CHAPTER 8

Unleashing Premium Entertainment with Hardware-Based Content Protection Technology

If there's content that can only be there if it's rights protected, we want to be able to have that content available to you... So in no sense are we hurting you, because if they're willing to make the content available openly, believe me, that's always the most wonderful thing. It's the simplest.

—Bill Gates

Fast-developing technologies have gradually changed many aspects of people's lifestyle, including the way we enjoy video and audio entertainment. Computers, from stationary workstations to mobile handhelds, are used increasingly as multimedia players. Readers have switched from paper books to e-books and audio books; CD, DVD, and Blu-ray rental and sales numbers have been declining in the last ten years because more and more people rent and purchase titles from the Internet.

Compared to traditional media, user experience for the new digital entertainment is not by any means worse: a lot of premium content is available at online stores at the same time or before the CD/DVD/Blu-ray releases; the picture quality of downloaded and streaming content is equal or higher than that of the disc; video and sound can be echoed from computers to big-screen TVs. What's more, the digitization of content further benefits consumers in several ways:

- *Convenience*: It is a human-oriented design for the users to sit comfortably on a couch in his living room, browse catalogs of thousands of titles, and watch interesting ones immediately. It saves the room that is needed for storing physical discs; digital contents can be stored on the cloud rather than local hard drives; sharing content with friends and family takes just a few clicks.

- *Lower cost*: Data transmission eliminates the material, manufacturing, and logistics expenses of discs, and eventually reduces the costs of content and benefits consumers. More flexible purchase options are now available. For example, instead of having to buy an entire music CD at a higher price, a music lover can pay for the individual songs of her choice.
- *Easy management*: Searching for specific content in a large collection is fast and accurate. No more worries about physical damages to a disc due to, for example, temperature and relative humidity.

Rights Protection

Alongside the widespread deployment of the entertainment feature on all forms of computers is the problem of copyright protection from diverse types of piracy, such as sharing, downloading, and counterfeiting. In the case of content protection, the device user is untrusted, and the content provider wants assurance from the device manufacturer that a user cannot bypass protections. This increases the scope of threats and attacks that have to be defended against, primarily physical and side-channel attacks.

It is commonly accepted that general-purpose computers with an open operating system and using software to handle content are not as robust as closed systems (for example, a dedicated Blu-ray player) because of inexpensive approaches that have been developed by researchers and hackers to defeat software protection schemes without involving advanced expertise or special equipment.

Obviously, profit loss because of piracy on computers is one of the biggest concerns of content providers (such as writers and film studios). Therefore, the content owners have to decide whether to demand rights protection for their content, and if so, what assets of the content shall be protected and at which level.

For example, most 4K ultra and 1080p high-definition movies sold at online stores today require the purchaser's device to feature appropriate hardware-based protection for rights management and playback of the content. Software-based protection measures may be deemed insufficient for such content. In other words, if the hardware of the user's computer does not meet minimum requirements, then the content owner and the service provider that distributes the content will not sell the title to the platform. On the other hand, the content of standard-definition formats usually requires only software-based protection or does not require protection at all. There also exists content that is free of any rights management, allowing consumers to make unlimited copies and share with others.

In any case, the choice and decision is solely the content owner's. Before buying or renting content, the customer should be well aware of all restrictions posed by the content.

■ **Note** The content provider decides the types and levels of rights protection that are required for its content.

Regarding what assets to protect for content, there are several considerations. Here are just a few examples:

- How many times can the content be played back?
- Can the consumer keep the content for good, or for only a limited period?
- Is the content permitted to be played back on the user's other devices in the same *domain*? If so, what is the maximum number of devices allowed?
- Is the user allowed to share the content with others? If so, what is the limitation? Is there a limitation on the number of times it can be copied and shared? What is the rights policy for the copied version of the content?
- Can the content be displayed on an external monitor or TV? If so, what type of protection is required on the link between the computer and the monitor or TV?

Such policy restrictions are recorded in a *license* (or *rights object*) file that is transmitted together with the content to the end user during a transaction.

It appears that the rights policy tightens what the user can do with the content. Why would software and hardware vendors spend resources to enforce them?

As a matter of fact, content right protection mechanisms implemented on a computer is *for*, and not against, users' interest, because they are necessary to satisfy people's raising demand for entertainment. It should be emphasized that, on Intel platforms, for example, the protection mechanisms

- Do not touch users' personal files
- Do not introduce constraints to content that does not come with rights protection mandated by the provider
- Do not enforce policies that have not been accepted by the user
- Do not impact or change anything the user would normally do with his computer

The protection mechanisms do one thing and only one thing: increase the difficulty and cost of unauthorized activities that violate the terms and conditions associated with the content to a level that meets the robustness requirements determined by the content provider, so that the content can be made available to end users. If a user is not happy with the content's terms and conditions, then he should not obtain the content, and the protection schemes simply will not be functioning at all.

It would be an annoying experience for a user to encounter an error window saying that the computer cannot play the movie because it lacks necessary hardware. Therefore, most modern computers today support the types and levels of protection mandated by the content providers. A platform without such capability will significantly limit what users can do for entertainment and degrade the user experience. To this end, Intel platforms shipped with the embedded management engine and select core processors feature a hardware-based content protection infrastructure that is able to satisfy the most

stringent protection requirements. This technology enables consumers to enjoy their favorite content seamlessly on Intel products and do not need to worry about compliance problems that prevent the content providers from delivering the content.

DRM Schemes

Digital rights management (DRM) controls the use of digital content. Discussions in this chapter will focus on video and audio content, but the protection mechanisms largely apply to rights management for textual publications and still images as well.

A number of DRM schemes exist in the market today. Content owners and distributors choose the desired schemes used for protecting their content. Many vendors support more than one scheme, so their content can be distributed on various types of systems and platforms. The following is a list of most popular DRMs used for online media:

- *Widevine*: The Widevine DRM was designed by a company of the same name, which was acquired by Google in 2010. Widevine is supported by Google TV, Google Chromecast, Google Play, and other media players on Android and other Google products.
- *PlayReady*: Microsoft first introduced PlayReady in 2007. It is widely used by Windows applications. In May 2010, Microsoft announced that Netflix had chosen PlayReady as its primary DRM technology.[1]
- *FairPlay*: Created by Apple, FairPlay is a built-in component of the QuickTime software, which is installed on all lines of Apple devices, such as iPhone, iPod, iPad, Apple TV, and iTunes.
- *Marlin*: The Marlin DRM is developed and maintained by the Marlin Developer Community, with founding members including Intertrust, Panasonic, Philips, Samsung, and Sony. Marlin is mainly deployed in the smart TV market. Especially popular in Europe and Asia, it is used by the national IPTV standard of Japan, as well as major online video streaming web sites in China, such as Baidu and PPTV.
- *OMA DRM*: OMA is an acronym for Open Mobile Alliance, created by major mobile phone manufacturers and carriers for developing open DRM standards to enhance interoperability among mobile devices.

This is not a complete list and there are other smaller DRMs too. These competing schemes realize, to a certain extent, similar rights management capabilities with distinct design details. In general, the three main elements of a DRM scheme is device key management, rights management, and content encryption.

The main building blocks of a platform's DRM implementation are graphically presented in Figure 8-1, where TRS is short for *tamper-resistant software*. The cryptography block includes a pseudorandom number generator (PRNG).

Not all components in the figure are applicable to all DRMs, and a DRM may support additional value-add features. The DRM client is responsible for performing device key provisioning and storage, enforcing rights management, and securing playback flows, per the defined policies.

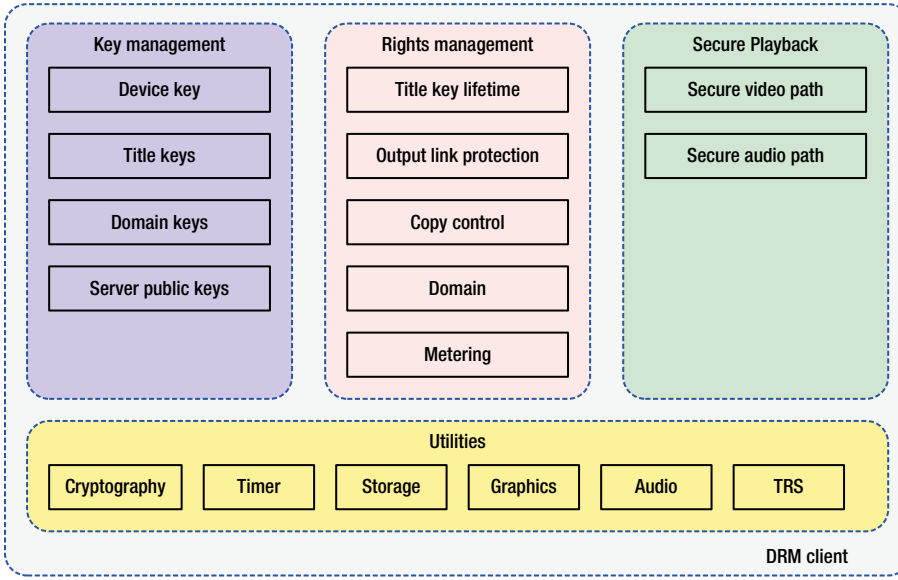

Figure 8-1. *The building blocks of a typical DRM client*

Owners of some DRM schemes release specifications with corresponding software development kits (SDKs) that contain reference code. Usually, SDKs implement DRM-specific functionalities, such as key management and rights management, and leave the utility implementations to adopters. An SDK not only remarkably simplifies the adopters' efforts for integration, but also unifies behaviors of different products on different systems, and hence reduces the chance of incompatibility.

Device Key Management

The device secret key is a critical asset for a content protection scheme. It can be a random AES[2] (*advanced encryption standard*) key or an asymmetric key pair. The device secret key is provisioned to each individual device during the manufacturing process, or remotely with a server before the DRM is invoked for the first time in the field. For most DRM schemes, the secret key on a specific device is unique, although a small number of DRMs still use global keys. Using unique keys makes it easier to implement revocation for compromised devices.

Once provisioned, the device secret key must be stored with confidentiality in nonvolatile memory of the device and never exposed in the clear or shared with other devices. A compromised device secret key could be used to decrypt content keys in

licenses, retrieve the decrypted content, and distribute or playback freely. Therefore, the key is more valuable than any single title. Compromising the key may result in the device being revoked and no longer eligible for content playback.

Besides the device secret key, a device is also provisioned with a content provider's public key, which is used in, for example, verifying digital signature of licenses. A content provider's public keys are not secrets and may be hardcoded in the device or provisioned in tandem with the device secret key. Integrity protection should be applied to public key storage.

Rights Management

The content license contains customized rights for the content for a specific transaction. The license must be integrity protected during transmission and storage. A DRM scheme that is designed for online content services generally features the following characteristics to enforce rights of content:

- *Protection levels*: This attribute specifies what level of protection is required for audio and video, respectively. For example, one level can be defined to requiring software-based protection such that clear content can be viewed by software, and another level requiring hardware-based protection where software is not allowed to see the clear content. Notice that if the audio and video of a title assume unequal protection levels, then the encryption keys for audio and video should be different. This attribute also indicates whether the stream is allowed to be played on external displays, and if so, which version of HDCP[3] (high-bandwidth digital content protection) is required.
- *Domain*: A user can create a domain on the server and enroll all his devices in the domain. Content may be copied from one device to another of the same domain. The domain members then share the content and a domain key, which is used to, for instance, securely transport title keys between two devices in the same domain.
- *Secure timer*: A DRM may leverage the system's trusted clock and secure timer to enforce playback duration. For example, a movie rental may expire after 24 hours after the first playback begins.
- *Playback metering*: A DRM may measure durations of the user's playback of content and report to the server. For example, a user earns free credits toward the purchase of premium movies after he has watched a certain amount of commercials, metered by the DRM.
- *Transaction tracer*: After downloading the content, the transaction tracer allows the user to start playback even if the computer is offline. The platform will record and report offline playback events to the rights server once network connection is resumed.

Notice that the rights policies are generated for each transaction, depending on the options selected by the customer.

Playback

The main goal of securing content playback is to protect clear content keys, as well as the video and audio paths. The content keys are sent to the platform encrypted with the device key or its derivative. The algorithm for encrypting the content key is DRM specific. The other attributes of the license should be at least integrity protected.

The entire flow of the content, especially after it is decrypted on the platform, must be safeguarded according to requirements specified in the license. A license should define the following policies:

- Whether the content keys (for video and audio, respectively) may be handled by software
- Whether clear compressed and uncompressed video data may be processed by software, respectively
- Whether clear compressed and uncompressed audio data may be processed by software, respectively
- What protection mechanism should be applied to the connection between the platform and an external display

Apparently, these playback requirements are specific to the title rather than individual transactions—they are the same for all issued licenses of the content.

UltraViolet

In early 2011, Intel announced the new digital media services[4] with the second-generation core processor family (code name "Sandy Bridge"). The services feature innovative hardware-based content protection mechanisms that enable consumers to enjoy UltraViolet[5] content "anytime and anywhere." Intel has partnered with content providers such as Best Buy CinemaNow and Walmart VUDU for UltraViolet content distribution. The security and management engine is one of the main elements of the solution.

UltraViolet is developed and maintained by the Digital Entertainment Content Ecosystem (DECE), an alliance of companies that include film studios, consumer electronics manufacturers, and vendors that form the ecosystem. In contrast to common misunderstandings, UltraViolet is, in fact, not a DRM. Instead, it is a free cloud-based content rights library that encompasses several existing DRMs, such as OMA. The ultimate goal of UltraViolet is allowing users to "*buy once, play everywhere*," regardless of the DRM scheme of the title and the device on which the purchase was made.

The coexistence of multiple DRM schemes in the market today is good for competition, but in the meanwhile results in inconvenience for consumers. Say, someone who owns an Android smartphone, an iPad tablet, and a Windows 8 laptop is not able to watch the TV programming he purchased from Google Play on his iPad or Windows laptop because Apple's FairPlay and Microsoft's PlayReady are not compatible with Google's Widevine. UltraViolet aims at resolving this problem by unifying content

encoding and format used by various DRMs and storing the proof-of-purchase on a centralized location—the UltraViolet cloud. Besides online content, new releases of DVD and Blu-ray discs from participating film studios are also UltraViolet-enabled. Once a user buys an UltraViolet-enabled disc, he can register with the cloud server and stream the same title on all of his UltraViolet-capable devices.

End-to-End Content Protection

The journey of protected content begins at the server that creates the encrypted content and finishes at a screen that displays the content. A number of steps and components are involved in the flow, as shown in Figure 8-2. They work together to realize complete end-to-end protection.

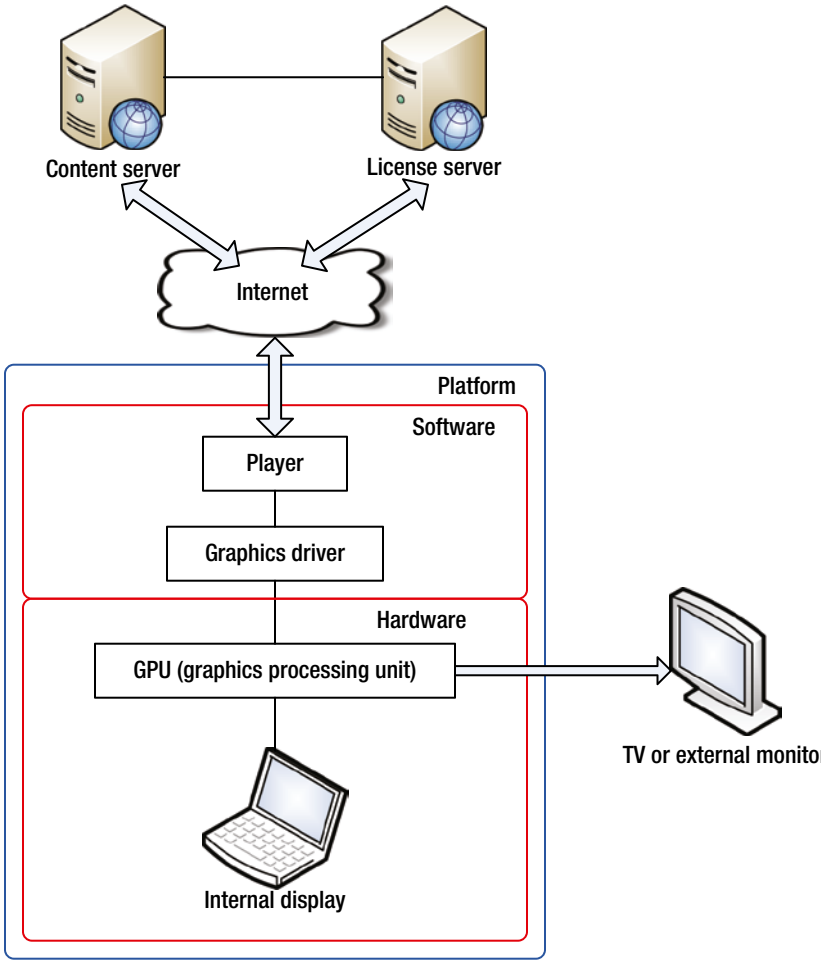

Figure 8-2. *End-to-end flow of content with software-based protection*

In a nutshell, the encrypted media and its license are delivered to the platform through open networks. The player receives them, installs the license, and performs operations per the license's permissions. For playback, the player produces the decrypted content and passes it to the graphics driver (user mode and kernel mode) for displaying on either the internal screen or an external device.

To guard the content from end to end, every link and every component must apply proper protections. This is because an attacker only needs to circumvent the weakest point in the path to acquire the protected content. The entire ecosystem is broken if one link or component is compromised, no matter how robust the others are. Let us first take a look at a few components, and then identify the weakest.

Content Server

A content server is responsible for encrypting clear encoded titles with the defined cryptographic algorithm and delivering encrypted content to users' platforms. Different schemes may deploy different algorithms. The most popular choices nowadays are AES in counter mode, CBC (cipher block chaining) mode, and CBC-CTS (ciphertext stealing) mode with 128-bit keys. Few DRM schemes also support 256-bit keys. Refer to Chapter 3 for an introduction on AES and these modes.

Notice that a long title (such as a 120-minute movie) may be divided into multiple sessions that are encrypted with different keys, respectively, with a key rotation algorithm. Also, different resolutions of a title should be encrypted with separate keys, because their desired rights managements may vary. In most cases, the title encryption is performed only once and the encrypted content will be consumed by all customer platforms. Re-encryption of a title is required under certain scenarios, such as compromise of the key.

The transmission from the content server to the platform's software is through an open Internet connection. Essentially, encrypted content alone is not a secret because it is useless without the decryption key. The content delivery may be real-time streaming as the consumer is watching, or the user can download the entire title and save it in local storage beforehand. Some distributors let clients proactively download content before viewing. Such options allow users to enjoy the entertainment even without high-speed network, such as when traveling on a plane.

License Server

A license server maintains databases for encryption keys of all titles and issues rights for content transactions. A license defines the use policies associated with the underlying content. Attributes regarding secure playback are the same for all users, for example, requiring HDCP when the content is sent to an external device for display; whereas other restrictions described in the license vary depending on the consumer's preference at the time of purchase. Naturally, if a user is willing to pay more, then the policy will generally enable more rights, such as sharing. The license server generates customized licenses for different transactions on the fly and transmits to end users' platforms.

Licenses must be integrity protected during the transmission for obvious reasons. The license server owns an asymmetric key pair, where the private key is used for signing licenses and the public key is for DRM clients to verify the signature.

The license server and the content server must be always in sync about the keys and initialization vectors used in title encryption. Although they are logically separate modules, in reality the two servers can be implemented on the same machine.

Software Stack

Besides implementing fundamental playback functionalities (such as start, stop, pause, fast-forward, slow motion, scene selection, and so forth) with the graphics driver, the player software also downloads content from the content server and, for rights-protected content, interacts with the license server to receive the license.

For content that does not mandate hardware-based protection, the player is also responsible for handling the device key, installing licenses, managing secure playback, and enforcing rights. This was the common practice before hardware-based content protection architectures were born, and it is still used today for nonpremium content. For example, Windows Media Player is able to enforce rights for Microsoft's PlayReady scheme. This includes unwrapping the content key in the license and decrypting the encrypted media file with the content key. In other words, the player has access to the entire clear compressed content. Additionally, a user's operation requests on the content are examined by the software and allowed only if permitted by the installed license.

To achieve meaningful protection, most software-based solutions utilize antitampering techniques, such as TRS, which make reverse-engineering and code modification harder. However, TRS is a type of security through obscurity and obfuscation with no provable robustness. What's worse, TRS not only enlarges the size of the software but also raises power consumption. Notice that both nonvolatile storage and battery life are critical performance vectors for mobile systems.

For hardware-based content protection, the player does not handle the processing of the license or decrypting of the content. Instead, it acts as a proxy and relies on the hardware infrastructure for enforcing the rights and decrypting the media for playback. As a result, there is no longer a necessity to apply TRS to the player because it has no secrets to hide.

External Display

Most rights management schemes require protection of the data transfer from the display controller to external display devices with the HDCP protocol, proposed by Intel in 2000. The protocol applies encryption on the link between the source (transmitter) and the sink (receiver or repeater). The link can be wired, for example, with an HDMI (high-definition multimedia interface) cable, or it can be wireless, such as through Intel's Wireless Display.[6]

Weak Points

From Figure 8-2 and the accompanying analysis, one may conclude that, for software-based content protection, the weakest points are on the client. For example, the player and the graphics driver on the platform's open operating system both possess plaintext content without provable security.

To harden the weak points to some extent, many DRM schemes require integrity assurance for the operating system by securing the boot process, so the platform is running in a known trusted state when it is playing premium content. Intel's Boot Guard technology introduced in Chapter 6 is used to satisfy such requirements.

Intel's Hardware-Based Content Protection

To robustly harden the weak points, the content protection mechanisms on the client side must be implemented in hardware, for all three aspects of the problem:

- Playback protection
- Device key management
- Rights management per policy

Figure 8-3 illustrates the high-level block diagram for Intel's hardware-based solution. The player software communicates with the embedded engine through the HECI (*host embedded communication interface*; see Chapter 3 for details) driver and offloads sensitive operations to the engine. Note that the engine does not use a network stack and it does not interact with remote servers directly. The graphics driver collaborates with the engine and GPU (graphics processing unit) for PAVP (*protected audio and video path*) session establishment and management.

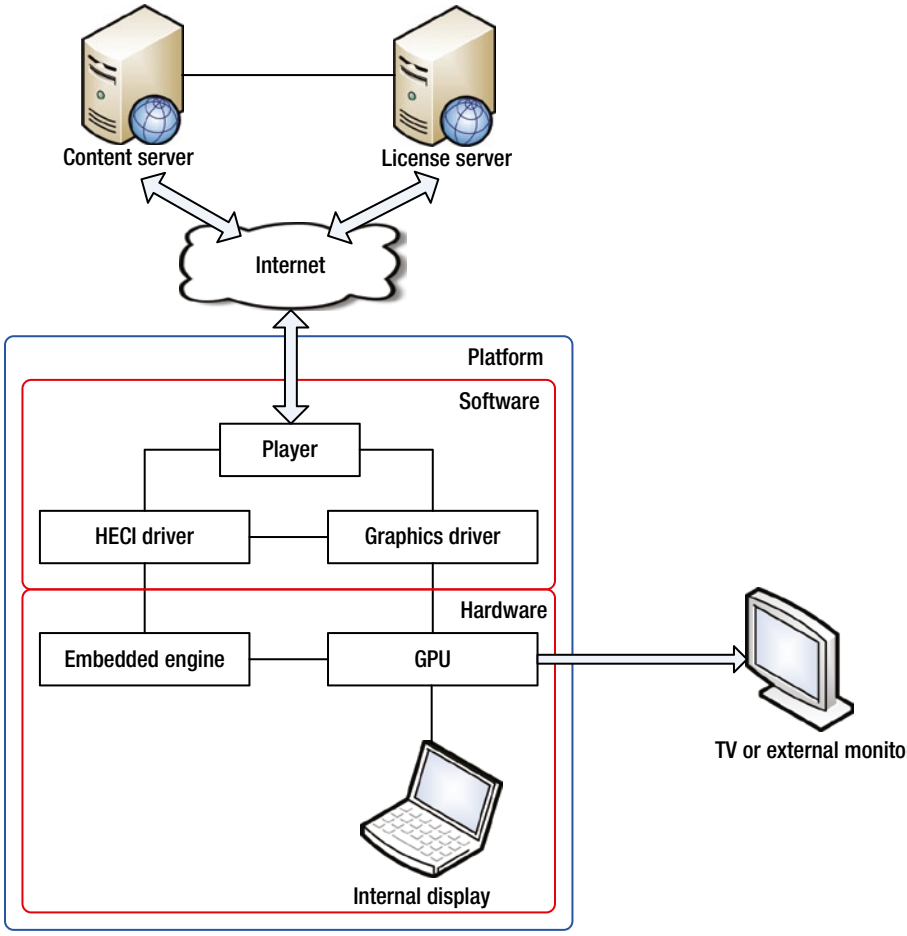

Figure 8-3. *End-to-end flow of content with Intel's hardware-based protection*

The embedded engine's firmware implements content protection functionalities in a dedicated *task*, logically isolated from other applications running on the engine. Refer to Chapter 4 for the task isolation mechanism of the firmware.

Intel's digital media services and the UltraViolet feature are backed by the hardware-based protection. However, notice that the solution is not specific to UltraViolet. It provides a generic foundation that is able to support any content protection schemes on any operating systems.

Protected Audio and Video Path (PAVP)

For a secure playback session, the player should not be in charge of content processing. The software stack on the host operating system, including ring-0 drivers, must not be able to access clear content or the content encryption key, and the clear content shall not

be present in memory that is visible to the host. The PAVP technology is designed to meet these requirements with native hardware blocks, so the capability can be invoked with minimum software development and integration effort.

The PAVP is an Intel proprietary technology that protects the audio and video flows in the GPU. The main idea is to have the GPU take encrypted content and the content key as input, perform decryption of the content in the hardware, and then display the content on screen. The security and management engine and graphics driver also participate in the flow. The embedded engine is responsible for finding out the content key and injecting it through an internal bus to the GPU; the graphics driver is responsible for programming frame metadata and initializing playback.

Figure 8-4 depicts the building blocks of PAVP and their connections. MMIO stands for memory-mapped input and output. For SoC (system-on-chip) platforms, the embedded engine, and the GPU communicate through the IOSF (Intel on-chip system fabric) sideband. For non-SoC, the channel is the DMI (direct media interface).

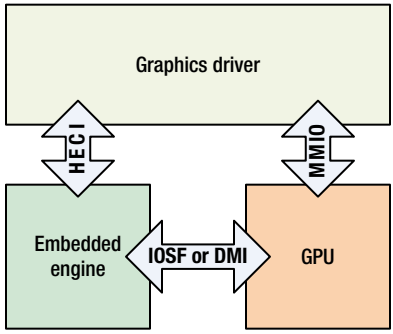

Figure 8-4. *The building blocks of PAVP*

The protection of the link between the platform and the external monitors is also managed by the embedded engine and the GPU using HDCP protocols. With this architecture, the entire content path, from the content server to the display, is protected with provable security.

Device Key Provisioning

For protection with a hardware root of trust, the device secret key must be secured by hardware and never exposed to the software stack. The security and management engine offers several approaches for device key installation. The computer manufacturers, after obtaining device keys from the DRM authority, can choose an appropriate installation process based on the requirements of the DRM scheme.

If the factory environment is trustworthy, then a device key may be provisioned in cleartext to devices and locked down before the conclusion of the manufacturing. The embedded engine's kernel provides a method for manufacturers to store nonvolatile data to the engine during manufacturing. The engine's firmware will convert secret nonvolatile data to secure blobs. It is also possible to send the clear device key to the engine through HECI and have the engine store in a secure blob.

Alternatively, if secure provisioning of the device key is required after the product leaves the factory, then the EPID (*enhanced privacy identification*; refer to Chapter 5 for technical details) algorithm can be leveraged. In this scenario, the platform and a remote provisioning server mutually authenticate each other and establish session keys that are then used to protect the device key when it is transmitted from the server to the platform. Note that the platform remains anonymous during the procedure. Also, recall that the EPID involves heavy mathematical calculations and may be slow on some platforms. The EPID-based provisioning can also be deployed in the factory environments, if necessary, so end users do not experience extra delays caused by device key initialization when they launch playback for the first time on new computers.

The EPID method is also advantageous in that the device does not have to be returned to the factory for refurbishment when its device key is lost (but not compromised). Reprovisioning of a new device key can be done remotely and conveniently by invoking EPID again.

The embedded engine saves device secret keys in its assigned partition of the system's flash memory, with protections for confidentiality, integrity, and optionally, anti-replay. The keys for protecting the device key blobs are device-specific, so copied blobs do not work on other platforms. See Chapter 3 for more information on the secure nonvolatile storage mechanism implemented by the engine's kernel.

Rights Management

The security and management engine is an ideal place for conducting rights enforcements for several reasons: first, it is equipped with necessary utilities, such as hardware-based PRNG, a cryptography driver, protected nonvolatile storage, a secure timer, and so forth; second, as discussed in previous chapters, its unique isolation nature makes it immune from attacks and threats from the host.

Intel Wireless Display

The HDCP is a specification developed by Intel to protect audio and video entertainment over digital interfaces. Today, HDCP protocol has become a popular choice across the industry for guarding the content transmission from a computer to a repeater or a receiver (a display device such as monitor or TV). A repeater functions as both a receiver and a transmitter for downstream HDCP repeaters and receivers. The latest HDCP version, 2.2, is an interface-independent specification that can be applied to any interface.

The Intel Wireless Display (WiDi) is a proprietary interface that adopts HDCP 2.2. It allows streaming movies or anything on the computer's screen from a WiDi-capable Intel platform to a TV or projector that supports WiDi or Miracast (requires WiDi 3.5 or newer). When the transmitted content requires link protection, HDCP 2.2 is the core on which the security of WiDi is built.

In the WiDi setup, the Intel platform is the HDCP transmitter. The transmitter side of the protocol is implemented jointly by WiDi software, the graphics driver, the GPU, and the security and management engine. The WiDi software running on the host talks

with the receiver through the operating system's Wi-Fi stack. The embedded engine sits behind the software and communicates with the software via HECI.

A transmitter does not receive content; hence it does not hold a device private key or certificate. A receiver (or repeater) owns a 1024-bit RSA (*Rivest, Shamir,* and *Adleman*) key pair. The receiver's unique device ID and RSA public key $kpub_{rx}$ are digitally signed by the HDCP governing authority, Digital Content Protection (DCP) LLC, in a certificate hardcoded in the device together with the device's RSA private key. Notice that the certificate is a binary format defined by the standard and not a generic X.509 certificate, and no intermediate certification authorities are involved.

Revocation for compromised receivers is realized by a system renewability message (SRM), which is a structure that comprises an SRM version number, the list of IDs of repeater and receiver devices that have been revoked, and the authority's signature. Because HDCP is designed to be an offline protocol that does not depend on Internet connectivity to function, the SRM is delivered to transmitters together with the content, if and only if the content mandates HDCP protection. A transmitter is required to reserve at least 5KB of nonvolatile memory for SRM storage, and the transmitter must keep the latest version of the SRM that it has ever encountered in its secure storage.

The HDCP 2.2 protocol is comprised of four stages:

1. *Authentication and key exchange (AKE)*: The transmitter verifies that the receiver has a valid RSA key pair endorsed by the DCP LLC, and it is not one of the devices revoked in the SRM. A master key *Km* is also exchanged in this phase if one does not exist on the transmitter for this receiver. The master key is specific for this pair of transmitter and receiver and it is used in lieu of the RSA key to expedite future HDCP handshakes between these two devices.

2. *Locality check*: The transmitter enforces the locality of the receiver by making sure that the time elapsed between sending a message to and receiving the response from the receiver is no longer than the specified duration.

3. *Session key exchange*: The transmitter generates a session key and initialization vector (IV) and sends securely to the receiver. The content is encrypted using a salted version of the session key and the IV with a 128-bit AES counter mode.

4. *Authenticating repeater*: If the receiver is a repeater, then the repeater assembles downstream device topology information and forwards it to the upstream transmitter for authentication.

As the root of trust, the security and management engine is responsible for performing all sensitive operations in the HDCP protocol, including but not limited to the following:

- Validating the receiver's certificate and checking its device ID against the revocation list.
- Managing SRM.

- Maintaining a database of master keys and corresponding repeater IDs. The database supports up to ten receivers. The oldest entry will be removed when an eleventh receiver is paired with the transmitter.
- Randomly generating the IV and the session key.
- Securely storing the HDCP global secret constant lc_{128}.
- Injecting the salted session key (session key exclusive-OR'ed with lc_{128}) to the GPU's PAVP block.
- Facilitating locality checks, with or without precomputed L and L'.

The GPU of the transmitter uses the salted session key and IV to encrypt the content before the encrypted content is transmitted to the receiver. On the receiver side, the same key is derived to decrypt and play the content.

Authentication and Key Exchange

To showcase the embedded engine's roles in the transmitter side of the protocol, Figure 8-5 replicates the AKE protocol flow chat (without stored *Km*) of the HDCP 2.2 specification and describes the operations that are offloaded to the firmware for processing. The firmware must hardcode DCP LLC's root public key and lc_{128}. The symbol (*data*)*key* denotes the ciphertext of *data* encrypted with *key*. Refer to the HDCP 2.2 specification for meanings of variables such as r_{tx}, r_{rx}, H, H', Kh, and so forth.

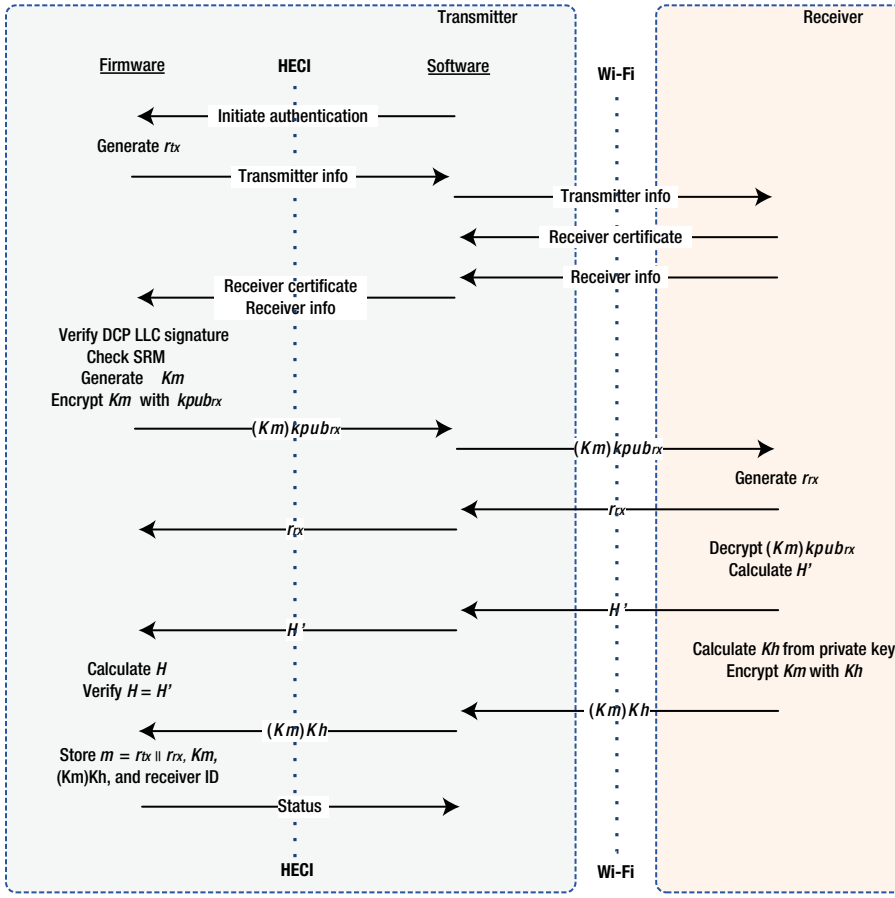

Figure 8-5. *Authentication and key exchange without stored Km; transmitter implemented in firmware*

The HDCP 2.2 protocol is made up of a few flows, and Figure 8-5 exhibits just one of them. Besides AKE without *Km*, the embedded engine participates in all other flows as well, to assure that HDCP's security objectives are satisfied at the hardware level.

Content Protection on TrustZone

The ARM processor architectures with TrustZone support DRM and hardware-based content protection. The TrustZone security framework allows software to execute sensitive DRM operations to the secure mode. The sensitive operations include content key processing and rights management. Conceptually, the separation for nonsecure and secure modes is similar to Intel's solution where the security and management engine handles sensitive operations.

However, there are two notable limitations of the TrustZone when it is used for content protection:

- *Persistent nonvolatile storage for device keys and installed licenses*: The TrustZone does not come with native secure nonvolatile storage. Integrators must deploy third-party extensions to realize this capability.
- *Secure audio and video path*: The Trust does not provide a native protection mechanism for decrypted content. The secure audio and video path has to be implemented by third-party extensions.

For example, the content protection architecture that is built in the Samsung's ARM-based Galaxy S III smartphone utilizes Discretix's hardware-assistant DRM solution, integrated with TrustZone and its trusted execution environments.[7]

On the other hand, Intel's hardware-based content protection is a self-contained and native solution that is equipped with all the necessary infrastructures. Platform manufacturers can simply invoke the defined hardware interface to take advantage of the technology. This significantly reduces system complexity and saves the resources required for deploying and maintaining additional extensions.

References

1. Microsoft, "Netflix Taps Microsoft PlayReady as Its Primary DRM Technology for Netflix Ready Devices and Applications," www.microsoft.com/en-us/news/press/2010/may10/05-25playreadynetflixpr.aspx, accessed on May 25, 2014.
2. National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, accessed on November 17, 2013.
3. Digital Content Protection LLC, "High-bandwidth Digital Content Protection System," www.digital-cp.com, accessed on May 10, 2014.
4. Intel, Unlock the World of UltraViolet with Intel Devices, www.intel.com/content/www/us/en/architecture-and-technology/intel-insider-for-premium-hd-home-entertainment.html, accessed on May 25, 2014.
5. DECE, UltraViolet, www.uvvu.com, accessed on May 25, 2014.
6. Intel Wireless Display, www.intel.com/go/widi, accessed on May 25, 2014.
7. Discretix, "Hardware-Assisted DRM with ARM TrustZone on the Samsung GALAXY S III Smartphone Secured by Discretix," www.discretix.com/press-release/hardware_assisted_drm_with_arm_trustzone_on_the_samsung_galaxy_s_iii_smartphone_secured_by_discretix, accessed on May 30, 2014.