US 20100191974A1

(19) **United States**

(12) **Patent Application Publication**

Dubhashi et al.

(10) **Pub. No.: US 2010/0191974 A1**

(43) **Pub. Date: Jul. 29, 2010**

(54) **SOFTWARE APPLICATION VERIFICATION**

(75) Inventors: **Kedarnath A. Dubhashi**, Redmond, WA (US); **John Bocharov**, Seattle, WA (US); **Hany Farag**, Issaquah, WA (US); **Gilles Khouzam**, Bothell, WA (US); **Kiran Kumar**, Bothell, WA (US)

Correspondence Address:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052 (US)

(73) Assignee: **MICROSOFT CORPORATION**, Redmond, WA (US)

(21) Appl. No.: **12/360,943**

(22) Filed: **Jan. 28, 2009**

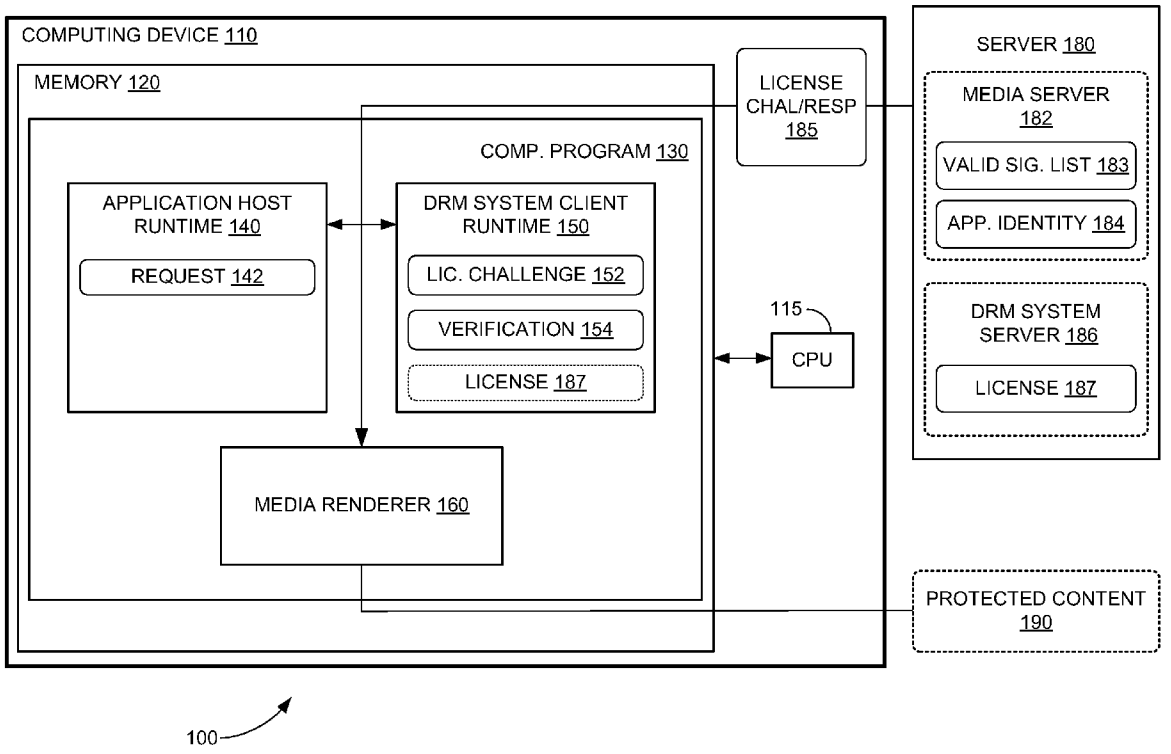
Publication Classification

(51) **Int. Cl.**
H04L 9/32 (2006.01)

(52) **U.S. Cl.** **713/176**

(57) **ABSTRACT**

Various embodiments for software application verification are disclosed. Software application verification applies digital rights management to applications that run protected content on a playback device. In this way, protected content may be provided to approved applications and withheld from applications that have not been approved to run the protected content.



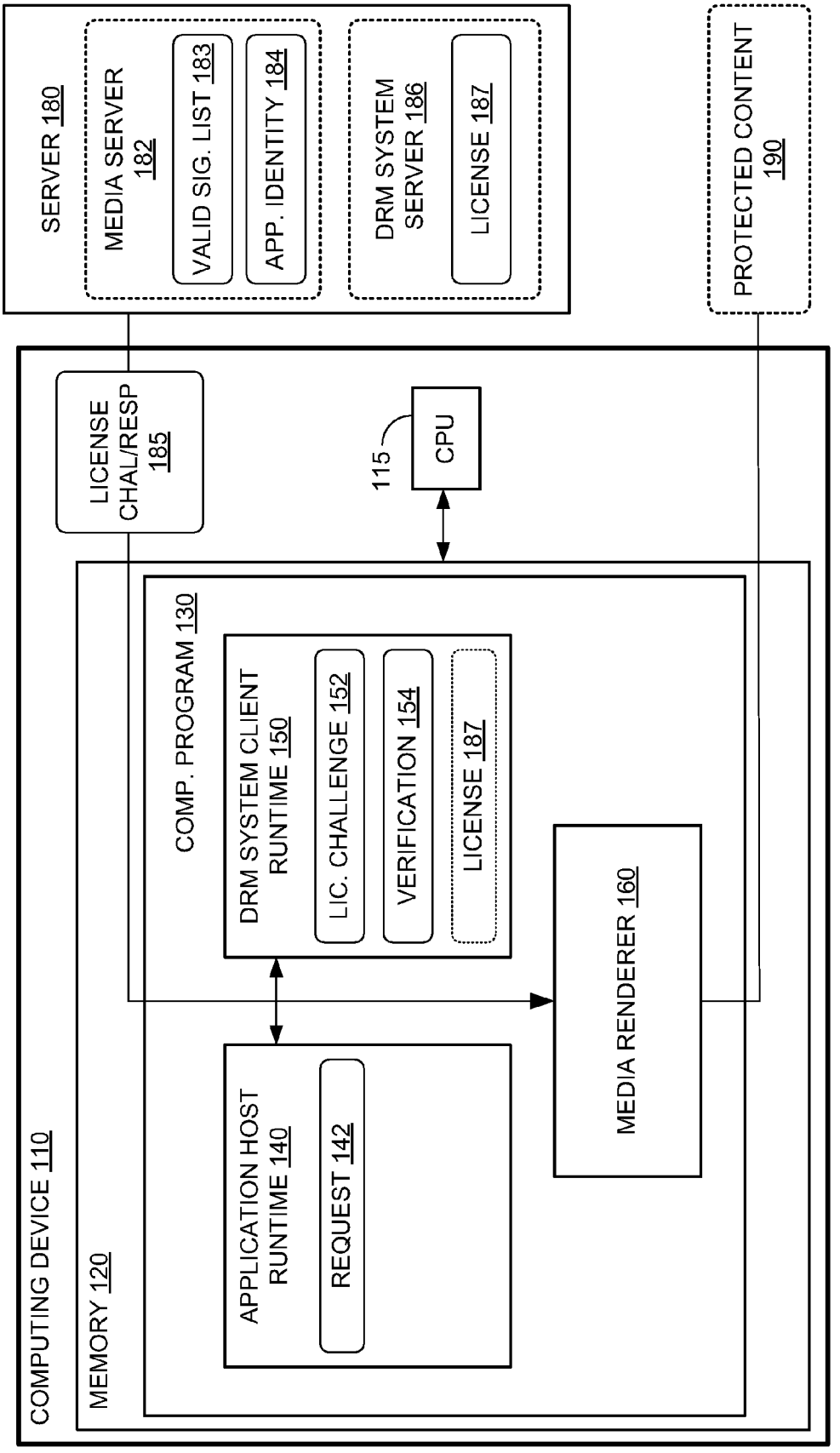

SERVER 180
MEDIA SERVER 182
VALID SIG. LIST 183
APP. IDENTITY 184
DRM SYSTEM SERVER 186
LICENSE 187
PROTECTED CONTENT 190
LICENSE CHAL/RESP 185
115
CPU
COMPUTING DEVICE 110
MEMORY 120
COMP. PROGRAM 130
DRM SYSTEM CLIENT RUNTIME 150
LIC. CHALLENGE 152
VERIFICATION 154
LICENSE 187
APPLICATION HOST RUNTIME 140
REQUEST 142
MEDIA RENDERER 160
100

FIG. 1

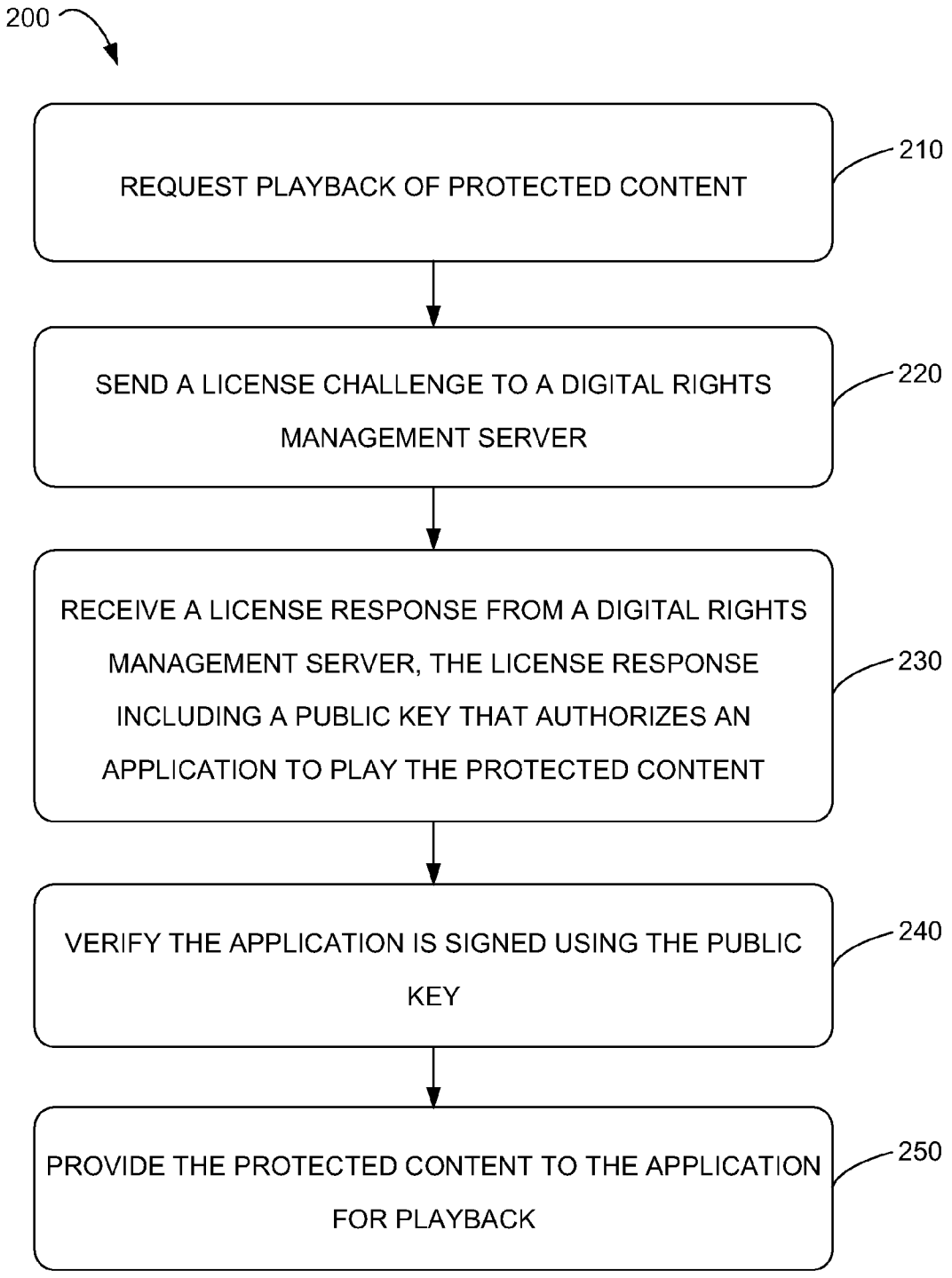

200
REQUEST PLAYBACK OF PROTECTED CONTENT
210
SEND A LICENSE CHALLENGE TO A DIGITAL RIGHTS MANAGEMENT SERVER
220
RECEIVE A LICENSE RESPONSE FROM A DIGITAL RIGHTS MANAGEMENT SERVER, THE LICENSE RESPONSE INCLUDING A PUBLIC KEY THAT AUTHORIZES AN APPLICATION TO PLAY THE PROTECTED CONTENT
230
VERIFY THE APPLICATION IS SIGNED USING THE PUBLIC KEY
240
PROVIDE THE PROTECTED CONTENT TO THE APPLICATION FOR PLAYBACK
250

FIG. 2

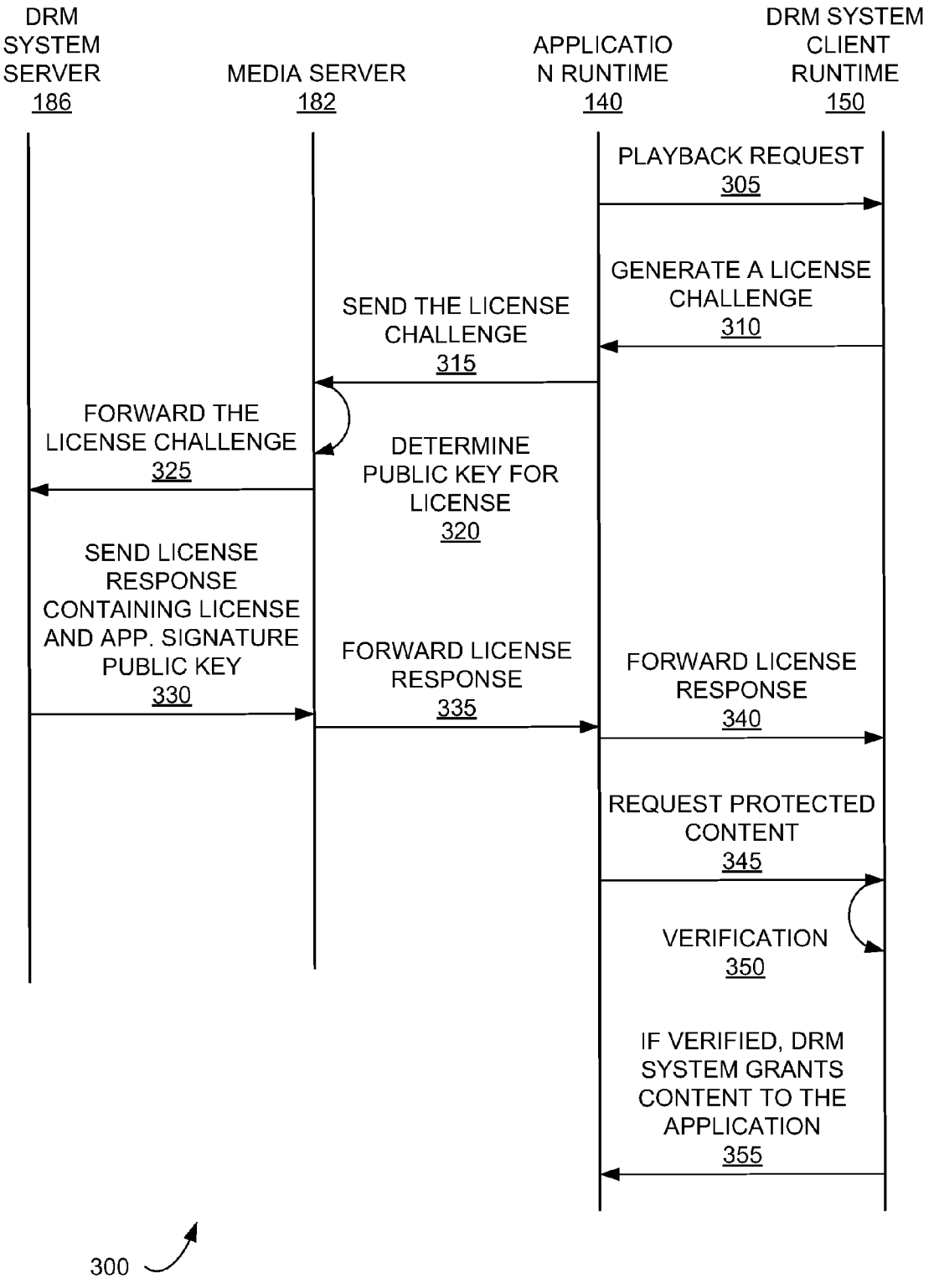

DRM SYSTEM SERVER 186
MEDIA SERVER 182
APPLICATIO N RUNTIME 140
DRM SYSTEM CLIENT RUNTIME 150
PLAYBACK REQUEST 305
GENERATE A LICENSE CHALLENGE 310
SEND THE LICENSE CHALLENGE 315
FORWARD THE LICENSE CHALLENGE 325
DETERMINE PUBLIC KEY FOR LICENSE 320
SEND LICENSE RESPONSE CONTAINING LICENSE AND APP. SIGNATURE PUBLIC KEY 330
FORWARD LICENSE RESPONSE 335
FORWARD LICENSE RESPONSE 340
REQUEST PROTECTED CONTENT 345
VERIFICATION 350
IF VERIFIED, DRM SYSTEM GRANTS CONTENT TO THE APPLICATION 355
300

FIG. 3

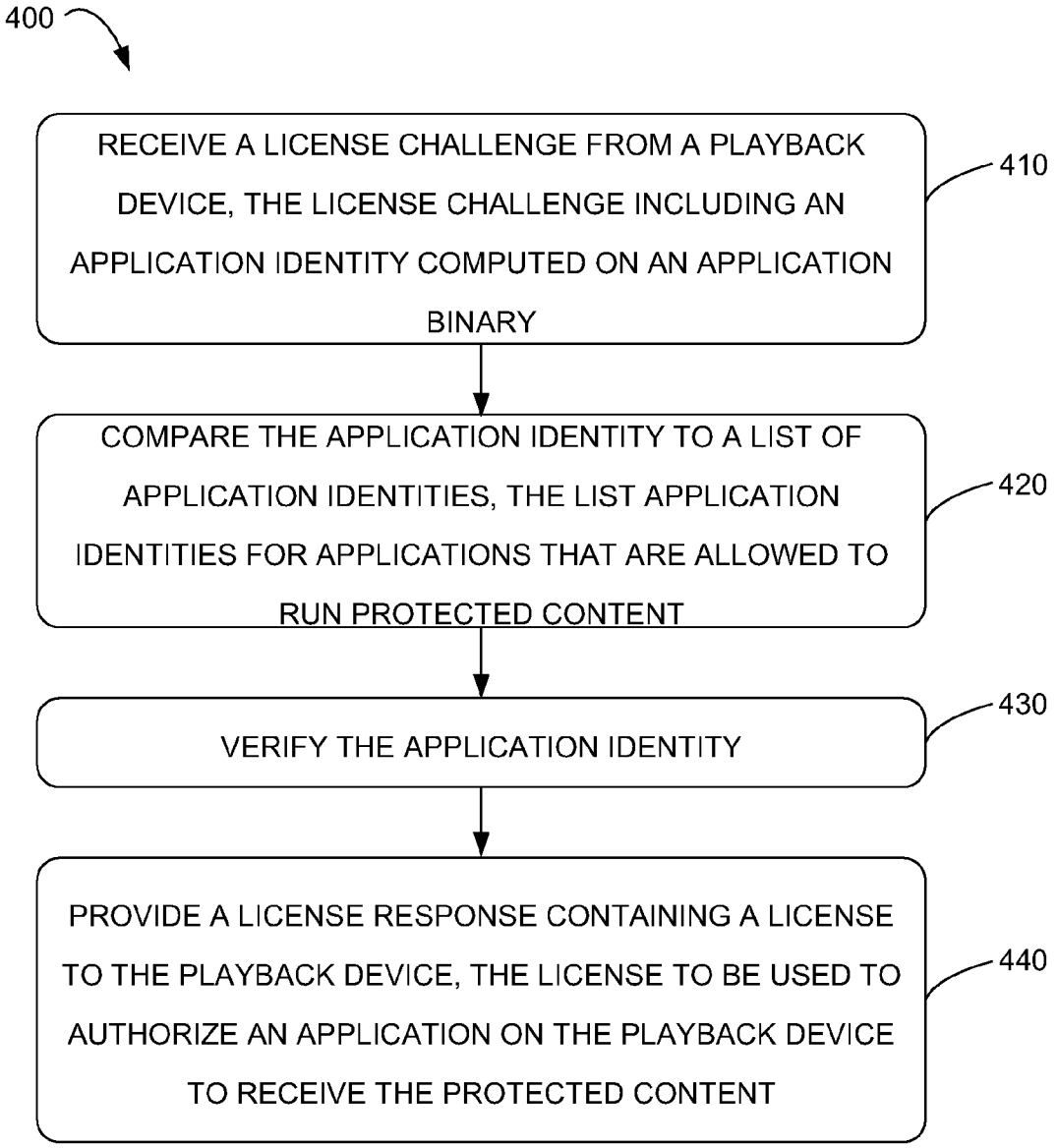

400
RECEIVE A LICENSE CHALLENGE FROM A PLAYBACK DEVICE, THE LICENSE CHALLENGE INCLUDING AN APPLICATION IDENTITY COMPUTED ON AN APPLICATION BINARY
410
COMPARE THE APPLICATION IDENTITY TO A LIST OF APPLICATION IDENTITIES, THE LIST APPLICATION IDENTITIES FOR APPLICATIONS THAT ARE ALLOWED TO RUN PROTECTED CONTENT
420
VERIFY THE APPLICATION IDENTITY
430
PROVIDE A LICENSE RESPONSE CONTAINING A LICENSE TO THE PLAYBACK DEVICE, THE LICENSE TO BE USED TO AUTHORIZE AN APPLICATION ON THE PLAYBACK DEVICE TO RECEIVE THE PROTECTED CONTENT
440

FIG. 4

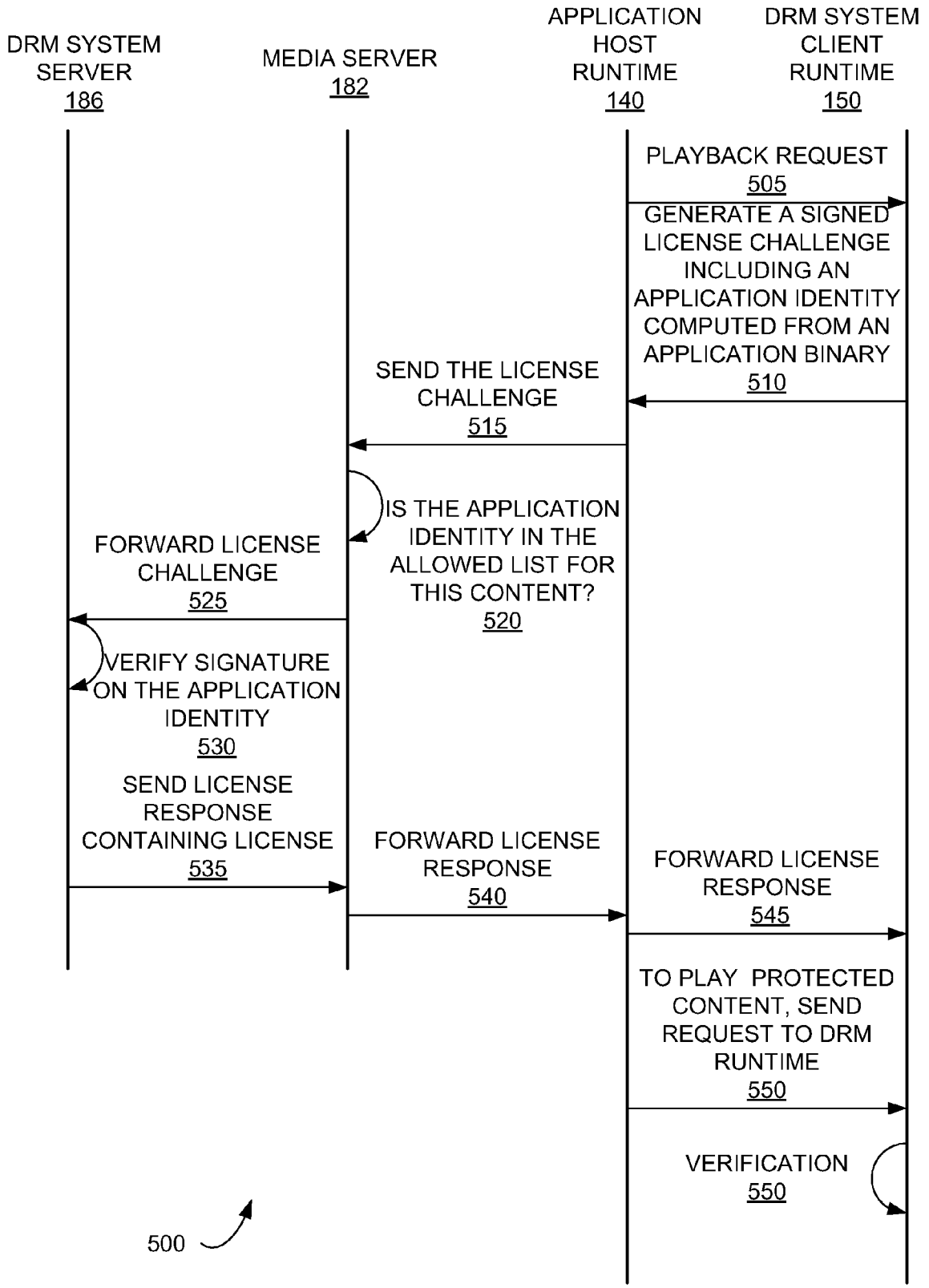

DRM SYSTEM SERVER 186
MEDIA SERVER 182
APPLICATION HOST RUNTIME 140
DRM SYSTEM CLIENT RUNTIME 150
PLAYBACK REQUEST 505
GENERATE A SIGNED LICENSE CHALLENGE INCLUDING AN APPLICATION IDENTITY COMPUTED FROM AN APPLICATION BINARY 510
SEND THE LICENSE CHALLENGE 515
IS THE APPLICATION IDENTITY IN THE ALLOWED LIST FOR THIS CONTENT? 520
FORWARD LICENSE CHALLENGE 525
VERIFY SIGNATURE ON THE APPLICATION IDENTITY 530
SEND LICENSE RESPONSE CONTAINING LICENSE 535
FORWARD LICENSE RESPONSE 540
FORWARD LICENSE RESPONSE 545
TO PLAY PROTECTED CONTENT, SEND REQUEST TO DRM RUNTIME 550
VERIFICATION 550
500

FIG. 5

SOFTWARE APPLICATION VERIFICATION

BACKGROUND

[0001] Digital Rights Management (DRM) allows content owners and distributors to not only deter unwanted copying of content but also to control how that content is used. For example, DRM allows content owners to control how many viewings are allowed for certain content, for example audio or video media data, how many copies may be made for the content, what device may receive the owned content, etc.

[0002] Currently, DRM protected content is associated with a machine, i.e. a PC, a portable device, a cellular phone, etc. This allows multiple applications on a machine to access the protected content once the machine has been granted access by a DRM system. Unfortunately, malicious users may circumvent this DRM approach and access content they are not authorized to access, change the way content is played back, etc.

[0003] One example of utilizing content in an unintended manner involves application spoofing of a Rich Internet Application (RIA). A spoofed application may be created that appears to a server like an application that is approved to access, play, store, etc., protected content. Therefore, the application may receive the content from the server and use it in a way a content owner does not intend. One example is advertisement funded video, where a content owner or distributor may intend a user to watch pre-roll advertisements, interstitial advertisements, or overlay advertisements, etc., in order to watch the premium content. Spoofed applications attempt to play only the video content without the advertisements, resulting in revenue loss for at least one party. Online banking provides another example of application spoofing where it is detrimental to allow an apparently approved RIA to use protected content in a manner not intended by the content owner, the bank, etc.

SUMMARY

[0004] Accordingly, various embodiments for software application verification are described below in the Detailed Description. Software application verification applies digital rights management to applications that run protected content on a playback device. In this way, protected content may be provided to approved applications and withheld from applications that have not been approved to run the protected content. Embodiments include client side software application verification, server side software application verification, distributed software application verification, etc.

[0005] One client side example embodiment comprises requesting playback of protected content, sending a license challenge to a digital rights management server, receiving a license response from a digital rights management server, the license response including a public key, verifying the application is signed using the public key, and if the application is signed with the public key, providing the protected content to the application for playback.

[0006] One server side example embodiment comprises receiving a license challenge from a playback device, the license challenge including an application identity such as a value computed on an application binary (for example a hash value), comparing the application identity to a list of approved applications, the list of application identities for applications that are allowed to run protected content, and providing a license response containing a license to the playback device, the license to be used to authorize an application on the playback device to receive the protected content.

[0007] This Summary is provided to introduce concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to implementations that solve any or all disadvantages noted in any part of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. **1** shows an example of an embodiment system for software application verification for an application on a computing device.

[0009] FIG. **2** shows a process flow depicting an embodiment of a method for client side software application verification.

[0010] FIG. **3** shows a signaling diagram of one embodiment of a method for client side software application verification.

[0011] FIG. **4** shows a process flow depicting an embodiment of a method for server side software application verification.

[0012] FIG. **5** shows a signaling diagram of one embodiment of a method for server side software application verification.

DETAILED DESCRIPTION

[0013] FIG. **1** shows an example of an embodiment system **100** for software application verification for an application **140** on a computing device **110**. System **100** also includes a server **180** including a media server **182** having a valid signature list **183** and a digital rights management server **186** that may contain a license to authorize an application on computing device **110** to play protected content **190**. In one embodiment, computing device **110** is configured to restrict playback of protected content to applications approved by a content provider, content distributor, etc. By extending DRM techniques to applications that play protected content, content owners have more control over how protect content is played, improving security, enforcing restrictions on advertisements and thus increasing advertisement based revenue, etc. We now describe example embodiment system **100** in more detail in the following paragraphs.

[0014] In the illustrated embodiment, computing device **110** includes a processor **115**, a memory **120**, at least one computer program **130**, and a media renderer **160**, which may be in software or hardware. Computing device **110** further includes an application host runtime, also called an application **140**, configured to play protected content, and a digital rights management system client runtime, also called a digital rights management module **150**, in communication with the application **140**.

[0015] In response to a request **142** to play protected content from the application **140**, the digital rights management module **150** is configured to send a license challenge **152** to a digital rights management server **186**, and to receive a license response **185** including license **187** having a public key that authorizes the application **140** to play the protected content **190** wherein once the application is verified **154** using the public key, the digital rights management module **150** allows the application **140** to play the protected content **190**. For

example, the digital rights management module **150** may be configured to prevent protected content from playing on a spoofed application, to enforce playback rules on the application, etc. Other embodiments may utilize other identifying information or security approaches than a public key. For example, other suitable approaches may use identifying information to authorize an application to play protected content without using public key cryptography. In some embodiments, the playback rules may enforce at least one of pre-roll advertising, interstitial advertising, or overlay advertising, but other embodiments are not so limited.

[0016] In some embodiments, the digital rights management module **150** may be configured to store the license **187** for offline verification of the application **140**, wherein the application **140** may be authorized to play protected content **190** while not being in communication with the digital rights management server **186**. In some embodiments, a public key that authorizes the application **140** to play protected content **190** may have also been used to sign the application **140** prior to a license challenge **152** being sent to a digital rights management server **186**. For example, the application **140** may be verified if the public key in the license **187** matches a private key used to sign the application. But other embodiments are not so limited and may utilize application verification approaches that do not use public keys, such as a code including a policy that states which applications are allowed to access protected content, as a non-limiting example.

[0017] Some embodiments may provide service side verification, for example where a digital rights management server examines an application identity before issuing a license to play protected content. Embodiment server side approaches are described in more detail below with reference to FIGS. **4-5**.

[0018] Continuing with the Figures, FIG. **2** shows a flow diagram of an embodiment of a method **200** for client side software application verification. First, as indicated in block **210**, method **200** comprises requesting playback of protected content. Method **200** also comprises sending a license challenge to a digital rights management server as indicated in block **220**. In this way, a license challenge including an application identity may be first sent to the media server and the media server can consult a valid list of application identities for applications that are allowed to play protected content. If an application is approved to play protected content, the media server can forward the license challenge including a request for the digital rights management server and the digital rights management server may then place a corresponding public key to an application in the approved signature list. In one embodiment, a server may include both a media or information server and a digital rights management server.

[0019] Next, method **200** comprises receiving a license response from the digital rights management server, the license response including a license having a public key that may be used to verify an application is authorized to play protected content, as indicated at **230**. In some embodiments, a private key that matches the public key that was used for application verification and authorization may have been used to sign the application prior to the license challenge. Additionally, method **200** may further comprise storing the license for offline verification of the application.

[0020] Method **200** also comprises verifying, using the public key, that the application is correctly signed, as indicated in block **240**. Such verifying may include, but is not limited to, verifying the application is signed, verifying the signature is correct, and verifying a private key used to sign the application matches a public key in the license.

[0021] Next, method **200** comprises and providing the protected content to the application for playback, as indicated at **250**. In some embodiments, providing the protected content to the application for playback further comprises preventing the protected content from playing on a spoofed application. Additionally, method **200** may further comprise enforcing playback rules on an application playing the protected content. In one example, enforcing playback rules may include at least one of enforcing playback of pre-roll advertising, interstitial advertising, or overlay advertising.

[0022] FIG. **3** shows a signaling diagram **300** of a detailed embodiment of a method for client side software application verification, as introduced generally with reference to FIG. **2**. Signaling diagram **300** includes communications between runtimes operable on a client including an application runtime, also called application **140**, and a DRM system client runtime, also called digital rights management module **150**, and server functionality including a media server **182** and digital rights management server **186**. The media server **182** and digital rights management server **186** may operate as separate programs on one server, may operate on separate servers, be distributed between a plurality of servers, etc.

[0023] Signaling diagram **300** begins with a playback request being sent from application **140** to digital rights management module **150** at **305**. In one example, a user may wish to run protected content in application **140** and in response to an input from the user, the playback request is sent to the digital rights management module **150**. Then, digital rights management module **150** generates a license challenge at **310** and sends it through application **140** to media server **182** at **315**. Media server **182** can then consult a configuration module on the media server **182** to determine which public key to put in a license to the protected content at **320**, wherein the configuration module stores which applications are allowed to play protected content.

[0024] Next, signaling diagram **300** involves the media server **182** forwarding the license challenge to the digital rights management server **186** at **325**, and requests the digital rights management server **186** server to include an application signature public key in the license in response. Then, digital rights management server **186** creates the license response containing the license and the application signature public key and sends the license response to media server **182** at **330**. Media server **182** then forwards the license response to application **140** at **335**, which in turn forwards the license response to digital rights management module **150** at **340**. In this way, when application **140** requests to play protected content at **345**, digital rights management module **150** can verify application **140** using a license in the license response at **350**, and if application **140** is authorized to play the protected content, the digital rights management module **150** grants the protected content to the application at **355**.

[0025] Continuing with FIG. **4**, a flow diagram of an embodiment of a method **400** for server side software application verification is illustrated. First, as indicated in block **410**, method **400** comprises receiving a license challenge from a playback device, the license challenge including a signed application identity computed on an application binary. Some embodiments may further comprise computing the application identity on an application binary, and signing the application identity with a public key of a digital rights management client on the playback device.

[0026] Method **400** also comprises comparing the signed application identity to a list of application identities, the list of application identities for applications that are allowed to run protected content, as indicated in block **420**. Next, method **400** comprises verifying the signed application identity, as indicated at **430**.

[0027] Next, method **400** comprises providing a license response containing a license to the playback device, the license to be used to authorize an application on the playback device to receive the protected content, as indicated at **440**. In some embodiments, a license may be provided to a playback device to prevent protected content from playing on a spoofed application. Additionally, a license may be provided to the playback device to enforce playback rules on an application playing the protected content. In this way, the license may be used to specify what are allowable uses of content, which applications may access content, including playing back content, exporting content, copying content, transferring content, etc. Some embodiments may further comprise the playback device then storing the license for subsequent requests to play protected content.

[0028] FIG. **5** shows a signaling diagram **500** of one embodiment of a method for server side software application verification, as introduced generally with reference to FIG. **4**. Signaling diagram **500** includes communications between runtimes operable on a client including an application runtime, also called application **140**, and a DRM system client runtime, also called digital rights management module **150**, and server functionality including a media server **182** and a digital rights management server **186**. The media server **182** and digital rights management server **186** may operate as separate programs on one server, may operate on separate servers, be distributed between a plurality of servers, etc.

[0029] Signaling diagram **500** begins with a playback request being sent from application **140** to digital rights management module **150** at **505**. In this example, digital rights management module **150** then generates a license challenge, computes an application identity of the application's binary, and adds the application identity to the license challenge. Then digital rights management module **150** returns the license challenge to application **140** at **510**. Application **140** then sends the license challenge to media server **182** at **515**, and media server **182** in turn consults a valid signature list to see whether the application identity is included in the valid signature list for requested protected content at **520**. If the application identity is not in the valid signature list, the application verification fails.

[0030] If the application identity is in the valid signature list, media server **182** forwards the full license challenge to digital rights management server **186** and asks it to validate the signature at **525**. In some embodiments, the media server **182** may provide the signature verification. Digital rights management server **186** then verifies the signature on the application identity at **530**, and sends a license response containing the license to media server at **535**. Media server **182** then forwards the license response to the application **140** at **540**, and the application **140** in turn forwards the license response to the digital rights management module **150** at **545**. In this way, when application **140** requests to play protected content, it sends a request to digital rights management module **150** at **550** and the digital rights management module **150** can verify if the application is authorized to play the protected content at **550**.

[0031] It will be appreciated that the embodiments described herein may be implemented, for example, via computer-executable instructions or code, such as programs, stored on a computer-readable storage medium and executed by a computing device. Generally, programs include routines, objects, components, data structures, and the like that perform particular tasks or implement particular abstract data types. As used herein, the term "program" may connote a single program or multiple programs acting in concert, and may be used to denote applications, services, or any other type or class of program. Likewise, the terms "computer" and "computing device" as used herein include any device that electronically executes one or more programs, including, but not limited to, media playback devices and any other suitable device such as personal computers, servers, laptop computers, hand-held devices, cellular phones, microprocessor-based programmable consumer electronics and other protected content playback devices.

[0032] It will further be understood that the configurations and/or approaches described herein are exemplary in nature, and that these specific embodiments or examples are not to be considered in a limiting sense, because numerous variations are possible. The specific routines or methods described herein may represent one or more of any number of processing strategies. As such, various acts illustrated may be performed in the sequence illustrated, in other sequences, in parallel, or in some cases omitted. Likewise, the order of any of the above-described processes is not necessarily required to achieve the features and/or results of the embodiments described herein, but is provided for ease of illustration and description.

[0033] The subject matter of the present disclosure includes all novel and nonobvious combinations and subcombinations of the various processes, systems and configurations, and other features, functions, acts, and/or properties disclosed herein, as well as any and all equivalents thereof.

1. A method of providing protected content to an authorized application, the method comprising:
- requesting playback of protected content;
- sending a license challenge to a digital rights management server;
- receiving a license response from the digital rights management server, the license response including a license having a public key that authorizes an application to play the protected content;
- verifying the application is signed using the public key; and
- providing the protected content to the application for playback.

2. The method of claim **1**, wherein providing the protected content to the application for playback further comprises preventing the protected content from playing on a spoofed application.

3. The method of claim **1**, wherein providing the protected content to the application for playback further comprises enforcing playback rules on an application playing the protected content.

4. The method of claim **3**, wherein enforcing playback rules further comprises at least one of enforcing pre-roll advertising, interstitial advertising, overlay advertising, or enforcing purchasing, subscription or rental rules.

5. The method of claim **1**, wherein the public key that authorizes the application to play the protected content was used to sign the application prior to the license challenge.

6. The method of claim **5**, further comprising storing the license for offline verification of the application.

7. The method of claim **5**, wherein verifying the application is signed using the public key further comprises:

verifying the application is signed;

verifying a signature for the application is correct; and

verifying a private key used to sign the application matches a public key in the license.

8. A method of providing protected content to an approved application, the method comprising:

receiving a license challenge from a playback device, the license challenge including a signed application identity computed on an application binary;

comparing the signed application identity to a list of application identities, the list of application identities for applications that are allowed to run protected content;

verifying the signed application identity;

providing a license response containing a license to the playback device, the license to be used to authorize an application on the playback device to receive the protected content.

9. The method of claim **8**, wherein the license is to be used on the playback device to prevent the protected content from playing on a spoofed application.

10. The method of claim **8**, wherein the license is to be used on the playback device to enforce rules on an application playing the protected content.

11. The method of claim **10**, wherein the rules include at least one of enforcing pre-roll advertising, interstitial advertising, overlay advertising or enforcing purchasing, subscription or rental rules.

12. The method of claim **8**, further comprising computing the application identity on the application binary, and signing the application identity with a private key of a digital rights management client on the playback device.

13. The method of claim **8**, wherein the application identity is a signed hash value.

14. A computing device for playing protected content in an approved application, the computing device comprising:

an application configured to play protected content; and

a digital rights management module in communication with the application, the digital rights management module configured to send a license challenge to a digital rights management server, and to receive a license response, wherein once the application identity is verified, the digital rights management module allows the application to play the protected content.

15. The computing device of claim **14**, wherein the digital rights management module is configured to prevent the protected content from playing on a spoofed application.

16. The computing device of claim **14**, wherein the digital rights management module is configured to enforce playback rules on the application.

17. The computing device of claim **16**, wherein the playback rules enforce at least one of pre-roll advertising, interstitial advertising, overlay advertising or enforcing purchasing, subscription or rental rules.

18. The computing device of claim **14**, wherein the application identity is verified using a public key included in the license response that matches a private key used to sign the application prior to the license challenge.

19. The computing device of claim **18**, wherein the digital rights management module is configured to store the license.

20. The computing device of claim **18**, wherein the application is verified with the public key if the public key in the license matches a private key used to sign the application.

* * * * *