**3GPP TSG- Meeting # *r01***

**Online, , -** revision of S4aI250014

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *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:*** | | **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 content  2. 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:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | | This revision 1 is updating the initial version to add substance.  It is submitted for endorsement as basis for future work | | | | | | | | |

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

### 4.0.4 Content preparation

The content preparation feature is applicable to both downlink media streaming (where is is provisioned as part of the content hosting feature introduced in clause 4.0.2) and uplink media streaming (where is 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.

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. Content Preparation may include encoding, packaging, encryption and protecting content using DRM.

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

There are cases for which the 5GMSd System runs a content encoding, encryption and/or packaging service. This may be needed in case the content is provided under restricted licence. In a typical deployment scenario, the License Server is external in the 5GMSd Application Provider domain, but the encoding, encryption and packaging is completed in the 5GMSd AS, for example to adapt the content to the mobile distribution needs. In order to complete the encryption tasks and manifest generation, the 5GMSd AS needs to communicate with the License Server for content encoding and packaging.

### 5.14.2 Procedure

A typical example workflow of encrypted content is shown in figure 5.14.2-1.

Msc-generator~|version=8.6.1~|lang=signalling~|size=1160x1553~|text=numbering=yes;~nhscale=auto;~ndefcolor lgrey=224,224,224;~n~n~nUE [label=~qMedia Player~q, fill.color=lgrey]{~n~4DRMS[label=~qDRM\nClient~q];~n~4MP[label=~qMedia\nPlatform~q];~n~4DC[label=~qDASH\nClient~q];~n};~n~n~nAS [label=~q5GMSd AS~q, fill.color=lgrey]{~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~nAppProvider [label=~q5GMSd Application Provider~q, fill.color=lgrey]{~n~4LS[label=~qLicense\nServer~q];~n~4AUS[label=~qAuthorization\nServer~q];~n};~n~n~2~nvspace 10;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IInitialisation {~nENP~l-~gLS~l-~gAUS: Exchange public keys;~n};~n~nvspace 10;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IContent Protection\nInformation construction {~nENP--ENP: Packager constructs\ncontent protection\ninformation;~nENP--ENP: Encryptor encrypts keys\nand adds to content\nprotection information;~n};~n~nvspace 10;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IContent Protection Information distribution {~nENP-~gLS-~gAUS: send content protection information;~nAUS--AUS: Decrypt keys\nand add data to\ncontent protection\ninformation;~nAUS-~gLS: Send content\nprotection information;~nLS--LS: Decrypt keys\nand add data to\ncontent protection\ninformation;~nLS-~gENP-~gMC: Send updated\ncontent protection information;~n};~n~nvspace 10;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IPresentation manifest and media segment generation {~nMC--MC: Generate\npresentation manifest\nand add content\nprotection information;~nMC-~gDP: Upload\npresentation\nmanifest;~nENP--ENC: Generate encrypted\nmedia segments\nand add content\nprotection information;~nENP-~gDP: Upload encrypted\nmedia segments;~n};~n~n...;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IClient requests and authorisation {~nDC~l-DP: Aquire\npresentation\nmanifest;~nDC~l-AUS: Acquire authorisation tokens;~nDC~l-LS: Acquire licence (possibly using authorisation tokens);~nDC-~gDRMS: Provide licence;~n};~n~nvspace 10;~n-- [number=no, line.color=none, line.corner=round, fill.color=lgrey,0.4]: \IContent delivery and decryption {~nDC~l-DP: Acquire\nencrypted\nmedia segments;~nDC-~gMP: Provide\nencrypted\nmedia segments;~nMP-~gDRMS: Provide\nencrypted samples;~nDRMS--DRMS: Decrypt samples\nusing licence\nand content keys;~nDRMS-~gMP: Provide\ndecrypted samples;~n};~n~|

Figure 5.14.2-1: Procedure for encoding, packaging and encrypting content

The following call flow is provided:

*Initialization:*

1. The Encryptor/Packager, License Server, and Authorization Server exchange public keys.

*Content Protection Information construction:*

2. The Packager constructs content protection information.

3. The Encryptor retrieves keys and adds them to the content protection information.

*Content Protection Information distribution:*

4a. The Encryptor/Packager sends the content protection information to the License Server

4b. The License Server sends the content protection information to the Authorization Server.

5. The Authorization Server decrypts the keys and adds data to the content protection information.

6. The Authorization Server sends the updated content protection information to the License Server.

7. The License Server decrypts the keys and adds data to the content protection information.

8. The License Server sends the updated content protection information to the Encryptor/Packager and the Manifest Creator.

*Presentation manifest and media segment generation:*

9. The Manifest Creator generates the presentation manifest (e.g. DASH MPD) and adds the content protection information.

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

11. The Encryptor/Packager generates encrypted segments and adds the content protection information.

12. The Manifest Creator uploads the encrypted segments to the Content Hosting.

*Client requests and authorisation:*

13. The DASH Client requests the presentation manifest from the Content Hosting.

14. The DASH Client requests authorisation tokens from the Authorization Server.

15. The DASH Client requests a DRM licence from the License Server, possibly using the authorisation tokens.

16. The DASH Client provides the DRM licence to the DRM Client.

*Content delivery and decryption:*

17. The DASH Client requests encrypted segments from the Content Hosting.

18. The DASH Client provides the encrypted segments to the Media Platform.

19. The Media Platform provides the encrypted samples to the DRM Client.

20. The DRM System decrypts the samples using the DRM licence and content keys.

21. The DRM System provides the decrypted samples to the Media Platform.