**3GPP TSG- Meeting #**

**, , -**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** |  | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** |  | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** |  |  | | | | | ***Release:*** | | |  |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | The work and call flow presented in the initial document is not fully aligned with common practices. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The call flow is updated to address the additional information. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Confusion for normative work phase. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 3.3, 5.10.3, 5.10.4, 6.10 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

## ===== CHANGE =====

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[187] Unified Streaming: "Introduction to CPIX", Unified Streaming online blog, https://docs.unified-streaming.com/documentation/drm/cpix\_intro.html

[188] castLabs: "FAQ: Digital Rights Management", https://castlabs.com/faq/drm/

## ===== CHANGE =====

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

ABR Adaptive Bit Rate

ACK ACKnowledgment

ACM Association for Computing Machinery

ALPN Application-Layer Protocol Negotiation

AMF Access and Mobility Management Function

ANBR Access Network Bitrate Recommendation

ANTS Advanced Network Technologies Symposium

API Application Programming Interface

AQM Active Queue Management

ARPA Advanced Research Projects Agency

ARQ Automatic Repeat reQuest

ASP Application Service Provider

ATSSS Access Traffic Steering, Switching, and Splitting

AUS AUthentication Server

AVC Advanced Video Coding

AWS Amazon Web Services

BBC British Broadcasting Corporation

BBR Bottleneck Bandwidth and Round-trip propagation time

BDT Background Data Transfer

BMFF Base Media File Format

BMSC Broadcast/Multicast Service Center

CAE Content-Aware Encoding

CBR Constant Bit Rate

CCF Call Control Function

CDF Cumulative Distribution Function

CDN Content Delivery Network

CDP Content Delivery Protocol

CERN European Organization for Nuclear Research

CIRR Carrier-Independent Routing Registry

CMAF Common Media Application Format

CMAS Commercial Mobile Alert System

CMCD Common Media Client Data

CMMF Coded Multisource Media Format

CMSD Content Media Server Data

CPI Content Protection Information

CPIX Content Protection Information Exchange

CPT Content Preparation Template

CRUD Create, Read, Update, Delete

CTA Consumer Technology Association

CWR Congestion Window Reduced

DANE DASH-Aware Network Element

DASH Dynamic Adaptive Streaming over HTTP

DCSM Data Collection and Storage Management

DNS Domain Name System

DRM Digital Rights Management

DSCP Differentiated Services Code Point

DTT Digital Terrestrial Television

DVB Digital Video Broadcasting

DVR Digital Video Recorder

EAS Edge Application Server

ECE Explicit Congestion Notification

ECN Explicit Congestion Notification

ECP Enhanced Content Protection

ECT Explicit Congestion Notification-Capable Transport

EDGE Enhanced Data rates for GSM Evolution

EDL Enhanced Data Link

EEC Edge-Enabled Client

EEL End-to-End Latency

EES Edge Enabler Server

EFDT Enhanced File Delivery Table

EME Encrypted Media Extensions

ENP Enhanced Network Performance

EPS Evolved Packet System

EPT Enhanced Packet Transport

EVEX EVent EXposure

FAR Forward Action Rule

FDIS Final Draft International Standard

FDT File Delivery Table

FEC Forward Error Correction

FHD Full High Definition

FLUS Framework for Live Uplink Streaming

FLUTE File Delivery over Unidirectional Transport

FQDN Fully Qualified Domain Name

GBR Guaranteed Bit Rate

GRO Generic Receive Offload

GSO Geostationary Satellite Orbit

GTP GPRS Tunneling Protocol

GUID Globally Unique Identifier

HDR High Dynamic Range

HEVC High Efficiency Video Coding

HLS HTTP Live Streaming

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IANA Internet Assigned Numbers Authority

ICCE International Conference on Consumer Electronics

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IMS IP Multimedia Subsystem

IOP Interoperability Points

IPC Inter-Process Communication

IPTV Internet Protocol Television

ISO International Organization for Standardization

JSON JavaScript Object Notation

JTC Joint Technical Committee

KID Key IDentifier

KPI Key Performance Indicator

LPT Low-Power Transceiver

LSD Low-Speed Data

LTE Long-Term Evolution

MABR Multicast Adaptive Bit Rate

MANO Management and Orchestration

MAR Multi-Access Rule

MBMS Multimedia Broadcast Multicast Service

MBS Multicast Broadcast Service

MBSF Multicast Broadcast Service Function

MBSTF Multicast Broadcast Service Transport Function

MHV Mile-High Video

MIME Multipurpose Internet Mail Extensions

MNO Mobile Network Operator

MoQ Media over QUIC

MPD Media Presentation Description

MPEG Moving Picture Experts Group

MPQUIC Multipath QUIC

MPTCP Multipath TCP

MQTT Message Queuing Telemetry Transport

MSE Media Source Extensions

MTSI Multimedia Telephony Service for IMS

NAT Network Address Translation

NBMP Network-Based Media Processing

NEF Network Exposure Function

NFV Network Functions Virtualization

NRF Network Repository Function

NRM Network Resource Model

NSACF Network Slice Admission Control Function

NSCALE Network Slice Capability Exposure

NSSAI Network Slice Selection Assistance Information

NWDAF Network Data Analytics Function

OAM Operations, Administration, and Maintenance

OMA Open Mobile Alliance

OTI Object Transmission Information

OTT Over-The-Top

PAS Publicly Available Specification

PCC Policy and Charging Control

PCF Policy Control Function

PDF Portable Document Format

PDR Packet Detection Rule

PDU Protocol Data Unit

PFCP Packet Forwarding Control Protocol

PFD Packet Flow Description

PFDF Packet Flow Description Function

PFS Packet Flow Set

PHB Per-Hop Behaviour

PING Packet Internet Groper

PLMN Public Land Mobile Network

PSA Public Service Announcement

PSDB Packet Set Delay Budget

PSER Packet Set Error Rate

PSIHI Packet Set Integrated Information

PSNR Peak Signal-to-Noise Ratio

PSS Packet-Switched Streaming

PSSH Protection System Specific Header

QER QoS Enforcement Rule

QFI QoS Flow Identifier

QLOG QUIC Logging

QoS (Network) Quality of Service

QPACK QUIC Header Compression

QRT QUIC RTP Tunnelling

QUIC Quick UDP Internet Connections

RAN Radio Access Network

RFC Request for Comments

RIST Reliable Internet Stream Transport

RLC Radio Link Control

ROM Receive-Only Mode

RSFEC Reed-Solomon Forward Error Correction

RTC Real-Time Communication

RTCP Real-Time Control Protocol

RTMP Real-Time Messaging Protocol

RTP Real-Time Protocol

RTT (Network) Round-Trip Time

SAND Server and Network Assisted DASH

SAP Service Access Point

SBI Service-Based Interface

SCTE Society of Cable Telecommunications Engineers

SDAP Service Data Adaptation Protocol

SDF Service Data Flow

SEAL Service Enabler Architecture Layer

SEI Supplemental Enhancement Information

SIB System Information Block

SIM Subscriber Identity Module

SLA Service Level Agreement

SMF Session Management Function

SMS Short Message Service

SMTP Simple Mail Transfer Protocol

SRT Secure Reliable Transport

SSD Seek Start-up Delay

SST Slice Selection Type

STS Secure Token Service

SVC Scalable Video Coding

SVTA Streaming Video Technology Alliance

TBD To Be Determined

TBF Token Bucket Filter

TCP Transmission Control Protocol

TEE Trusted Execution Environment

TEID Tunnel Endpoint Identifier

TLS Transport Layer Security

TMGI Temporary Mobile Group Identity

TOI Transmission Object Identifier

TSG Technical Specification Group

TSI Transport Stream Identifier

TZM Time Zone Management

UDP User Datagram Protocol

UDR Unified Data Repository

UHD Ultra-High Definition

UMTS Universal Mobile Telecommunications System

UPF User Plane Function

URI Uniform Resource Identifier

URL Uniform Resource Locator

URN Uniform Resource Name

URR Usage Reporting Rule

USIM Universal Subscriber Identity Module

UTC Coordinated Universal Time

UUID Universally Unique Identifier

VBR Variable Bit Rate

VSF Video Services Forum

WAVE Web Application Video Environment

WDD Workflow Description Document

XML Extensible Markup Language

ZSM Zero-touch network and Service Management

## ===== CHANGE =====

### 5.10.3 Deployment Architectures

The core components of a DRM workflow are provided in figure 5.10.3-1 based on Part 6 of the DASH-IF Interoperability Points specification [144]. Similar workflows apply also for other streaming technologies, but DASH is used in the below analysis for illustrative purposes.

A diagram of a software system

Description automatically generated

Figure 5.10.3-1: Core elements in content protection according to DASH-IF IOP Part 6 [144]

The definition of the functions is as follows:

- **Authorization Server**: provides authorization tokens that may be required for requesting a license from a license server.

- **DASH client**: a function using the Media platform and the DRM system to playback encrypted content.

- **DASH Presentation**: A server hosting DASH resources, i.e. MPDs and Segments primarily, and includes information on the used DRM System.

- **License Server**: A license server provides licenses that are data structures in a DRM system specific format that contains one or more content keys and associates them with a policy that governs the usage of the content keys (e.g. expiration time).

- **Media Platform**: enables playback of encrypted content while protecting the decrypted samples and content keys against potential attacks.

- **DRM System**: an implementation of content keys management cooperating with the device’s media platform to enable playback of encrypted content while protecting the decrypted samples and content keys against potential attacks, consisting of two main components: a license server and a DRM client.

- **DRM Client:** processes licenses and enforcing the associated policies. Either the DRM client handles the decryption of samples, or the DRM client interacts with the hardware elements that address the decryption.

A DRM system cooperates with the device’s media platform to enable playback of encrypted content while protecting the decrypted samples and content keys against potential attacks. The same encrypted DASH presentation can be decrypted by different DRM systems if a DASH client is provided the DRM system configuration for each DRM system, either in the MPD or at runtime. A content key is a key used by a DRM system to make content available for playback. A content key and its identifier can be shared between all DRM systems, whereas the mechanisms used for key acquisition and content protection are largely DRM system specific. DASH adaptation sets are often protected by different content keys. The encapsulated content keys are typically encrypted and only readable by the DRM system.

A more detailed DRM workflow is provided in figure 5.10.3-2 based on the Content Protection Information Exchange Format (CPIX) as specified in ETSI TS 103 799 [143]. It complements Part 6 of the DASH-IF Interoperability Points specification [144] by putting more emphasis on the back-end aspects. The following additional functions are defined:

- **Content Provider**: A publisher who provides the rights and rules for delivering protected media, also possibly source media (mezzanine format, for transcoding), asset identifiers, key identifiers (KID), content key values, encoding instructions, and content description metadata.

- **Encoder**: A function that encodes media in a specified set of formats with different bitrates and resolutions etc., possibly determined by the publisher.

- **Packager/Encryptor**: A function that who encrypts and packages media, inserting DRM signalling and metadata into the media files.

- **Manifest Creator**: A function that generates the media manifests which group the various media files into a coherent presentation. These manifest files may contain DRM signalling information.

- **DRM Client**: Obtains information from different sources: media manifest files, media files, and DRM licenses.

An example architecture is provided in figure 5.10.3-2.

A diagram of a system

Description automatically generated

Figure 5.10.3-2: Example backend architecture and workflow for encrypted live content  
based on ETSI TS 103 799 [143]

In this case, content is continuously received, transcoded in the desired format and encrypted if any type of protection is required. One or multiple content keys can be used regardless of whether key rotation is used or not. Keys are generated by the encryption engine or the DRM system and are available to all DRM systems and encryption engines to support multi-DRM with a shared key The MPD Generator requests to the DRM systems their specific signalling, if any, to be added in the MPD. Encrypted segments and the media manifest can be uploaded on a CDN making it available to users.

Figure 5.10.3-3 illustrates the usage of the encrypted content in a realistic workflow comprising multiple cooperating components. In ETSI TS 103 799 [143], a standardised data format for content protection information exchange is defined, collected in a document that can be signed.

A black background with a black square

Description automatically generated with medium confidence

Figure 5.10.3-3: Incremental update and extension of the document based on ETSI TS 103 799 [143]

Also, in ETSI TS 103 799 [143] a workflow for use with CPIX is presented for which multiple producers are included. This workflow is shown in figure 5.10.3-4.

A diagram of a system

Description automatically generated

Figure 5.10.3-4: Multiple producer example steps based on ETSI TS 103 799 [143]

Based on the collection of data above, as well as further deployment guidelines, the following functions are defined when distributing DRM-protected content using multiple DRMs and common encryption:

- **DRM system**: an implementation of content key management cooperating with the device’s media platform to enable playback of encrypted content while protecting the decrypted samples and content keys against potential attacks. It is comprised of two main components: a License Server and a DRM Client.

- **Authorization Server**: Provides authorization tokens that may be required for requesting a license from a License Server.

- **License Server**: Provides data structures in a DRM system-specific format that includes one or more content keys and associates them with a policy that governs the usage of these content keys (e.g. expiration time).

- Interacts with the Authorization Server to verify user credentials.

- Interacts with the Key Server to obtain the necessary decryption keys.

When a user requests playback of DRM-protected content, the media player's request is sent to the License Server. The License Server verifies the user's credentials and entitlements, and if authorized, it retrieves the decryption keys from the Key Server and issues a license to the media player.

- **Key Server**: Responsible for generating, storing, and managing encryption keys used to encrypt the content. Provides the encryption keys to the Encryptor during the content encryption process. These keys are then used to encrypt the content before distribution. The Key Server typically interacts with the Encryptor and the License Server to ensure that the correct keys are used for both encryption and decryption processes.

- **Encoder:** Converts raw video and audio content into a compressed format suitable for streaming. Typically generates multiple versions of the content at different bit rates to support adaptive streaming. May also produce different variants of the content, for example an HD version, a UHD version, and HDR version, etc. The encoder does not interact with the DRM system.

- **Encryptor**: Applies encryption to the encoded content using encryption keys. This ensures that the content is protected and can only be accessed by authorized users. Communicates with the Key Server to obtain the encryption keys needed to encrypt the content.

- **Packager**: Receives encrypted content from the Encryptor and formats it into streaming protocols like DASH (MPD) and HLS (M3U8). Also inserts DRM metadata into the packaged media segments.

- **Media Entry Point creator**: Generates presentation manifest documents (e.g., DASH MPD, HLS M3U8) that describe how the content is organized and how it should be accessed and played back. These presentation manifests include DRM metadata that informs the media player about the encryption and how to obtain the decryption keys.

- **Content Hosting Server**: A server hosting media streaming resources, i.e. primarily Media Entry Points and Segments and media segments, and which includes information about the used DRM system.

- **Media Platform**: Enables playback of encrypted content while protecting the decrypted samples and content keys against potential attacks. Summarises the functions of media decapsulation, media decryption, media decoding and media rendering as defined in clause 4.2.2 of TS 26.501 [15].

- **DRM Client:** Processes licenses and enforces the associated policies. Either handles the decryption of samples, or interacts with hardware decryption elements.

- **Media Player**: A function using the Media Platform and the DRM system to play back encrypted content.

NOTE: In many DRM workflows, only the key identifiers (KIDs) are exchanged rather than the actual encryption keys. This approach enhances security by minimising the exposure of the actual keys.

A typical example workflow of encrypted content is shown in figure 5.10.3-5.

Msc-generator~|version=8.6.1~|lang=signalling~|size=1409x1944~|text=numbering=yes;~nhscale=auto;~ndefcolor lgrey=224,224,224;~n~nhide UE [label=~qUE~q, fill.color=lgray]{~n~4APP[label=~qApplication~q]; ~n~4Player [label=~q5GMSd Media Player~q, fill.color=lgrey]{~n~8MP[label=~qMedia\nPlatform~q];~n~8DC[label=~qMedia Access\nClient~q];~n~8DRMC[label=~qDRM\nClient~q];~n~4};~n};~n~nCS [label=~q5GMSd AS~q, fill.color=lgray]{~n~4DP[label=~qContent\nHosting~q];~n~4MC[label=~qManifest\nCreator~q];~n~4ENP[label=~qEncryptor/\npackager~q];~n~4ENC[label=~qEncoder~q];~n};~n~nFAppProvider [label=~q5GMSd Application Provider~q, fill.color=lgray]{~nDRMS [label=~qDRM System~q, fill.color=lgrey]{~n~4LS[label=~qLicense\nServer~q];~n~4KS[label=~qKey\nServer~q];~n};~n~nAP [label=~qApplication Service Provider~q, fill.color=lgrey]{~n~4CP [label=~qContent\nProvider~q];~n~4AUS[label=~qAuthorization\nServer~q];~n};~n};~n~n~nvspace 10;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iContent preparation\i~q {~n~4CP-~gENC-~gENP-~gMC: Encoding instructions and usage rules\n\_M1d/M3d\_[strong]; ~n~4CP-~gENC: Raw content\n\_M2d\_; ~n};~n~nvspace 10;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iTrust establishment\i~q {~n~4ENP~l-~gLS~l-~gKS~l-~gAUS: Exchange public keys\n\_M1d/M3d\_[strong];~n};~n~nvspace 5;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iContent Protection\nInformation construction\i~q {~n~4ENP--ENP: Packager constructs\nContent Protection Information\n\_identification of the receivers and usage rules\_;~n~4ENP-~gKS: Request one or several content keys\n\_M2d\_[strong]; ~n~4KS--KS: Generate\ncontent keys;~n~4KS-~gENP: Content keys [strong];~n~4ENP--ENP: Extract and store\ncontent keys;~n};~n~nvspace 5;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iContent Protection Information distribution\i~q {~n~4ENP-~gLS: Content Protection Information\n\_M2d\_[strong];~n~4LS--AUS: Update Content Protection Information\nwith DRM Metadata;~n~4LS-~gENP-~gMC: Updated Content Protection Information\n\_M2d\_[strong];~n};~n~nvspace 5;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iPresentation manifest and media segment generation\i~q {~n~4MC--MC: Generate\npresentation manifest\nand add Content\nProtection Information;~n~4MC-~gDP: Upload\npresentation\nmanifest;~n~4hide MC;~n~4ENP--ENC: Generate encrypted\nsegments and add\nContent Protection\nInformation;~n~4hide ENC;~n~4ENP-~gDP: Upload\nencrypted segments;~n};~nhide ENP;~n~n...;~nshow UE;~nvspace 10;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iClient request and authorisation\i~q {~n~9~3~n~4APP~gCP: User Authentication\n\_M8d\_[strong];~n~4CP~gAPP: Provide authorisation tokens\n\_M8d\_[strong];~n~4hide CP;~n~4APP~gDC~gDRMC: Provide authorisation tokens;~n~4hide APP;~n~4DC~l-~gDP: Aquire\npresentation\nmanifest\n\_M4d\_[strong];~n~4DC-~gLS: Acquire licence\n(possibly including authorisation tokens)\n\_M13d\_[strong];~n~4LS-~gAUS: Verify user;~n~4AUS-~g LS [number=no]: Verified;~n~4hide AUS;~n~4LS-~gKS: Request decryption keys;~n~4KS-~gLS [number=no]: Decryption keys; ~n~4LS-~gDRMC-~gDC: Licence with decryption keys\n\_M13d\_[strong];~n~4hide LS, KS;~n};~n~nvspace 5;~nbox .. [fill.color=lgrey,0.4, line.corner=round, line.color=none, number=no]: ~q\iContent delivery and decryption\i~q {~n~4DC-~gDP: Acquire\nencrypted Segments\n\_M4d\_[strong];~n~4DP-~gDC [number=no]: Encrypted Segments [strong];~n~4hide DP;~n~4DC-~gDRMC: Encrypted\nsamples;~n~4hide DC;~n~4DRMC--DRMC: Decrypt samples\nusing licence and\ncontent keys;~n~4DRMC-~gMP: Decrypted samples;~n};~n~|

Figure 5.10.3-5 Typical workflow for generating, distributing and playing back encrypted content

The following call flow is provided:

*Content preparation* phase*:*

1. A content provider provides encoding instructions and rules for using the content to the content preparation and hosting system via reference points M1d/M3d as part of Content Preparation Template.

2. The content provider provides raw content to the content preparation and hosting system via reference point M2d.

*Trust establishment* phase*:*

3. The Encryptor/Packager and the DRM System (typically the License and Authorization Server as well as the Key Server) exchange public signing keys to establish a trusted communication via reference points M1d/M3d as part of Content Preparation Template.

*Content Protection Information construction* phase*:*

4. The Packager generates initial Content Protection Information that includes identification of the receivers and the various stream encoding criteria (usage rules).

5. The Encryptor retrieves this information from the packager and requests one or several content keys from the key server by sending the signed Content Protection Information to a key server via reference point M2d.

6. The Key Server generates content keys according to the request.

7. The Key Server adds these content keys to the Content Protection Information, signs the Content Protection Information and sends it back to the encoder/packager via reference point M2d.

8. The encryptor/packager extracts the content keys and stores them.

*Content Protection Information distribution* phase*:*

9. The Encryptor/Packager sends the Content Protection Information to the DRM System via reference point M2d.

10. The License Server verifies the Content Protection Information and imports keys or key identifiers into its database.

11. The License Server sends the updated Content Protection Information to the Encryptor/Packager and the Manifest Creator including associated DRM System information via reference point M2d.

*Presentation manifest and media segment generation* phase*:*

12. The Manifest Creator generates the presentation manifest (e.g. DASH MPD) and adds the Content Protection Information (DRM System specific information, key identifiers, etc.).

13. The Manifest Creator uploads the presentation manifest to the Content Hosting.

14. The Encryptor/Packager generates encrypted segments and adds the Content Protection Information (e.g. DRM System specific information).

15. The Encryptor/Packager provides the encrypted segments to the Content Hosting.

*Client requests and authorisation* phase*:*

16. The application requests an authentication for the user with the content provider via M8d.

17: The application provider providers authorisation tokens via M8d.

18: The application provides the tokens to the DASH Player and the DRM client.

19: The Media Access Client requests the presentation manifest including Content Protection Information from the Content Hosting function on the 5GMSd AS via reference point M4d.

20: The Media Access Client requests one or several licenses from the licensing server using key identifiers, possibly including authorization tokens via reference point M13d.

21: The License Servers asks for user verification to the Authorisation Server and the user is verified by the Authorisation Server.

22: The License Server requests the decryption keys from the key server based on the key identifiers and the Key Server provides the relevant content keys in response.

23. The License Server provides a DRM license with one or multiple keys to the Media Access Client and DRM Client via reference point M13d.

*Content delivery and decryption* phase*:*

24. The Media Access Client requests encrypted segments from the Content Hosting function of the 5GMSd AS via reference point M4d and the Media Access Client provides the encrypted segments in response.

25. The Media Access Client extracts encrypted samples from the encrypted segments and provides them to the DRM Client.

26. The DRM Client decrypts the samples using the DRM licence and content keys.

27. The DRM System provides the decrypted samples to the Media Platform for playback.

As mentioned above, ETSI TS 103 799 [143] provides a format that permits a DRM system to provide Content Protection Information to an Encryptor, Packager and manifest creator according to step 10 above. An example CPIX document is shown in listing 5.10.3-1.

Figure 5.10.3-1 Example CPIX document

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <CPIX xmlns="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc">  <ContentKeyList>  <ContentKey kid="e82f184c-3aaa-57b4-ace8-606b5e3febad">  <Data>  <pskc:Secret>  <pskc:PlainValue>wvr2bihSzExKdR8KKpQf2w==</pskc:PlainValue>  </pskc:Secret>  </Data>  </ContentKey>  <ContentKey kid="087bcfc6-f7a5-5716-b840-6aa6eba3369e">  <Data>  <pskc:Secret>  <pskc:PlainValue>8281ce8db9083697d9770d87db962835</pskc:PlainValue>  </pskc:Secret>  </Data>  </ContentKey>  </ContentKeyList>  <DRMSystemList>  <DRMSystem kid="e82f184c-3aaa-57b4-ace8-606b5e3febad" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">  <PSSH>  AAAAxnBzc2gBAAAA7e+LqXnWSs6jyCfc1R0h7QAAAAINw+xPdoNUi4HnPGTlguE2FEe37S9mVyu9EwbOfPNhDQAAAIISEBRHt+0vZlcrvRMGznzzYQ0SEFrGoR6qL17Vv2aMQByBNMoSEG7hNRbI51h7rp9+zT6Zom4SEPnsEqYaJl1Hj4MzTjp40scSEA3D7E92g1SLgec8ZOWC4TYaDXdpZGV2aW5lX3Rlc3QiEXVuaWZpZWQtc3RyZWFtaW5nSOPclZsG  </PSSH>  </DRMSystem>  <DRMSystem kid="087bcfc6-f7a5-5716-b840-6aa6eba3369e" systemId="9a04f079-9840-4286-ab92-e65be0885f95">  <PSSH>  AAAAxnBzc2gBAAAA7e+LqXnWSs6jyCfc1R0h7QAAAAINw+xPdoNUi4HnPGTlguE2FEe37S9mVyu9EwbOfPNhDQAAAIISEBRHt+0vZlcrvRMGznzzYQ0SEFrGoR6qL17Vv2aMQByBNMoSEG7hNRbI51h7rp9+zT6Zom4SEPnsEqYaJl1Hj4MzTjp40scSEA3D7E92g1SLgec8ZOWC4TYaDXdpZGV2aW5lX3Rlc3QiEXVuaWZpZWQtc3RyZWFtaW5nSOPclZsG  </PSSH>  </DRMSystem>  </DRMSystemList>  <ContentKeyUsageRuleList>  <ContentKeyUsageRule kid="e82f184c-3aaa-57b4-ace8-606b5e3febad">  <VideoFilter minPixels="0" maxPixels="442368"/>  <BitrateFilter maxBitrate="500000"/>  </ContentKeyUsageRule>  <ContentKeyUsageRule kid="087bcfc6-f7a5-5716-b840-6aa6eba3369e">  <VideoFilter minPixels="442369" maxPixels="2073600"/>  </ContentKeyUsageRule>  </ContentKeyUsageRuleList>  </CPIX> |

The example provides the following:

- **ContentKeyList**: This element defines the content keys used to encrypt the content. Each **ContentKey** element includes a @kid attribute for the Key ID and the actual encryption key in the **PlainValue** element.

- **DRMSystemList**: This element defines the DRM systems associated with each content key. Each **DRMSystem** element includes a @kid (Key ID) and a @systemId that identifies the DRM system (e.g., Widevine™, PlayReady™). The **PSSH** element includes the Protection System Specific Header (PSSH) containing DRM-specific data required for license acquisition and playback.

- **ContentKeyUsageRuleList**: This element defines usage rules for each content key. Each **ContentKeyUsageRule** element includes a @kid (Key ID) and filters that specify how the content can be used. For example, the **VideoFilter** and **BitrateFilter** elements define restrictions on video resolution and bit rate respectively.

- Usage rules and filters:

**- VideoFilter**: Specifies the range of video resolutions that can be decrypted with the associated key. For example, @minPixels="0" and @maxPixels="442368" restrict the key to standard definition (SD) video.

**- BitrateFilter**: Specifies the maximum bit rate that can be decrypted with the associated key. For example, @maxBitrate="500000" restricts the key to content with a bit rate of up to 500 kbps.

This CPIX document ensures that different keys are used for different content tracks and that specific usage rules are enforced, enhancing the security and flexibility of the DRM-protected content.

Step 8 above may include only a subset of the information, i.e. the DRM system information is absent and is only added in step 9.

## ===== CHANGE =====

### 5.10.4 Mapping to high-level call flows

Based on the generalised Media Delivery architecture from TS 26.501 [15], as reproduced in figure 5.15.1-1 below, different mapping options of the components of the above DRM architecture to the downlinkdownlink media streaming architecture are provided in table 5.10.4-1.



NOTE Since this Key Issue deals only with downlink media streaming, figure 4.2.1-2 from TS 26.501 may be applicable as well, with "5GMSd" instead of "Media" throughout.

Figure 5.10.4-1 Media Delivery architecture as defined in figure 4.1.2.2-1 of TS 26.501 [15]

Table 5.10.4-1 provides different deployment options on how the DRM network functions are mapped to the downlink media streaming functions.

Table 5.10.4-1 Possible deployment options to map DRM network functions  
to downlink media streaming functions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DRM Function | 5GMSd Application Provider | 5GMSd AS | 5GMSd AF | Media Player | Media Session Handler |
| Authorization Server | 1, 2, 3, 4, 5, 6 | 2 |  |  |  |
| Key Server | 1, 3, 4 | 2 | 5, 6 |  |  |
| License Server | 1, 3, 4 | 2 | 5, 6 |  |  |
| Encoder | 1, 4, 6 | 2, 3, 5 |  |  |  |
| Encryptor/packager | 1, 4, 6 | 2, 3, 5 |  |  |  |
| Manifest Creator | 1, 6 | 2, 3, 4, 5 |  |  |  |
| Content Hosting |  | 1, 2, 3, 5, 6 |  |  |  |
| DRM Client |  |  |  | 1, 2, 3, 4, 5, 6 |  |
| DASH Client |  |  |  | 1, 2, 3, 4, 5, 6 |  |
| Media Platform |  |  |  | 1, 2, 3, 4, 5, 6 |  |

The following different deployment options are discussed:

Option 1: The 5GMSd Application Provider runs all DRM and packaging related functions, and the 5GMSd AS only caches the DASH Presentation as a CDN. For DRM acquisition, the UE contacts the 5GMSd Application Provider.

NOTE: This option corresponds to downlink media streaming in Release 18 in which DRM is external to the 5GMSd AS. Other options address the internalization of certain DRM-related functions.

Option 2: The 5GMSd System runs a DRM and packaging service, including a License Server, on the 5GMSd AS. Content is ingested by the 5GMSd AS and all functions of licence hosting, content encoding, content encryption and so on run in the 5GMSd AS.

Option 3: The 5GMSd System runs a content encoding and packaging service, but the DRM system (including License and Key Servers) is deployed externally in the 5GMSd Application Provider domain. The 5GMSd AS needs to communicate with the License Server for content encoding and packaging.

Option 4: The 5GMSd System generates the presentation manifest, but content encoding, content packaging and content encryption are external in the 5GMSd Application Provider. The 5GMSd AS needs to acquire relevant information for presentation manifest generation.

Option 5: This is similar to Option 2 but, being quite specific in operation, the License Server is handed to the 5GMSd AF alongside the Authorization Server. The main communication is between 5GMSd AS and 5GMSd AF at reference point M3d.

Option 6: This is similar to option 5, but only the License Server is offered by the 5GMSd System, hosted in the 5GMSd AF. Similar to option 4, content encoding, content packaging and content encryption are external in the Media Application Provider. The main communication is between the 5GMSd Provider and the 5GMsd AF at reference point M1d for the exchange of Content Protection Information.

Based on current deployments, licence acquisition is a user plane communication and no APIs in the media access client exist to delegate the licence acquisition to a third-party function, such as the Media Access Function (e.g. Media Player). Hence, options 5 and 6 are not further discussed.

Now, for the different options 1-4, table 5.10.4-2 maps the steps in figure 5.10.3-5 to components and/or interfaces defined in the media delivery architecture as shown in figure 5.10.4-1.

Table 5.10.4-2: Mapping of steps in figure 5.10.3-5 to functions and/or reference points defined in the media delivery architecture as shown in Figure 5.10.4-1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Option 1 (External DRM and packaging) | Option 2 (Internal DRM and packaging) | Option 3 (Internal packaging, external DRM) | Option 4 (only manifest internal) |
| 1 | 5GMSd AP | M1d/M3d (Content Template) | M1d/M3d (CT) | M1d/M3d (CT) |
| 2 | 5GMSd AP | M2d (Ingest) | M2d (Ingest) | M2d (Ingest) |
| 3 | 5GMSd AP | 5GMSd AS | M1d/M3d | 5GMSd AP |
| 4 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AP |
| 5 | 5GMSd AP | 5GMSd AS | M2d (CPI) | 5GMSd AP |
| 6 | 5GMSd AP | 5GMSd AS | M2d (CPI) | 5GMSd AP |
| 7 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AP |
| 8 | 5GMSd AP | 5GMSd AS | M2d (CPI) | 5GMSd AP |
| 9 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AP |
| 10 | 5GMSd AP | 5GMSd AS | M2d (CPI) | M2d (CPI) |
| 11 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AS |
| 12 | M2 (Ingest) | 5GMSd AS | 5GMSd AS | 5GMSd AS |
| 13 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AP |
| 14 | M2 (Ingest) | 5GMSd AS | 5GMSd AS | M2d (ingest) |
| 15 | M8d | M8d | M8d | M8d |
| 16 | M8d | M8d | M8d | M8d |
| 17 | Media Player | Media Player | Media Player | Media Player |
| 18 | M4d | M4d | M4d | M4d |
| 19 | New (License Request) | M2d | New (License Request) | M4d |
| 20 | 5GMSd AP | M2d | 5GMSd AS | 5GMSd AP |
| 21 | 5GMSd AP | M2d | 5GMSd AS | 5GMSd AS |
| 22 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AS |
| 23 | 5GMSd AP | 5GMSd AS | 5GMSd AS | 5GMSd AS |
| 24 | new | M4d | New (License Response) | New |
| 25 | M4d | M4d | M4d | M4d |
| 26 | M4d | M4d | M4d | M4d |
| 27 | Media Player | Media Player | Media Player | Media Player |
| 28 | Media Player | Media Player | Media Player | Media Player |
| 29 | Media Player | Media Player | Media Player | Media Player |

Of the options documented in table 5.10.4-2, Option 1 is most prominent because it is the default option for downlink media streaming in Release 18. Of the remaining options:

- In particular Option 3 is expected to be of interest for 5G Media Streaming deployments, because it addresses the scenario for which a specific encoding or transcoding for 5G Media Streaming is carried out by the Application Server.

- Option 4 is a subset of Option 3, and hence not discussed explicitly.

- We also exclude option 2 for now, as DRM servers are generally handled outside MNO networks.

NOTE: The mapping to 5GMS via MBS/MBMS is for further study

## ===== CHANGE =====

## 6.10 Support for distributing DRM-protected, encrypted and high-value content

DRM and Conditional Access are commonly used by third-party streaming services. However, in case streaming is done through MBS or MBMS, a more careful management of the encryption/decryption keys needs to be checked. Scalability of key delivery is an issue. The support for encrypted content in unicast/multicast and broadcast is relevant. Integration of Content Protection interfaces in the provisioning of 5G Media Streaming – for example using CPIX back-end interfaces – is of high relevance for the industry and needs to be addressed accordingly. With the inclusion of DRM support in 5GMS, and the support of 5GMS via MBS and MBMS, these use cases need to be supported not only for unicast but also for multicast/broadcast User Services.

Based on the considerations in clause 5.10, it is recommended to:

- Support the Content Protection Information Exchange Format (CPIX) as specified in ETSI TS 103 799 [143] at reference point M2d by specifying the necessary stage-2 and stage-3 extensions in TS 26.501 [15] and TS 26.512 [16] respectively.

- Support the DASH-IF Interoperability Points specified in [144] at reference point M4d for both DASH and HLS by specifying the necessary stage-2 and stage-3 extensions in TS 26.501 [15] and TS 26.512 [16], respectively.

- Specification of a Content Preparation Template format in TS 26.512 [16] or TS 26.510 [108] that can configure encryption content preparation tasks in the 5GMS AS.

NOTE: Maintaining functional symmetry with uplink media streaming (where applicable) is generally considered important, but DRM is considered to be applicable only for 5GMS downlink.

In particular, stage 2 extensions are needed to address the open issues:

- Definition of a new reference point between the Media Player and the 5GMS Application Provider (see also clause 6.19).

- Functional updates to the definition of the 5GMS AS to support:

- Ingest, delivery, and contribution of encrypted content

- Content preparation tasks for:

- Decrypting content ingested at reference point M2d.

- (Re-)encrypting content prior to distribution at reference point M4d.

- Applying usage filters and restrictions.

- Updates to the definitions of reference points to support:

- Carriage of Content Protection information at reference point M2d.

- Delivery of Content Protection information in presentation manifests at reference point M4d.

Stage 3 aspects are provided above.