、字幕流、图形动画流等。

[0065] 图 16 示出了怎样把来自不同提供者 Organization_id_1、Organization_id_2 的内容绑定到虚拟文件系统，即如果提供者使用例如相同的光盘标识符 Disc_ID_0 的话。它需要在工作台之间是唯一的，并且可以是例如通用唯一标识符 (UUID)。

[0066] 图 17 解释了怎样把共享的片尾与只读光盘相关联。

[0067] 本发明可以用于针对可移动存储介质的记录和 / 或重放设备，它可以对其它可重写存储介质 - 例如 HDD- 进行访问。本发明可以用于例如 Blu-ray 光盘的高密度数据载体。

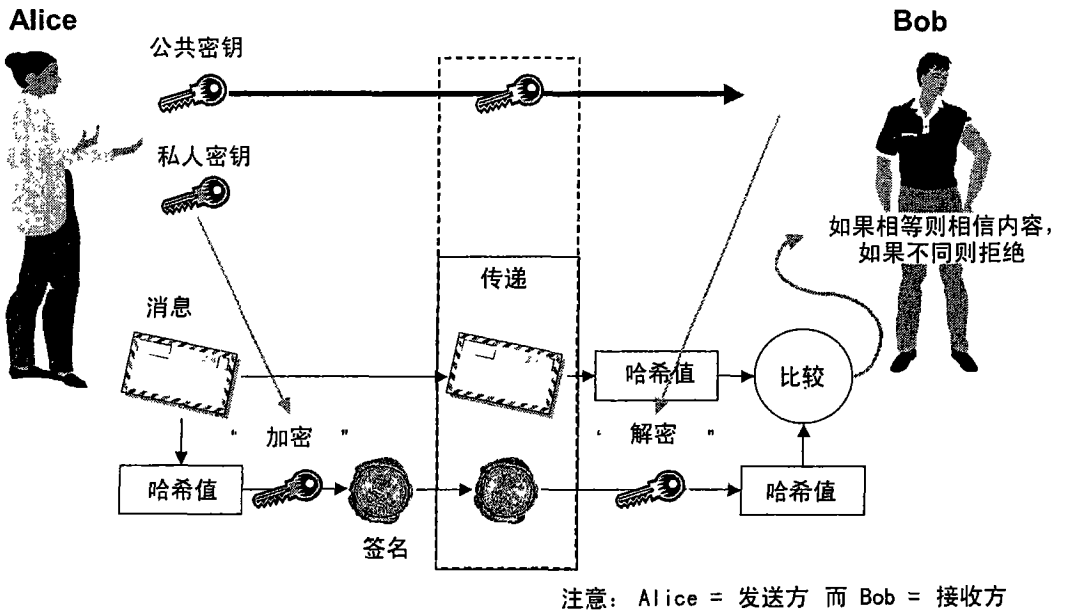

Alice
Bob
公共密钥
私人密钥
消息
传递
如果相等则相信内容，
如果不同则拒绝
哈希值
比较
“ 加密 ”
“ 解密 ”
哈希值
哈希值
签名
注意：Alice = 发送方 而 Bob = 接收方

图 1

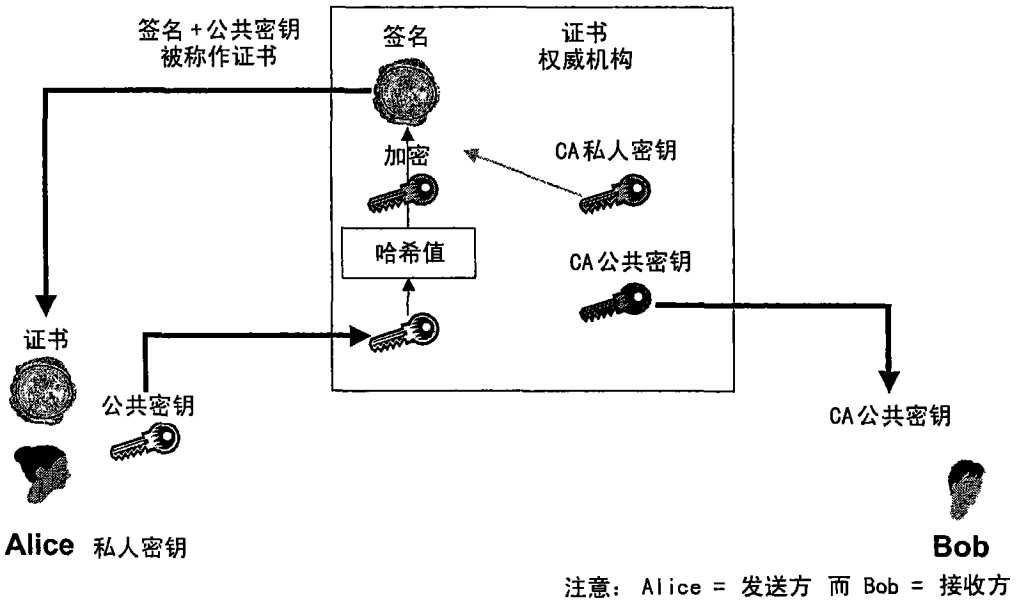

签名 + 公共密钥
被称作证书
签名
证书
权威机构
加密
CA私人密钥
哈希值
CA公共密钥
证书
公共密钥
CA公共密钥
Alice 私人密钥
Bob
注意：Alice = 发送方 而 Bob = 接收方

图 2

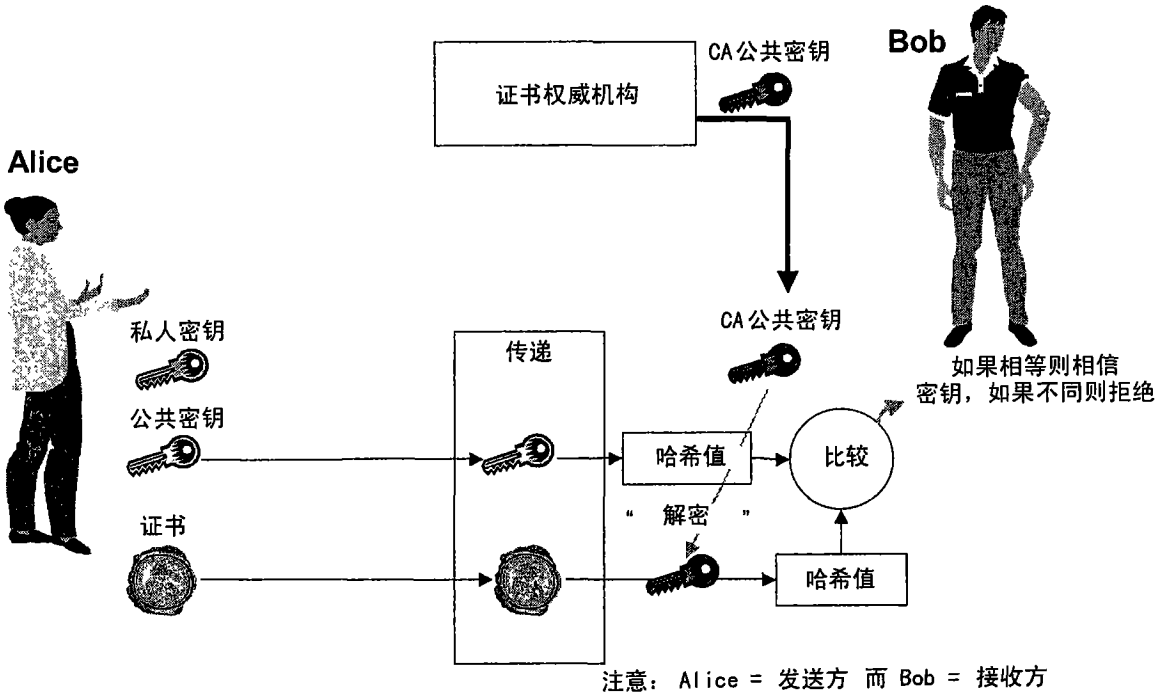

证书权威机构
CA公共密钥
Bob
Alice
私人密钥
公共密钥
证书
传递
CA公共密钥
哈希值
比较
“ 解密 ”
哈希值
如果相等则相信
密钥，如果不同则拒绝
注意： Alice = 发送方 而 Bob = 接收方

图 3

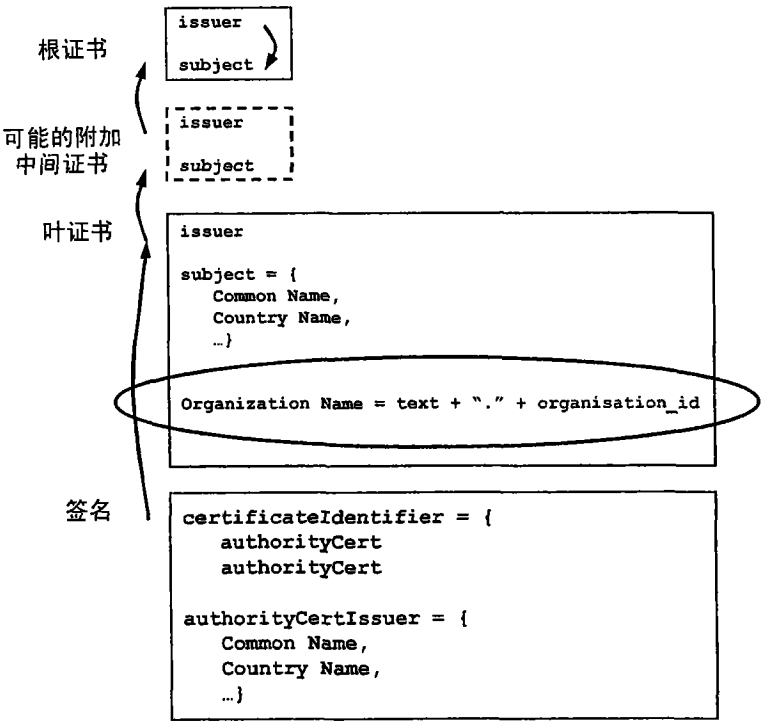

根证书
issuer
subject
可能的附加
中间证书
issuer
subject
叶证书
issuer
subject = {
Common Name,
Country Name,
…}
Organization Name = text + "." + organisation_id
签名
certificateIdentifier = {
authorityCert
authorityCert
authorityCertIssuer = {
Common Name,
Country Name,
…}

图 4

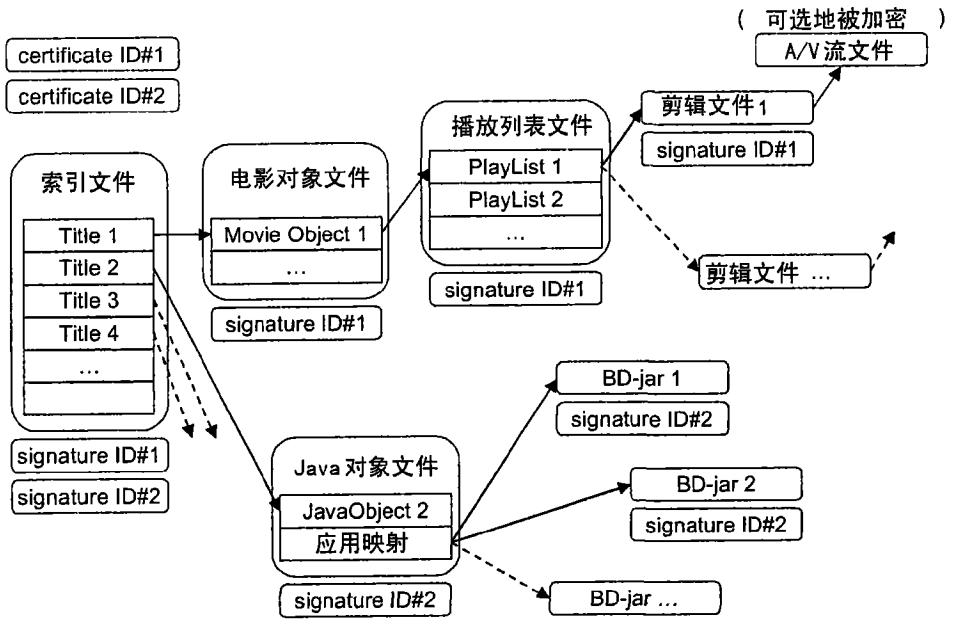

(可选地被加密)
A/V流文件
certificate ID#1
certificate ID#2
剪辑文件1
signature ID#1
播放列表文件
PlayList 1
PlayList 2
...
signature ID#1
索引文件
Title 1
Title 2
Title 3
Title 4
...
电影对象文件
Movie Object 1
...
signature ID#1
剪辑文件 ...
signature ID#1
signature ID#2
Java对象文件
JavaObject 2
应用映射
signature ID#2
BD-jar 1
signature ID#2
BD-jar 2
signature ID#2
BD-jar ...

图 5

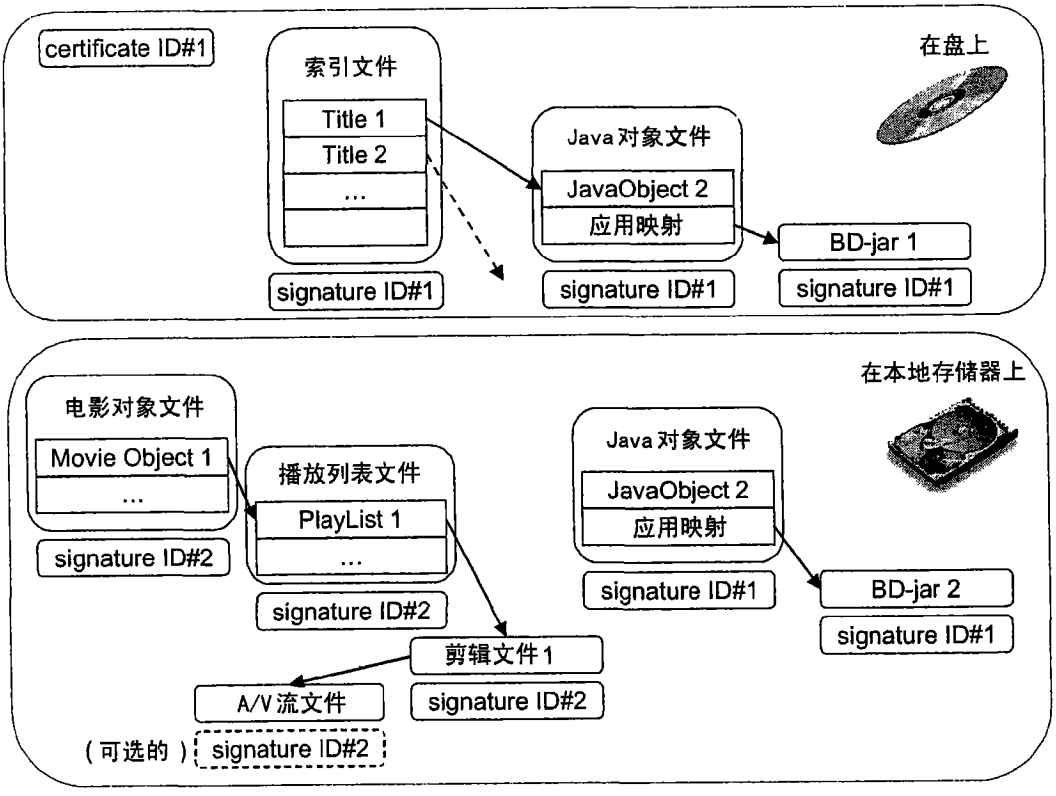

certificate ID#1
索引文件
Title 1
Title 2
...
signature ID#1
Java对象文件
JavaObject 2
应用映射
signature ID#1
BD-jar 1
signature ID#1
在盘上
在本地存储器上
电影对象文件
Movie Object 1
...
signature ID#2
播放列表文件
PlayList 1
...
signature ID#2
剪辑文件1
signature ID#2
A/V流文件
(可选的) signature ID#2
Java对象文件
JavaObject 2
应用映射
signature ID#1
BD-jar 2
signature ID#1

图 6

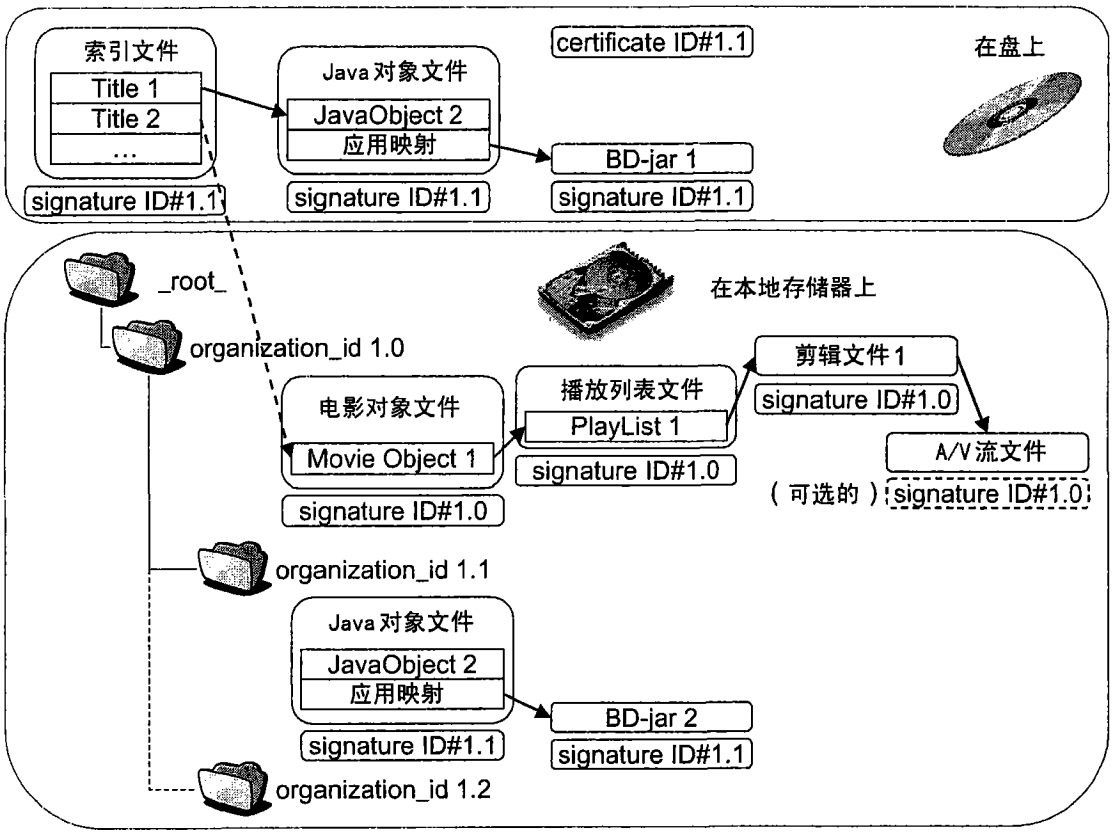

索引文件
Title 1
Title 2
...
signature ID#1.1
Java 对象文件
JavaObject 2
应用映射
signature ID#1.1
certificate ID#1.1
BD-jar 1
signature ID#1.1
在盘上
root
在本地存储器上
organization_id 1.0
电影对象文件
Movie Object 1
signature ID#1.0
播放列表文件
PlayList 1
signature ID#1.0
剪辑文件 1
signature ID#1.0
A/V 流文件
(可选的) signature ID#1.0
organization_id 1.1
Java 对象文件
JavaObject 2
应用映射
signature ID#1.1
BD-jar 2
signature ID#1.1
organization_id 1.2

图 7

角色和职责

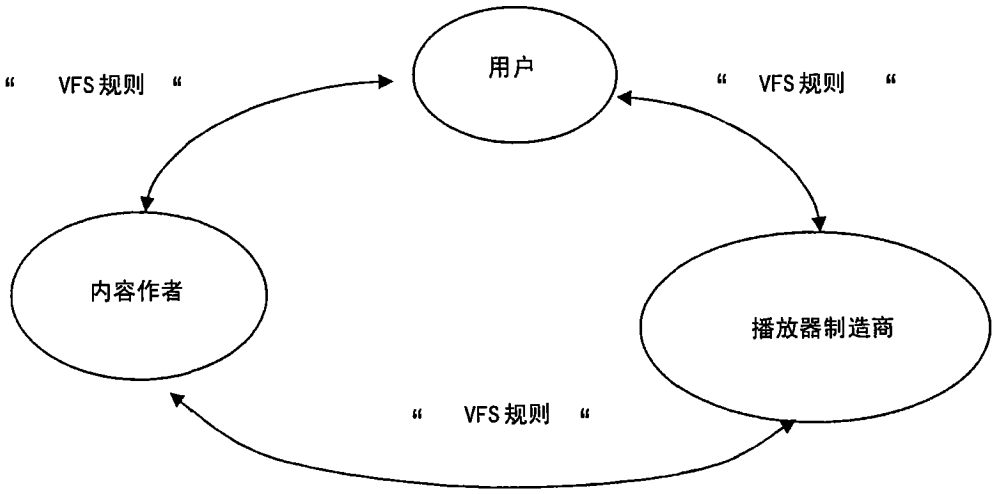

用户
“ VFS 规则 “
“ VFS 规则 “
内容作者
播放器制造商
“ VFS 规则 “

图 8

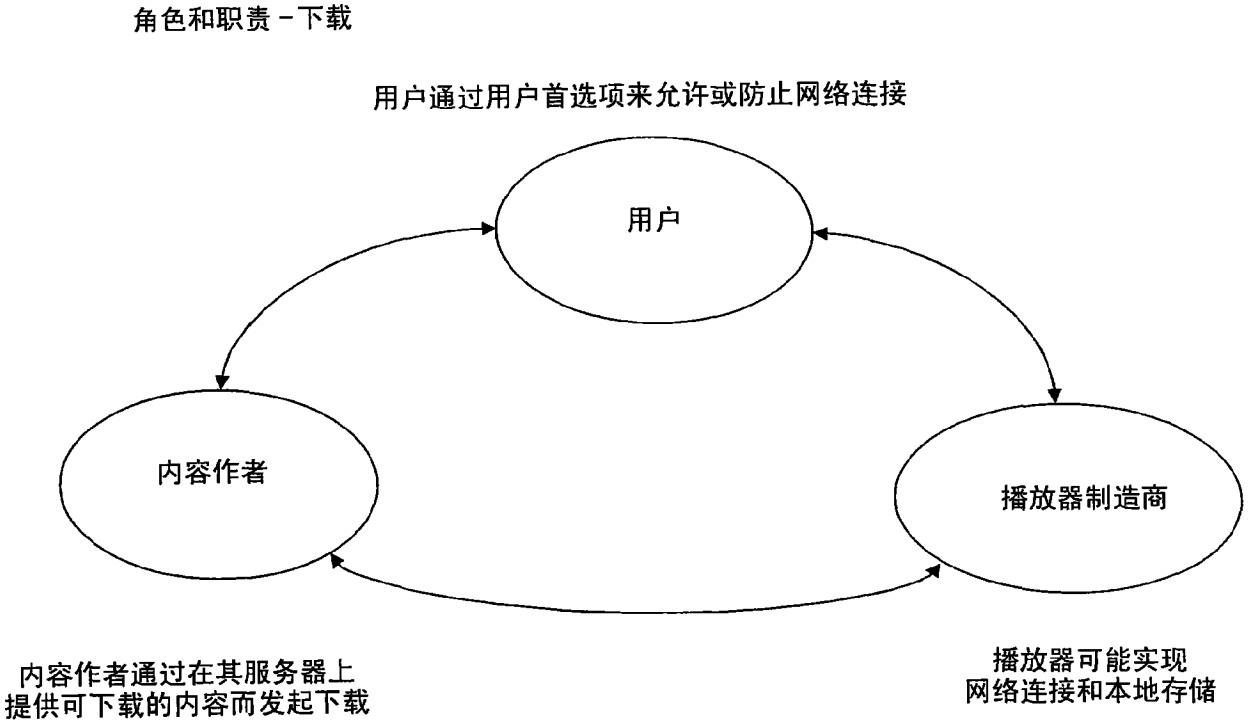

角色和职责 -下载
用户通过用户首选项来允许或防止网络连接
用户
内容作者
播放器制造商
内容作者通过在其服务器上
提供可下载的内容而发起下载
播放器可能实现
网络连接和本地存储

图 9

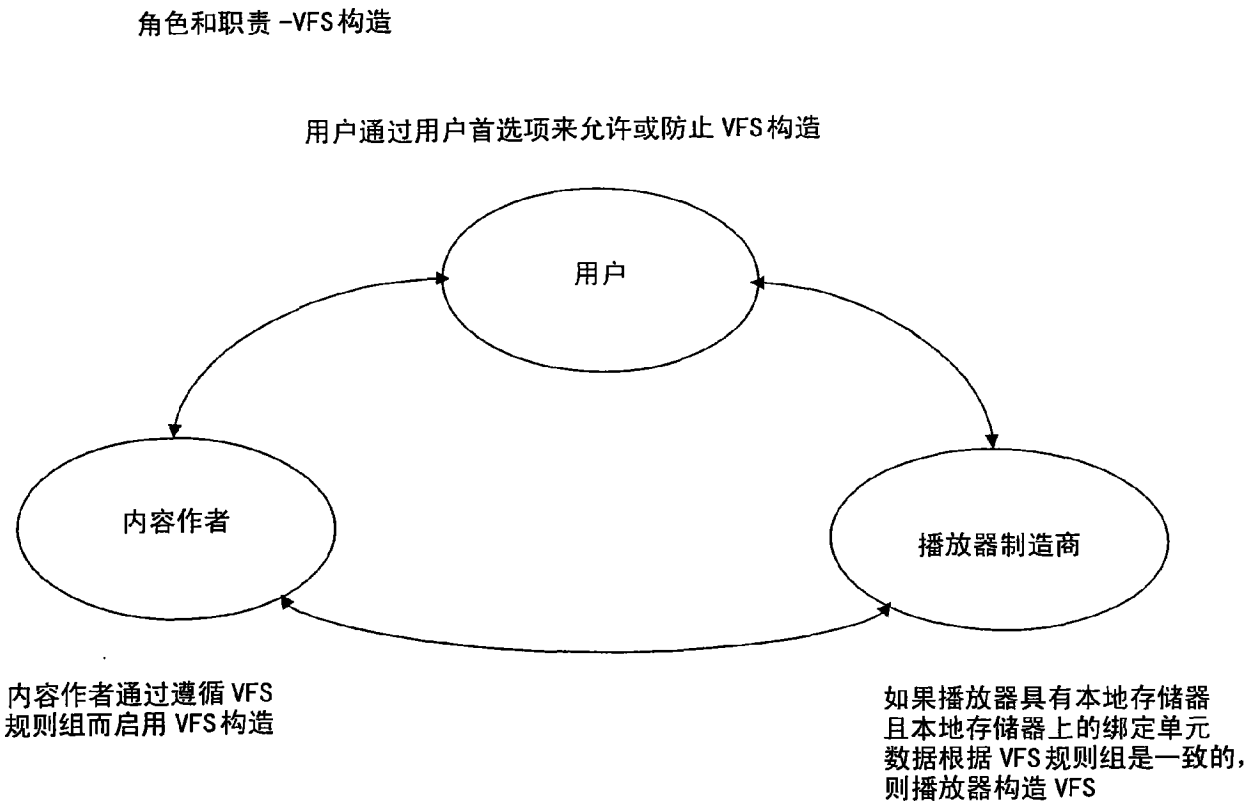

角色和职责 -VFS构造
用户通过用户首选项来允许或防止 VFS构造
用户
内容作者
播放器制造商
内容作者通过遵循 VFS
规则组而启用 VFS构造
如果播放器具有本地存储器
且本地存储器上的绑定单元
数据根据 VFS规则组是一致的，
则播放器构造 VFS

图 10

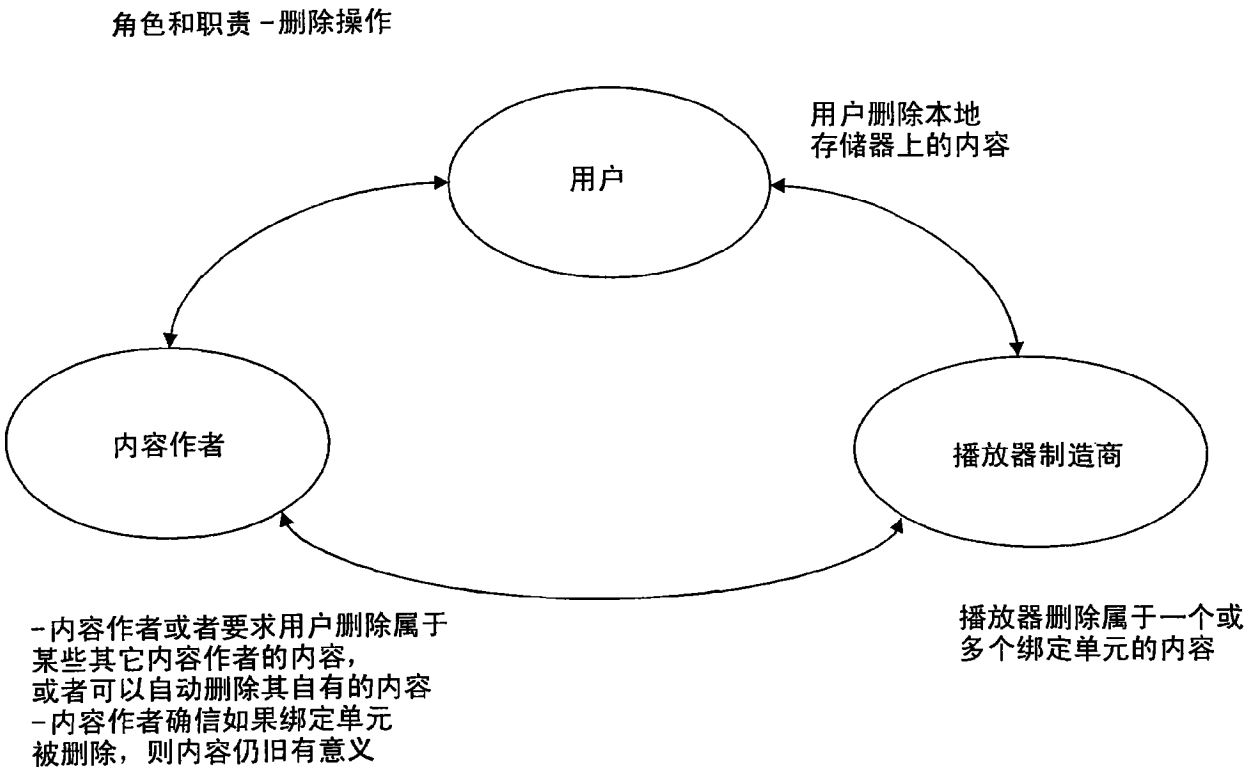

角色和职责 -删除操作
用户
用户删除本地
存储器上的内容
内容作者
播放器制造商
-内容作者或者要求用户删除属于
某些其它内容作者的内容，
或者可以自动删除其自有的内容
-内容作者确信如果绑定单元
被删除，则内容仍旧有意义
播放器删除属于一个或
多个绑定单元的内容

图 11

何时构造 VFS？

- 在盘插入时
  - 从播放器实施方式的观点来看是最简单的，但是在盘被插入期间不允许下载和下载内容的随后重放。
- 在盘插入和标题改变时
  - 良好的选择，因为标题改变通常不是无缝的，不论怎样都需要加载播放列表、剪辑信息和电影（？）/Java对象。
- 在重放开始时
  - 对于播放列表、剪辑信息和 M2TS 文件有效，但对于已下载的电影对象和 BD-J 对象则要求不同的 VFS 构造机制。
- 动态（API 调用）
  - 如果不存在可能被中断的重放运行，则可能用于 BD-J，但是不能用于 HDMV。

图 12

什么是良好的绑定单元？

- 按照盘
  - 从播放器实施方式的观点来看是最简单的，但是在盘被插入期间不允许下载和下载内容的随后重放。
- 按照标题
  - 最适于基于标题的 VFS 构造架构
- 按照播放列表
  - 对于播放列表、剪辑信息和 M2TS 文件有效，但对于电影对象、 BD-J 对象文件的更新则要求分离绑定单元的概念
- 动态（ API 调用）
  - 如果不存在可能被中断的重放运行，则可能用于 BD-J，但是不能用于 HDMV。

图 13

怎样检查 VFS 的完整性?

- 每次构造 VFS 之前，播放器都需要检查 VFS 的完整性

— 目标：

- 避免使用被破坏的数据（例如在下载期间被破坏）
- 避免使用部分下载的内容（不完整的绑定单元）
- 确保仅有被授权的内容用于 VFS（不要把工作室 A 的内容窃取到工作室 B 的 VFS 中）

— 可能的解决方案

- 哈希码可以很好地检测单一文件中的破坏
- 拥有属于绑定单元的所有文件的列表
- 签名可以用于授权

图 14

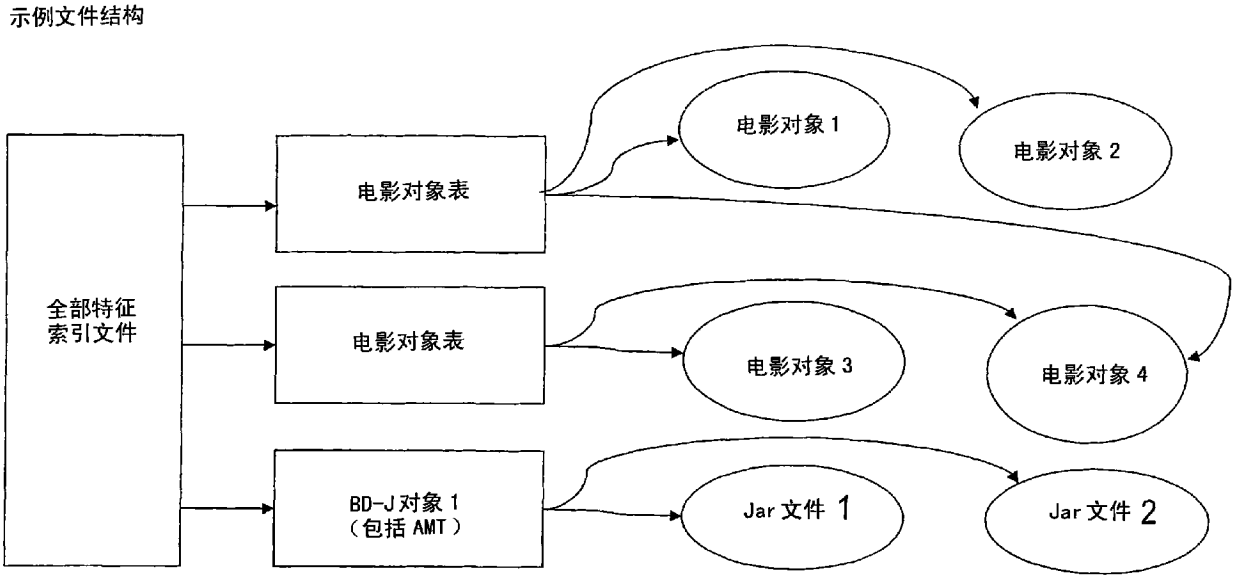

示例文件结构
全部特征
索引文件
电影对象表
电影对象表
BD-J 对象 1
（包括 AMT）
电影对象 1
电影对象 2
电影对象 3
电影对象 4
Jar 文件 1
Jar 文件 2
• 电影对象表包含由全部特征 HDMV 标题所使用的所有电影对象文件和播放列表文件的列表
• BD-J 对象包含由 BD-J 标题所使用的所有 JAR 文件和播放列表文件的列表

图 15

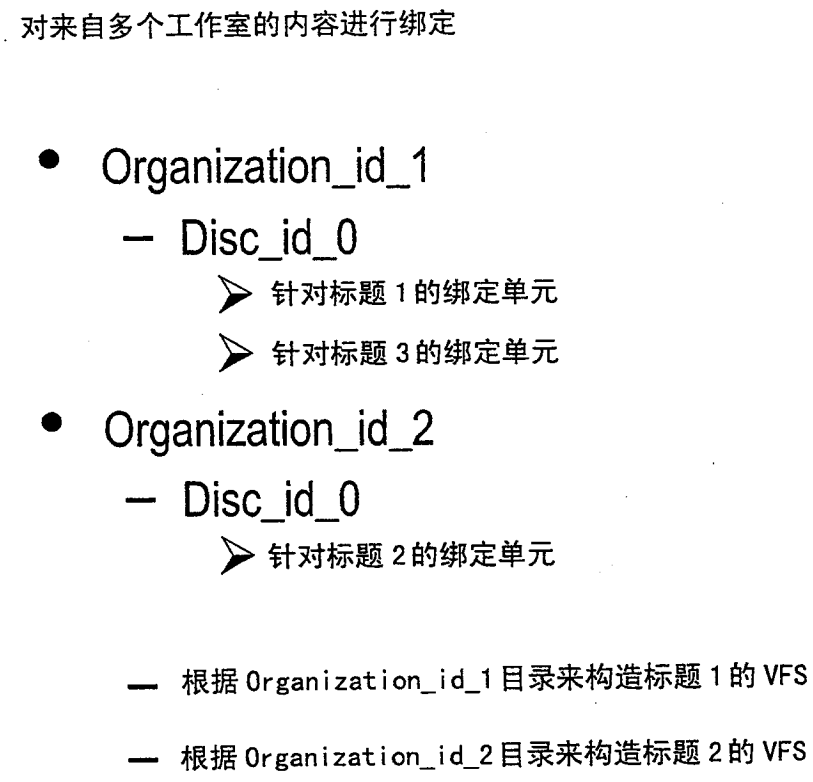

对来自多个工作室的内容进行绑定
Organization_id_1
Disc_id_0
针对标题 1 的绑定单元
针对标题 3 的绑定单元
Organization_id_2
Disc_id_0
针对标题 2 的绑定单元
根据 Organization_id_1 目录来构造标题 1 的 VFS
根据 Organization_id_2 目录来构造标题 2 的 VFS

图 16

把共享的片尾绑定至盘

- 向盘增加片尾需要对索引文件进行更新
  - 这可以针对本地存储器上的每一个盘而单独完成
- 内容作者具有 2 种选择：
  - 为每一个盘下载新的索引文件
    - 索引文件是非常短 -> 快的下载
  - 使用 Java 应用程序来改变每一个盘的索引文件
    - 索引文件结构是非常简单 -> 不重要的

图 17