**3GPP TSG- Meeting #**

**, , - revision of S4-250022**

|  |
| --- |
| *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 canbe 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:*** | **DRM and Conditional Access**: 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 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, for example using CPIX back-end interfaces is of high relevance for the industry and should accordingly be studied. The impacts of these on media plane (reference points M2 and M4) as well as the media session handling APIs (reference points M3, M5) should also be studied. For details see TR 26.804 clause 5.10 and the conclusions in clause 6.10. |
|  |  |
| ***Summary of change:*** | For *distributing encrypted and high-value content* as introduced in clause 5.10 of TR 26.804:i. Functional updates to the definition of the 5GMS AS to support:1. Ingest, delivery, and contribution of encrypted content2. Content preparation tasks for:3. Decrypting content ingested at reference point M2d.4. (Re-)encrypting content prior to distribution at reference point M4d.ii. Updates to the definitions of reference points to support:1. Carriage of Content Protection information at reference point M2d.2. Delivery of Content Protection information in presentation manifests at reference point M4d. |
|  |  |
| ***Consequences if not approved:*** | Feature not supported |
|  |  |
| ***Clauses affected:*** | 4.0.4, 5.14 (new) |
|  |  |
|  | **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:*** |

|  |  |  |  |
| --- | --- | --- | --- |
| [**S4-250022**](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_131_Geneva/Docs/S4-250022.zip) | [AMD-ARCH-MED] Distributing encrypted and high-value content | Qualcomm Germany | Thomas Stockhammer |

**Revisions**: none**Presenter**: Thomas Stockhammer**Online Discussion**:* r01 version presented.
* Richard: We will probably have to add key rotation.
	+ Thomas: This can be in key distribution.
	+ Richard: It could be a good idea to have a separate diagram.
	+ Thomas: I didn’t want to add this during this meeting but we can add a note.
	+ Frederic: This can be managed with a fix.

**Decision**: r01 agreeable with a note about key rotation. Revised and the revision will be agreed without presentation.[S4-250022](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_131_Geneva/Docs/S4-250022.zip) is **revised to S4-250263**. |

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

### 4.0.4 Content preparation

The content preparation feature is applicable to both downlink media streaming (where is provisioned as part of the content hosting feature introduced in clause 4.0.2) and uplink media streaming (where is provisioned as part of the content publishing feature introduced in clause 4.0.3). The content preparation feature enables a 5GMS Application Provider to specify content manipulation by network-side components of the 5GMS System according to provisioned Content Preparation Templates. Content preparation may include encoding, transcoding, packaging, encryption and protecting content using DRM.

When a 5GMSd Application Provider has provisioned the content preparation feature for downlink media streaming:

1. Network-side components of the 5GMS System may manipulate ingested media content and may cache the manipulated content prior to serving it to the 5GMSd Client in the UE.

When a 5GMSu Application Provider has provisioned the content preparation feature for uplink media streaming:

1. Network-side components of the 5GMS System may manipulate the media content ingested from the 5GMSu Client in the UE and may cache the manipulated content prior to egesting it to the 5GMSu Application Provider.

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

## 5.14 Content preparation of DRM-protected content

### 5.14.1 Scenario

A typical scenario is the following: The 5GMSd Application Server runs a content encoding and/or packaging service. However, in many cases, the content provider requires that the content is distributed with specific content and usage rights, referred to a Digital Rights Management (DRM). The DRM system is 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. The DRM system is comprised of two main components: a DRM server and a DRM client.

In a typical deployment scenario, the DRM system –including both key server and license server – is deployed externally to the 5GMS System in the 5GMSd Application Provider domain, but the encoding, encryption and packaging is assigned to the 5GMSd AS. This may, for example, be the case to adapt the content to the needs of mobile distribution, in combination with Content Preparation.

In order to complete the encryption tasks and presentation manifest generation, the 5GMSd AS needs to communicate with the DRM system to obtain all information for manifest generation, content encoding and packaging. In this case *Content Protection Information* is exchanged across 5GMSd reference points.

NOTE 1: The scenario does not address key rotation, i.e. updating keys during an ongoing media streaming session. This aspect is for further study.

In order to address the scenario above, the following sub-functions of the 5GMSd Application Provider, the 5GMSd Application Server and the 5GMSd Client are defined:

- 5GMSd Application Provider:

- **Content Provider**: a content provider providing content for distribution via 5GMSd with the provided content having assigned specific content and usage rights.

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

- **DRM Server**: a component of the DRM system that manages and enforces the access and usage rights of digital content. It typically includes a License Server and a Key Server. In certain cases, the DRM System can have embedded entitlement logic to decide for which request to grant a license and for which not, supported by an Authorization 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.

NOTE 2: Multiple License Servers may be deployed in case of a multi-DRM system with common encryption.

- **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.

- 5GMSd Application Server:

- **Encoder:** Converts ingested media content into a format suitable for 5GMSd distrbution. 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.

- **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 to comply with streaming protocols. 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 at reference point M4d and played back. Those 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 downlink media streaming resources, i.e. primarily Media Entry Points and media segments, and which includes information on the used DRM system.

- 5GMSd Client:

- **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.

- **DRM Client:** Processes licenses and enforcing 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 3: 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.

### 5.14.2 Procedure

A typical example workflow for the scenario described in clause 5.14.1 is shown in figure 5.14.2-1.

 ![Msc-generator~|version=8.6.1~|lang=signalling~|size=1409x1959~|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=~qMedia\nEntry Point\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 5.14.2-1: Procedure for encoding, packaging and encrypting content

The steps are as follows, highlighting in **bold** relevant information exchanged via 5GMSd reference points:

*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 Media Access Client 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.