DRM Integration Specification

Contents

Introduction

The present document contains several different types of material concerning HbbTV and DRM.

- Clauses 5 and 6 are technical background material concerning DRM and the DRM APIs in TS 102 796 [1].
- Clause 7 is recommendations and guidelines for various stakeholders in the HbbTV ecosystem. The primary focus of this clause is looking perhaps 2-3-4 years into the future from when the present document was created. However, recommendations and guidelines for stakeholders given the current status of the ecosystem are also included.
- Clause 8 is DRM-system independent requirements for terminals to play back DRM encrypted content.
- Clause 9 lists criteria to be met in order for additional DRM systems to be added to future revisions of the present document.
- Annexes A and B are optional (conditional mandatory) descriptions of how specific DRM systems are to be used with TS 102 796 [1] if those systems are supported. Annex C is a placeholder for a DRM system that has been used with TS 102 796 [1] in the past.
- Annex D is data on the use of DRM with TS 102 796 [1] in 2022 and 2023 while the present document was being compiled.

1 Scope

The present document addresses the integration between HbbTV (as defined in TS 102 796 [1]) and broadband content protection technologies - "DRM". This integration has not previously been included in TS 102 796 due to commercial sensitivities although experience shows this contributes to inter-operability problems in the market.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 796: "Hybrid Broadcast Broadband TV".

[2] ETSI TS 103 285: "Digital Video Broadcasting (DVB); MPEG-DASH Profile for Transport of ISO BMFF Based DVB Services over IP Based Networks"

[3] ETSI TS 101 154: "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcast and Broadband Applications"

[4] ISO/IEC 23000-19: "Common Media Application Format"

[5] ISO/IEC 23001-7 (2016): "Information technology - MPEG systems technologies - Part 7: Common encryption in ISO base media file format files".

[6] Open IPTV Forum Release 2 specification, volume 5 (V2.3): "Declarative Application Environment".

[7] Open IPTV Forum Release 2 specification, volume 2 (V2.3): "Media Formats"

[8] W3C Recommendation 18 September 2017: "Encrypted Media Extensions"

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] DASH-IF IOP-6 V5.0.0 (2022-01): "DASH-IF Interoperability Points; Part 6: Content Protection"

[i.2] Google, “Getting Started with Widevine™ DRM”

[i.3] Google, “Widevine™ DRM Architecture Overview”

[i.4] Google, “Getting Started Widevine™ DRM on Devices”

[i.5] Google, “Widevine™ DRM Proxy Integration”

[i.6] Google, “Widevine™ Pssh Data proto format”, 2016

[i.7] ETSI TR 103 824 V1.1.1: Unit Test Descriptions for Hybrid Broadcast Broadband TV

[i.8] HbbTV Association: "Reference Video Application"

[i.9] Microsoft: "DASH Content Protection using Microsoft PlayReady™", Version 1.3, February 15, 2018

[i.10] Microsoft: “PlayReady™ Test and Documentation Server: Query String Syntax Documentation”

[i.11] Microsoft: "HbbTV Content Protection Using Microsoft PlayReady™".

NOTE: This document is only available to PlayReady™ licensees.

[i.12] Smart TV Alliance, “Widevine™ API Mapping”, Version 1.0, 7th January 2013

[i.13] LG, “DRM Messages”

[i.14] Microsoft: “Content Packaging and Delivery”

[i.15] Microsoft “PlayReady Header Specification”

[i.16] Microsoft: “License Persistence”

[i.17] Microsoft: “Security Level”

[i.18] Microsoft: “PlayReady Integration to HbbTV”, Version 0.95, February 2012

[i.19] Recommendation ITU-T X.667: "Information technology - Procedures for the operation of object identifier registration authorities: Generation of universally unique identifiers and their use in object identifiers".

[i.20] WHATWG: "HTML Living Standard"

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in TS 102 796 [1] and the following apply:

HbbTV® Test Suite: The set of unit test descriptions from TR 103 824 [i.7].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TS 102 796 [1] and the following apply:

CDM	Content Decryption Module
DASH-IF	DASH Industry Forum
EME	Encrypted media extensions
HD	High definition
HDR	High dynamic range
HFR	High Frame Rate
MSE	Media Source Extensions
OEM	Original Equipment Manufacturer
OIPF	Open IPTV Forum
SD	Standard definition
TA	Trusted application and/or targeted advertising
TEE	Trusted execution environment
US	United States of America
VOD	Video on Demand

4 Background (informative)

Generally, DRM systems require authentication, encryption and digital rights management (DRM). Authentication is used to ensure the content is being delivered to an approved device. Encryption is used to ensure the content is secure and cannot be accessed or tampered with by unauthorized parties. DRM is used to protect the content from illegal copying and distribution, with restrictions defined in dedicated service levels and output protection levels of the DRM systems. It also enables content providers to control who can access the content in a defined resolution, and the length of time it can be accessed.

Content licensors, especially the Hollywood studios, have been defining increasingly stringent content protection requirements and obligations, e.g. for HD 1080p, UHD 2160p / HDR format content or early release window content, which then have to be implemented by service providers / licensees. This may include the obligation to use approved high-level DRM systems that are already integrated with a hardware based Trusted Execution Environment (TEE), such as Microsoft PlayReady™'s SL3000 hardware-based security. End user devices using Microsoft PlayReady™ DRM and requesting UltraHD resolution licensed content are often required to ensure a SL3000 and HDCP 2.2 on digital outputs. As content protection technologies aren't standing still, further advanced profiles of modern DRM can be expected in the future.

While the terminals used to play HbbTV applications are generally equipped with current versions of DRM systems (as they are generally required by the non-HbbTV applications also supported on the terminals), many terminals have historically not fully integrated the HbbTV implementation and the DRM system, which prevented end users using HbbTV applications to access high value content.

5 DRM features (informative)

5.1 Basic features

The most basic use of DRM is to protect individual items of on-demand content. Specifically;

- The content is encrypted and identified as such (e.g. in a manifest or in the file format)
- Attempting to play content identified as encrypted content triggers a request for a license from a DRM license server.

NOTE: Normally this request would be handled by an application however in some cases a native DASH player may make a request without involving an application as explained in clause 5.4.

- The DRM license server delivers a license to the DRM implementation in the media consumption device. Normally this license will pass through an application however in some cases a native DASH player may process this response without involving an application.
- The DRM license contains at a minimum the content encryption key, and generally includes the terms of use or required protection for the content and a license validity duration
- The DRM implementation validates the license and extracts the key to decrypt the content
- The media decoding pipeline in the media consumption device is configured to include a decryption element that in turn is configured using the key
- The content is played

Some variations on the theme above include the following;

- The license may be pre-provisioned before attempting to play the content
- Only the video is encrypted and not the audio

Protecting live content is more complex than protecting on-demand content and often involves advanced features like key rotation addressed in clause 5.6.2 below.

5.2 Common encryption & multi-DRM

Traditionally, distributing digital content to multiple devices and platforms with different DRM technologies required content providers to separately encrypt their content for each DRM technology and device. This process created management complexities and increased costs for content owners. To address these challenges, common encryption was developed. Common encryption allows content owners to encrypt their content once using a single encryption key, which can be used across multiple devices and platforms with different DRM technologies. This approach simplifies content distribution management, reduces costs, and ensures a consistent user experience across different platforms. License acquisition and processing remains specific to each different DRM system.

Multi-DRM builds on top of common encryption by adding specific signalling for each supported DRM in a standardized format to the content encrypted with the common encryption key. This allows content to be accessed and decrypted using the appropriate matching DRM implementation on the device. This approach helps content owners to cater to different user preferences and device capabilities, while still providing a consistent user experience. Additionally, multi-DRM helps content providers to comply with industry standards and regulations for content protection, which is crucial in preventing piracy and unauthorized distribution of digital content.

Common encryption is defined by MPEG in ISO/IEC 23001-7 [5]. More information about using MPEG common encryption can be found in clause 8 of ETSI TS 103 285 [2] and DASH-IF "Interoperability Points; Part 6: Content Protection" [i.1].

5.3 AES encryption modes

ISO/IEC 23001-7 [5] defines a number of options for encryption of media data of which the most commonly used are 'cenc' (clause 10.1 of that document) and 'cbcs' (clause 10.4). Historically most DRM systems used 'cenc' except for Apple Fairplay that used 'cbcs' with an encrypt:skip pattern of 1:9 as defined in clause 10.4.2 of that document.

More recently, DRM systems have started supporting Apple-compatible 'cbcs' as well as 'cenc' however, at the time of writing, there is still a large installed base of devices only supporting 'cenc'. Supporting 'cbcs' in addition to 'cenc' is a trend in the industry as it enables unifying support across all devices with a common media format, Apple and others.

TS 102 796 [1] from V1.6.1 onwards defines an extension to the XML capabilities mechanism that enables applications to discover support for cbcs 1:9. The working draft of EME [8] has, since December 2019, defined an `encryptionScheme` of 'cbcs-1-9' for use in `MediaKeySystemMediaCapability` to discover support for cbcs 1:9.

NOTE: The reference video application [i.8] includes tests for cbcs 1:9 as well as 'cenc'. Clause **A.1.2** includes data on the extent of support for cbcs 1:9 in a sample of current and previous terminals.

5.4 Carriage of DRM information in ISO BMFF files and DASH manifests

5.4.1 DRM Information in ISO BMFF files

5.4.1.1 'pssh' box

The DRM may need a license from which it can generate one or more keys, identified by key IDs, to be used to decrypt the protected content. This license may be carried in one or more 'pssh' boxes, provided by one of the DRM APIs described in the present document or provided by some other means.

'pssh' boxes may be carried in the following locations:

- Within a ContentProtection element in the MPD, as described in clause 5.4.2. This is the preferred location for the 'pssh' box as it allows players to perform advance license acquisition, which is beneficial for the license servers as well to reduce peak loads.
- In the 'moov' box in the initialization segment. The same 'pssh' box is carried in this location for all Representations in the AdaptationSet. If a 'pssh' box for a DRM system is present within a `ContentProtection` element in the MPD, all 'pssh' boxes for that DRM system in the 'moov' box are ignored.
- In the 'moof' box in the media segment. The same 'pssh' box is carried in this location co-timed for all Representations in the AdaptationSet. This location is used when key rotation is employed, usually in association with a 'pssh' box in a `ContentProtection` element in the MPD.

If the 'pssh' box is carried in the media segments, it has to be present in every media segment in every Representation to ensure that the player is able to start playing the content from any location.

The DRM may require information from the 'pssh' boxes in both the initialization segment (or MPD, as appropriate) and the media segment in order to obtain license and keys.

Once a player has selected a DRM to use to decrypt the protected content, it can filter all of the 'pssh' boxes based on the UUID for that DRM, matching this against the value in the `SystemID` field in the 'pssh' box.

There may be one or more 'pssh' boxes in each location. After applying the above filtering, the player needs to pass the remaining 'pssh' box(es) to the DRM to enable it to acquire the license and generate the required keys.

5.4.1.2 Initialization Vector

The initialization vector for use when decrypting a media sample is carried in the `InitializationVector` field of the Sample Auxiliary Information, which is located using the corresponding byte offset and size stored in the Sample Auxiliary Information Offset ('saio') and Sample Auxiliary Information Size ('saiz') boxes. The Sample Auxiliary Information, which includes the initialization vector and NAL subsample byte ranges, may be stored in a Sample Encryption ('senc') box.

If the Sample Auxiliary Information is not present for a sample, there may be a default value in the 'tenc' box, for when pattern-based encryption is in effect, in the `default_constant_IV` field.

NOTE: Clause 10.4.1 of ISO/IEC 23001-7 [5] requires that constant IVs be used with cbcs encryption and excludes the use of sample IVs.

5.4.1.3 Key ID

Every protected media sample is encrypted using a key, which is identified by a key ID (KID). This is a 128 bit number, recommended to be generated as a UUID to ensure global uniqueness.

The key ID is carried in the sample group description 'seig' box in the KID field. This box also indicates whether or not the sample is encrypted (in the `isProtected` field). If the sample is not a member of a group or no 'seig' box is present for the group, the default key ID carried in the 'tenc' box in the `default_KID` field is used. This value may also be found in the MPD as described in clause 5.4.2.

5.4.2 DRM information in the MPD

In the MPD, protected content has at least one `ContentProtection` element on each AdaptationSet which contains protected media.

Each AdaptationSet has a generic `ContentProtection` element to advertise that it contains content encrypted according to CENC (ISO/IEC 23001-7) [5]. This element will have the `schemeUri` attribute set to "urn:mpeg:dash:mp4protection:2011" and the `value` attribute set to "cenc" or "cbcs", which is the same string as in the `scheme_type` field in the scheme type box 'schm' which is in the protection scheme information box 'sinf' box. The `value` attribute allows the player to determine the encryption mode without having to read any of the media segments.

For each DRM that could be used to obtain a key to decrypt the content, each AdaptationSet also has a `ContentProtection` element which may contain some of the elements and attributes defined by CENC (ISO/IEC 23001-7) [5] and/or elements as defined by the DRM provider. Such a `ContentProtection` element has the `schemeUri` attribute containing a UUID URN identifying the DRM.

NOTE 1: UUID strings are not case sensitive and may be expressed in upper or lower case. Refer to Recommendation ITU-T X.667 [i.19] for more information about how to format and construct a UUID in a URN.

A player can use the DRM-specific `ContentProtection` elements to determine whether there is a match with any embedded DRMs that it supports. Alternatively, it may have a priori knowledge that it contains a suitable DRM.

The elements and attributes defined by CENC (ISO/IEC 23001-7) [5] are:

- The attribute default_KID which may be added to the generic ContentProtection element. It is a copy of the value from the 'tenc' box. It contains a single key ID.
- The element pssh which may be added as a child of the DRM-specific ContentProtection element and contains a binary copy of the 'pssh' box that would allow a license/key server to generate the license bound to the requesting device and associated with the default key ID and content. This is normally the same as (and takes precedence over) any 'pssh' box that might be present in an initialization segment.

NOTE 2: The DASH-IF content protection and security specification [i.1] defines a dashif:laurl that can be used to add a license request URL to a DASH MPD in a DRM-system independent way. This is not believed to be used in the context of native DASH players in TS 102 796 [1] although it may be used included in JavaScript DASH players.

- Clause 5.8.4.1.2 of DASH [] defines an optional @robustness attribute that may be present in a content protection element to indicate the minimum robustness value required for playing back the associated Representations.

5.5 Clear Key

Clear Key is a mechanism defined in W3C EME [8] that provides some protection for content crossing the network to a media consumption device. As the name implies, the keys to decrypt the content are not protected except by whatever mechanism an HbbTV® application implements so Clear Key is inherently less secure than a commercial DRM.

Clear Key is understood not to meet the requirements of global content providers however it may be suitable for a broadcaster making their own content available.

When using Clear Key, the manifest only needs to include the cenc and clearkey drm elements. The init segment should signal just a generic cenc pssh (guid=1077efec-c0b2-4d02-ace3-3c1e52e2fb4b) box or no pssh boxes at all in the init segment. e.g.

```
<ContentProtection cenc:default_KID="44332211-1234-1234-1234-112233441236"
schemeIdUri="urn:mpeg:dash:mp4protection:2011" value="cenc"/>

<ContentProtection schemeIdUri="urn:uuid:e2719d58-a985-b3c9-781a-b030af78d30e" value="ClearKey1.0">
```

Combining Clear Key and regular commercial DRM in common encryption would be a serious concern for several organisations and is bad security practice. It would weaken the security of the other (commercial) DRM systems. Organisations considering doing this should explicitly confirm with their content and technology suppliers before doing so.

NOTE 1: It has been suggested that, for at least one commercial DRM, it would be incompatible with the agreements governing its use.

NOTE 2: This is explicitly prohibited by clause 8 of DASH-IF "Interoperability Points; Part 6: Content Protection" [i.1].

5.6 Advanced features

5.6.1 Persistent licenses

Typically, a new request for a license is made each time protected content starts to be played. Licenses would not survive beyond the duration of a session of media playback. DRM systems may support a feature often referred to as "persistent licenses" where DRM licenses persist beyond the end of a media playback session. This is particularly attractive where licenses are used that last a relatively long period such as 24 or 48 hours.

The combination of long-lasting licenses and persistence may significantly reduce the number of requests made to the DRM license servers. Since DRM license server operators may charge per license request, reducing the number of such requests has been felt to have a significant financial impact by some advertising funded broadcasters.

Long-lasting persistent licenses also enable delivery of licenses for popular content in advance. This reduces the delay between when content is requested and when it becomes visible (and audible) to end-users.

5.6.2 Key rotation and license rotation

In license rotation, a new license is obtained while playback continues. For DASH, this would occur at a Period boundary. See clause 9.2 of DASH-IF IOP 5 part 6 [i.1]. This is also sometimes referred to as "Periodic re-authorization".

In key rotation, the key needed to decrypt content is periodically updated without needing a query to be made to a license server. A number of variations of key rotation exist;

-
- New key/license information is carried in a pssh box in the media segments. See clause 9.3 of DASH-IF IOP 5 part 6 [i.1].
- There may be a hierarchy of keys/licenses where a root key/license authenticates a number of leaf keys/licenses that rotate to decrypt the content. See clause 9.4 of DASH-IF IOP 5 part 6 [i.1].

NOTE: Root / leaf key rotation is believed to be relatively poorly supported both technically and in the industry.

See also clause 8.2.2.3 of CMAF [4] and clause 8.5.1 of TS 103 285 [2].

5.6.3 Hardware DRM / trusted execution environment / security levels

DRM systems are designed to secure the access to media content, providing access only to authorized entities. To protect against access by unauthorized entities, DRM systems must ensure that the processing of DRM content licenses, content keys or the content itself is secured and cannot be easily bypassed or spied upon.

To access Premium content, Content Owners typically mandate that the DRM systems and the decrypted content are isolated in a separate, trusted execution environment (TEE), and that the isolation between such trusted execution environment and the main OS and applications is robust and enforced by hardware. The hardware guarantees that the resources used by the trusted environment are not accessible by the main OS and applications.

Various implementations of such trusted execution environments exist in the market today. The DRM implementations run as trusted applications (TAs) within the TEE. Beyond executing the DRM TAs, the TEE also provides a Secure Media Path that protects the content from decryption up to the display, or up to a protected output such as HDMI/HDCP or an encrypted recording.

Figure 1: Illustration of multiple security levels

The level of protection and robustness offered by a given DRM and content protection implementation is assessed and materialized through the Security Level assigned to a particular platform or device. A different dedicated Security Level is used for development or test devices, for production devices protected mostly by software means, and for production devices implementing a TEE and Secure Media Path protected by hardware means. This is shown in figure 1.

At the time that the content is protected, based on the requirements set for that particular content by Content Owners, the operator will set the minimum required device Security Level as well as the applicable content usage rules. This information will be embedded and carried in the content licence by the DRM.

NOTE: As explained in clause 10.2.11 of TS 102 796 [1], protection of audio with a license requiring a high Security Level may prevent audio from memory played via the Web Audio API from being heard when played simultaneously.

EXAMPLE: If audio is part of a DASH stream and is protected such that PlayReady SL3000 is needed for decryption then audio from memory may not be heard.

5.6.4 Transitioning between encrypted and unencrypted content

Transitioning between unencrypted and encrypted content or vice-versa is typically needed where adverts have been inserted or replaced as adverts are rarely encrypted. Typically, if content playback started with encrypted content then the media decoding pipeline will also be able to cope with unencrypted content, generally without artefacts. It is however important that a transition from encrypted to unencrypted content does not result in the license being discarded and having to be requested at a following transition back to encrypted.

If content playback started with unencrypted content (e.g. a pre-roll advert) then the media decoding pipeline may have been setup without including decryption logic. That would result in the media decoding pipeline having to be dismantled and rebuilt on the first occasion that playback of encrypted content is required. This may result in an extended period of black, e.g. at the transition from the pre-roll advert to the main content.

NOTE: In DASH, transitioning from encrypted content to unencrypted content or vice-versa at a Period boundary may be accompanied with other changes that stress implementations. Examples include changes in the number and perhaps id of audio tracks and/or subtitles tracks.

5.6.5 Key per track or key per resolution

Because of the varied types of devices deployed in the field, with different DRM implementations and security levels, content owners or content distributors may wish to protect the highest resolutions or the best features like HDR, Next

Generation Audio, or various audio languages with different keys and licenses. Using different keys and licenses for various resolutions or features, it becomes possible to restrict their use only on devices with the highest security level and/or to monetize them with a higher fee.

- In key per track, a different key is used for the various elements of the content (generally one key for video and another one for audio tracks).
- In key per resolution, a different key is used for the highest video resolution(s) (typically UHD), compared to other resolution (typically SD/HD).
  In a DASH manifest, key per resolution is indicated as multiple Adaptation Sets with the supplemental property "urn:mpeg:dash:adaptation-set-switching:2016" although support for this is not required by TS 103 285 [2].

5.7 Authenticating license requests

DRM license servers may require requests for a license to be authenticated using some kind of access token.

- Where the EME API (see 6.3.3) is used, the application is responsible for sending the request to the license server and can add the access token as required for the license server concerned.
- Where the oipfDrmAgent API (see 6.3.2) is used, the DRM agent sends the request to the license server. How the access token is supplied is DRM system and/or license server specific.

6 Media streaming and DRM integration in HbbTV (informative)

6.1 Native DASH players

TS 102 796 has, since version 1.2.1, required support for a DASH player as part of the terminal implementation, generally referred to as the "native" DASH player. This is required to support content according to DVB-DASH, TS 103 285 [2].

6.2 MSE DASH players

TS 102 796 has, since version 1.6.1, required support for the W3C Media Source Extensions (MSE) **Error! Reference source not found.**. These enable DASH players to be provided as part of an HbbTV application, e.g. as a JavaScript library. Examples of DASH players that are JavaScript libraries include dash.js (https://dashjs.org/, https://github.com/Dash-Industry-Forum/dash.js) and Shaka (https://github.com/shaka-project/shaka-player).

6.3 DRM APIs

6.3.1 Introduction

TS 102 796 [1] includes two APIs for DRM, the older oipfDrmAgent API (see clause 6.2) and the newer Encrypted Media Extensions (see clause 6.3). The oipfDrmAgent API can also be used for broadcast conditional access as well as DRM. Applications should only use one of these for DRM. Using both APIs for DRM during the lifetime of the same application may result in unpredictable behaviour.

Use of oipfDrmAgent for broadcast conditional access should be independent of use of that API for DRM. Applications should be able to use Encrypted Media Extensions for DRM simultaneously with using oipfDrmAgent for broadcast conditional access.

TS 102 796 [1] defines one DRM-related API call, the `setActiveDRM` method. When terminals support more than one DRM system, this method enables applications to control which of the possible DRM systems is used and which are not used. Applications can disable the terminal-specific algorithm for selecting which DRM system and instead force a particular DRM system to be used. This is particularly relevant when DASH MPDs include one or more license request URLs and some DASH players may support obtaining a license without the involvement of an application.

6.3.2 oipfDrmAgent

The Open IPTV Forum (OIPF) JavaScript API for content protection is built round the `application/oipfDrmAgent` embedded object defined in clause 7.6.1 of the OIPF DAE specification [6]. The key features are as follows;

- Applications send messages to DRM systems using the method `sendDRMMessage( String msgType, String msg, String DRMSystemID )` where `DRMSystemID` identifies the DRM system, `msgType` is a DRM system-specific identifier for the type of message being sent and `msg` is the body of the message.
- Responses from the DRM system to the application are sent to the function `onDRMMessageResult( String msgID, String resultMsg, Integer resultCode )` or the equivalent `DRMMessageResult` event.
- DRM systems may send messages to the application that are not a response to `sendDRMMessage` using function `onDRMSystemMessage( String msg, String DRMSystemID )` or the equivalent `DRMSystemMessage`. DRM systems may not use this mechanism.

Applications are able to query the supported DRM systems by searching the XML capabilities for <`drm`> elements. See clause 10.2.4.7 of TS 102 796 [1] and clause 9.3.10 of the OIPF DAE specification [6].

This is a very old API dating back before the beginning of HbbTV®. It is widely supported for Microsoft PlayReady™ and, to a lesser extent, Intertrust Marlin.

6.3.3 Encrypted Media Extensions

The W3C Encrypted Media Extensions API (EME) [8] are a much more recent API than oipfDRMAgent. Since 2018 the EME API is deployed in almost all web browsers.

NOTE: See https://caniuse.com/eme

The API design and architecture are very different to oipfDRMAgent. The architecture features a Content Decryption Module (CDM) that contains all the DRM system functionality (referred to as Key Systems in that document). All communication between the CDM and license servers goes through the application. In contrast to the oipfDRMAgent API, the CDM is prohibited from communicating directly on the network (see clause 8.1 of EME [8]).

Application use of EME requires calling a number of methods in a specific order. Example code is found in clause 13 of EME [8], particularly clause 13.2.

Applications need to create and disable the `oipfDRMAgent` object when using EME by calling the `setActiveDRM` method of the `oipfDRMAgent` with an argument value of "urn:hbbtv:oipfdrm:inactive".

Applications should close the EME key session once the DRM license has been acquired to limit the memory footprint of multiple EME sessions and prevent `QuotaExceeded` exceptions being raised when reloading an EME session for persistent licenses. Licenses can still be used to decrypt media after the EME session has been closed. Applications should listen for the `keystatuseschanged` event and close the session when any `MediaKeyStatus` carries a value of usable.

Where the EME session is of type persistent-licence, the Application needs to store each EME `sessionId` in persistent storage such as WebStorage or IndexedDB after the EME session has been created. When the Application document is reloaded, the Application or EME supported media player needs to retrieve the previously persisted EME `sessionId` from persistent storage and create a new EME session using `MediaKeys.createSession()`. The Application or media player needs to then invoke the `load` method of the returned `MediaKeySession` with the `sessionId` argument.

6.4 Device capabilities

Applications are able discover DRM capabilities of terminals as follows;

- With the XML capabilities (see clause 10.2.4.7 of TS 102 796 [1] and clause 9.3.11 of the OIPF DAE specification [6]), support for the DRM feature is required to be indicated by a DRM element and adding a DRM attribute to `video_profile` and `audio_profile` elements.

NOTE: Only the DRM element is used in practice and not the DRM attribute.

- With the EME API (see clause 3.2.1 of EME [8]) the `requestMediaKeySystemAccess` method is used to request access to a particular DRM system - which is required to be denied if that DRM system is not supported.

Applications should not assume that a DRM system only supported via the EME API (i.e. not via oipfDrmAgent) is included in the XML capabilities. Since all DRM systems are supported by EME, the opposite does not apply. Using the XML capabilities is however more likely to give an accurate answer on the installed base.

7 Selection of a DRM system and version

7.1 General

The present document does not choose a DRM system or version. This is a commercial choice for stakeholders.

NOTE 1: Clause A.1.3 explains that, in 2022 and 2023 terminals, Microsoft PlayReady version 3.0 or greater was found to be supported in the context of HbbTV by 95% and Widevine by 65%. It is expected that, when measured across the installed base, the PlayReady % figures will keep increasing over time. Any increase in the Widevine percentage would likely depend on adoption of that technology in combination with HbbTV by content providers, platforms and operators.

NOTE 2: A survey of 19 multi-DRM service providers in the CTA WAVE project identified that only 3 DRM systems were pervasively supported, FairPlay, PlayReady and Widevine. 3 other DRM systems were found that were only supported by multi-DRM service providers were owned by the same parent company as a DRM system provider. One other DRM system was found that was supported by only 2 multi-DRM service providers without any obvious ownership relationship.

7.2 Content providers, platforms and operators

Application developers, content providers, platforms and operators (and any other organization that is in the position of influencing the features of terminals in a given market) should note that better interoperability can be expected if they decide on a DRM system for which a detailed and open integration specification for HbbTV exists. The present document enables a more consistent and inter-operable integration between HbbTV implementations and DRM by providing such an integration specification for the DRM systems that it includes.

Obviously, there are many factors that content providers and/or platforms need to consider when deciding which DRM system (and which version) to use for HbbTV services. While some of these are obvious, the following are worth highlighting.

- The requirements from the various content owners and content providers for protecting the content (see clause 8.2.2). Specifically requiring support for hardware DRM (see clause 5.6.3) as a condition for access to any content will prevent many users with older terminals from accessing the content.
- The extent to which its's important to reach the widest installed base of HbbTV terminals.

NOTE: The two factors above may conflict to some extent.

- Requiring the support of advanced features as a condition for access to any content will prevent many users with older terminals from accessing the content.

Decisions will need to be communicated and may need consultation with other stakeholders in the industry. If consultation is needed and a content provider operates in a market where there is a platform or operator or national trade association then this consultation may be organised through that body. Otherwise the content provider may have to organise this themselves. If the content provider is one of the first to use a particular advanced feature with HbbTV then careful consideration of test applications and test streams (see clause 8.2.3.3) becomes even more important.

Once a final decision is reached, this decision needs to be documented and communicated to implementers. If the content provider or platform already has a specification for access to their services then it could be included in there. In theory a national trade association with a specification could document such a decision in their document however historically some DRM suppliers found that controversial. It is anticipated that part of documenting such a decision would be to make at least one optional annex of the present document be mandatory for services and terminals within the stakeholder's scope.

7.3 Terminals and terminal implementers

Neither the present document nor TS 102 796 [1] require implementations to support a specific DRM system in combination with HbbTV® applications. TS 102 796 [1] V.1.8.1 and above require at least one of the DRM systems in the present document be supported.

Terminal implementers should determine which DRM systems and versions are used in combination with HbbTV® applications and services in the markets where the specific terminal may be used. This information should be made available by broadcasters, platforms and operators.

Terminal implementers may choose one or more than one of the DRM systems included in the present document to support in combination with HbbTV®.

- In the present document, it is optional for terminals to support PlayReady™ in combination with HbbTV® applications. If this option is supported, then the terminal requirements specific to this system are found in Annex B.
- In the present document, it is optional for terminals to support Widevine™ in combination with HbbTV® applications. If this option is supported, then the requirements terminal requirements specific to this system are found in Annex C.
- Terminal implementers may consider supporting more than one DRM system in combination with HbbTV® as an insurance policy, particularly where the terminal has to support that DRM system for a feature, function or application unrelated to HbbTV®.

Terminal implementers should note that application developers, content providers, platforms and operators are likely to decide on a DRM system for which a detailed and open integration system for HbbTV exists. In combination with testing, they would expect better interoperability from such a choice. The present document enables a more consistent and inter-operable integration between HbbTV and DRM by providing such an integration specification for the DRM systems it includes.

8 Recommendations for stakeholders (informative)

8.1 Introduction

Most of the recommendations in this clause are forward looking and aspirational. They are based on the hoped evolution of the ecosystem over time following the publication of the present document, of V1.8.1 of TS 102 796 and related actions. Recommendations for stakeholders based on the current status of the ecosystem can be found in clause 8.4.

8.2 Recommendations for content providers, platforms and operators

8.2.1 Introduction

This clause contains recommendations for content providers intending to use DRM with HbbTV® for the first time or to make significant changes to already deployed HbbTV® applications using DRM. Obviously, some of the contents of this clause will be well known to some organisations but it is expected that some will find some of this clause useful and helpful.

8.2.2 Identifying the content protection requirements

Clarity about which requirements apply to which content and the consequences is critical.

- US produced content typically has the highest requirements, particularly for UHD. It may be that only devices produced since 2022 - 2023 meet the content protection requirements for UHD. These requirements may become more demanding over time.
- Conversely content owned by the content provider themselves may not even need protection by commercial DRM, ClearKey may be sufficient. Enabling content providers to avoid paying for commercial DRM if the content owner does not require DRM is why ClearKey was added to the HbbTV® specification.

Reconciling these requirements with the DRM support in the installed base and commercial objectives to reach that installed base may be challenging. It is possible that only SD content can be distributed to devices from before 2015 - 2020, even ones supporting HD that were high end devices at the time of purchase. The latter will have consequences, e.g. for how the service is marketed.

8.2.3 Risk assessment and mitigation

8.2.3.1 Risk assessment

Content providers planning a new HbbTV® service or a significant change or enhancement to an existing service should do a risk assessment whether DRM is involved or not. For services involving DRM, some example risk factors that should be considered include the following;

- Is the service on-demand (which is lower risk) or Live/Linear (which is higher risk)?
- Does the service require any of the advanced DRM features (see clause 5.6)? Are they widely implemented on a range of different HbbTV® equipment?
- How much experience does the supplier, or the development team have with DRM on HbbTV®?

NOTE: Suppliers with Smart TV experience but without HbbTV® experience should not be considered low risk. The HbbTV® ecosystem is more diverse than individual Smart TV ecosystems.

- Are the terminals targeted by the service covered by a certification regime, e.g. operated by an operator or platform or on behalf of them? Does that certification regime certify the DRM functionality or feature (see clause 8.2.3.5)?
- How much testing is planned for the service before market introduction (see clause 8.2.3.3)?
- How much would the reputation of the content provider be damaged if there would be problems with the service or application?

Here are some examples:

- Low Risk
  A VOD service only using features already known to work in a wide range of HbbTV® implementations with a supplier with experience delivering to the variety of HbbTV® terminals.
- Medium Risk
  A VOD service is unlikely to be more than medium risk unless it needs to use features not previously used or implemented or unless the supplier or development team have little or no HbbTV® experience.
- High Risk
  A live service with immature features, a supplier with limited experience of DRM on HbbTV® and/or there being limited time for testing.

Medium and high risks can be mitigated by choosing a supplier that is more experienced with HbbTV® and/or measures as listed below.

Deploying services in markets where there is no certification regime for terminals is inherently higher risk than deploying them in markets where there is a such certification regime. Clearly this is not something that is easily changed but it is important to recognise that testing of implementations is essentially voluntary without the presence of a certification regime.

8.2.3.2 Mitigation - Communication with implementers

NOTE: This clause is very important for high-risk services, useful for medium-risk services and may be reviewed for low-risk services.

Communication with implementers is important. A non-exclusive list of points that should be considered include the following:

- Unless the DRM system and supplier for encoding, packaging & encryption are already present in that market, implementers should be given notice of new services or significant revisions to existing services such as adding DRM where there was no DRM previously or changing a supplier for DASH encoding, packaging, encryption and DRM license serving.
- High risk changes (as above) that may not work with existing deployed terminals should be communicated with significantly more notice than lower risk changes. In extreme cases, more than a year’s notice could be needed if a content provider would be the first HbbTV® service in the world to use a particular complex feature or function.

- Content providers should plan to have resources available to respond to questions and feedback from manufacturers. Where parts of the workflow and/or system are outsourced, content providers should ensure their suppliers have resources available as well.

8.2.3.3 Mitigation - Test applications and test streams

NOTE: This clause is very important for high-risk services, useful for medium-risk services and may be reviewed for low-risk services.

There is no substitute for a content provider having a test application and test streams that use the exact same supplier, equipment, tools and settings as their real service uses or will use. The importance of this rises as the risk services associated with the service rises. These enable voluntary testing by the many attentive, conscientious manufacturers.

- A representative test stream and license request application should be made available to manufacturers a reasonable time before a new or updated service is made available to users.
  - License requests for such a test stream could either be hidden in an existing live app behind a special key sequence or use a new dedicated app. In the latter case, it would be helpful for manufacturers if a transport stream signalling that app could be provided and not just rely on each manufacturer building their own such stream including a URL provided by the content provider.
  - For test applications, it is necessary to provide instructions for a journey through the application and to document the expected results at each step.
- It is helpful for test streams to continue to be made available after a service has been launched for regression testing as part of an annual product cycle and when manufacturers introduce new product families.

NOTE: The HbbTV DASH-DRM reference application [i.8] and unit tests are not generally a substitute for the above except, to some extent, where a content provider has chosen to use the exact same supplier(s) including for encoding and packaging.

8.2.3.4 Mitigation – Enhanced testing by the application / service developer

NOTE: This clause is very important for high-risk services, useful for medium-risk services and may be reviewed for low-risk services.

Many application / service developers will test applications and services on a small number of terminals themselves. One way to reduce risk is increasing the number of terminals on which they test the app or service. This may be done by increasing the number of terminals in-house or obtaining the services of a so-called “receiver zoo”. For more information and information on receiver zoos, see https://www.hbbtv.org/hbbtv-zoo-directory/ .

This testing would be under the developer’s control and would be possible to plan unlike voluntary testing by terminal implementers.

8.2.3.5 Mitigation – Certification of Terminals

NOTE: This clause is very important for high-risk services, useful for medium-risk services. The cost and overhead are probably excessive for low-risk services.

Content providers and platforms may operate a certification scheme for terminals, probably linked to some kind of ‘allow list’ of certified terminals. Certifying a terminal may require testing of specific applications on the terminal and passing specific unit tests from those referred to by TR 103 824 [i.7].

If a content provider believes it appropriate to certify terminals as running a specific application, factors that need to be considered include the following:

- What level of proof will be required of implementers? Is it sufficient for them to deliver a test report demonstrating tests were passed or a recording of an app running? If so, who will scrutinise those?
- How will incoming requests to access a service be checked that they come from a tested / certified device? Is a list of certified HTTP user agent strings sufficient? While in theory TLS client certificates can be used to authenticate an incoming request to access a service, in practice the administrative overhead is significant and very few service providers use these. How will apps obtain the list of certified HTTP user agents or equivalent?

If content providers or platforms believe the set of unit tests described by TR 103 824 is not sufficient, it is recommended that they either;

- Commission tests to be donated to that set or
- Produce unit test descriptions so that set can be extended following its normal lifecycle and process

8.3 Confirming the Security of the Application Environment

In addition to a DRM (which secures the content distribution) some commercial use cases may also require some 'hardening' of the HbbTV® application environment to help prevent DRM licenses being granted to unauthorised users. HbbTV® specification provides some mechanisms such as clause 11.8 of TS 102 796 [1] to protect application assets (for example stored cookies) from being extracted from devices.

It should be noted that this isn't tested by the HbbTV® Test Suite and there is no other verification mechanism provided by HbbTV®. Service providers should arrange to check with individual manufacturers to confirm that their HbbTV® implementation on the models that their application is deployed to is sufficiently secure for their purposes.

8.4 Recommendations given the 2024 status of the ecosystem

NOTE: This clause contains recommendations for content providers, platforms, operators, national trade associations and terminal implementers based on the status of the ecosystem at the time this document is prepared – 2023 / 2024 and based on version 1.7.1 and earlier of TS 102 796.

At the time of writing, the state of the ecosystem is such that deploying a new HbbTV app or service using DRM is almost never low risk in spite of what is described in 8.2.3.1. As a consequence, the most basic DRM service or application is at least medium risk and some of risk mitigation measures are almost always needed. For example:

- Simplifying the scope of the application, at least initially, to fit within the scope of what is demonstrated to work by the DASH-DRM reference application (see clause A.1.2)
- Increasing investment in test applications, test streams and communication with manufacturers (see clause 8.2.3.3)
- Testing on a wider range of terminals than might otherwise be done (see clause 8.2.3.4)
- For high risk or high-profile applications, certifying the application on terminals and operating an 'allow list' of certified terminals (see clause 8.2.3.5).

The following is applicable if a platform or operator (or indeed a content provider) wanted to extend the set of unit tests referenced from TR 103 824 [i.7] to include testing of DRM.

- A simple approach is to refer to an existing set of DRM unit tests used by other operators or platforms. In the short term, this reduces the initial investment by the platform or operator and may be quick to put in place. How it would work in the medium term could be organisationally complex.
- A more complex approach is for the platform or operator to develop (or have developed) a set of functional requirements for DRM unit tests and obtain their own unit tests. Depending on the supplier, these may be based on unit tests for another operator or platform. How the maintenance of these tests would work in the medium term also needs consideration.
- In either case, the platform or operator needs to consider the issues mentioned in clause 8.2.3.5 concerning implementers providing evidence of passing unit tests and there being someone to scrutinise that evidence on behalf of the platform or operator.
- Some platforms have multiple alternative suppliers of unit tests for DRM.

9 Requirements for terminals

9.1 DRM-system independent requirements

9.1.1 Media playback

The following media playback scenarios shall be supported with DRM encrypted content for each combination of DRM system, media playback API and license request API supported by a terminal.

- Playback of encrypted video with encrypted audio
    - Where supported by a DRM system, this shall include both scenarios with the same key for video and audio and ones with different keys (see clause 5.6.5)
- Playback of encrypted video with unencrypted audio
- Random access to encrypted video starting from the start of a CMAF fragment that is not the start of a piece of content
- Random access to encrypted video starting from a time that is not the start of a CMAF fragment
- Random access to encrypted video and audio starting from a time that is not the start of a CMAF video fragment
- Playback starting with encrypted content, switching to unencrypted content (e.g. adverts or interstials) and back to encrypted content (repeatedly)
- Playback starting with encrypted content, switching to a different piece of encrypted content and then returning to the original piece of encrypted content (repeatedly).
- Playback that switches between encrypted DASH Representations within an Adaptation Set.

NOTE: Although simple and obvious to many, the above requirements have been made explicit to ensure unit tests are properly justified by clear specification language.

- Discarding any encrypted Adaptation Set or Representation where a suitable DRM system is supported but a valid license cannot be obtained or which does not meet the DRM @robustness level.
- Switching between encrypted Periods and unencrypted Periods and vice-versa, repeatedly, including where the number and ids of Adaptation Sets change at Period boundaries, e.g., different Periods have different numbers of Adaptation Sets for each of video, audio and subtitles.
- Live DASH including where the playback point is behind the live edge, e.g. by a number of seconds.

Terminals shall support decryption of all video and audio codecs that are supported for content delivery by broadband in a fragmented mp4 container.

9.1.2 Advanced playback of encrypted content

The following scenarios shall be supported with DRM encrypted content for each combination of DRM system, media playback API and license request API supported by a terminal.

- Key changes within a piece of content including:
    - Decryption stops, the app obtains a new license and re-starts playback and decryption, decoding and presentation. (repeatedly)
    - A new license being obtained proactively before decryption has stopped (repeatedly)
- Persistent licenses (where supported by the particular DRM system)
    - Support for persistent licenses shall mean not requesting a new license when a terminal already has one that can be used.

- Terminals shall store persistent licenses that have not expired in a way that shall survive routine power cycling of the terminal.

NOTE: Persistent licenses need not survive first-time installation or similar terminal factory-reset options.

- Terminals shall be able to store at least 16 persistent licenses.
- When there is insufficient space to store a persistent license, terminals shall discard any expired licenses first. If there is still insufficient space then the policy is outside the scope of the present document.

- The following shall apply for terminals that support hardware DRM (see clause 5.6.3):
  - AVC video shall be supported with hardware DRM at resolutions up to and including 1080p.
  - For terminals that support both HEVC video and UHD, hardware DRM shall be supported at resolutions up to and including 2160p.
  - Terminals shall support hardware DRM for video with a lower security level for audio – hence different KIDs. The present document does not require that terminals supporting hardware DRM for video also support for hardware DRM for audio.

9.1.3 Media profile and bandwidth requirements

Terminals supporting DRM with HbbTV applications shall support presentation of DRM protected content carried over HTTP/TLS for which the combined channel bandwidth does not exceed the bitrates listed in clause 7.3.1.2 of TS 102 796 [1].

9.1.4 Transition behaviour

The transition behaviour defined in clause 9.6.3 (and annex J) of TS 102 796 [1] shall apply where either one of the two media elements refers to content protected using DRM delivered via MPEG DASH or a MediaSource object. It is optional where both elements refer to content protected using DRM. If the second media element refers to content protected using DRM then the 250ms requirement only applies if a license that can decrypt the content has already been obtained,

9.1.5 Requirements from TS 102 796

The requirements in the following clauses from TS 102 796 [1] are particularly critical.

- Section 9.6.7 in relation to playback failure due to lack of a license that can decrypt DRM protected content
- Section 10.2.4 in relation to reporting the presence of one or more DRM systems using the capabilities reporting mechanism
- Annex F in relation to DRM Integration

9.1.6 API specific requirements

9.1.6.1 oipfDrmAgent

On terminals where the oipfDrmAgent API (see clause 6.3.2) is supported in combination with HbbTV applications, the following requirements are highlighted.

- If a license server does not return a license in response to `sendDRMMessage` being called then `onDRMMessageResult` shall be called with `resultCode`=2 and `resultMsg` being a DRM system specific value (see clause 7.6.1.1 of the OIPF DAE specification [6]).
- When using an HTML5 video element and the native DASH player, failure to decrypt content shall result in the `HTMLVideoElement` generating an 'error' event with its `'error'` property set to a `MediaError` object with code `MEDIA_ERR_DECODE` (see clause 9.6.7 of TS 102 796[1]). Examples of reasons for this include the following;
  - The available license does not give the rights to play video and audio

- The terminal does not support any DRM system that can decrypt that content
- If the terminal supports a DRM system that can decrypt the content but the terminal has no license and no way of obtaining one
- If the terminal has a license that has expired before playing the content, while playing the content or while playback of the content is paused
- Additional DRM-system specific reasons may apply.

9.1.6.2 EME

On terminals where the EME API (see clause 6.3.3) is supported in combination with HbbTV applications, the following requirements are highlighted.

- The present document does not require support for use of EME with the native DASH player and DRM (see clause 6.1)

NOTE: Support for EME with the native DASH player is required by TS 102 796 [1] for the ClearKey key system.

- Changing of initialization segments over time is required to be supported including changing between initialization segments that indicate encrypted media and ones that indicate non-encrypted media and vice-versa (see EME [8] clause 7.5.2 "Initialization Data Encountered").
- Expiration of licenses is required to be supported (see `MediaKeySession.expiration` in EME [8]).
- Where a DRM system supports persistent licenses, requesting these using a `MediaKeys` object with a `MediaKeySessionType` of `"persistent-license"` (see EME [8] clause 5), the CDM shall not request a new license if it already has one that can be used to decrypt content to be played.
- Providing messages (including licenses) to the CDM is required to be supported (see `MediaKeySession.update` in EME [8]).
- Requesting a supported key system configuration (see the Get Supported Configuration algorithm, clause 3.2.2.1 in EME [8]).
- When using an HTML5 video element and MSE, failure to decrypt content shall result in both i) an event named `error` being fired at the `HTMLVideoElement` with its `'error'` being property to a `MediaError` object with code `MEDIA_ERR_DECODE` (see “If the media data is corrupted” in clause 4.8.115 of HTML [i.20]) and ii) the `MediaKeyStatus` of the key concerned being set as defined in clause 6.3.2 of EME [8].
- If `MediaKeySystemMediaCapability.robustness` is specified by the application and the value is supported by the terminal then it shall be followed. If `robustness` is not specified by the application then the most secure robustness level supported by the terminal shall be selected.

9.2 DRM-system specific requirements

In addition to the DRM-system independent requirements in the previous clause, DRM-system specific requirements shall also apply for each of the DRM systems are supported in combination with HbbTV applications. These requirements are found in the annex to the present document for each DRM system concerned.

10 Criteria for adding DRM systems

Further DRM systems may be considered for addition in a future revision of the present document subject to meeting the following requirements;

- There is some recent existence proof of an integration of the DRM system with an HbbTV terminal implementation.

NOTE 1: A proof of content inside an engineering organisation is sufficient to meet this requirement. Commercial introduction of a product is not required.

- Unit tests have been defined and their descriptions independently reviewed that cover the integration between the DRM system and an HbbTV terminal implementation. The unit test descriptions are in the XML format used by TR 103 824 [i.7].

NOTE 2: Creation of unit tests according to the reviewed descriptions is not a requirement for this clause.

- Some (limited) publicly available documentation on the DRM system exists.

NOTE 3: It is fully accepted that detailed technical documentation on a DRM system will be confidential and only available under license.

- The DRM system has been added to the HbbTV DASH-DRM reference app [i.8] in a way that is compatible with the text that is proposed to be added to the present document.

NOTE 4: This addition could be in a fork owned by the DRM system supplier and does not have to be in the trunk however, it is expected that there would be no technical or legal barriers to merging the fork into the trunk if the DRM system would be added to the present document.

- There is a documented conformance / certification process for devices that implement the DRM system that operators, platforms and content providers can rely on.

NOTE 5: It is expected that details of such a process are confidential, but they need to exist.

- There is a request to include the DRM system by at least two stakeholders that are independent of the owner of the DRM system, at least one of which is a platform or content provider that would use the DRM system as specified in the present document.

Annex A (informative): Status of the ecosystem

A.1 Market data

A.1.1 Methodology

In 2017, the HbbTV Improving Interoperability Task Force initiated the development of an extensive testing tool specifically designed for conducting Digital Adaptive Streaming over HTTP (DASH) Digital Rights Management (DRM) tests in HbbTV terminals. The primary objective of this tool is to facilitate comprehensive testing procedures for ensuring optimal functionality and compatibility in HbbTV-enabled devices.

In the course of executing these testing procedures, a comprehensive collection of test results has been amassed. This repository of data serves as a valuable resource, providing market insights to HbbTV member companies. To ensure privacy and confidentiality, the shared data is anonymized, thereby safeguarding the identities of the involved entities.

The responsibility for managing and overseeing this undertaking falls under the purview of the HbbTV Improving Interoperability Task Force, which assumes the leadership role in coordinating and guiding the project's development and execution.

Usage statistics are collected and presented for each month, and historical data is available since January 2019.

Around 80 devices from 2017 – 2024 are involved in the daily testing cycle. Tests include basic DRM tests with different configurations, and also more advanced tests, including Security Level 3000 for PlayReady™ and L1-3 for Google Widevine™, multi-DRM signalling, DASH events, persistent licenses, multiple audio and subtitle tracks and so on. There are three profiles:

- "OIPF"
  - Uses the A/VObject and the OIPF DRM object for video playback – i.e. what is supported on terminals implementing TS 102 796 V1.2.1 or greater.
- "HTML5"
  - Uses the HTML5 video element and the OIPF DRM object for video playback – i.e. what is supported on terminals implementing TS 102 796 V.1.4.1 or greater.
- "MSE-EME"
  - Uses MSE-EME extensions and dash.js player (https://dashjs.org/, https://github.com/Dash-Industry-Forum/dash.js) for playback – i.e. what is supported formally in terminals implementing TS 102 796 V1.6.1 or greater and informally in many earlier terminals.
  - This is also compatible with PC browsers, Xboxes and other general web-connected media consumption devices.

A comprehensive tool set ('dasher') is included to create DRM content with various profiles and configurations, including multi-period content for creating streams that simulate the results of SSAI having happened.

Documentation and information on how to use the tools can be found in the public repository of the project at https://github.com/HbbTV-Association/ReferenceApplication [i.8].

A.1.2 Results

Based on the data from running the DASH DRM Reference application tests, table D.1 presents the state of the various DRM features in commercially available devices from 2017 onwards.

Table D.1: State of DRM features in TS 102 796 implementations

Test	MSE-EME		HTML5 Video object + OIPF DRMAgent	AVObject + OIPF DRMAgent
	2017-2023	2017-2024		
PlayReady™ CENC encryption	41%	64%	92%	92%
PlayReady™ Different KID for A/V	38%	52%	47%	45%
PlayReady™ Security Level 3000	34%	46%	40%	40%
Widevine™ SL1 and SL3	39 %	43%	4 % -> 9%	8%
PlayReady™ CBCS encryption	22%	28%	0%	0%
Widevine™ CBCS scheme	27%	45%	0%	0%
Multiperiod DRM	75%	78%	7%	--
Multi-DRM with PR and WV (*)	55%	55%	77%	78%
Persistent Licenses	--	5%	43%	44%

Key;

- PR = “PlayReady™”; WV = “Widevine™”
- Green is >= 67%
- Yellow is < 67% but > 33%
- Red is <= 33%
- Grey is no data available yet or feature not supported on enough devices to be statistically useful.

Numbers are not weighted by market significance.

(*) The `setActiveDRM` method is not used

It should be noted that there is an upward trend in better support for PlayReady™ SL3000 and Widevine™ in EME mode for example in more recent devices.

A.1.3 2022, 2023 and 2024 feature support

Most terminals from 2022 and 2023 meet the following requirements:

- PlayReady™ version 3.0 or better is supported in 95% of devices (PlayReady™ can be missing because of licensing or specific market issues)

NOTE 1: When considering 2023 and 2024 devices, this increases to 97%.

- The oipfDRMAgent API is supported with the following specific features working:
  - CENC
  - Separate KID values for video and audio
  - Security Level 2000 for the above
  - Security Level 3000 is supported in 70% of devices

NOTE 2: When considering 2023 and 2024 devices, this increases to 75%.

- The EME API is supported in 84 % of the devices, with the following specific features working:

NOTE 3: When considering 2023 and 2024 devices, this increases to 87%.

  - PlayReady™ CENC, all devices, all devices supporting EME API

  - Separate KID values for video and audio, all devices supporting EME API
  - PlayReady™ Security Level 2000, all devices supporting EME API
  - When considering 2023 and 2024 devices, 85% support the PlayReady™ Security Level 3000 with the EME API.
- Widevine™ SL1 and SL3, supported in 65% of the devices supporting EME API

NOTE 4: When considering 2023 and 2024 devices, this falls to 60%.

A.2 Known issues

A.2.1 Persistent licenses

Tests have indicated that many implementations incorrectly implement persistent licenses. Specifically, that they always request a new license even if they already have a license that can decrypt the content concerned. This breaks the specific reason for using persistent licenses – reducing the number of license requests to DRM license servers.

A.2.2 MediaKeySession.expiration

In EME, the `expiration` property of the `MediaKeySession` can be used by the Application to proactively acquire a new license prior to the expiry of the license in order to avoid a discontinuity in playback whilst a new license is retrieved. In some circumstances, the `expiration` property may return a value of `NaN`, in which case applications should not rely on this property and determine the expiration time of the license out of band from the `MediaKeySession` API.

A.2.3 setActiveDRM

Implementations of the `setActiveDRM` method are believed to have issues on many existing implementations. Content providers and applications should avoid situations where this method would be useful, specifically to avoid including license request URLs in 'pssh' boxes - see clause 5.5.

A.2.4 Version 1 of the ‘pssh’ box

The pssh box is defined in ISO/IEC 23001-7 [5]. Version 0 is widely used with commercial DRM systems and version 1 with ClearKey. At the time of writing, version 1 has started appearing in streams generated for commercial DRM systems however it is known that some (unknown) proportion of terminals do not support pssh version 1 with an empty KID loop. Use of pssh version 0 is recommended to reach the largest number of terminals.

A.2.5 Failure to reload an EME session

Implementations exist where reloading an EME session for a persistent license results in a `QuotaExceeded` exception. Although it contradicts the EME specification, as a work-around, applications may try listening for the `keystatuseschanged` event and close the session when any `MediaKeyStatus` carries a value of `usable`. Licenses may still be usable to decrypt media after this.

A.2.6 Failure to present content when robustness is specified

Some terminals have been observed to fail to present content when `MediaKeySystemMediaCapability.robustness` is specified. This is particularly true for implementations prior to the present document and / or implementations not distributed in the UK.

Annex B (normative): PlayReady™

B.1 General

The present document does not require support for PlayReady™. This annex is optional but shall be supported where PlayReady™ is available to HbbTV applications and normative statements in this annex only apply when that is the case.

When PlayReady™ is available to HbbTV applications, at least version 4.4 shall be supported.

In case of any conflict between the contents of this annex and the PlayReady™ requirements, the latter take precedence.

B.2 Background information and context

Information about the various features that can be specified when making a license request from a PlayReady™ license server can be found in Microsoft: “PlayReady™ Test and Documentation Server: Query String Syntax Documentation” [i.10].

Information about PlayReady messages and headers can be found in “PlayReady Header Specification” [i.15].

Two modes of license acquisition are defined for PlayReady in combination with TS 102 796[1]:

- “Pre-acquisition” that is triggered by an HbbTV application
- Automatic “post-acquisition” triggered by the media player

PlayReady includes support for all the advanced features listed in clause 5.6.

PlayReady includes support for both ‘cenc’ and çbcs’as listed in clause 5.3.

B.3 HbbTV APIs

B.3.1 oipfDrmAgent

B.3.1.1 General

Terminals shall support use of PlayReady™ with the oipfDrmAgent API (see clause 6.3.2). This is defined in "HbbTV Content Protection Using Microsoft PlayReady™" [i.11] and summarised below. Some publicly available information may be found at “DRM Messages” [i.13].

PlayReady™ in combination with the `oipfDrmAgent` object shall be supported for content played using the A/V control object and using the HTML5 Media Element with the native media player (i.e., where the `src` attribute of the media element is an HTTPS URL referring to a DASH manifest).

NOTE: The present document does not define the `oipfDrmAgent` object to be usable where the content is obtained by the HbbTV application using the fetch or XMLHttpRequest APIs and passed to the terminal using MSE. Applications should not make any assumptions about whether it works or how it would fail if it does not work.

B.3.1.2 sendDRMessage

Terminals shall support applications sending messages to the DRM system using the `sendDRMessage( String msgType, String msg, String DRMSystemID)` method as follows:

- `msgType` shall be “application/vnd.ms-playready.initiator+xml”
- `msg` shall be a PlayReady Initiator string, UTF8 encoded
- `DRMSystemID` shall be “urn:dvb:casystemid:19219”. This is case-insensitive and may be in capitals.

- the return value shall be as defined in the OIPF DAE specification [6]

An example PlayReady Initiator string is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<PlayReadyInitiator xmlns= "http://schemas.microsoft.com/DRM/2007/03/protocols/">
<LicenseAcquisition>
   <Header>
   <WRMHEADER xmlns= "http://schemas.microsoft.com/DRM/2007/03/PlayReadyHeader" version="4.0.0.0">
      <DATA>
         <PROTECTINFO>
           <KEYLEN>16</KEYLEN>
           <ALGID>AESCTR</ALGID>
         </PROTECTINFO>
         <LA_URL>http://rm.contoso.com/rightsmanager.asmx</LA_URL>
         <KID>lFmb2gxg0Cr5bfEnJXgJeA==</KID>
         <CHECKSUM>P7ORpD2IpA==</CHECKSUM>
      </DATA>
   </WRMHEADER>
   </Header>
   <CustomData>AuthZToken XYZ</CustomData>
</LicenseAcquisition>
</PlayReadyInitiator>
```

The WRMHEADER element is defined in the "PlayReady Header Specification" [i.15]. As the CHECKSUM attribute of the KID element is optional, terminals shall support both its presence and its absence.

The following PlayReady Initiators are defined.

Initiator	Mandatory (M) / Optional (O)	Description	Reference
<LicenseAcquisition>	M	License pre-acquisition	
<JoinDomain>	O	Not defined in the scope of the present document.	
<LeaveDomain>	O	Not defined in the scope of the present document.	
<Metering>	O	Not defined in the scope of the present document.	
<LicenseServerUriOverride>	M	After this message, all further license post-acquisition need to send the license challenge to the specified LA_URL and not to the default LA_URL contained in the WRMHEADER of the content. Pre-acquistions shall not be effected.	
<SetCustomData>	M	Sets a value to be used by the PlayReady Client stack for all the further license post-acquisitions as the CustomData value.	
<LicenseQuery>	M	Query the existing rights for the content.	

B.3.1.3 onDRMMessageResult

The function `onDRMMessageResult( String msgID, String resultMsg, Integer resultCode )` of the `oipfDrmAgent` object shall be supported as defined in the OIPF DAE specification [6].

The `resultMsg` is an XML string eformatted as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="PlayReadyResponse">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="DRM_RESULT" type="xs:unsignedLong" minOccurs="1" maxOccurs="1" />
          <xs:element name="CustomData" type="xs:string" minOccurs="0" maxOccurs="1" />
          <xs:element name="LicenseQueryState" minOccurs="0" maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="DRM_LICENSE_STATE" type="xs:string" minOccurs="1" />
                <xs:element name="DATETIME" type="xs:dateTime" minOccurs="0" maxOccurs="2" />
                <xs:element name="VAGUE" type="xs:int" minOccurs="0" maxOccurs="1" />
              </xs:sequence>
            </xs:complexType>
```

```
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Typical responses are much simpler than the above schema.

Possible values for the elements in the result code are in table 2.

Table 2: `onDRMMessageResult.resultMsg` **elements**

Element	Value	Meaning
DRM_RESULT		Shall be set to 0x8004C600 when a PlayReady licence server cannot perform a licence delivery in response to a call to `sendDRMMessage`.
CustomData	CustomData inserted in the license response by the server.	
DRM_LICENSE_STATE	NORIGHT	No rights found
	UNLIM	Rights found, with no restriction
	FROM	Rights found, with a fixed begindate restriction
	UNTIL	Rights found, with a fixed expiration date restriction
	FROM_UNTIL	Rights found, with a fixed begindate and a expiration date restriction
	EXPIRATION_AFTER_FIRSTUSE	Rights found, with a expiration period set after the first playback – first playback hasn't occurred yet
DATETIME	xs:dateTime	Date and Time related to the time restrictions of the license (UTC)
VAGUE	xs:int	'1' means that the ights returned are result of the aggregation of multiple rights from multiple licenses and may not reflect the whole complexity of all the rights

B.3.2 EME

Terminals shall support use of PlayReady™ with the EME API (see clause 6.3.3) where the content is obtained by the HbbTV application using the `fetch` or `XMLHttpRequest` APIs and passed to the terminal using MSE. The following additional requirements shall apply:

- Terminals shall support and applications should use the `com.microsoft.playready.recommendation` key system in order to ensure greatest compatibility between the CDM and the referenced EME specification.
- Applications should proactively set the expected robustness level when requesting access to the key system to avoid indeterministic behaviour when the default robustness value is chosen.

Terminals shall support and applications should signal PlayReady™ security levels with the security level as a number wrapped in a string e.g. "3000" within the `robustness` property of the `audioCapabilities` or `videoCapabilities` object.

- Applications or media players need to be tolerant to messages returned by the CDM using UTF-8, UTF-16 or UTF-32 character encoding.
- Applications or media players need to be tolerant to CDM messages wrapped in an XML document.

B.3.3 HTML video element using either oipfDrmAgent and native DASH player or EME and MSE

When content presented using an HTML video element fails to be presented due to PlayReady for any of the PlayReady-specific below reasons, the following shall apply:

- An event named `'error'` shall be fired at the `HTMLVideoElement` and its `'error'` property set to a `MediaError` object with code `MEDIA_ERR_DECODE`.

- Where the combination of EME and MSE are used, the `mediaKeyStatus` shall be set as defined in clause 5.3.2 of EME [8].

Examples of PlayReady-specific reasons include the following:

- If 'scalable key rotation' is being used (see clause B.6.2) and any of the following apply:
  - the root license / key expires while content is being presented
  - the root license / key has expired before content is presented
  - the leaf license / key expires while media is presented

NOTE: The PlayReady™ documentation uses the terms "root license" and "root key" as well as "leaf license" and "leaf key".

B.4 DASH

The usage of PlayReady™ DRM with DASH is defined in [i.9].

B.5 Basic features

With reference to the list of "Required PlayReady Functionalities" in clause 2.6 of [i.18], the present document requires support for the following functionalities when PlayReady is supported according to this Annex.

- Direct License Pre-Acquisition (through `sendDRMMessage`)
- Direct License Automatic Post-Acquisition
- LicenseServerUriOverride, Set CustomData
- Persistent and Non Persistent Licenses
- LicenseQuery
- Trusted Client Clock (anti-rollback clock or secure clock)
- Respond to Server Error Codes

The present document does not require support for the following PlayReady functionalities, although other documents may require support for these.

- Domain support (Domain Join, Leave, Renewability)
- Bind embedded license
- Embed license in content
- Metering
- Indirect License Acquisition (through a PC host, e.g. via USB MTP)
- Indirect License Synchronization through initiators

B.6 Advanced features

B.6.1 Persistent licenses

NOTE: See also see clause 5.6.1

PlayReady supports persistent and non-persistent (transient) licenses. See clause 2.5 of PlayReady Integration to HbbTV [] and License Persistence [i.16].

B.6.2 Key rotation and license rotation

NOTE: See also clause 5.6.2.

PlayReady supports an advanced version of key rotation called “scalable key rotation”. In this option, there is a root license and key for a stream or track that authenticates multiple leaf licenses / keys. This is described in the “Content Packaging and Delivery” document [i.14].

The present document does not require support for “scalable key rotation” although other documents may require support for this.

B.6.3 Hardware DRM / trusted execution environment / security levels

NOTE: See also clause 5.6.3.

PlayReady defines Hardware DRM as “Security Level” 3000. This is described in Security Level [i.17]. Terminals shall support SL3000. A description of how SL3000 can be supported with the EME API is found in clause B.3.2. .

B.6.4 Transitioning encrypted content to unencrypted & back

NOTE: See also clause 5.6.4.

Terminals shall support transitioning from encrypted content to unencrypted content and vice-versa.

B.6.5 Key per track or per resolution

NOTE: See also clause 5.6.5.

Terminals shall support different KIDs for video and for audio. Terminals shall support different security levels for video and for audio, e.g. SL3000 for video and SL2000 for audio.

Seamless switching between different resolutions is only supported within a single Adaptation Set. Hence "key per resolution" would require encrypting HD content twice, once for an HD Adaptation Set and once for a UHD Adaptation Set.

B.7 Authenticating license requests with oipfDRMAgent

Providing an access token to authenticate license requests with the oipfDRMAgent API may be license server specific. One possibility is to include the access token in custom data that is passed to the DRM system using the "<SetCustomData>" initiator referred to in clause B.3.1.2 above.

Annex C (normative): Widevine™

C.1 General

The present document does not require support for Widevine™. This annex is optional but shall be supported where Widevine™ is available to HbbTV applications. Normative statements in this annex only apply when that is the case.

When Widevine™ is available to HbbTV applications, at least Widevine™ DRM version 7 (specifically the Widevine™ Modular version) shall be supported.

In case of any conflict between the contents of this annex and the Widevine™ requirements, the latter take precedence.

C.2 Background information and context

Widevine™ DRM is Google's content protecting system. It provides the way to securely distribute and protect playback of premium contents on consumer devices. The Widevine™ DRM platform today uses standards-based royalty-free solutions for encryption, adaptive streaming, transport, and player software [i.2].

Most specifications and details are provided by Widevine™ only to Certified Widevine™ Implementation Partners (CWIP) or at least under the MLA (Master License Agreement).

Widevine™ Modular is the Widevine™ DRM version required, providing support for open standards such as DASH, HLS, Common Encryption (CENC), Encryption Media Extension (EME) and Media Source Extensions (MSE).

The following diagram shows a Widevine™ architecture overview [i.3]. All the client architecture components, such as OEM Crypto Module and Content Decryption Module (CDM) are defined in [i.4].

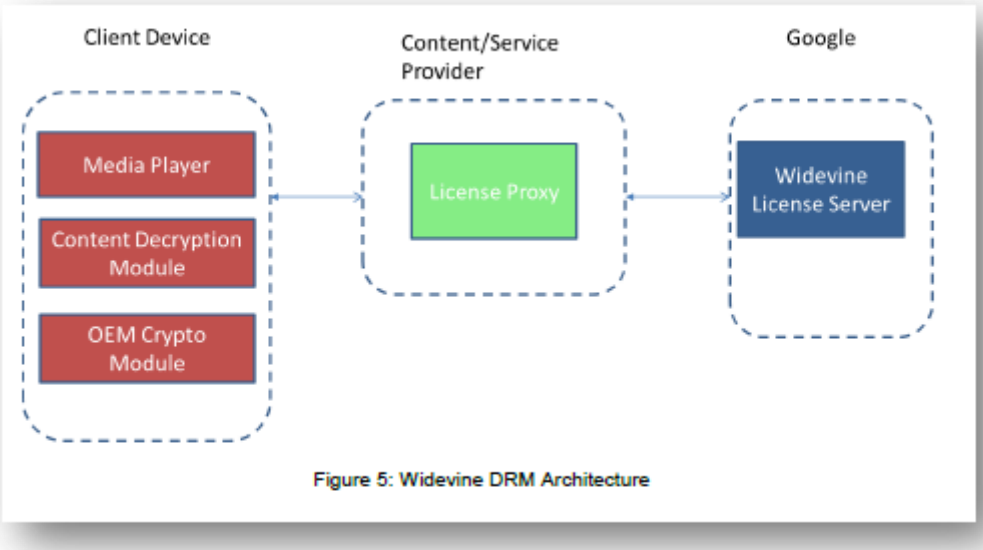

Figure 2: Widevine™ DRM architecture

With respect to the figure 2.

- All the interactions between client devices and Widevine™ License Server are via a license proxy. The license proxy component can be implemented only by a Certified Widevine™ Implementation Partner (CWIP).
- All the interactions between client and proxy and license server are via HTTPS.
- The Widevine™ License Server interface is defined by Widevine™.
- The interface between client devices and license proxy is not defined by Widevine™ but, being it handled at application level, interoperability between services and client devices can be achieved without specifying it here.

Table 3 summarizes the Widevine™ Device Security Level and HDCP Version required to protect contents at different resolutions.

Table 3: Mapping between content resolutions, Widevine™ Security Level and HDCP version

Content resolution	Widevine™ Security Level	HDCP Version
Less than 720p	L3	1.4
720p to 1080p	L1	1.4
4k or UHD	L1	2.2

The interface between client devices and license proxy is not defined by Widevine™ but, being it handled at application level, interoperability between services and client devices can be achieved without specifying it in the present document.

C.3 HbbTV APIs

C.3.1 oipfDrmAgent

The present document does not require support for the use of Widevine™ with the oipfDrmAgent API (see clause 6.3.2). Specifically, there is no requirement to support the Smart TV Alliance, "Widevine™ API Mapping" [i.12].

C.3.2 EME

Terminals shall support use of Widevine™ with the EME API (see clause 6.3.3) where the content is obtained by the HbbTV application using the fetch or XMLHttpRequest APIs and passed to the terminal using MSE (see 6.2). There is no requirement to support Widevine™ with the native DASH player (see 6.1).

C.4 DASH

Delivery of Widevine™ DRM protected contents with MPEG-DASH SHALL be compliant to [2].

If the schemeIdUri attribute of one of the <ContentProtection> elements in the MPD equals to "urn:uuid:edef8ba9-79d6-4ace-a3c8-27dcd51d21ed", the protected media segments are signaled as Widevine™ DRM.

For Widevine™ DRM protected content, the Common Encryption for ISO Base Media file format SHALL be used as defined in [5]. The coding of the PSSH box defined in [i.6] shall be supported, in particular with reference to the algorithm, key_id, provider, content_id and protection_scheme fields.

C.5 Advanced features

The advanced features defined in clause 5.6 shall be supported as defined in table 4.

Table 4: Advanced features for Widevine™

Feature	Reference	Requirement
Persistent licenses	5.6.1	Widevine requires persistent and non-persistent licenses. The duration of validity and validity after first usage can be defined. A renewal of an expired license is possible. Widevine does not have a specific name for persistent licenses, licenses can be configured with status flag "persistent"
Key rotation and license rotation	5.6.2	Widevine supports license rotation but does not support key rotation.
Hardware DRM / trusted execution environment	5.6.3	An OEMCrypto component that complies with Level 1 (L1) Widevine™ Device Security shall be supported.
Transitioning encrypted content to unencrypted content & back	5.6.4	Terminals shall support transitioning from encrypted content to unencrypted content and vice-versa. This shall be reliable when performed repeatedly (see clause 9.1.1).
Key per track or per resolution	5.6.5	Key per track shall be supported. Switching between different keys is not required to be seamless.
'cbcs' encryption	5.3	Both cbcs 1:9 and cenc shall be supported.

Annex D (informative): Marlin

Marlin DRM is not included in the present document.

While Marlin DRM is included in the Reference Video Application [i.8], this is a historical legacy.

Annex (informative): Change history

Date	Version	Information about changes
<Month year>	<#>	<Changes made are listed in this cell>

History

Document history		
<Version>	<Date>	<Milestone>

Latest changes made on 2023-04-20