**3GPP TSG-SA3 Meeting #101-e *S3-202880-r1***

**Online, 9th Nov 2020 - 20th Nov 2020**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.180** | **CR** | **0151** | **rev** | **1** | **Current version:** | **16.4.0** |  |
|  | | | | | | | | |
| *For* [***HELP***](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 |  | Radio Access Network |  | Core Network | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | [33.180] R16 Fix terminology | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Airbus, Sepura Ltd. as a supporter | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | MCXSec | | | | |  | ***Date:*** | | | 2020-10-29 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | | Rel-16 |
|  | *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-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Changing terms to create consistency | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | 5.1.2.1 Term correction: IdM client to MCX client  B.3 Term correction: MCX client to client  B.4.1 Term correction: MCX client to client  B.4.2.2 Term corrections:  - scope: MCX client to client  - redirect\_uri: deleting “MCX", as this cannot work if the scope is for multiple MC services; replacing “user agent” as it is the OpenID connect equivalent of IdM client; deleting “to the MCX client”  - state: MCX client to client; authorization to authentication, deleting “back to the MCX client”  - acr\_values: authorization to authentication  - code\_challenge: “that is sent” intended to apply to the code challenge.  B.4.2.3 Term corrections: MCX client to client; NGMI API client” to “client; authorization to authentication  B.4.2.4 Term corrections: MCX client to client; authorization to authentication  3.2 The abbreviation NGMI is deleted as it does not exist in any document after the proposed change. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Readers might face inaccuraties which may lead to misunderstanding. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.1.2.1, B.3, B.4.1, B.4.2.2, B.4.2.3, B.4.2.4, 3.2 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  |  | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  |  | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  |  | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

**\*\*\*\* START OF CHANGES \*\*\*\***

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5.1.2.1 Identity management functional model

The mission critical Identity Management functional model is shown in figure 5.1.2.1-1 and consists of the identity management server located in the MCX common services core and the identity management client located in the MCX UE. The IdM server and the IdM client in the MCX UE establish the foundation for MCX user authentication and user authorization.

Note that use of the term "IdM client" in this document is generically used to represent any identity management service endpoint within an MC UE that communicates with the IdM Server (authorization endpoint or token endpoint) over the CSC-1 reference point for MC identity management services. It does not imply any specific client implementation of the client-side identity management service.

The CSC-1 reference point, between the IdM client in the UE and the Identity Management server, provides the interface for user authentication. CSC-1 is a direct HTTP interface between the IdM client in the UE and the IdM server and shall support OpenID Connect 1.0 ([19], [20] and [21]).

The OpenID Connect profile for MCX shall be implemented as defined in annex B. MCX user authentication, MCX user service authorization, OpenID Connect 1.0, and the OpenID Connect profile for MCX shall form the basis of the identity management architecture.

In alignment with the OpenID Connect 1.0 [21] and OAuth 2.0 standards [19] and [20], CSC-1 shall consist of two identity management interfaces; the authorization endpoint and the token endpoint. These endpoints are separate and independent from each other, requiring separate and independent IP addressing. The authorization endpoint server and the token endpoint server may be collectively referred to as the IdM server in this document.

The HTTP connection between the Identity Management client and the Identity management server shall be protected using HTTPS.

****

**Figure 5.1.2.1-1: Functional Model for MC Identity Management**

To support MCX user authentication, the IdM server (IdMS) shall be provisioned with the user's MC ID and MC service IDs (the MC service ID may be the same as the MC ID). A mapping between the MC ID and MC service ID(s) shall be created and maintained in the IdMS. When an MCX user wishes to authenticate with the MCX system, the MC ID and credentials are provided via the UE IdM client to the IdMS (note that the primary authentication method used to obtain the MC ID and credentials is out of scope of the present document). The IdMS receives and verifies the MC ID and credentials, and if valid returns an ID token, refresh token, and access token to the UE IdM client specific to the credentials. The MCX client learns the user's MC service ID(s) from the ID token. Table 5.1.2.1-1 shows the MCX tokens and their usage.

**Table 5.1.2.1-1: MC tokens**

|  |  |  |
| --- | --- | --- |
| **Token Type** | **Consumer of the Token** | **Description (See Annex B for details)** |
| ID token | UE client(s) | Contains the MC service ID for at least one authorised service (MCPTT ID, MCVideo ID, MCData ID). Also may contain other info related to the user that is useful to the client. |
| Access token | KMS, MCPTT server, etc. (Resource Server) | Short-lived token (definable in the IdMS) that conveys the user's identity. This token contains the MC service ID for at least one authorised service (MCPTT ID, MCVideo ID, MCData ID). |
| Refresh token | IdM server (Authorization Server) | Allows UE to obtain a new access token without forcing user to log in again. |
| Security token | Partner IdM server (Authorisation server) | Short-lived token (definable in the IdMS) that conveys the user's identity to an Identity management server in a partner MC domain. User access to services within the partner domain are based on the validation of this token. |

In support of MCX user authorization, the access token(s) obtained during user authentication is used to gain MCX services for the user. MCX user service authorisation is defined in clause 5.1.3.

To support the MCX service identity functional model, the MC service ID(s) shall be:

- Provisioned into the IdM database and mapped to MC IDs.

- Provisioned into the KMS and mapped to identity associated keys.

- Provisioned into the MCX user database and mapped to a user profile; and

- Provisioned into the GMS(s) and mapped to Group IDs.

Further details of the user authorization architecture are found in clause 5.1.3.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

B.3 Client registration

Before a client can obtain ID tokens and access tokens (required to access MCX resource servers) it shall first be registered with the IdM server of the service provider as required by OpenID Connect 1.0. The method by which this is done is not specified by this profile. For native clients, the following information shall be registered:

- The client is issued a client identifier. The client identifier represents the client's registration with the authorization server, and enables the IdM server to reference parameters associated with that client's registration when being requested for an access token by the client.

- Registration of the client's redirect URIs.

Other information about the client such as (for example): application name, website, description, logo image, legal terms to be consented to, may optionally be registered.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### B.4.1 General

Once a client has been successfully registered with the IdM server of the MCX service provider, the client may request ID tokens and access tokens (as required to access MCX resource servers such as PTT, Video, Data and KMS). MCX Connect will support a number of different client types, including: native, web-based, and browser-based. Only native MCX clients are defined in this version of the MCX Connect profile. The exact method in which a client requests the access token depends upon the client profile. The client profiles, along with steps required from them to obtain OAuth access tokens, are explained in technical detail below.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### B.4.2.2 Authentication request

As described in OpenID Connect 1.0, the IdM client constructs a request URI by adding the following parameters to the query component of the authorization endpoint's URI using the "application/x-www-form-urlencoded" format, redirecting the user's web browser to the authorization endpoint of the IdM server. The standard parameters shown in table B.4.2.2-1 are required by the MCX Connect profile. Other parameters defined by the OpenID Connect specification are optional.

Table B.4.2.2-1: Authentication Request standard required parameters

|  |  |
| --- | --- |
| Parameter | Values |
| response\_type | REQUIRED. For native clients the value shall be set to "code". |
| client\_id | REQUIRED. The identifier of the client making the API request. It shall match the value that was previously registered with the IdM server of the MCX service provider. |
| scope | REQUIRED. Scope values are expressed as a list of space-delimited, case-sensitive strings which indicate which MCX resource servers the client is requesting access to (e.g. MCPTT, MCVideo, MCData, KMS, etc.). If authorized, the requested scope values will be bound to the access token returned to the client.  The scope value "openid" is defined by the OpenID Connect standard and is mandatory, to indicate that the request is an OpenID Connect request, and that an ID token should be returned to the client.  This profile further defines the following additional authorization scopes:  - *"*3gpp:mc:ptt\_service*"*  - *"*3gpp:mc:video\_service*"*  - *"*3gpp:mc:data\_service*"*  *- "*3gpp:mc:ptt\_key\_management\_service*"*  *- "*3gpp:mc:video\_key\_management\_service*"*  *- "*3gpp:mc:data\_key\_management\_service*"*  *- "*3gpp:mc:ptt\_config\_management\_service*"*  *- "*3gpp:mc:video\_config\_management\_service*"*  *- "*3gpp:mc:data\_config\_management\_service*"*  *- "*3gpp:mc:ptt\_group\_management\_service*"*  *- "*3gpp:mc:video\_group\_management\_service*"*  *- "*3gpp:mc:data\_group\_management\_service*"*  *- "*3gpp:mc:location\_management\_service*"*  Others may be added in the future as new MCX resource servers are introduced by 3GPP (see note). |
| redirect\_uri | REQUIRED. The URI of the client to which the IdM server will redirect the authentication response in order to return the authorization code. The URI shall match the redirect URI registered with the IdM server during the client registration phase. |
| state | REQUIRED. An opaque value used by the client to maintain state between the authentication request and authentication response. The IdM server includes this value in its authentication response. |
| acr\_values | REQUIRED. Space-separated string that specifies the acr values that the IdM server is being requested to use for processing this authentication request, with the values appearing in order of preference. For minimum interoperability requirements, a password-based ACR value is mandatory to support. "3gpp:acr:password" as per the OpenID Connect 1.0 specification [21]. |
| code\_challenge | REQUIRED. The base64url-encoded SHA-256 challenge derived from the code verifier, to be verified against later. |
| code\_challenge\_method | REQUIRED. The hash method used to transform the code verifier to produce the code challenge. This profile current requires the usage of "S256" |
| NOTE: The order in which they are expressed does not matter. | |

An example of an authentication request for MCX Connect might look like:

EXAMPLE:

GET/as/authorization.oauth2?response\_type=code&client\_id=idm\_client&scope=openid

3gpp:mc:ptt\_service&redirect\_uri=<http://3gpp.mcptt/cb&state=abc123&acr_values=3gpp:acr:password&code_challenge=0x123456789abcdef&code_challenge_method=S256>

HTTP/1.1  
Host: IdMS.server.com:9031  
Cache-Control: no-cache  
Content-Type: application/x-www-form-urlencoded

Upon receiving the authentication request from the client, the IdM server performs user authentication. Note that user authentication should be completely opaque to the MC services on the client.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### B.4.2.3 Authentication response

The authorization endpoint running on the IdM server issues an authorization code and delivers it to the client. The authorization code is used by the client to obtain an ID token, access token and refresh token from the IdM server. The authorization code is added to the query component of the redirection URI using the "application/x-www-form-urlencoded" format. The authorization code standard parameters are shown in table B.4.2.3-1.

Table B.4.2.3-1: Authentication Response standard required parameters

|  |  |
| --- | --- |
| Parameter | Values |
| code | REQUIRED. The authorization code generated by the authorization endpoint and returned to the client via the authentication response. |
| state | REQUIRED. The value shall match the exact value used in the authentication request. If the state does not match exactly, then the client is under a Cross-site request forgery attack and shall reject the authorization code by ignoring it and shall not attempt to exchange it for an access token. No error is returned. |

An example of an authentication response for MCX Connect might look like.

EXAMPLE:

HTTP/1.1 302 Found

Location:[http://mcptt\_client/cb](http://www.google.com/url?q=http%3A%2F%2Fmcptt_client%2Fcb&sa=D&sntz=1&usg=AFQjCNF48ZKSADf-rMYVIRxf3IXZWWtmRw" \t "_blank)?

code=SplxlOBeZQQYbYS6WxSbIA

&state=abc123

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 5 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 6 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### B.4.2.4 Access token request

In order to exchange the authorization code for an ID token, access token and refresh token, the client makes a request to the authorization server's token endpoint by sending the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. Note that client authentication is REQUIRED for native applications (using PKCE IETF RFC 7636 [53]) in order to exchange the authorization code for an access token. If client secrets are used, the client secret is sent in the HTTP Authorization Header as defined in IETF RFC 6749 [19]. The access token request standard parameters are shown in table B.4.2.4-1.

Table B.4.2.4-1: Access token request standard required parameters

|  |  |
| --- | --- |
| Parameter | Values |
| grant\_type | REQUIRED. The value shall be set to "authorization\_code". |
| code | REQUIRED. The authorization code previously received from the IdM server as a result of the authentication request and subsequent successful authentication of the MCX user. |
| client\_id | REQUIRED. The identifier of the client making the API request. It shall match the value that was previously registered with the OAuth Provider during the client registration phase of deployment, or as provisioned via a development portal. |
| redirect\_uri | REQUIRED. The value shall be identical to the "redirect\_uri" parameter included in the authentication request. |
| code\_verifier | REQUIRED. A cryptographically random string that is used to correlate the authentication request to the token request. |

An example of an access token request for MCX Connect might look like this.

EXAMPLE:

POST /as/token.oauth2 HTTP/1.1  
Host: IdM.server.com:9031  
Cache-Control: no-cache  
Authorization: Basic cnA33hpsb25nABClY3VyZS1yYW5kb20tc2VjdnV0

Content-Type: application/x-www-form-urlencoded  
grant\_type=authorization\_code

&code=SplxlOBeZQQYbYS6WxSbIA

&client\_id=myNativeApp

&code\_verifier=0x123456789abcdef

&redirect\_uri=http://3gpp.mcptt/cb

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 6 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Start of change 7 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

3.2 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].

CMS Configuration Management Server

CS Crypto Session

CSB-ID Crypto Session Bundle Identifier

CSC Common Services Core

CSK Client-Server Key

CSK-ID Client-Server Key Identifier

DPCK MCData Payload Cipher Key

DPPK MCData Payload Protection Key

DPPK-ID MCData Payload Protection Key Identifier

GBA Generic Bootstrapping Architecture

GMK Group Master Key

GMK-ID Group Master Key Identifier

GMS Group Management Server

GUK-ID Group User Key Identifier

IdM Identity Management

IdMS Identity Management Server

InK Integrity Key

InK-ID Integrity Key Identifier

InterKMRec Interworking Key Management Record

InterKMRec-ID Interworking Key Management Record Identifier

InterSD Interworking Security Data

IWF InterWorking Function

JSON JavaScript Object Notation

JWS JSON Web Signature

JWT JSON Web Token

KDF Key Derivation Function

KFC Key For Control Signalling

KFC-ID Key for Floor Control Identifier

KMS Key Management Server

MBCP Media Burst Control Protocol

MCData Mission Critical Data

MCPTT Mission Critical Push to Talk

MCVideo Mission Critical Video

MCX Mission Critical Services

MKFC Multicast Key for Floor Control

MSCCK MBMS subchannel control key

MSRP Message Session Relay Protocol

MuSiK Multicast Signalling Key

MKI Master Key Identifier

NTP Network Time Protocol

NTP-UTC Network Time Protocol – Coordinated Universal Time

OIDC OpenID Connect

PCK Private Call Key

PCK-ID Private Call Key Identifier

PKCE Proof Key for Code Exchange

PSK Pre-Shared Key

SEG Security Gateway

SeGy Security Gateway

SPK Signalling Protection Key

SRTCP Secure Real-Time Transport Control Protocol

SRTP Secure Real-Time Transport Protocol

SSRC Synchronization Source

TBCP Talk Burst Control Protocol

TGK Traffic Generating Key

TrK KMS Transport Key

TrK-ID KMS Transport Key Identifier

UID User Identifier for MIKEY-SAKKE (referred to as the 'Identifier' in RFC 6509 [11])

XPK XML Protection Key

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* End of change 7 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* END OF CHANGES \*\*\*