[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
H04N 7/16 (2006.01)
H04N 5/00 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200680025692.9

[43] 公开日 2008 年 7 月 30 日

[11] 公开号 CN 101233755A

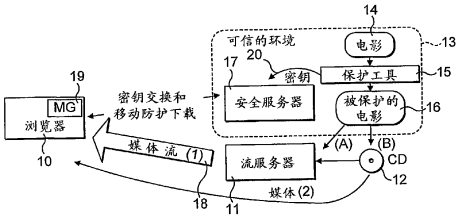
[22] 申请日 2006.7.14
[21] 申请号 200680025692.9
[30] 优先权
[32] 2005. 7.14 [33] GB [31] 0514492.8
[86] 国际申请 PCT/GB2006/002619 2006.7.14
[87] 国际公布 WO2007/007112 英 2007.1.18
[85] 进入国家阶段日期 2008.1.14
[71] 申请人 塞库斯特里姆技术公司
地址 挪威特隆赫姆
[72] 发明人 吉斯勒·格里曼 克里斯蒂安·芒什
[74] 专利代理机构 北京集佳知识产权代理有限公司
代理人 李德山 李春晖
权利要求书 5 页 说明书 19 页 附图 3 页

[54] 发明名称

多媒体数据保护

[57] 摘要

本发明提供一种用于将诸如电影之类的媒体作品传送给客户端的方法，包括步骤：(a)使用与文件的各个时间分隔的片段相对应的不同密钥的序列来对作品进行加密；(b)将包含算法的软件代码从安全服务器传送给客户端，算法具有作为客户端的状态的函数的结果；(c)在客户端执行代码，并将结果返回给安全服务器；(d)确定该结果是否指示是未修改的客户端；并进一步包括步骤：(e)将片段从服务器传送给客户端；(f)将与所传送的片段相对应的密钥从安全的远端服务器安全地流传输给客户端；(g)使用所获得的媒体密钥来对片段进行解密；(h)如果步骤(d)指示了修改的客户端，则阻止更多的密钥被传送，否则重复步骤(e)到(g)以及重复步骤(b)到(d)。



1. 一种用于将媒体作品传送给客户端的方法，包括步骤:

（a）使用与作品的各个时间分隔的片段相对应的不同密钥的序列对作品进行加密;

（b）将第一密钥从安全服务器安全地传送给客户端，并将相应的片段从服务器传送给客户端;

（c）在客户端中，使用第一密钥来对相应的片段进行解密;

（d）在浏览器中，呈现解密的部分；以及

（f）关于其他片段和密钥而重复步骤（b）到（d）。

2. 如权利要求 1 所述的方法，其中，各密钥在密码上彼此独立，使得没有一个密钥可以被用于解密超过一个片段。

3. 如权利要求 1 或 2 所述的方法，其中，仅在客户端被授权接收文件的检查之后才提供密钥。

4. 如权利要求 1、2 或 3 所述的方法，其中，密钥被用于加强安全服务器与客户端之间的合作。

5. 如任何前述权利要求所述的方法，其中，每个密钥与预定长度的片段相对应。

6. 如任何前述权利要求所述的方法，其中，安全服务器在客户端的远端。

7. 如任何前述权利要求所述的方法，其中，密钥是使用密钥交换协议来传输的，所述密钥交换协议使用随机数据发生器和安全服务器的公共密钥，所述公共密钥对于客户端来说是已知的。

8. 如任何前述权利要求所述的方法，其中，每个密钥必须由客户端来单独地请求。

9. 如任何前述权利要求所述的方法，进一步包括步骤：检查客户端的完整性以确保其没有被篡改。

10. 如权利要求 9 所述的方法，其中，安全服务器被安排成在检测到客户端修改的情况下和/或者如果客户端完整性检查没有成功，停止密钥的供应。

11. 如权利要求 9 或 10 所述的方法，其中，客户端的完整性由移动防护（如这里所限定的）来检查。

12. 如权利要求 11 所述的方法，其中，每个密钥仅在已经成功地校验了客户端完整性的移动防护的信任间隔期间被传送。

13. 如权利要求 9、10 或 11 所述的方法，包括使用随机产生的算法，所述随机产生的算法仅在客户端是未修改的情况下才返回正确的结果。

14. 如任何前述权利要求所述的方法，其中，片段独立于相应的密钥而被传送，使得即使是密钥的供应被停止的情况下，传送也可以继续进行。

15. 如任何前述权利要求所述的方法，其中，媒体作品是记录品。

16. 如权利要求 1 到 14 的任何一个所述的方法，其中，媒体作品是实况表演。

17. 如任何前述权利要求所述的方法，其中，媒体作品从远端服务器被流传输给客户端。

18. 一种将数据传送给客户端的方法，包括步骤：

（a）将数据传送给客户端；

（b）将包含算法的软件代码从安全服务器发送到客户端，该算法具有作为客户端的状态的函数的结果；

（c）在客户端执行代码，并将结果返回安全服务器；以及

（d）确定结果是否指示是未修改的浏览器。

19. 如权利要求 18 所述的方法，其中，数据是媒体作品。

20. 如权利要求 18 或 19 所述的方法，其中，直到步骤（d）已经被执行之后，客户端才可以使用所述数据。

21. 如权利要求 18、19 或 20 所述的方法，进一步包括步骤（e），在所述结果指示不是未修改的浏览器的情况下，停止传输数据和/或对数据进行解密所必需的密钥。

22. 如权利要求 18 到 21 的任何一个所述的方法，其中，代码包含校验和计算，客户端的程序代码和/或存储器映射被输入到该校验和计算中。

23. 如权利要求 22 所述的方法，其中，校验和计算包括将随机数输入到该校验和计算中。

24. 如权利要求 23 所述的方法，其中，客户端用安全服务器的公共密钥对随机数进行加密；加密的随机数和对媒体密钥的请求、以及计算的校验和一起被传送给安全服务器；安全服务器解密所述随机数并使用它对媒体密钥进行加密；安全服务器使用所述随机数更新自己的校验和的计算；然后，安全服务器比较校验和的这两个数值。

25. 如权利要求 24 所述的方法，其中，如果也只有在所述两个数值相等的情况下，加密的媒体密钥才能被传送给客户端以使得客户端可以对媒体密钥进行解密。

26. 如权利要求 25 所述的方法，其中，如果所述两个数值不相等，则确定客户端浏览器已经被篡改了。

27. 如权利要求 18 到 26 的任何一个所述的方法，其中，代码在浏览器上执行伪装任务。

28. 如权利要求 27 所述的方法，其中，伪装任务包括：将执行中的浏览器的存储器映射进行随机化。

29. 如权利要求 27 所述的方法，其中，伪装任务包括下述的一个或多个：代码重定位、代码多样化、数据重定位、以及数据隐藏，都如这里所限定的那样。

30. 如权利要求 18 到 29 的任何一个所述的方法，其中，所述移动防护被伪装。

31. 一种用于对执行中的浏览器进行伪装的方法，包括对执行中的浏览器的存储器映射进行随机化。

32. 一种用于将媒体作品传送给客户端的方法，包括步骤：

（a）使用与作品的各个时间分隔的片段相对应的不同密钥的序列来对作品进行加密；

（b）将包含算法的软件代码从安全服务器传送给客户端，算法具有作为客户端的状态的函数的结果；

（c）在客户端执行代码，并将结果返回给安全服务器；

（d）确定该结果是否指示是未修改的客户端；

（e）将片段从服务器传送给客户端；

（f）如果所述结果指示是未修改的客户端，则将与所传送的片段相

对应的密钥从安全的远端服务器安全地流传输给客户端;

（g）使用密钥对片段进行解密。

33. 如权利要求 32 所述的方法，其中，每一包含算法的软件代码具有相关联的信任间隔，并且在所述信任间隔期间多个密钥被流传输给客户端。

34. 如权利要求 32 或 33 所述的方法，进一步包括步骤（h），其中步骤（b）到（g）被重复。

35. 一种用于将媒体作品传送给客户端的方法，包括步骤:

（a）使用与文件的各个时间分隔的片段相对应的不同密钥的序列来对作品进行加密;

（b）将包含算法的软件代码从安全服务器传送给客户端，算法具有作为客户端的状态的函数的结果;

（c）在客户端执行代码，并将结果返回给安全服务器;

（d）确定该结果是否指示是未修改的客户端;

并进一步包括步骤:

（e）将片段从服务器传送给客户端;

（f）将与所传送的片段相对应的密钥从安全的远端服务器安全地流传输给客户端;

（g）使用所获得的媒体密钥来对片段进行解密;

（h）如果步骤（d）指示修改的客户端，则阻止更多的密钥被传送，否则重复步骤（e）到（g）。

36. 如权利要求 35 所述的方法，进一步包括步骤（i），其重复步骤（b）到（d）。

37. 一种用于检查客户端程序的完整性的方法，包括步骤：将软件代码从安全的源传送给正运行客户端程序的客户端计算机，软件代码包含依赖于客户端程序的状态的算法；执行软件代码；以及将结果返回给源，从而所述源能够确定客户端程序的完整性。

38. 如权利要求 37 所述的方法，其中，客户端程序被用于下列情况中的一个：互联网银行业、在线游戏、以及分布计算。

39. 一种被配置成根据任何前述权利要求的方法来操作的装置。

多媒体数据保护

技术领域

本发明涉及诸如电影、TV 展示、音频文件等的具有时间尺度的多媒体作品的安全分发。本发明尤其涉及用于以阻止用户获得作品的未授权拷贝的方式来安全地传递此类作品的系统。本发明的各方面还具有在诸如在线银行、游戏等的其他服务器-客户端情形中的应用。

背景技术

艺术作品的非法拷贝是一个长期的问题。在电影工业的早期，虽然进行胶片的未授权拷贝是可能的，然而这样做是很昂贵的，也不是很容易实现，除非那些人使用专业设备。随着家庭录像机的出现，电影和其他记录节目的新市场对制片商来说变得可以利用，同时，对那些记录品进行非法拷贝和分发也变得可能了。

如今，提供更高质量回放以及更方便和紧凑的数据承载体的 DVD 格式快速地取代了录像。另外，随着可承受的宽带互联网连接的出现，目前就已经出现了将电影和其他媒体从远端服务器下载或流传输到家庭计算机上的新兴市场。

媒体作品一旦被下载，其拷贝就被存储在计算机的硬盘驱动器上，并且与观看录像类似可以正常地被用户重复观看。流内容，不论是实况的还是记录的，在其被传送到计算机上时都可以（如同传统的 TV 节目一样）被几乎实时地观看（由于需要提供一些缓冲而存在短时间的延迟）。众所周知，一些电台和 TV 台用这种方式来提供他们的内容。

虽然这些技术上的进步允许媒体公司开发有前途的新市场，然而在防止作品的未授权拷贝的生产和分发方面也存在相应的问题。目前，谁都知道，即使是廉价的家庭计算机也具有将内容记录到 DVD 上的能力。

所以，以防止这种拷贝为目的技术已经产生。在传统的方法中，媒体提供商，这里是指“内容提供商”，拥有编码的媒体作品，例如电影，通常被称为是“媒体作品”。这些以不允许用户创建编码的媒体作品的拷贝的方式

而被分发和提供到用户的客户端程序/浏览器(viewer)。这种传递可以通过基于网络的流传输来实现，或者通过将物理介质，例如 DVD 传递给客户端来实现。

当作品基于网络传输时，其通常通过加密方式来保护，以便保护该作品不被第三方截取和拷贝。我们称这种加密方式为“传输加密”（加密与编码不同，是一种安全措施，从而，作品被转换并通常被压缩成可以容易且有效地传输的形式）。加密技术发展得非常好，并且基于计算机网络的通信安全可以以合适的方式来保护。

在媒体作品被传递到客户端之前，内容所有者使用加密方式来保护编码的媒体文件。在安全的“提供商环境”中，加密工具被使用，以便使用“媒体密钥”加密作品，从而创建加密的编码媒体作品，“加密的作品”。

其意图是，只有在客户端具有允许其对作品进行解密的媒体密钥的情况下才可以使用该作品。这可以被嵌入到客户端程序/浏览器/播放器和/或媒体中，例如 DVD 播放器和 DVD 中（客户端程序/浏览器/播放器可以是自立的设备或计算机上的浏览器软件程序）。

另一种选择，其被示意地说明在图 1 中，是根据要求从许可证服务器 1 得到媒体密钥。这允许媒体作品的流传输。为了支持这种模型，加密工具 2 将媒体密钥和附加信息一起打包在许可证 3 中并将其传送到许可证服务器 1。然后，客户端从流服务器 5 接收加密的编码媒体流 4，媒体流 4 在可以被呈现给客户端之前必须在浏览器 6 中被解密。为了观看加密的电影 7，浏览器从许可证服务器请求包含媒体密钥的许可证（见图 1 中的“启动阶段”）。

浏览器一旦接收到了许可证 3（和随之的媒体密钥），它就连接到流服务器 5，并从流服务器 5 接收加密的编码媒体流 4。浏览器使用媒体密钥来对加密的编码媒体流进行解密并将其呈现给客户端（见图 1 中的“流传输阶段”）。

上述情形中的主要问题是，浏览器是在由客户端控制的主机上执行的。因此，浏览器不是在可信的环境 8（电影 9 最初在那里被加密）中执行的。所以，存在客户端可以修改浏览器的风险。即使浏览器通常将只解密和解码部分媒体流，然而，在整个呈现处理期间，编码的媒体流的每一部分在某种程度上都将被呈现在浏览器存储器-映射(memory-image)中。另一种风险是，由于浏览器的存储器映射必须包含密钥，所以，用户可以提取媒体密钥，在这种情况下，他可以创建未加密的编码媒体的拷贝。

修改的问题存在于纯基于软件的浏览器中，也存在于基于硬件的浏览器中，例如指定的 DVD 播放器中。尽管修改基于硬件的浏览器比修改基于软件的浏览器要更加困难，但也不是不可能。所以需要能够解决这些缺陷的系统。

对任何有效的保护机制的总体要求包括如下：对于破解（break）,它应当是资源密集型的，使得破解成本至少与作品成本的量值相当。任何成功的攻击都不应当是可归纳的，所以它还可以被应用在别处。更好地，它还应当有助于检测。下述描述的本发明的各个方面分别致力于这些要求，本发明的优选形式提供了一种能够满足所有上述要求的系统。

发明内容

在下面的讨论中，媒体作品是具有时间特征的作品，即，它包含许多呈现步骤，这些步骤必需以合适的次序来执行。这些步骤通常在计算上是彼此独立的，并且可以被独立地处理。在大多数情况下，完全呈现要用大量的时间，在一部电影的情况下是很多分钟或几小时。

根据本发明的一个方面，提供了一种将媒体作品传送到客户端的方法，包括步骤：

（a）使用与各个在时间上分隔的作品片段相对应的不同密钥的序列来对作品进行加密，

（b）将第一密钥从安全的服务器安全地传送到客户端，并将相应的片段从服务器传送到客户端，

（c）在客户端中，使用第一密钥来解密相应的片段，

（d）在浏览器中，呈现解密的部分，

（e）关于更多的片段和密钥而重复步骤（a）~（d）。

本发明可以被应用到具有时间特征的任何类型的媒体作品（如上面定义的），对于分发电影，例如基于互联网进行流传输来说尤其有用。

通过将文件分割成片段序列，使得拷贝超过一小部分也变得不现实了，这是因为每个密钥只能解密一个片段，即，各密钥在功能上是独立的。因此，在一段时间只能拷贝电影的一个片段。此外，不应当存在能够解锁其他密钥的主密钥，即，各密钥在结构上最好应当是独立的。最好是，为典型长度的电影使用成千上万的不同密钥，使得每一个密钥与预定长度的片段相对应，

例如，仅与几秒钟的相对应，假定少于两或三秒钟，最好是一秒钟或更短。大部分类型的媒体作品仅在其基本完整的情况下才有重要的价值。例如，即便仅最后几分钟缺失的电影也通常具有很小的价值。因此，想要非法拷贝电影的人必须解密每一个片段。

为了维持数据解密的连续流，在一些实施例中，客户端可以请求当前的密钥和接下来的密钥，并在存储器中缓存小数量（例如2、3、4等）的密钥。

通常，安全的服务器在浏览器的远端，此处被称为“安全服务器”。

电影通常在可信的环境中被加密。最好是，在加密期间产生的密钥被提供给安全服务器，这是在可信的提供商环境中进行的。尽管密钥随后从安全服务器被传送给了浏览器，然而，电影或其他作品可以从其他地方传送。例如，可以从可信的环境之外的单独的服务器流传输。因此，在一个优选的实施例中，一旦电影已经在可信的提供商环境中被加密了，随后它就被提供给非安全的流服务器。

因此，在这种布置中，客户端可以是运行在远端计算机（例如用户的PC）上的软件浏览器程序，与安全服务器进行通信来接收密钥（被称为媒体密钥），并与单独的流服务器进行通信。

媒体密钥最好跟随来自客户端的请求而被传送给客户端，并且这最好使用密钥交换协议来进行，所述密钥交换协议使用随机数据发生器以及对浏览器来说是已知的安全服务器的公共密钥。

在一种实现中，当需要获得下一个媒体密钥时，客户端就产生随机数据，并使用安全服务器的公共密钥对其进行加密。然后，加密的数据可以被包括在下一个媒体密钥的请求中，最好是与用以识别客户端的数据一起被包括在下一个媒体密钥的请求中，传送给安全服务器。一旦接收到该请求，安全服务器就检查客户端是否被授权来接收媒体作品，解密和提取随机数据并使用它和请求的密钥来执行一函数(function)以便使用随机数据来加密密钥。在一个实施例中，它们可以被“异或”(XOR)运算。然后结果被传送回客户端。一旦客户端接收到结果，就可以通过执行相应的函数而从结果中提取所请求的密钥，例如，通过将所述结果与初始密钥请求中提供的相同的随机数据进行“异或”运算来进行。用这种方式，加密的编码媒体流就可以被解密，而不用在浏览器的源代码中隐藏任何秘密的密钥。

最好是，公共密钥是被包括在校验和计算中，以便防止公共密钥被调换的“中间人(man in the middle)”攻击。

在优选的协议形式中，还采取步骤来确保由移动防护检查的客户端与产生随机数据的客户端是相同的一个。这可以通过将输入扩展到校验和以便包括用于请求媒体密钥的随机数据来进行。因此，到校验和的输入可以包括：来自客户端的代码、安全服务器的公共密钥、以及与密钥请求一起传送的随机数据。

由于外部熵源可以被监视，所以用于产生随机数的熵源可以是由执行环境自身以任务如何被制定和中断的形式而产生的熵源。因此，随机产生过程可以包括：创建几个线程（thread）和执行移动防护，其中所述几个线程工作于不同的计算任务，所述不同的计算任务可以和来自浏览器的当前状态的数据一起被输入到安全哈希（hash）算法。

对于需要客户端接收密钥的连续序列可以被用于加强用户合作。因此，如果提供商请求的一定步骤没有被客户端执行，那么，密钥的提供可以被停止。如下面将要进一步讨论的那样，该步骤可以是客户端的完整性检查，并且最好是，仅在所谓的“移动防护”指示浏览器是未修改的时候，对新密钥的请求才被响应。

当使用移动防护时，可以产生在上述优选密钥交换协议中使用的随机数，而不是实际的客户端浏览器/播放器程序中所使用的。

可以理解，这种合作的加强是可能的，这是因为内容提供商控制着媒体作品，并且由于作品的时间特性，作品可以以小部分的方式提供给被请求合作的用户，以便接收随后的各部分。这比起许可证解锁整个文件的传统系统有效地利用了时间属性。

在本发明的一个可选实施例中，可信的环境被扩展使得流服务器被包括在其中。当这样做时，可以使流服务器产生媒体密钥和加密运行中的媒体流。这确保每个媒体流被使用唯一的媒体密钥组来加密。那意味着，泄漏的媒体密钥不能被用于对相同电影的不同拷贝进行解密。为了有助于媒体密钥的分发，流服务器将其传送到安全服务器，安全服务器将如上所述地将其分发给浏览器。下载侧是需要被信任的另一个实体，并且，下载侧运行中的加密是计算昂贵的。因此，在此需要在一方面非常高的安全性与另一方面可信的环境的复杂性和计算成本之间折中。

本发明不限于文件从远端服务器流传输的布置。因为文件被加密了，所以文件能够以任何方便的方式来分发。因此，加密的文件可以从本地服务器提供到客户端，或提供到物理介质上（例如 DVD）。然后，文件可以从本地

服务器或物理介质传送到浏览器，并以如前述的相同的方式来解密。

尽管这种布置比先前技术的系统提供了显著的改进，然而，浏览器可以被篡改的风险依然存在，使得解密的作品（电影等）可以被记录和拷贝。因此，最好是，本发明还包括用于检查浏览器的完整性的设备，以确保其不被篡改。这可以通过编程以规则间隔和/或在请求密钥时传送诸如校验和的信号到安全服务器来实现。这种信号将被设计成依赖于浏览器的状态，使得对浏览器的任何修改都将改变所述信号。

然而，存在一种风险，即，这种措施可以通过对修改的浏览器进行编程而不顾其真实状态来传送“正确”信号而被破坏。因此，最好是，该方法进一步要求安全服务器随时间的变化使用许多不同的测试来询问浏览器。在一种具体优选的形式中，各测试包括使用随机产生的算法，仅在浏览器未修改时返回正确的结果。此外，未能响应或响应中不适当的延迟可以被认为指示浏览器修改。

因此，最好是，安全服务器被安排成：在检测到浏览器修改的情况下和/或如果这种浏览器完整性检查不成功，则停止密钥供应。

最优选的安排是将算法以软件代码（例如机器代码）的形式由安全服务器传送到客户端。软件可以被命名为“移动防护”并在下文中进一步描述。

这种完整性检查的系统被认为其本身是创造性的，因此从其他方面来看，提供了一种将数据传送给客户端的方法，包括步骤：

（a）将数据传送给客户端，

（b）将包含算法的软件代码从安全服务器传送到客户端，该算法具有作为客户端的状态的函数的结果，

（c）在客户端执行代码，

（d）将结果返回到安全服务器，以及

（e）确定结果是否指示是未修改的浏览器。

数据可以是例如基于互联网流传输到客户端的媒体作品，或数据可以从如上所述的本地服务器、DVD 或其他媒体提供。然而，如将要在下面更全面地讨论的那样，所述数据可以是能够在服务器和客户端之间传送的任何类型的数据。客户端可以是运行在计算机上的程序或诸如 TV 机顶盒之类的硬件设备。步骤（b）中所指的算法可以在文件的任何部分已经被传送之前来传送，或者，整个或部分作品可以在算法之前被传送。最好是，直到步骤（d）

已经被执行之后，文件才可以被观看。

依赖于步骤（d）中的结果，可以采取适当的动作。在作品正被流传输时，如果发现浏览器是未修改的，那么作品的传送以及解密该作品所需的任何密钥的传送都将正常地被允许继续进行。然而，还可以包括步骤（e），在结果指示不是未修改的客户端的情况下，停止作品的传输和/或解密该作品所需的密钥的传送。优选地，如果从移动防护没有返回结果，那么也认为指示客户端已经被修改了。

当作品正在从诸如本地服务器、DVD 等的本地源传送时，其动作可以是停止传送解密文件所需的密钥。

可选地，如果发现客户端已经被修改，那么可以采取其他动作。例如，传输可以被允许继续进行，并搜集证据来识别用户。如果想要采取法律或调查行动，例如为了检测犯罪活动或为了防止将来对文件的非法拷贝，这可能是合适的。

如上面所指出的，响应于修改的客户端的识别所采取的动作可以是停止解密密钥的传送。因此，可以理解，本方法可以进一步包括将作品分割成多个在时间上分隔的片段，使用不同的密钥来对所述片段进行加密。这些密钥可以被顺序地分发给客户端，最好是如上所述地进行。因此，如果它们的分发被停止，那么作品的剩余部分就不能被解密。

本方法最好是使用以软件代码形式（如上面所提及的那样）随机产生的秘密算法来执行。这些所谓的加强算法产生依赖于客户端（例如浏览器程序）的状态的结果，但是由于随机特点，因此对用户来说正确结果是不可猜的。最好是，它们包含校验和计算，浏览器程序的代码被输入到所述校验和计算中。尽管算法作为一个整体是秘密的，然而校验和计算可以是已知的一种，例如消息摘要算法 5（MD5）（RFC1321，www.faqs.org/rfcs/rfc1321.html），其可以结合随机化的输入修改来使用。

输入修改是指修改器（modifier）的随机创建，所述修改器对将要被输入到校验和的数据进行排列。在一种实现中，当软件代码（这里被称为“移动防护”）被产生时，随机序列就被确定。当算法被执行时，来自浏览器的输入代码就被分割成相同尺寸的 n 块。然后，这些块被混编在上面提及的随机序列中，并且将结果输入到校验和算法中。尽管在这种安排中校验和算法本身是公开的，然而其结果是 n 块被输入到其中的次序（order）的函数。对安全服务器来说该次序是已知的，所以安全服务器可以确定返回来的结果

是否指示为完整的浏览器。

用来创建输入-过滤的校验和的可选的方法是：分解已知的校验和算法，然后以按照给定序列读取输入的方式来将其重新组合。

取代使用输入过滤，可以通过擦除（scratch）来产生校验和函数。因此，输入可以被分裂成 l 字（32 位），并且创建函数 f，其从输入读取 l 字，从输出字的变量区读取 m 字。该函数可以包括随机数目的任务，所述任务被接连地执行，校验和可以是 f 的应用的所有结果模 2^{32} 的和。

组合（compose）函数具有的优点是：几乎所有的校验和算法的代码被随机地创建，导致代码更多的结构多样性。由于构造块很小，所以允许与其他算法进行更容易的交织。

最好是软件代码还包含附加的算法，可以是秘密的或非秘密的。它们最好是功能上和/或空间上牵连着秘密算法。以这种方式，客户端的计算机/浏览器可以被强迫执行附加的算法，这是因为如果不这样做，那么秘密算法将不会被执行。例如，附加算法可以被用于检查浏览器硬件的完整性。

由于移动防护与浏览器存在于相同的环境，所以它潜在地容易被攻击。用户可能试图修改它，以便攻破它所执行的保护方法。如上面所讨论的那样，通过确保移动防护被部分地随机创建，就可以防止对其进行自动攻击。另外，伪装（obfuscate）变形可以被应用于移动防护。移动防护可以将校验和隐藏在不透明的数据结构中，所述不透明的数据结构以专属于移动防护的方式与校验和交织在一起。各变量可以随机地位于移动防护的存储器中，另外，移动防护的指令可以被分裂成块，也随机地位于存储器中。这最好包括到移动防护中的入口点。事实上，用于一个移动防护的入口点可以通过前一个来提供。

如果这些步骤被执行，那么在可以开始任何自动攻击之前，需要人的攻击来战胜这些伪装。该途径不可避免地消耗了大量的时间，由此假定在连续的移动防护之间的“信任间隔”足够短，则该途径不会有效。换句话说，由于移动防护被频繁地替换，所以没有足够的时间来做这种事情。因此，伪装处理保护了移动防护，使其在被另一个移动防护替换之前的时间间隔中免受攻击。

存在观察器窥探计算机中存储解密电影数据的存储器位置的风险。如果已知的存储器位置被使用，那么数据就可能被拷贝。所以，通过识别一定存储器位置（基于位置的识别）来定位代码而可实行的方法是不理想的，最好

是一旦各位置已经被使用，它们不应当被重新使用。而且，基于模式的识别最好也被阻止，在所述基于模式的识别中，代码可以通过寻找类似 MPEG 头之类的序列来寻找。

所以最好是通过移动防护来保护浏览器，使其状态不能通过窥探来确定。为此，移动防护最好还包括一个或更多个保护算法来保护免受此类攻击。可以在客户端上（例如在浏览器程序上）执行伪装任务，这在下面被称为“运行时间浏览器伪装”，即，当浏览器运行时在浏览器上执行迷惑。这改变了运行中的浏览器的存储器映射。

这种运行时间浏览器伪装被认为是另一个创造性概念，因此根据另一方面，本发明提供了一种对执行中的浏览器进行伪装的方法，包括将执行中的浏览器的存储器映射进行随机化。

运行时间伪装可以包括一个或更多个下述的技术。

代码重定位包括在存储器内移动代码块。在程序运行时，移动防护将把代码移动到存储器的其他部分，然后它们将在稍后被执行。这种算法最好是与校验和计算紧密地交织在一起。

最好是通过如下步骤来执行代码重定位，即：（1）识别程序中的所有基本构造块，并将其分割成小的可重定位的片段；（2）在移动防护的执行期间，这些片段可以被安排到存储器中各随机的位置；以及（3）修改所有跳转指令，以便与各新的代码位置相对应。结果，在移动防护执行期间，攻击者将面临变化的存储器映射。由于各片断的位置是通过安全服务器提供的移动防护来确定的，所以这对于攻击者来说是不可预测的，于是攻击者将无法依赖于特定存储器位置包含特定数据的假定。

数据重定位包括移动数据和改变用于访问数据的指令。而且，新位置可以被随机地确定。

数据隐藏致力于解决基于模式的识别和定位问题。一种途径是应用双路（two-way）功能来改变数据的外观，有效地将其进行掩饰。最好是使用简单的一次垫衬（one-time pad）方法。这可以包括新创建的模函数，所述模函数创建一个到随机数据阵列的索引。随机数据可以被用于通过在随机和敏感部分之间应用“异或”运算子来改变敏感数据。最好是被应用于这些和部分地址的敏感数据之间。

一种途径是对数据进行加扰（掩码）和解扰，使得敏感数据被存储成加扰的形式，在需要时解扰，然后重加扰或删除。然而，这在数据被解扰时留

下一个短暂的窗口。

然而，可以利用流处理的优点来延迟解扰，直到数据处于处理器的寄存处中。

因此在需要新数据时，实际内容解码器可以被修改，以便执行最后的解密操作。这意味着，在主存储器中将根本不存在任何解密的数据。可以使用下述步骤来提供:

a）移动防护根据需要修改解码器来执行最后的解密步骤;

b）获得下一个加密片段;

c）获得用于加密片段的媒体密钥;

d）根据解码器被修改的情况产生解密流，并将解密流置于存储器的随机位置中;

e）然后，解码器将根据需要每次读取一字节或一字并将其进行解密。

代码多样化包括在执行期间由移动防护对客户端程序所执行的操作。所执行的各操作改变代码，使其包含不同的指令而不改变其语义。这是为了防止基于模式的识别。可以执行下述步骤的一个或更多个。

可以插入上下文(context)独立的指令。这些指令的输入上下文可以与程序中的上下文共享，但这些指令的输出上下文却不同于程序中的任何输入上下文。由于它们不能改变程序的任何输入上下文，所以无需关心它们在处理什么。

上下文依赖的指令可以被执行相同功能的指令来代替。可以理解，这是更难以实现的，但也是更有效的，这是因为它们不能通过数据流分析来识别。

可以进行的功能独立的改变包括：改变各指令的执行次序；与时间变量一起或不与时间变量一起插入各指令；在存储器中对各指令进行重排序；以及进行控制流程变化。

功能依赖的改变需要注意以保持功能和副作用(side effect)的完整。它们包括：用功能等价物来替换各指令；引入同一性功能；引入运算子，使得文字值被某些指令替换，所述某些指令对所述值任意地进行初始化并执行改正所述值以匹配最初文字的运算。而且，可以引入变量，使得对目标的拷贝被替换成新创建的变量的拷贝。

在一个使用了基于硬件的浏览器方案的实施例中，例如 TV 机顶盒，浏

览器的发行商不但控制浏览器软件，还控制浏览器环境，即，硬件和操作系统。所以，基于硬件的浏览器与基于纯软件的方案相比，通常能够以完全得多的方式被移动防护来检查。在该实施例中，移动防护中的校验和算法不限于仅检查浏览器软件，而是还可以检查操作系统和硬件的不同方面。

因此，与基于硬件的浏览器一起可以两种方式来使用本系统。第一，可以被用于替代基于昂贵的防攻击硬件的方案。第二，可以提供在防攻击硬件无力的情况下而起作用的额外的安全措施。

可以看出，本发明优选地涉及单独加密的文件片断的组合以及“移动防护”概念的使用。因此，从更进一步的方面来看，本发明提供了一种用于将媒体作品传送给客户端的方法，包括步骤:

（a）使用与作品的各自时间分隔的片段相对应的不同密钥的序列来对作品进行加密，

（b）将包含算法的软件代码从安全服务器传送给客户端，算法具有作为客户端的状态的函数的结果，

（c）在客户端执行代码，并将结果返回给安全服务器，

（d）确定该结果是否指示是未修改的浏览器，

（e）将片段从服务器传送给浏览器，

（f）如果结果指示是未修改的浏览器，将与所传送的片段相对应的密钥从安全的远端服务器安全地流传输给浏览器，

（g）使用密钥来对片段进行解密。

可以理解，尽管各步骤能够以上面给出的次序来执行，然而，至少一些步骤能够以不同的次序来执行或同时执行。例如，步骤（e）可以与步骤（b）、（c）、（d）或（f）同时执行，使得各片段可以在密钥之前、与密钥一起或在密钥之后被传输。然而，在片段被解密之前，密钥必须是可获得的。

在一个实施例中，本方法包括另一个步骤（h），其中，步骤（b）到（g）被重复。

然而，通常所传送的软件代码具有一定的“寿命”或“信任间隔”，例如，少于 30 秒钟。另一方面，与软件代码的寿命相比，各片段通常被传送得更频繁，例如，每秒一片段。如此，就不需要在每次传送片段时都传送新的软件代码，而是通常只需要在当前软件代码的寿命已经到期时才传送一次。因此，通常步骤（e）到（g）将被重复，直到新的软件代码被请求为止，

那时步骤（b）将被重复。以此方式，一片软件代码（移动防护）就保护了很多密钥的传递。

尽管对于每片软件代码来说，代码的执行以及确定结果是否指示是未修改的浏览器（步骤 c 和 d）可以被执行超过一次，然而通常在软件代码的寿命期间只需要执行一次。如此，通常仅在步骤（b）已经被重复之后才重复步骤（c）和（d）。

从再一个方面来看，本发明提供一种用于将媒体作品传送给客户端的方法，包括步骤:

（a）使用与作品的各自时间分隔的片段相对应的不同密钥的序列来对作品进行加密，

（b）将包含算法的软件代码从安全服务器传送给客户端，算法具有作为客户端的状态的函数的结果，

（c）在客户端执行代码，并将结果返回给安全服务器，

（d）确定该结果是否指示是未修改的浏览器，

并进一步包括步骤:

（e）将片段从服务器传送给浏览器，

（f)将与所传送的片段相对应的密钥从安全的远端服务器安全地流传输给浏览器，

（g）使用所获得的媒体密钥来对片段进行解密，

（h）如果步骤（d）指示了修改的浏览器，就阻止更多的密钥被传送，否则就重复步骤（e）到（g)。

最好是，本方法进一步包括重复步骤（b）到（d）的步骤（i)。

可以理解，尽管各步骤能够以上面给出的次序来执行，然而至少一些步骤能够以不同的次序来执行或同时执行。事实上，一些步骤可以比其他步骤执行更多次。

步骤（b）到（d）可以独立于步骤（e）到（h）而被执行，最好是与之同时执行。如前面所提及的那样，软件代码通常具有容许传送很多片段和密钥的寿命。如此，步骤（b）到（d）的重复（在步骤（i）中被提及）通常没有步骤（e）到（g）的重复（在步骤（h）中被提及）执行得频繁。最好是，步骤（i）只有在软件代码的寿命已经到期的情况下才被执行。

本发明还扩展到被配置成如上所述运行的装置，包括被配置来接收流传输的媒体的客户端以及服务器布置，二者组合地或单独地提供。因此，根据另一个方面，可以提供一种用于将媒体作品传递给客户端的系统，包括:

（a）用于将作品传送给客户端的装置，

（b）用于将包含算法的软件代码从安全服务器传送给客户端的装置，算法具有作为客户端的状态的函数的结果，以及

（c）与安全服务器相关联的装置，用于接收结果并确定该结果是否指示是未修改的浏览器。

另一方面提供了一种客户端，例如，用于播放诸如电影之类的作品的浏览器，该客户端被安排成：从远端源接收作品并接收包含算法的软件代码；在客户端上执行算法；以及将算法的结果返回给远端源，从而向远端源说明客户端的完整性并启动文件的播放。

客户端最好使用由远端源提供给它的密钥，通过对作品进行解密、或对作品的片段进行解密来启动作品的播放。最好是，客户端被配置来请求密钥的序列，并使用序列形式的密钥来对作品的连续部分进行解密，然后被作为连续的呈现来播放。最好是，如上面所讨论的那样，密钥的供应依赖于客户端向源说明其完整性。

本发明还扩展到用于如上所述传递作品的系统与客户端的组合，从而文件被传递到客户端，并且只有在浏览器向源说明其完整性的情况下，文件才可以被播放。

可以看出，与先前技术的软件方案相比，本发明不依赖于使用户可获得的、被包含在数据中的秘密，不管是在程序代码中还是在媒体文件中。本发明允许拷贝企图的早期检测，并允许内容提供商在媒体文件的实质部分可以被拷贝之前启动计数测量。

还可以看出，使用移动防护来检查系统完整性的概念具有超越到客户端的文件传输（如所限定的那样）的其他应用。一般地可以被用于校验运行在对输入数据执行计算的未控制环境中的代码的完整性和真实性。也可以被用于防止某一方改变数据处理的方式而不会被检测到。因此，上面关于媒体浏览器的讨论可以被应用到任何客户端程序。应用包括游戏、银行业、音频等。

因此，从进一步的方面来看，本发明包括：将软件代码（诸如移动防护之类）从安全的源传送到运行客户端程序的客户端计算机，软件代码包括具有依赖于客户端程序的状态的结果的算法；执行软件代码并将结果返回给

源，从而源可以确定客户端程序的完整性。本发明还扩展成装置，所述装置被安排成根据这种方法来运行。

本发明的这一方面可以使用上面讨论过的任何或全部优选的特征，尤其是关于移动防护的特征。对媒体作品的上述引用同样适用于在服务器与客户端之间传送的时间有效载荷数据。因此，服务提供商能够以相同的方式来加强用户的客户端的合作，并且如果检测到合作停止或篡改就可以停止进一步的有效载荷数据。

因此可以理解，与服务器通信的任何客户端可以使其完整性以连续的方式被检查。因此，本发明允许在未控制的环境中运行的客户端能够被信任。如果得知客户端的完整性已经失效，就会采取行动。例如，与客户端的通信可以被中止，解密密钥的供应被暂停（如对待上面讨论过的媒体流传输应用那样）和/或采取步骤来搜集证据（例如，在针对银行业系统的可疑欺诈攻击的情况下）。

本发明在分布计算的环境中是有用的，在那种环境中尽管通常机密性和欺诈并不重要，然而软件的正确执行却很重要。因此，移动防护可以被用来保护以免客户端的故意的或非故意的修改，如果需要，既可以保护软件，也可以保护硬件。因此，进行分布计算作业的实例可以使用移动防护来检查执行计算的、远端节点处的客户端的正确操作。

在在线游戏的环境中，客户端程序的修改可以使欺骗成为可能，如果失控，就会引起客户不满和导致收入损失。因为所涉及的数据不是机密，并且记录也没有意义（如对待媒体作品那样），所以通常仅校验客户端软件的完整性就足够了。其中，游戏是基于客户端-服务器方式来运行的，移动防护可以如前面所讨论的那样来被应用。如果用户不允许与移动防护进行合作，那么就可以拒绝升级整体游戏状态。

在家庭银行业的情形中，移动防护可以被用于确保第三方不能访问机密数据。尽管正常用户通常不会对修改他的客户端程序感兴趣，然而他可能会成为中间人攻击的牺牲品。所以，银行业服务器可以使用移动防护来校验家庭银行业客户端的完整性和真实性，并且可以包含银行业服务器的公共密钥。该公共密钥被用于对从家庭银行业客户端到银行业服务器传递的所有数据进行加密，并且由于移动防护的完整性是有保证的，所以用户可以确保他的数据是保持机密的。

本发明可扩展成装置，所述装置被配置来使用上述的方法，本发明也可

扩展成包含指令的软件产品，使得计算机以此方式来运行。本发明还可以扩展成服务器-客户端组合和/或网络，其中，数据根据上面讨论的本发明的各方面而被提供给客户端。

附图说明

现在，仅仅通过例子参考附图来描述本发明的某些实施例，其中：

图 1 是如上所述的先前技术的媒体流系统的示意图；

图 2 是本发明的第一实施例的示意总图；

图 3 是示出了如实施例中所使用的、随机产生校验和算法的组件的示意图；

图 4 是示出了实施例的操作的流程图；以及

图 5 是实施例中使用的服务器算法的流程图。

具体实施方式

如可以从图 2 中看出的那样，客户端装备有浏览器 10，浏览器 10 可以用于观看媒体（例如电影），所述媒体是从流服务器 11 或可选地从本地存储媒体例如 CD 12 流传输的。系统的这些组件的每一个都在可信任的环境 13 之外。在可信任的环境之内的是：未加密的电影 14；保护工具 15，用于产生被保护的电影 16；以及安全服务器 17。

如图 1 中所说明的先前技术的系统那样，内容所有者在将编码媒体文件传递给客户端之前保护编码媒体文件。然而，不是使用单个媒体密钥，保护工具 15 用大量（成千上万）的媒体密钥 20 来对电影进行加密。这种过程产生加密的、编码的媒体，即，被保护的电影 16。

在媒体资源的呈现期间，媒体密钥 20 被分发，使得媒体资源被及时展开，如下面将要被描述的那样，媒体资源按照请求间隔每次一块地被安全流传输给客户端。媒体本身单独地被流传输。每个密钥仅包含很少字节（大约 16 个），所以用于流传输密钥所需的资源产生很少的开销。

由于每个密钥仅可以被用于解密大约一秒钟、或最多几秒钟的电影，所以仅仅获得单个密钥是没有什么价值的。

在本发明的第一实施例中，被保护的电影以数据流的形式经由路径 A、

流服务器 11 和媒体流 18 而被传递给客户端。在其他实施例中，使用了有形的媒体，例如 CD 或 DVD 12。

浏览器 10 在客户端的主机上被执行，并被安排成经由媒体流 18 从流服务器 11（或者，在其他实施例中是从 CD/DVD）接收被保护的电影 16。在呈现处理过程期间，浏览器 10 与安全服务器 17 进行通信，以便下载必要的媒体密钥 20 来对被保护的电影 16 进行解密。

另外，浏览器 10 还以大约 30 秒的规则间隔来下载被称为移动防护 19 的代码片。这些代码片的每一片的内部都嵌有在安全服务器 17 中创建的算法形式的秘密信息。这些算法的执行对于流传输的数据 18 的使用来说是必要的。当每一个移动防护 19 被传递到浏览器中时，执行由秘密算法确定的计算，并将结果返回给安全服务器。以如下方式构造移动防护，即，只有浏览器没有被破坏时，计算的结果才是正确的。秘密算法的结果包含有校验和，向安全服务器证明浏览器的完整性。

移动防护还可以具有在功能上和空间上牵连着秘密算法的其他附加算法。以这种方式，客户端的计算机/浏览器可以被强迫执行附加算法，这是因为如果不这样做，那么秘密算法将不会被执行。以此方式，浏览器可以全面地被检查。

如果由移动防护返回给安全服务器 17 的结果与期望的结果不相匹配，那么安全服务器就停止向浏览器分发媒体密钥 20。如果浏览器 10 拒绝移动防护，或者如果正确结果没有在一定的时间之内到达，那么安全服务器也停止向浏览器分发媒体密钥 20。密钥交换协议将在稍后被更详细地进行解释。

秘密算法是基于校验和的计算，对于检测被检查的数据（即，浏览器代码）中的变化具有很高的概率。如可以从图 3 中看出的那样，随机产生的校验和算法 21（用于移动防护中）使用的校验和计算被分成两步：随机化的输入修改 22；以及对修改的输入进行的已知的校验和计算 23。这些步骤一起形成了随机化的和秘密的校验和算法。

输入修改是指修改器的随机创建，所述修改器排列将要被输入到校验和计算 23 的数据。在移动防护被安全服务器产生时，就确定了一个随机序列。在校验和算法 21 被浏览器执行时，来自浏览器的输入程序代码就被分割成相同尺寸的 n 块。然后，这些块在输入修改阶段 22 中被混搅到上面提及的随机序列中。其结果随后被输入到校验和计算阶段 23。这使用已知的信息摘要算法 5（MD5）。然后，校验和计算被执行，其结果被返回给安全服务器。

可以理解，尽管校验和算法本身是公开的，然而其结果是所述 n 块被输入到校验和算法的次序的函数。该次序对于安全服务器来说是已知的，因此安全服务器可以确定返回到安全服务器的结果是否指示了完整的浏览器。

移动防护需要被保护以避免篡改和避免对其内部工作方式的窥探。移动防护的保护的第一方面是，每当浏览器需要被检查时，就随机地创建新的版本。其次，浏览器环境中的移动防护在被使用时，其寿命很短（少于 30 秒）。尽管人（即，智能的，相对于自动的）对移动防护的攻击在理论上是可能的，但是将花费大量的时间。通过使每个移动防护具有几秒钟的到期时间，人协助的攻击就因此变得实质上不可能了，这是因为在任何攻击可以被完成之前，该移动防护早就是多余的了。

如前面所讨论的那样，移动防护被伪装，以便抵御自动的攻击。

移动防护对运行中的浏览器的存储器映射进行随机化，这里被称为“运行时间浏览器伪装”。浏览器的代码和数据区域被交换，并且堆栈被加扰。这将在下面更全面地讨论。

运行时间浏览器伪装的效果是确保对浏览器的运行时间映射只能进行智能攻击，这是因为将解密的编码流的存储器位置进行了随机化并从而将其隐藏。

为了对存储器访问的位置进行随机化，移动防护对浏览器代码和数据区域的结构进行修改。代码和数据区域被分裂成逻辑片段。需要留意的是，各片段边界不要位于操作代码内。

在新下载的移动防护接收到控制之后并且在对流的解密进行启动之前，移动防护将各片段重定位到新的位置。该过程包括代码片段的修改，这类似于由动态链接器执行的重定位，以便确保:

1. 跳转和分支指令将控制转移到重定位的位置。

2. 读取和写入指令访问重定位的位置处的数据。

在对各片段进行重定位之后，移动防护就执行其操作，直到它被下一个移动防护替代为止。

移动防护需要知道浏览器中某些函数的入口点。各片段的新位置被安全服务器获知并被提供给移动防护。以此方式，在客户端侧上不需要在两个移动防护之间传递信息。

关于堆栈加扰，堆栈包含有对前面的函数调用所返回的地址。这可以被

用于通过改变堆栈上的返回地址来窥探出控制流，或改变浏览器的控制流。在这种攻击中，当程序将要跳转回调用函数时，它就可能被替换为将控制转移给可能的敌方代码。

为了保护堆栈免受这类攻击，使用如下方法，即，随着新的返回地址被加入到堆栈而逐步地加扰堆栈。在函数调用之后，被检查的代码将把控制传递给移动防护中的加扰函数，这将在把控制返回调用函数之前加扰堆栈上新的返回地址。为了对堆栈进行解扰，在使用任何返回地址之前调用移动防护中相应的解扰函数。

加扰函数的实现利用了如下事实的优点，即，移动防护根据需要来创建，以便检查浏览器。这使得在每个移动防护中将创建唯一的加扰和解扰函数。加扰函数基本上是由随机数据的集合组成的，所述随机数据是由安全服务器创建并包含在移动防护中，与浏览器的堆栈上的返回地址进行“异或”运算。为了选择使用哪部分的随机数据，简单的数学函数被用来计算到随机数据的集合的索引（index）。

从而，浏览器被移动防护保护，以避免其状态（包括控制流的位置和变量内容）被通过窥探而确定（如前面所讨论的那样）。

媒体密钥以大约每秒一个的速率被传送给浏览器。这是使用密钥交换协议来进行的，所述密钥交换协议使用随机数据发生器和对于浏览器来说是已知的安全服务器的公共密钥。在有必要获得下一个媒体密钥时，浏览器 10 就产生 16 字节的随机数据并用安全服务器 17 的公共密钥进行加密。然后，加密的数据被包括在对密钥的请求中传送给安全服务器。

安全服务器检查所述请求，只有在移动防护指示浏览器中一切都正确的情况下才会同意所述请求。如果移动防护指示一切都正确，安全服务器提取随机数据，将其与所请求的密钥进行“异或”运算，并将结果传送回浏览器。

在浏览器接收到结果时，通过将结果与在对密钥的初始请求中提供的随机数据相同的随机数据进行“异或”运算来从所述结果提取所请求的密钥。

该协议提供了一种途径来对加密的编码媒体流进行解密，而不必在浏览器的源代码中隐藏任何秘密密钥。密钥的寿命只有几秒钟，这就避免了安全流传输处理在提取一个或几个秘密密钥的情况中出现单点失效。

可以理解，存在两个有效的独立线程，由客户端来执行，它们被总结在图 4 的流程图中。

第一线程是校验。客户端接收移动防护，然后校验客户端程序。一旦校

验已经被确认，就可以在随后的信任间隔中接收 n 个密钥，直到移动防护到期。然后，线程必须使用新的移动防护重复。

与此并行运行的是呈现线程。针对每个密钥，媒体流的片段被接收、解密和呈现。

图 5 总结了服务器的操作。当接收到密钥请求时，如果并且也只有在移动防护仍然起作用（即，如果仍然在那个移动防护的信任间隔之内）的情况下，才给客户端传送密钥。如果移动防护已经到期，那么新的移动防护就被传送给客户端用于校验客户端。如果结果是不正确的，客户端就被认为已经被篡改了，于是密钥传输就被停止。如果结果是令人满意的，那么就开始一个新的信任间隔，并在该新的信任间隔中将密钥传送给客户端。

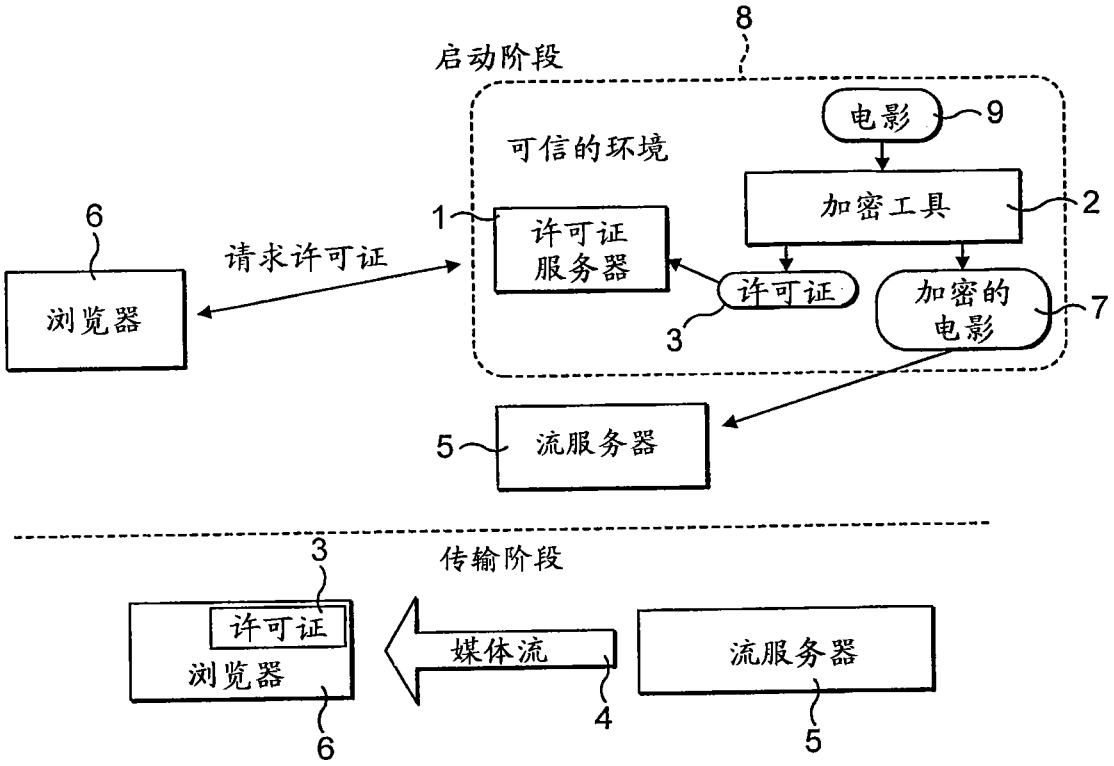

8
启动阶段
电影
9
可信的环境
加密工具
2
6
1
许可证
服务器
请求许可证
浏览器
许可证
加密的
电影
7
3
5
流服务器
3
传输阶段
许可证
浏览器
媒体流
流服务器
4
6
5

图1

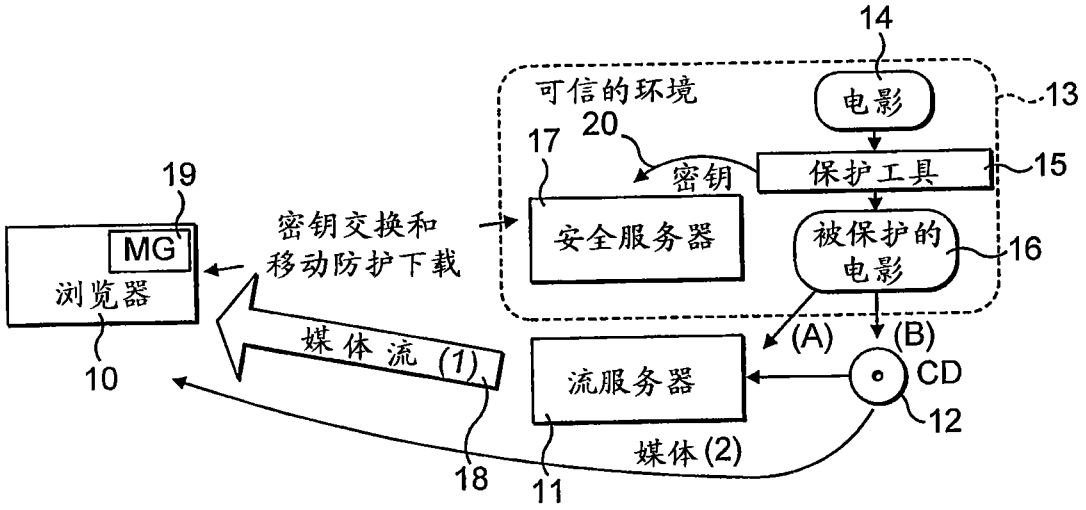

14
可信的环境
电影
13
20
17
密钥
保护工具
15
19
密钥交换和
移动防护下载
安全服务器
被保护的
电影
16
MG
浏览器
媒体流 (1)
(A)
(B)
流服务器
CD
12
媒体(2)
10
18
11

图2

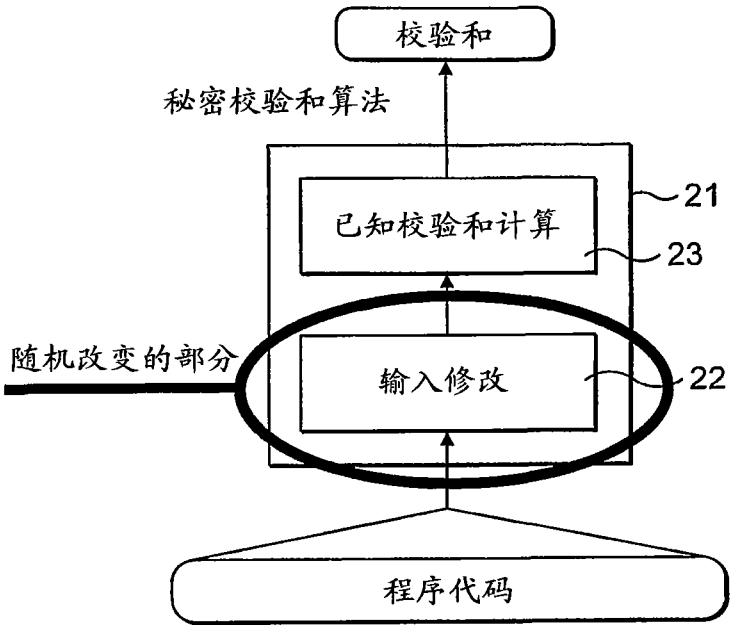

校验和
秘密校验和算法
21
已知校验和计算
23
随机改变的部分
输入修改
22
程序代码

图3

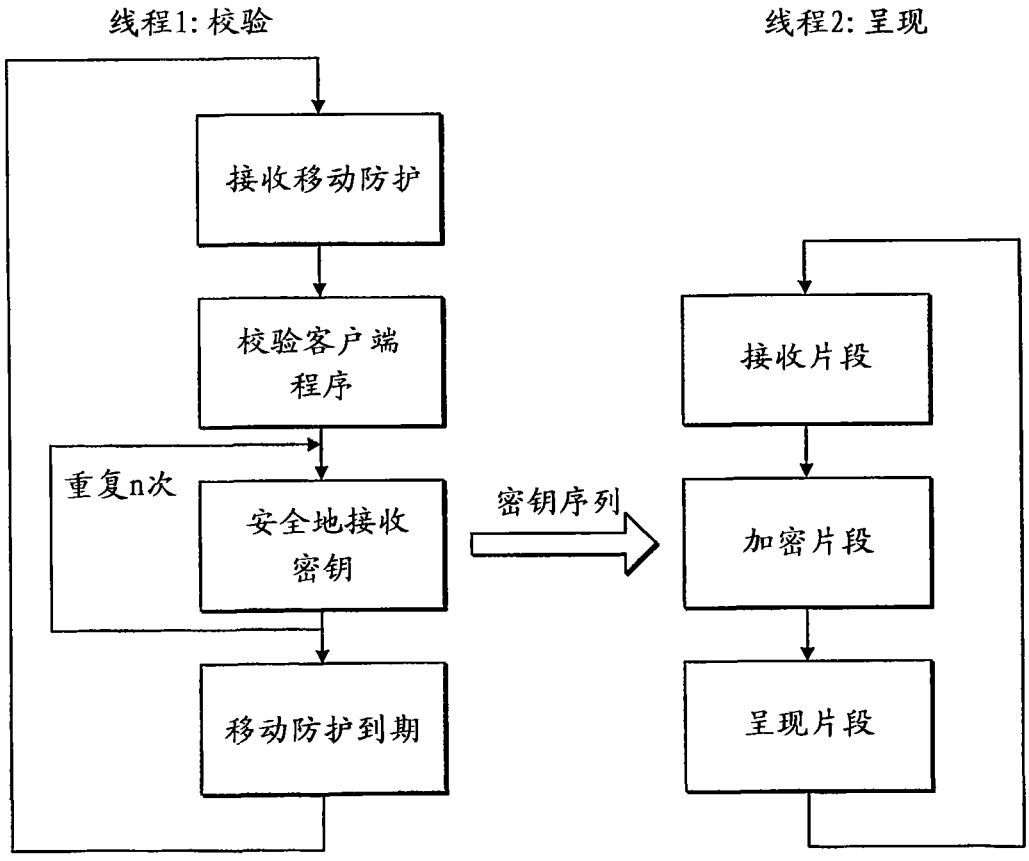

线程1: 校验
线程2: 呈现
接收移动防护
校验客户端程序
重复n次
安全地接收密钥
密钥序列
移动防护到期
接收片段
加密片段
呈现片段

图4

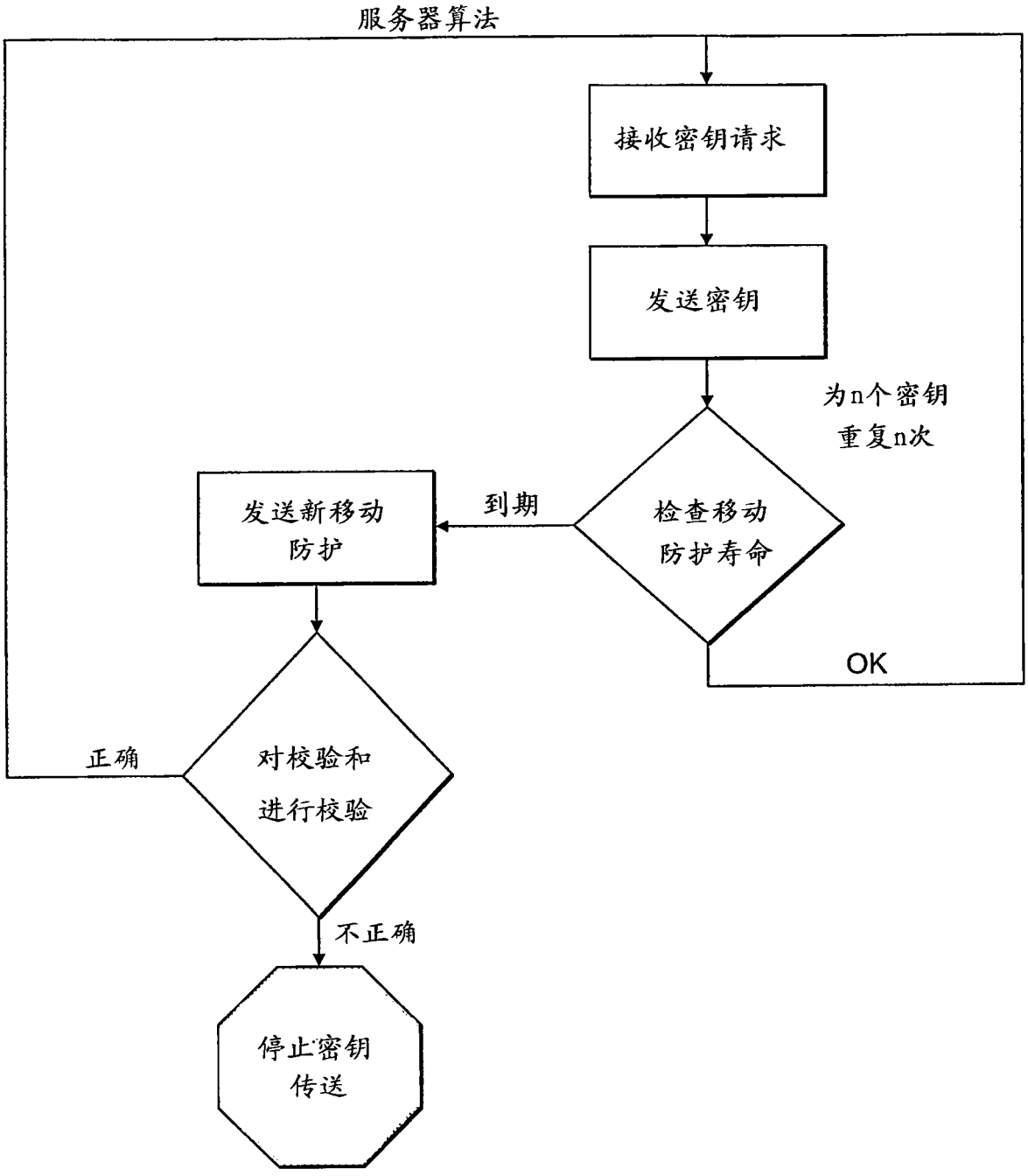

服务器算法
接收密钥请求
发送密钥
为n个密钥
重复n次
检查移动
防护寿命
到期
发送新移动
防护
OK
对校验和
进行校验
正确
不正确
停止密钥
传送

图5