发明名称
将数字版权绑定到软件应用

摘要
本发明公开了一种计算机实施的用于将数字版权管理信息绑定到软件应用的方法和系统。所述方法和系统包括用于在软件应用组件与永久数据存储组件之间提供数字版权管理 (DRM) 组件：截取来自所述软件应用组件的访问所述永久数据存储组件的请求：将 DRM 组件数据与软件应用组件数据绑定在绑定数据集中；以及，将所述绑定数据集存储在所述永久数据存储组件中。
1. 一种方法，包括：
在软件应用组件与永久数据存储组件之间插入数字版权管理组件；
截取来自所述软件应用组件的访问所述永久数据存储组件的请求；
将数字版权管理组件数据与软件应用组件数据绑定在绑定数据集中；
以及
将所述绑定数据集存储在所述永久数据存储组件中。

2. 如权利要求 1 所述的方法，其中，所述永久数据存储组件远程连接到所述数字版权管理组件。

3. 如权利要求 1 所述的方法，其中，利用以下处理中的一种或多种来绑定所述绑定数据集：所述处理包括利用密码加密、加密、隐写式隐藏。

4. 如权利要求 2 所述的方法，其中，利用以下处理中的一种或多种来绑定所述绑定数据集：所述处理包括利用密码加密、加密、隐写式隐藏。

5. 如权利要求 2 所述的方法，其中，所述数字版权管理组件数据包括指示特定用户的信息，针对所述特定用户的数字版权管理组件数据的存储不影响先前针对不同用户存储的数字版权管理组件数据。

6. 如权利要求 1 所述的方法，其中，所述绑定数据集为以下类型中的一种或多种：所述类型包括数据块、流式数据。

7. 一种方法，包括：
在软件应用组件与永久数据存储组件之间插入数字版权管理组件；
截取来自所述软件应用组件的访问所述永久数据存储组件的请求；
从所述永久数据存储组件检索绑定数据集；以及
从解绑定数据集中的数字版权管理组件数据恢复软件应用组件数据。

8. 如权利要求 7 所述的方法，其中，所述永久数据存储组件远程连接到所述数字版权管理组件。

9. 如权利要求 7 所述的方法，其中，利用以下处理中的一种或多种来恢复所述绑定数据集：所述处理包括利用密码解密、去扰、暴露隐写。
式隐藏的数据。

10. 如权利要求8所述的方法，其中，利用以下处理中的一种或更多种来恢复所述绑定数据集：所述处理包括利用密码解密、去扰、暴露隐写式隐藏的数据。

11. 如权利要求7所述的方法，其中，所述解绑定数据集为以下类型中的一种或更多种：所述类型包括数据块、流式数据。

12. 如权利要求7所述的方法，还包括：从数字版权管理组件数据恢复软件应用组件数据，而不修改所述永久数据存储组件中的所述绑定数据集。

13. 一种实施为包括数据的机器可存取介质的制品，当由机器访问时，使所述机器用于：
- 在软件应用组件与永久数据存储组件之间插入数字版权管理组件；
- 截取来自所述软件应用组件的访问所述永久数据存储组件的请求；
- 将数字版权管理组件数据与软件应用组件数据绑定在绑定数据集中；
- 将所述绑定数据集存储在所述永久数据存储组件中。

14. 如权利要求13所述的制品，其中，所述永久数据存储组件远程连接到所述数字版权管理组件。

15. 如权利要求13所述的制品，其中，利用以下处理中的一种或更多种来绑定所述绑定数据集：所述处理包括利用密码加密、去扰、隐写式隐藏。

16. 如权利要求13所述的制品，其中，利用以下处理中的一种或更多种来绑定所述绑定数据集：所述处理包括利用密码加密、去扰、隐写式隐藏。

17. 如权利要求13所述的制品，其中，所述绑定数据集被存储在所述制品中。

18. 如权利要求13所述的制品，其中，所述数字版权管理组件数据包括指示特定用户的信息，针对所述特定用户的数字版权管理组件数据的存储不影响先前针对不同用户存储的数字版权管理组件数据。

19. 如权利要求13所述的制品，其中，所述绑定数据集为以下类型
中的一种或更多种：所述类型包括数据块、流式数据。

20. 一种实施为包括数据的机器可存取介质的制品，当由机器访问时，使所述机器用于：
在软件应用组件与永久数据存储组件之间插入数权管理组件；
截取来自所述软件应用组件的访问所述永久数据存储组件的请求；
从所述永久数据存储组件检索绑定数据集；以及
从解绑绑定数据集中的数据权管理组件数据恢复软件应用组件数据。

21. 如权利要求20所述的制品，其中，所述永久数据存储组件远程连接到所述数据权管理组件。

22. 如权利要求20所述的制品，其中，利用以下处理中的一种或更多种来恢复所述绑定数据集：所述处理包括利用密码解密、去扰、暴露隐写式隐藏的数据。

23. 如权利要求21所述的制品，其中，利用以下处理中的一种或更多种来恢复所述绑定数据集：所述处理包括利用密码解密、去扰、暴露隐写式隐藏的数据。

24. 如权利要求20所述的制品，其中，所述解绑定数据集为以下类型中的一种或更多种：所述类型包括数据块、流式数据。

25. 如权利要求20所述的制品，还包括：从数权管理组件数据恢复软件应用组件数据，而不修改所述永久数据存储组件中的所述绑定数据集。

26. 一种系统，包括：
处理器；
永久数据存储组件，用于存储数权管理数据和软件应用组件数据；以及
数字权管理组件，用于在软件应用组件与永久数据存储组件之间插入数权管理组件，以截取来自所述软件应用组件的访问所述永久数据存储组件的请求，将数字权管理组件数据与软件应用组件数据绑定在绑定数据集中，以及将所述绑定数据集存储在所述永久数据存储组件中，其中所述数字权管理组件数据能够用于管理对所述软件应用组件数据的访问。
27. 如权利要求 26 所述的系统，其中，所述永久数据存储组件远程连接到所述数字版权管理组件。

28. 如权利要求 26 所述的系统，其中，利用以下处理中的一种或更多种来绑定所述绑定数据集：所述处理包括利用密码加密、加扰、隐写式隐藏。

29. 如权利要求 26 所述的系统，其中，所述数字版权管理组件数据包括指示特定用户的信息，针对所述特定用户的数字版权管理组件数据的存储不影响先前针对不同用户存储的数字版权管理组件数据。

30. 一种系统，包括：

处理器；

永久数据存储组件，用于存储数字版权管理数据和软件应用组件数据；以及

数字版权管理恢复组件，用于截取来自软件应用组件的访问所述永久数据存储组件的请求，从所述永久数据存储组件检索绑定数据集，以及从解绑定数据集中的数字版权管理组件数据恢复软件应用组件数据，其中所述数字版权管理组件数据能够用于管理对所述软件应用组件数据的访问。

31. 如权利要求 30 所述的系统，其中，利用以下处理中的一种或更多种来恢复所述绑定数据集：所述处理包括利用密码解密、去扰、暴露隐写式隐藏的数据。

32. 如权利要求 30 所述的系统，还包括：从数字版权管理组件数据恢复软件应用组件数据，而不修改所述永久数据存储组件中的所述绑定数据集。
将数字版权绑定到软件应用

背景介绍

1. 优先权声明

本 PCT 申请要求于 2007 年 1 月 29 日提交的题为 “COMPUTER-IMPLEMENTED METHOD AND SYSTEM FOR BINDING DIGITAL RIGHTS MANAGEMENT INFORMATION TO A SOFTWARE APPLICATION”的美国专利申请序列号 11/699,679 的申请日的权益，特此根据 35 U.S.C. § 120 或 365 (c) 要求该优先权，而该美国专利申请要求于 2006 年 10 月 9 日提交的欧洲专利申请 (EPO) 第 06121995.2 号的申请日的优先权权益，上述各个专利申请的全部内容通过引用结合于此。

1. 技术领域

本公开内容涉及数字版权管理方法和系统。更具体而言，本公开内容涉及将数字版权管理信息绑定到软件应用。

2. 相关技术

数字版权管理 (DRM) 方案需要在客户机上保存试用 (trial) 信息。此信息的用途是存储给定的受保护应用的当前试用状态，以便每次启动该应用时更新试用状态。此外，每次启动游戏时均检查试用准则，使得当试用结束时（例如，使用三次之后），该受保护应用可以终止。当试用准则终止该受保护应用时，再次使用该应用的唯一途径是付费订购或购买该应用的完整版本。常规方法是以传统方式将此试用数据保存为可由客户操作系统 (OS) 使用所提供的应用程序接口 (API) 进行保存和恢复的永久信息。常规系统中所使用的普通持续方法的常见示例包括：1）将数据保存到硬盘驱动器中的文件内，以及从存储在硬盘驱动器上的那些文件恢复所述信息；2）访问 OS 注册表以保存和恢复信息；或者 3）访问和修改该应用中的一些已知文件（或基本 OS 安装中的已知文件），因此，利用例如
隐写（steganographic）方法，可以以用户不易察觉的方式（例如，变更图像、音乐或视频中的最低几位）来保存和恢复信息。这些常规方法依赖于以下事实：在下一次执行该受保护应用时，所有保存的试用信息都将被恢复，因此可以方便地更新试用状态。常规方法的一个问题是，普通攻击通过收集 DRM 访问和修改的所有永久信息来重置 DRM 试用状态信息，使得该应用退出时，将所述信息恢复到执行该应用之前所保存的状态。这些对 DRM 的攻击通常试图避免对原始受保护应用所保存或更新的任何永久信息进行修改。以此方式，可以绕过 DRM 的有效性，并且可以无限制地使用该受保护应用。

因此，需要一种计算机实施的用于将数字版权管理信息绑定到软件应用的方法和系统。

附图说明

下文通过举例来描述本发明的实施例，并且本发明的实施例不限于附图中的各个图，在附图中：

图 1 图示了常规的受保护软件应用程序。
图 2 图示了黑客绕过 DRM 组件的有效性情况下图 1 所示的常规系统。
图 3 图示了常规的受保护软件应用程序的替换实施方式。
图 4 图示了示出包括软件应用部分和数字版权管理（DRM）部分的受保护软件应用程序的示例性实施例。
图 5 和 6 为图示了各种实施例中的处理步骤的流程图。
图 7 和 8 为本发明的实施例可运行和驻留于其中的计算系统的框图。

具体实施方式

本发明公开了一种计算机实施的用于将数字版权管理信息绑定到软件应用的方法和系统。在以下描述中，陈述了大量的具体细节。然而，应理解，可以在没有这些具体细节的情况下实施本发明的实施例。在其它情况下，未详细示出公知的过程、结构和技术，以便本说明书变得不清楚。
各种实施例包括用于将数字版权管理信息绑定到应用（主机软件）而不需要对所述应用进行代码改变的机制。各种实施例努力改善主要可执行软件（host executable）与DRM信息之间的绑定，同时保持不需要对主机进行源码级修改的益处。

在本文所述的各种实施例中，利用与受保护软件应用所使用的用以保存其自身数据的相同永久数据通道来保存永久DRM信息。以此方式，黑客极难将DRM信息与受保护应用软件信息分开。在下文将更为详细描述的各种实施例中，受保护软件应用进行操作系统（OS）调用或物理介质访问，以便通过DRM接入层来保存和检索数据。由于DRM试用状态信息和软件应用信息均通过同一DRM接入层来传递，因此不可能变更此类信息以仅去除DRM信息而不影响永久应用程序信息。为了保存DRM信息，重要的是使用与受保护软件应用所使用的相同输入/输出（I/O）数据通道，通过使用加密方法来存储和检索数据，从而安全地将DRM信息与软件应用程序数据绑定。下文更为详细地描述各种实施例。

图1图示了常规的受保护软件应用程序100，其包括应用部分104和数字版权管理（DRM）部分102。软件应用104表示用于销售或许可的任何常规的软件应用程序、软件游戏、商业或企业软件以及类似的市场软件产品。DRM部分102表示用于管理以各种受控或受限方式对应用104的访问的常规软件组件。例如，DRM组件102可用于通过常规方式向用户提供应用104的试用样本。在这种试用版本中，DRM组件102向用户提供对应用104的有限访问。例如，DRM组件102可以向用户提供对应用104的有限时长、有限使用次数的访问，或提供功能受限版本的应用104。在典型的常规配置中，用户通过DRM组件102对应用104进行访问。对于用户每次这样的访问，DRM组件102可将永久DRM信息存储在非易失性数据存储组件106中。类似地，对于应用软件104的执行，应用104可将永久软件应用数据存储在非易失性数据存储组件108中。本领域的普通技术人员将易于理解，可以比如硬盘驱动器、闪速存储器、磁性介质等常规存储设备实施数据存储组件106和108。当用户访问应用104时，DRM组件102将此访问记录在永久数据存储组件106中。一旦特定的使用样本到期或者当对应用软件104的许可证期满或到期时，DRM组件102可防止用户对应用104进行随后的访问。在替换实施例中，DRM组件102还可记录用户标识符、用户名、设备标识符、软件许可证/注册号等，使得永久DRM信息可以与特定用户和/或特定设备相关联。以此方式，各种实施例允许在不同的用户或设备之间共享文件或软件应用试用版。这将使
第一用户能够与第二用户共享第一用户所保存的游戏或其它软件应用，而不会使第一用户的游戏或其它软件应用的试用状态到期。类似地，反之，第二用户可与第一用户共享第二用户所保存的游戏或其它软件应用，而不会使第二用户的游戏或其它软件应用的试用状态到期。

现在参照图 2，其示出了在黑客绕过 DRM 组件 102 的有效性情况下，图 1 所示的系统。在该示例中，黑客攻击了存储在永久数据存储组件 106 中的 DRM 信息。如果存储在永久数据存储组件 106 中的 DRM 信息被去除或被不准确数据替代，则可绕过 DRM 组件 102 在保护对应用 104 的访问方面的有效性。以此方式，黑客可修改或去除永久数据存储组件 106 中的 DRM 信息，由此获得对应用 104 的无限制访问。因而，现有技术的 DRM 实施方式易于受到比如如上所述的攻击。

现在参照图 3，其示出了常规的受保护软件应用程序的替换实施方式。在图 3 的常规实施方式中，应用程序 107 通过 DRM 组件 103 将所有的输入/输出（I/O）经由 I/O 路径 105 引导到应用数据存储组件 108。在典型的常规配置中，用户通过 DRM 组件 103 进行对应应用 107 的访问。对于用户每次这样的访问，DRM 组件 103 可将永久 DRM 信息存储在非易失性数据存储组件 106 中。类似地，对于应用软件 107 的执行，应用 107 可将其自身的永久软件应用数据存储在非易失性数据存储组件 108 中。本领域的普通技术人员将易于理解，可以比如硬盘驱动器、快速存储器、磁性介质等常规存储设备实施数据存储组件 106 和 108。当用户访问应用 107 时，DRM 组件 103 将此访问记录在永久数据存储组件 106 中。然而，图 3 所示的常规实施方式仍易于受到黑客攻击。如果存储在永久数据存储组件 106 中的 DRM 信息被去除或被不准确数据替代，则可绕过 DRM 组件 103 在保护对应用 107 的访问方面的有效性。以此方式，黑客可修改或去除永久数据存储组件 106 中的 DRM 信息，由此获得对应用 107 的无限制访问。因而，现有技术的 DRM 实施方式易于受到比如如上所述的攻击。

现在参照图 4，其图示了一种解决现有技术系统中的问题的实施方式。在如图 4 所示的改进实施例中，图 4 图示了包括软件应用部分 114 和数字版权管理（DRM）部分 112 的受保护软件应用程序 110。软件应用 114 表示用于销售或许可的任何常规的软件应用程序、软件游戏、商业或企业软件以及类似的市售软件产品。DRM 部分 112 表示用于管理以各种受控或限制方式对应用 114 的访问的改进的数字版权管理软件组件。在图 4 所示的示例中，输入/输出（I/O）数据通道或数据路径 113 提供应用 114
向永久数据存储组件 116 传输应用信息的方法。在一个实施例中，DRM 112 在应用 114 与常规操作系统（OS）之间提供软件层。在其它实施例中，DRM 112 的组件可替代应用 114 的各种组件、系统驱动器或 OS 组件，以在应用 114 与 OS 之间提供软件层或在应用 114 与硬件之间直接提供软件层。在这些配置中，DRM 112 可截取应用 114 进行的对 OS 的任何函数调用。因为常规的操作系统函数调用是公知的接口，所以 DRM 112 可被配置为预计并截取应用 114 对 OS 的这些 I/O 函数调用。以此方式，无论何时应用 114 需要访问永久数据存储组件 116，应用 114 都能访问 DRM 112。在通过应用 114 来服务这些 I/O 请求的过程中，应用数据通过 DRM 112 以及经由数据路径 113 和 115 来传送到永久数据存储组件 116 或从永久数据存储组件 116 传送出出去。

为了其自身的保持永久 DRM 信息的目的，DRM 112 也经由数据路径 115 来访问永久数据存储组件 116。DRM 112 的这些访问可用于存储和检索与用户对应用 114 的受限制使用或试用样本相关的 DRM 信息。在这些情况下，DRM 信息也经由数据路径 115 传送到永久数据存储组件 116/从永久数据存储组件 116 传送出出去。因此，在正常运行中，所有的永久应用数据和永久 DRM 数据均经由数据路径 115 传送到永久数据存储组件 116/从永久数据存储组件 116 传送出出去。应用特定的信息经由数据路径 113 和 115 传送到应用 114 或从应用 114 传送出出去。

在图 4 所示的实施例中，永久 DRM 信息和永久应用 114 的信息在永久数据存储组件 116 中组合，并经由公共数据路径 115 进行传输。当应用数据或 DRM 数据将要保存在永久数据存储组件 116 中时，DRM 112 利用多种技术将应用数据和 DRM 数据绑定在一起。在一个实施例中，利用密码将应用数据和 DRM 数据组合并加密。在另一个实施例中，对应用数据和 DRM 数据进行组合和加扰、混合、哈希（hash）或隐写式隐藏，以创建组合的应用数据和 DRM 数据的绑定数据集，这种绑定数据集极其难解密或去扰。隐写术是写入隐藏消息，使得除预定接收者外无人知道该消息的存在。这种方法与加密术相反，在加密术中，消息本身的存在不被隐蔽，但内容被模糊。可利用公知的隐写技术来实现 DRM 数据与应用的隐写式隐藏。这些技术可用于创建组合的应用数据和 DRM 数据的绑定数据集。所述绑定数据集可以由数据块或一组流式数据。随后，将此绑定数据集写入永久数据存储组件 116。当应用 114 和/或 DRM 112 需要读取存储在永久数据存储组件 116 中的绑定数据集时，DRM 112 读取该绑定数据集，并在将解绑定的数据发送到应用 114 或在 DRM 112 内保留和使用解
绑定的数据之前绑定数据集进行解密或去扰。因而，图 4 所示的实例提出了一种黑客很难绕过的配置。因为在永久数据存储组件 116 中以极难解密或去扰的方式将应用特定的信息和永久 DRM 信息绑定在一起，所以黑客再也不能方便地仅去除 DRM 永久信息而不影响永久应用 114 的信息。因此，使得 DRM 112 和应用 114 更能抵御黑客攻击。

图 5 图示了各种实施例的一个示例中所采用的处理流程的流程图。在处理块 410 中，在软件应用组件与操作系统组件之间插入 DRM 组件。此插入的 DRM 组件在软件应用与操作系统之间创建软件层。当软件应用组件请求访问永久数据存储组件时，使用 DRM 组件来截取来自软件应用组件的访问永久数据存储组件的请求（处理块 412）。DRM 组件如上所述地将 DRM 数据与应用数据绑定在绑定数据集中（处理块 414）。最后，DRM 组件将绑定数据集（包括 DRM 数据和应用数据）存储在永久数据存储组件中（处理块 416）。以此方式，应用特定的信息和永久 DRM 信息一起绑定在永久数据存储组件 116 中。

图 6 图示了各种实施例的另一个示例中所采用的处理流程的流程图。在处理块 510 中，在软件应用组件与操作系统组件之间插入 DRM 组件。此插入的 DRM 组件在软件应用与操作系统之间创建软件层。当软件应用组件请求访问永久数据存储组件时，DRM 组件截取来自软件应用组件的访问永久数据存储组件的请求（处理块 512）。DRM 组件从永久数据存储组件中检索绑定数据集（包括 DRM 数据和应用数据）（处理块 514）。最后，DRM 组件如上所述地从应用数据恢复 DRM 数据以创建解绑定数据集（处理块 516）。本领域的普通技术人员将易于理解，可利用传输到易失性存储器并在其中进行处理的绑定数据集副本将 DRM 数据从应用数据解绑定。以此方式，在解绑定过程中不会修改永久数据存储组件中所保持的绑定数据集（包括 DRM 数据和应用数据）。永久数据存储组件中所保持的绑定数据集保持绑定，直到旧版本的绑定数据集被更新的版本重写。这防止了黑客访问存储在永久数据存储组件中的绑定数据集的解绑定版本。随后，可将恢复的应用数据发送到应用，并且 DRM 可使用恢复的 DRM 特定数据。以此方式，应用特定信息和永久 DRM 信息可在永久数据存储组件 116 中绑定在一起，并且随后在需要时由应用和/或 DRM 组件进行恢复。

图 7 和 8 图示了计算机系统 200 的一个示例，该示例图示了可实施本示例性实施例的特征的示例性客户机或服务器计算机系统。计算机系统
200 由用于传送信息的总线或其它通信装置 214 和 216 以及用于处理信息的处理装置比如与总线 214 耦合的处理器 220 组成。计算机系统 200 还包括耦合到总线 214 的、用于存储信息和将由处理器 220 执行的指令的随机存取存储器（RAM）或其它动态存储设备 222（通常称为主存储器）。主存储器 222 也可用于存储处理器 220 执行指令期间的临时变量或其它中间信息。计算机系统 200 还包括耦合到总线 214 的、用于存储静态信息和处理 220 的指令的只读存储器（ROM）和/或其它静态存储设备 224。

还可将可选数据存储设备 228 比如磁盘或光盘及其相应的驱动器耦合到计算机系统 200 以便存储信息和指令。计算机系统 200 还可经由总线 216 耦合到显示设备 204，比如阴极射线管（CRT）或液晶显示器（LCD），以便向计算机用户显示信息。例如，可在显示设备 204 上向用户呈现信息的图像、文本、视频或图形描绘。通常，将包括字母数字和其它按键的字母数字输入设备 208 耦合到总线 216，以便向处理器 220 传送信息和/或命令选择。另一类用户输入设备为光标控制设备 206，比如常规的鼠标、轨迹球或其它类型的光标方向键，用于向处理器 220 传送方向信息和命令选择，并用于控制显示 204 上的光标移动。

还可将通信设备 226 耦合到总线 216，以便经由例如因特网来访问远程计算机或服务器，比如网络服务器或其它服务器。通信设备 226 可包括调制解调器、网络接口卡或其它公知的接口设备，比如用于与以太网、令牌环网、无线网或其它类型的网络接口的接口设备。以此方式，在任何情况下，计算机系统 200 均可通过常规的网络基础设施耦合到多个服务器。

如上所述，一个示例性实施例的系统包括软件、信息处理硬件以及各种处理步骤。可以以机器或计算机可执行指令来实施示例性实施例的特征和处理步骤。所述指令可用于与以所述指令编程的通用或专用处理器执行示例性实施例的步骤。或者，可通过包含用于执行所述步骤的硬接线逻辑的特定硬件组件或通过经编程的计算机组件和定制硬件组件的任意组合来执行所述特征或步骤。尽管参照因特网描述了上述实施例，但是本文所述的方法和装置同样可应用于其它网络基础设施或其它数据通信系统。

应注意，不是必须以所描述的顺序或以任何特定的顺序来执行本文所述的方法。此外，可以以重复、同时、递归、串行或并行的方式来执行针对本文所确定的方法来描述的各种行为。可通过通信设备 226、以一个或更多个载波的形式来发送和接收包括参数、命令、操作数以及其它数据的信息。
在阅读和理解本公开的内容时，本领域的普通技术人员将了解可从基于计算机的系统中的计算机可读介质启动软件程序以执行上述软件程序中所定义的功能的方式。本领域的普通技术人员还将了解可用于创建用于实施和执行本文所公开方法的一个或更多个软件程序的各种编程语言。可利用面向对象的语言比如 Java、Smalltalk 或 C++来将所述程序构造成面向对象的格式。或者，可利用过程语言比如汇编语言或 C 语言来将所述程序构造成面向过程的格式。可利用本领域的普通技术人员所熟知的大量机制中的任何一种来传送软件组件，所述机制比如是应用程序接口或进程间通信技术，包括远程过程调用。各种实施例的教导不局限于任何特定的编程语言或环境，包括 HTML 和 XML。

因此，可实现其它实施例。例如，图 7 和 8 图示了根据各种实施例的制品的框图，比如计算机 200、存储系统 222、224 和 228、磁盘或光盘 212、一些其它存储设备 228 和/或任何类型的电子设备或系统。制品 200 可包括：耦合到计算机可读介质 212 的计算机 202 (具有一个或更多个处理器)；和/或具有相关信息（例如，计算机程序指令和/或数据）的存储设备 228 (例如，固定和/或可移动存储介质，包括具有导体、光导体或电磁导体的有形存储器) 或通过通信设备 226 的载波。当所述相关信息由计算机 202 执行时，使计算机 202 执行本文所描述的方法。

已描述了各种实施例。特别是，可以描述具有各种类型和格式的用户接口表示的实施例的使用。本领域的普通技术人员将易于理解，可采用本文所描述的实施方式的替换实施例，并且这些替换实施例仍落入在下文陈述的权利要求的形式内。在本文的详细描述中，各种实施例被描述为在计算机实施的处理逻辑中进行实施，所述处理逻辑在本文中有时被表示为“软件”。然而，如上所述，所要求保护的本发明并不局限于纯软件实施方式。

因此，本发明公开了一种计算机实施的用于将数字版权管理信息绑定到软件应用的方法和系统。尽管已根据多个示例性实施例描述了本发明，但是本领域的普通技术人员将认识到，本发明并不局限于所描述的实施例，而是可在所附权利要求的精神和范围内进行各种修改和变化。因此，本文的描述应被认为是说明性的而不是限制性的。
图2（现有技术）
图5

将数字版权管理信息绑定到软件应用的处理逻辑

- 400 -

在软件应用组件与永久数据存储组件之间插入DRM组件

- 410 -

使用DRM组件来截取来自软件应用组件的访问永久数据存储组件的请求

- 412 -

使用DRM组件将DRM数据与应用数据绑定在绑定数据集中

- 414 -

使用DRM组件将绑定数据集存储在永久数据存储组件中

- 416 -
将数字版权管理信息从软件应用解绑定的处理逻辑

500

在软件应用组件与永久数据存储组件之间插入DRM组件

510

使用DRM组件截取来自软件应用组件的访问永久数据存储组件的请求

512

使用DRM组件从永久数据存储组件中检索绑定数据集

514

使用DRM组件从解绑定数据集中的DRM数据恢复应用数据

516

退出

图6