ETSI TS 133 180 V16.4.0 (2020-08)

LTE;
Security of the mission critical service
(3GPP TS 33.180 version 16.4.0 Release 16)

Reference
RTS/TSGS-0333180vg40

Keywords
LTE,SECURITY

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.
All rights reserved.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
oneM2M™ logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.
GSM® and the GSM logo are trademarks registered and owned by the GSM Association.

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document specifies the security architecture, procedures and information flows needed to protect the mission critical service (MCX). The architecture includes mechanisms to protect the Common Functional Architecture and security mechanisms for mission critical applications. This includes Push-To-Talk (MCPTT), Video (MCVideo) and Data (MCData). Additionally, security mechanisms relating to on-network use, off-network use, roaming, migration, interconnection, interworking and multiple security domains are described.

This specification complements the Common Functional Architecture defined in TS 23.280 [36], the functional architecture for MCPTT defined in 3GPP TS 23.379 [2], the functional architecture for MCVideo defined in 3GPP TS 23.281 [37] and the functional architecture for MCData defined in 3GPP TS 23.282 [38].

The MC service can be used for public safety applications and also for general commercial applications e.g. utility companies and railways. As the security model is based on the public safety environment, some MC security features may not be applicable for commercial purposes.

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*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.379: "Functional architecture and information flows to support Mission Critical Push To Talk (MCPTT); Stage 2".

[3] 3GPP TS 22.179: "Mission Critical Push To Talk (MCPTT); Stage 1".

[4] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[5] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[6] 3GPP TS 33.203: "3G security; Access security for IP-based services".

[7] 3GPP TS 33.179 Release 13: "Security of Mission Critical Push To Talk (MCPTT) over LTE".

[8] 3GPP TS 33.328: "IP Multimedia Subsystem (IMS) media plane security".

[9] IETF RFC 6507: "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)".

[10] IETF RFC 6508: "Sakai-Kasahara Key Encryption (SAKKE)".

[11] IETF RFC 6509: "MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)".

[12] IETF RFC 3550: "RTP: A Transport Protocol for Real-Time Applications".

[13] IETF RFC 3711: "The Secure Real-time Transport Protocol (SRTP)".

[14] 3GPP TS 33.401: "3GPP System Architecture Evolution (SAE); Security architecture".

[15] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[16] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".

[17] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[18] NIST FIPS 180-4: "Secure Hash Standard (SHS)".

[19] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[20] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

[21] OpenID Connect 1.0: "OpenID Connect Core 1.0 incorporating errata set 1", http://openid.net/specs/openid-connect-core-1_0.html.

[22] IETF RFC 3830: "MIKEY: Multimedia Internet KEYing".

[23] IETF RFC 3602: "The AES-CBC Cipher Algorithm and Its Use with IPsec".

[24] IETF RFC 4771: "Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)".

[25] IETF RFC 6043: "MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY)".

[26] IETF RFC 7714: "AES-GCM Authenticated Encryption in the Secure Real-time Transport Protocol (SRTP)".

[27] W3C: "XML Encryption Syntax and Processing Version 1.1", https://www.w3.org/TR/xmlenc-core1/.

[28] W3C: "XML Signature Syntax and Processing (Second Edition)", http://www.w3.org/TR/xmldsig-core/.

[29] IETF RFC 5905: "Network Time Protocol Version 4: Protocol and Algorithms Specification".

[30] IETF RFC 5480: "Elliptic Curve Cryptography Subject Public Key Information".

[31] IETF RFC 6090: "Fundamental Elliptic Curve Cryptography Algorithms".

[32] IETF RFC 7519: "JSON Web Token (JWT)".

[33] IETF RFC 7662: "OAuth 2.0 Token Introspection".

[34] IETF RFC 3394: "Advanced Encryption Standard (AES) Key Wrap Algorithm".

[35] IETF RFC 7515: "JSON Web Signature (JWS)".

[36] 3GPP TS 23.280: "Common functional architecture to support mission critical services; Stage 2".

[37] 3GPP TS 23.281: "Functional architecture and information flows for mission critical video; Stage 2".

[38] 3GPP TS 23.282: "Functional model and information flows for Mission Critical Data".

[39] 3GPP TS 23.002: "Network Architecture".

[40] IETF RFC 2045: "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies".

[41] IETF RFC 2392: "Content-ID and Message-ID Uniform Resource Locators".

[42] NIST Special Publication 800-38D: "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC".

[43] IETF RFC 5116: "An Interface and Algorithms for Authenticated Encryption".

[45] IETF RFC 7521: "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants".

[46] IETF RFC 7523: "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants".

[47] 3GPP TS 22.280: " Mission Critical Services Common Requirements; Stage 1".

[48] 3GPP TS 23.283: " Mission Critical Communication Interworking with Land Mobile Radio Systems; Stage 2".

[49] 3GPP TS 24.379: "Mission Critical Push To Talk (MCPTT) call control; Protocol specification."

[50] 3GPP TS 24.282: "Mission Critical Data (MCData) signalling control; Protocol specification. "

[51] IETF RFC 3711 Errata ID 3712, https://www.rfc-editor.org/errata/eid3712.

[52] IANA: "Multimedia Internet KEYing (MIKEY) Payload Name Spaces", https://www.iana.org/assignments/mikey-payloads/mikey-payloads.xhtml.

[53] IETF RFC 7636: "Proof Key for Code Exchange by OAuth public clients".

3 Definitions and abbreviations

3.1 Definitions

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

Authorised Identity: An application identity given to an authorised user or network entity (e.g. MC Service ID) containing authorisation information.

External KMS: The KMS which is the root of trust for a specific External Security Domain.

External Security Domain: A security domain that the user is not a member of, but with which the user may communicate.

Floor: Floor(x) is the largest integer smaller than or equal to x.

Home KMS: The KMS that is the root of trust of the Home Security Domain.

Home Security Domain: The MCX user's primary security domain.

Identity Management Domain: The MC clients and MC functions that share an Identity Management Server (IdMS). To be specific, the MC clients request access tokens from the same primary IdMS, and the MC functions accept access tokens from this IdMS.

KMS Certificate: A certificate containing the security parameters for a security domain. This is required to support identity-based cryptography and differs from X.509 certificates used for traditional PKI. See Annex D.3.1 for details.

KMS URI: A unique identifier for a security domain, or equivalently, a logical KMS.

MCX: Mission critical services where “MCX” may be substituted with the term “MCPTT”, “MCVideo”, “MCData”, or any combination thereof.

Migration KMS: The KMS that is the root of trust of a specific Migration Security Domain.

Migration Security Domain: A security domain that a user is a (temporary) member of, and may be keyed to use, but is not the user's Home security domain.

Partner domain: A secondary MC domain which may support MC services for MC users who are home to a different MC domain. See also External Security Domain.

Primary domain: The "home" MC domain where MC users receive their primary identity management and MC services. See also Home Security Domain.

Privileged signalling: Signalling which is performed by an authorised user and allows the authorised userto cause an intrusive action on a target client without the target user's permission.

Security Domain: A security domain is a group of MCX users who share common security requirements and policies for their communications. From a technical perspective, users within a security domain share a KMS and KMS certificate. MCX users may be members of one or more security domains.

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
NGMI	Next Generation Mobile Intelligence
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

4 Overview of Mission Critical Security

4.1 General

The mission critical security architecture defined in this document is designed to meet the security requirements defined in Annex A. The security architecture provides signalling and application plane security mechanisms to protect metadata and communications used as part of the MC service. The following signalling plane security mechanisms are used by the MC service:

- Protection of the signalling plane used by the MC Service, defined in clause 6.1 and 6.2.
- Protection of inter/intra domain interfaces, defined in clause 6.3.

The following application plane security mechanisms are used by the MC service:

- Authentication and authorisation of users to the MC Service, defined in clause 5.1.
- Protection of sensitive application signalling within the MC Service, defined in clause 9.
- Security of RTCP (e.g. floor control, transmission control) within the MC Service, defined in clause 9.
- Security of data signalling within the MCData Service, defined in clause 8.
- End-to-end security of user media within the MC Service. Defined in clause 7 for MCPTT and MCVideo services and defined in clause 8 for the MCData service.

Security mechanisms in the signalling and application plane are independent of each other, but may both be required for a secure MC system.

4.2 Signalling plane security architecture

Within a MC system, signalling plane security protects the interfaces used by the MC application. Figure 4.2-1 provides an overview of these interfaces.

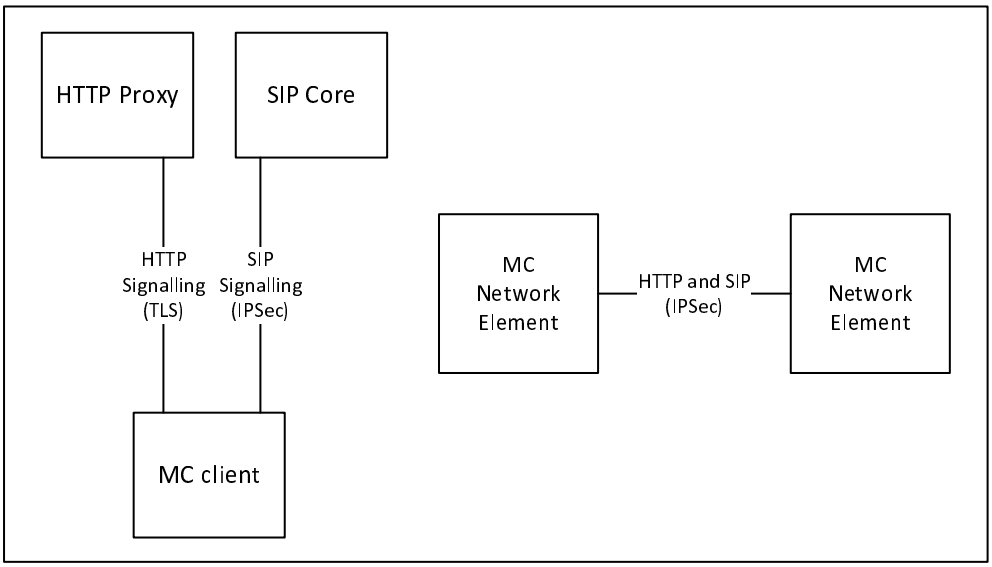

Figure 4.2-1: Signalling plane security architecture

Signalling from the MC client is passed over both HTTP and SIP. The signalling plane security mechanisms for client to server interfaces and between network elements are defined in clause 6.

4.3 MC system security architecture

4.3.1 General

The MC system security architecture provides protection both between MC clients, between the MC client and the MC domain, and also between MC domains. MC system security on the client is bound to the MC user associated with the client and not to the MC UE. Consequently, user authentication and authorisation to the MC domain is required prior to access to the majority of MC services.

Application plane signalling security allows protection of MC-specific signalling from all entities outside of the MC system (potentially including the SIP core). Application plane signalling security is applied from the MC client to the client's primary MC domain. It may also be applied between MC domains.

Media security allows protection of MC media within the MC system. It is applied end-to-end between MC clients or in some cases from the MC client to the MCX server (e.g. One-to-server video push or one-from-server video pull). Under normal operation however, MC network entities such as the MCX Servers are typically unable to decrypt the media.

Additionally, signalling plane protection is applied to all HTTP and SIP connections into the MC domain. While signalling plane protection and signalling plane entities are not shown in this subclause, including the SIP core and HTTP proxy, it is assumed that signalling plane protection mechanisms are in use.

4.3.2 User authentication and authorisation

Prior to connecting to the MC domain, the MCX user application requires a 'token' authorising its access to MC services. To obtain authorisation token(s), the MCX user application authenticates the MC user to an Identity Management Server which provides the authorisation token.

The authorisation token is provided to MCX network entities, such as the MCX Server, over an MCX signalling interface (either a HTTP interface or SIP interface). The MCX network entity will provide access to MCX services based upon the token provided.

The architecture for user authentication and authorisation is shown in Figure 4.3.2-1.

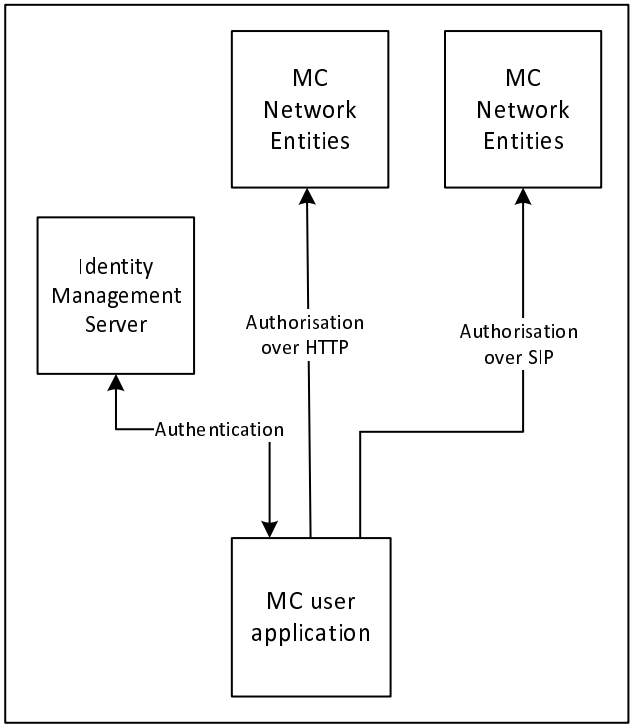

Figure 4.3.2-1: User authentication and authorisation

While the HTTP proxy and SIP core is not shown in Figure 4.3.2-1, authorisation occurs over HTTP or SIP and hence uses signalling plane protection to encrypt authorisation requests carried over HTTP to a HTTP proxy and authorisation requests carried in SIP messages through the SIP core to the MCX domain.

The mechanism to perform user authentication and authorisation is defined in clause 5.1.

4.3.3 Identity keying of users and services

Once a MC client has obtained user authorisation to access the MCX domain, the client may obtain key material associated with the user's identity using the authorisation token. Identity keys are required to support key distribution for application signalling, floor control, transmission control and media. Identity key material is obtained via an HTTP request to a Key Management Server as shown in Figure 4.3.3-1.

Identitiy keying is repeated periodically (e.g. monthly). This ensures that user identities are regularly verified and that users that are no longer part of the MCX domain are removed from the system.

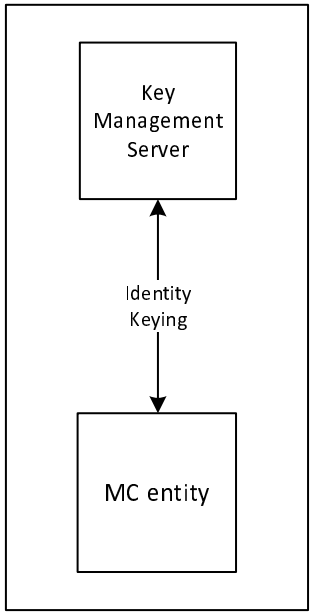

Figure 4.3.3-1: Identity keying of MC entities

While not shown in Figure 4.3.3-1, the UE connection to the KMS is over HTTP and hence is secured using TLS directly between the MC client and KMS or between the MC client and the HTTP proxy or directly to the KMS. When the HTTP proxy is in the path between the MC client and the KMS, key material is wrapped using a transport key (TrK) distributed out-of-band (reference clause 5.3.2). The TrK or a shared Integrity key (InK) may be used to sign the key material.

A number of MC network entities also require identity key material including the MCX Server and Group Management Server. This key material is obtained via the same HTTP interface.

The mechanism to perform identity keying is defined in clause 5.3.

4.3.4 Protection of application plane signalling

4.3.4.1 Application plane signalling security

Application plane signalling security protects application signalling between the MC client and the MCX server. Initial key distribution for application signalling is performed by sending a client-server key (CSK) from the MC client to the MCX Server over the SIP interface. The key is secured using the identity key material provisioned by the Key Management Server. Following initial key distribution, the MCX server may perform a ‘key download' procedure to update key material, and to key the client to allow multicast signalling to be protected.

There are a variety of types of application plane signalling, including:

- XML signalling within SIP payloads
- Control signalling (e.g. RTCP for floor control or transmission control).
- MCData signalling payloads within SIP payloads.

In each case, the same root key material is used to protect the signalling when the signalling is unicast on the uplink or downlink. Should the signalling be multicast on the downlink, the MCX Server will distribute key material for this purpose and use this key material to protect multicast signalling.

The security architecture is shown in Figure 4.3.4.1-1.

The mechanisms to provide application plane signalling security are defined in clause 9.

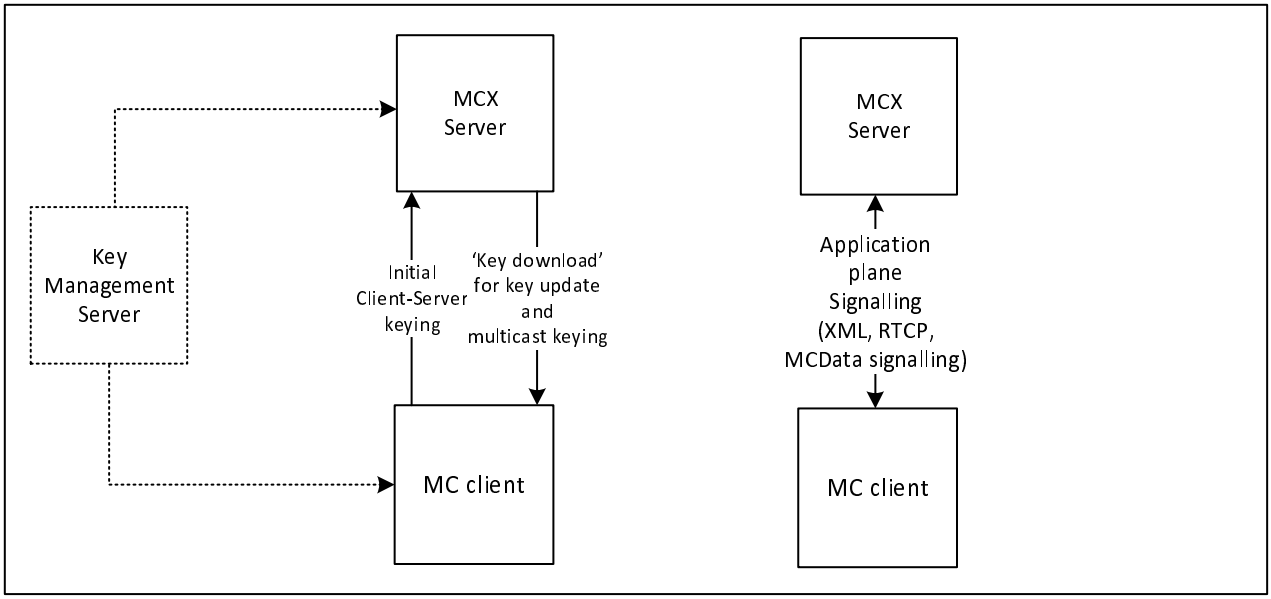

Figure 4.3.4.1-1: Application plane signalling security

Application plane signalling security can also be applied between MCX servers. In this case the MCX servers are keyed manually. While not shown in Figure 4.3.4-1, application plane signalling uses SIP and HTTP and hence is also secured up to the SIP core and HTTP proxy respectively.

4.3.4.2 Security enforcement at the network edge

Clause 4.3.4.1 describes the application plane signalling security functions between the MC client and MCX Servers and between MCX Servers. These security functions can be enforced by the MCX Servers themselves as described in Clause 4.3.4.1.

However, in some scenarios, there may be value in applying application plane signalling security at the edge of the MC Domain. This deployment option involves moving security functions out of the MCX Servers and into Signalling Proxies at the edge of the MC Domain as shown in Figure 4.3.4.2-1.

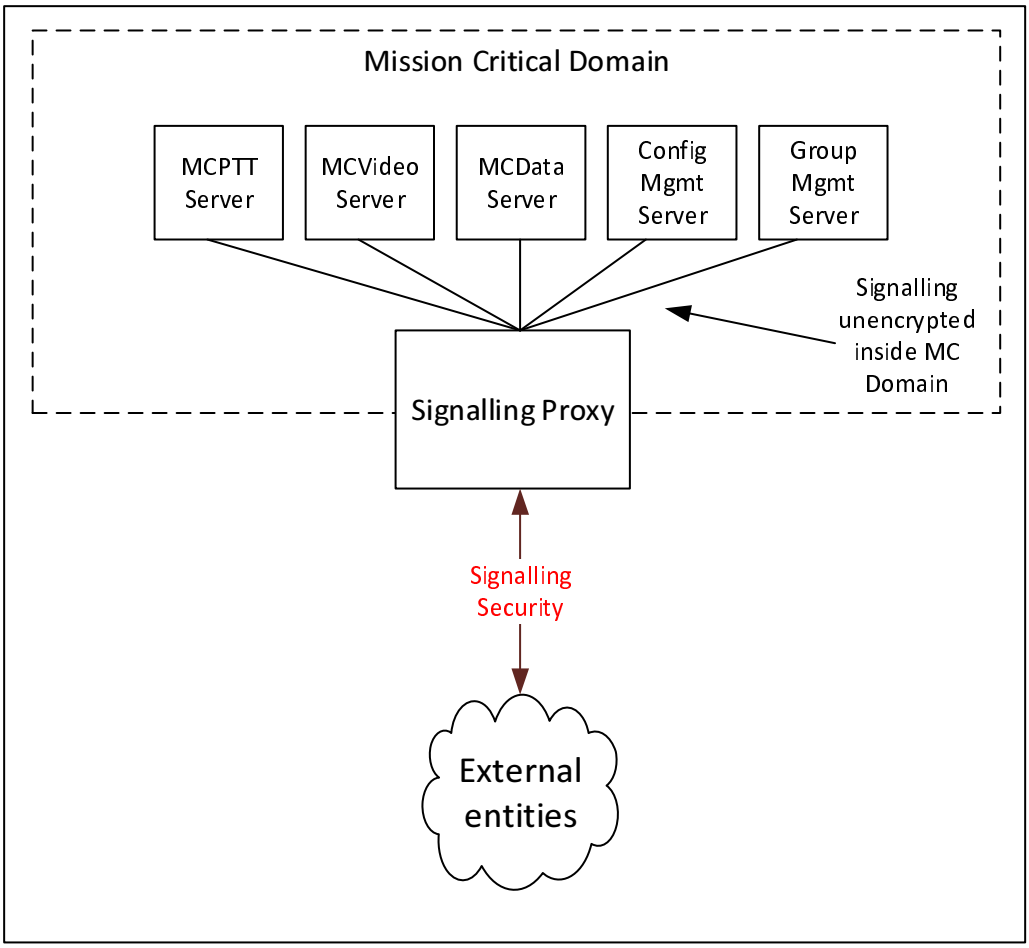

Figure 4.3.4.2-1: Signalling Proxies

There are two types of Signalling Proxy:

- Client Signalling proxy (CS Proxy), which controls security towards the MC clients.
- Interconnection Signalling Proxy (IS Proxy), which controls security towards other MC Domains.

Full details of both types of Signalling Proxy are provided in Annex I. The use of signalling proxies has the following advantages:

- The mission critical core network architecture is not exposed to Mission Critical clients or other external entitites. The client no longer needs to know the SIP URI of each distinct MCX Server.
- Intrusion detection within the XML signalling link is possible at the network edge.
- Policies can be assigned to signalling on entry to the Mission Critical network.
- The number of signalling protection keys required by the client and the MC Domain are reduced.
- Multicast bearers can be shared across multiple MCX Servers.

Effectively, for XML-protected application signalling, the Signalling Proxy is able to perform equivalent functions to a Session Border Controller (as defined in RFC 5853 [24]), or IMS IBCF (as defined in Annex I of 3GPP TS 23.228 [23]).

4.3.5 Media security

4.3.5.1 General

Media security establishes an end-to-end security context between MC users to support group communications and private communications for the MCPTT, MCVideo or MCData services. The intention is for media to be able to be encrypted end-to-end between MC clients, irrespective of whether the media is routed unicast via the media distribution server, multicast via the media distribution server, or transmitted over a direct or IOPS connection.

Key distribution for groups is performed by the Group Management Server. Key distribution for private calls is performed by the initiating MC client. Once a security context is established, the media is protected using the distributed key material. Aditionally, when MC UEs are off-network, the security context that is used to protect media security is also used to protect control signalling (e.g. RTCP).

4.3.5.2 Media security for group communications.

Media security for groups is secured by establishing a shared group security context between group members. Key distribution for the group security context is performed by a Group Management Server. The Group Management Server creates and sends group keys and group security parameters over SIP as part of group management.

Group keys and security parameters are encrypted by the Group Management Server to the identity of the individual MC users that are members of the group.. MC users and MCX servers require identity keying by a KMS prior to performing group management.

Figure 4.3.5.2-1 provides an overview of the group keying process. Details of the process may be found in clause 5.7.

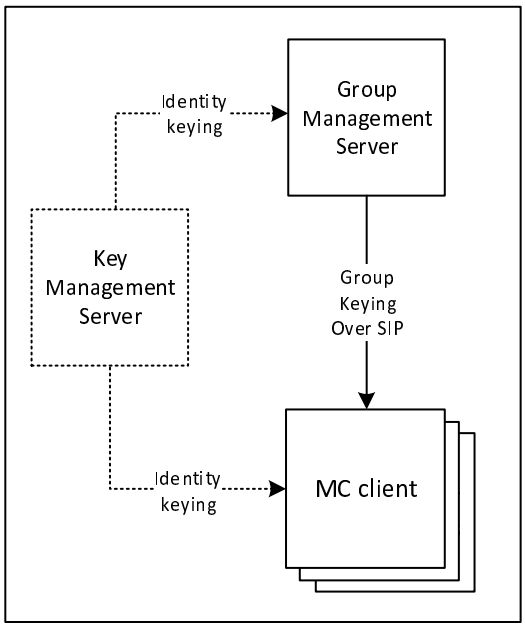

Figure 4.3.5.2-1: Group keying for media security

Once a group key has been shared with MC users, keys are derived from that group key to protect media (and control signalling when the UE is off-network).

For MCPTT and MCVideo (specifically RTP), key derivation is based on the MCPTT or MCVideo user's identity, hence every member of the group encrypts media using a different key. Media is encrypted using the SRTP protocol in this case. For MCData, the user-specific key derivation is not required. Media is encrypted within a MCData data payload in this case.

When the MC UE has a network connection the encrypted media is routed to other MC clients via the media distribution function in the MCX Server. Media from an MC client is distributed to group members by the MCX Server over either unicast or multicast. When the MC UE is off-network, the encrypted media is routed directly to MC clients

on other MC UEs. The security procedure for protecting media is the same in either case. Details of media encryption are provided in clause 7 for MCPTT and MCVideo, and clause 8 for MCData.

Unlike media, control signalling (such as floor control or transmission control) is protected differently when the UE has a network connection and when it is off-network. When the UE has a network connection, control signalling traffic is encrypted to the identity of the MC Domain. When it is off-network, control signalling is encrypted directly to UEs using a key derived from the root key for the group or private communication. Details of control signalling encryption is provided in clause 9.4.

Figure 4.3.5.2-2 provides an overview of how media is protected for group communications.

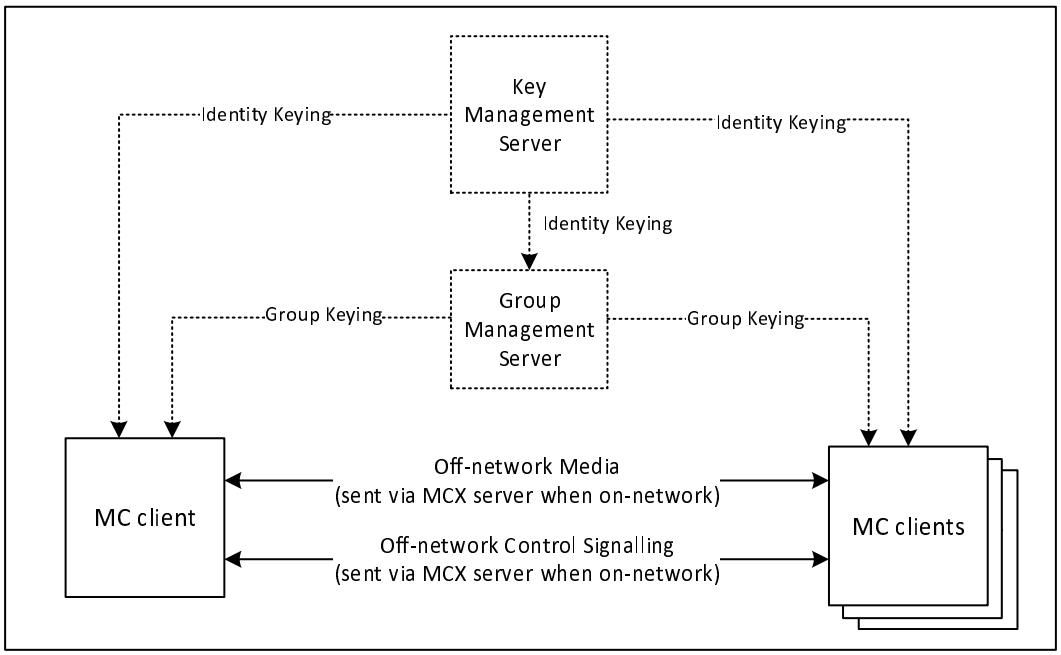

Figure 4.3.5.2-2: Group media protection

4.3.5.3 Media security for private calls

As part of setting up a private call, the call initiator provides the session key to the terminating client. The key is encrypted to the MC user that is currently registered on the terminating client. As a result, MC users require identity keying by a KMS prior to performing private communications.

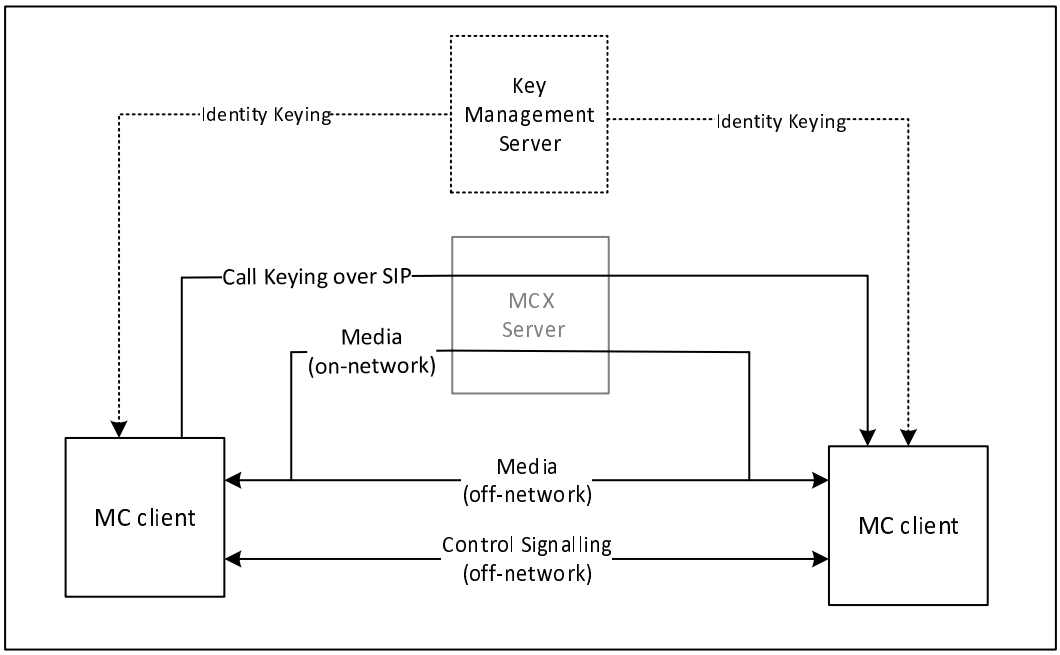

Figure 4.3.5.3-1: Media security for private calls

Figure 4.3.5.3-1 provides an overview of media protection for private calls. For clarity, MC network entities do not have access to the private call key material and hence are not able to decrypt the media for the private call communication (unless the monitoring function is specifically authorised for either user).

Details of private call key distribution are provided in clause 5.6, specific MCPTT and MCVideo procedures are described in clause 7 and specific MCData procedures are in clause 8.

Once private call key distribution has been completed, control signalling and application signalling are used to setup and control the media transport of a private communication. Media will be routed via the media distribution function in the MCX Server when the UE is online, and directly when the UE is off-network. Details of media protection are found in clauses 7 and 8, control signalling protection is found in clause 9.4 and application signalling protection is found in clause 9.3.

The media security context shall also be used to protect control signalling (e.g. floor control) when the MC UE is off-network.

5 Common mission critical security framework

5.1 User authentication and authorization

5.1.1 General

The generic steps for MCX user authentication and authorisation is shown in figure 5.1.1-1.

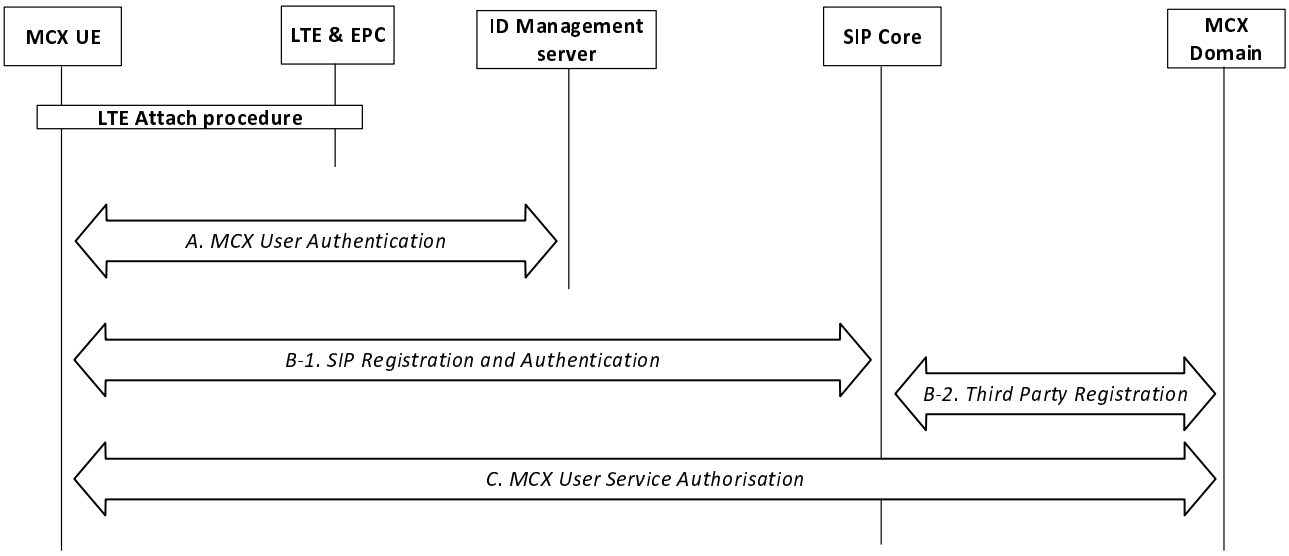

Figure 5.1.1-1: MCX authentication and authorisation

At UE power-on, the MCX UE performs LTE authentication as specified in TS 33.401 [14]. The MCX UE then performs the following steps to complete authentication of the user, authorisation of the user, MCX service registration, and identity binding between signalling layer identities and the MC service ID(s).

- A: MCX user authentication.
- B: SIP Registration and Authentication.
- C: MCX Service Authorization.

These procedures are described in more detail in subsequent clauses.

Steps A and B may be performed in either order or in parallel. For scenarios where this order has an impact on the identity bindings between signalling layer identities and the MC service ID(s), a re-registration (Step B) to the SIP Core may be performed to update the registered signalling layer identity.

If an MCX UE completes SIP registration in Step B prior to performing MCX user authentication in Step A and MCX user service authorization as part of Step C, the MCX UE shall be able to enter a 'limited service' state. In this limited state, where the MCX user is not yet authorized with the MCX service, the MCX UE shall be able to use limited MCX services (e.g. an anonymous MCX emergency communication). The MCX Server is informed of the registration of the MC UE with the SIP core though Step B-2.

Additionally, an HTTP-1 authentication mechanism is used.

NOTE: Mechanisms for confidentiality and integrity protection (not defined in this clause) may be combined only with certain authentication procedures.

5.1.2 User authentication

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.

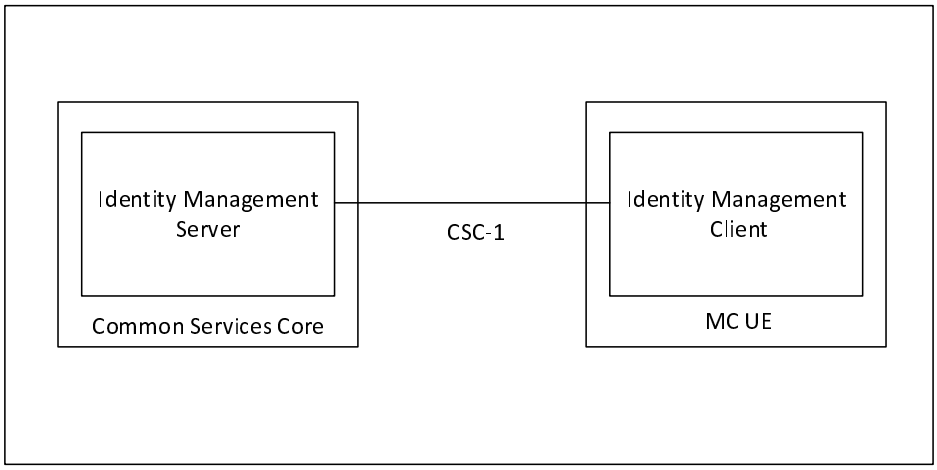

Figure 5.1.2.1-1: Functional Model for MC Identity Management

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 IdM 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.

5.1.2.2 User authentication framework

The framework utilises the CSC-1 reference point as depicted in Figure 5.1.2.2-1.

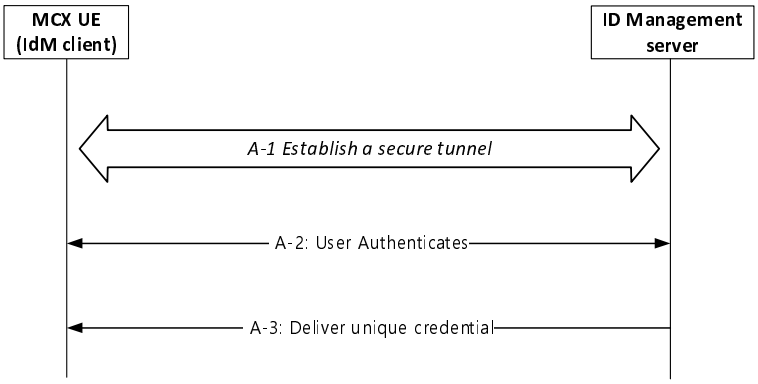

Figure 5.1.2.2-1: MCX User Authentication Framework

The User Authentication procedure in Step A of Figure 5.1.1-1 is further detailed into 3 sub steps that comprise the MCX user authentication framework:

- A-1 - Establish a secure tunnel between the MCX UE and Identity Management (IdM) server. Subsequent steps make use of this tunnel.
- A-2 - Perform the User Authentication Process (User proves their identity).
- A-3 - Deliver the credential(s) that uniquely identifies the MCX user to the IdM client.

Following step A-3, the MCX client uses the credential(s) obtained from step A-3 to perform MCX user service authorization as per procedure C in figure 5.1.1-1.

The framework supporting steps A-2 and A-3 shall be implemented using OpenID Connect 1.0 ([19], [20] and [21]).

NOTE: MCX service authorization in step C of Figure 5.1.1-1 is outside the scope of the User Authentication framework.

5.1.2.3 OpenID Connect (OIDC)

5.1.2.3.1 General

Figure 5..1.2.3.1-1 describes the MCX User Authentication Framework using the OpenID Connect protocol. Specifically, it describes the steps by which an MCX user authenticates to the Identity Management server (IdMS), resulting in a set of credentials delivered to the UE uniquely identifying the MC service ID(s). The means by which these credentials are sent from the UE to the MCX services are described in clause 5.1.3. The authentication framework supports extensible user authentication solutions based on the MCX service provider policy (shown in step 3), with username/password-based user authentication as a mandatory supported method. Other user authentication methods in step 3 (e.g. biometrics, secureID, etc.) are possible but not defined here. A detailed OpenID Connect flow can be found in annex C.

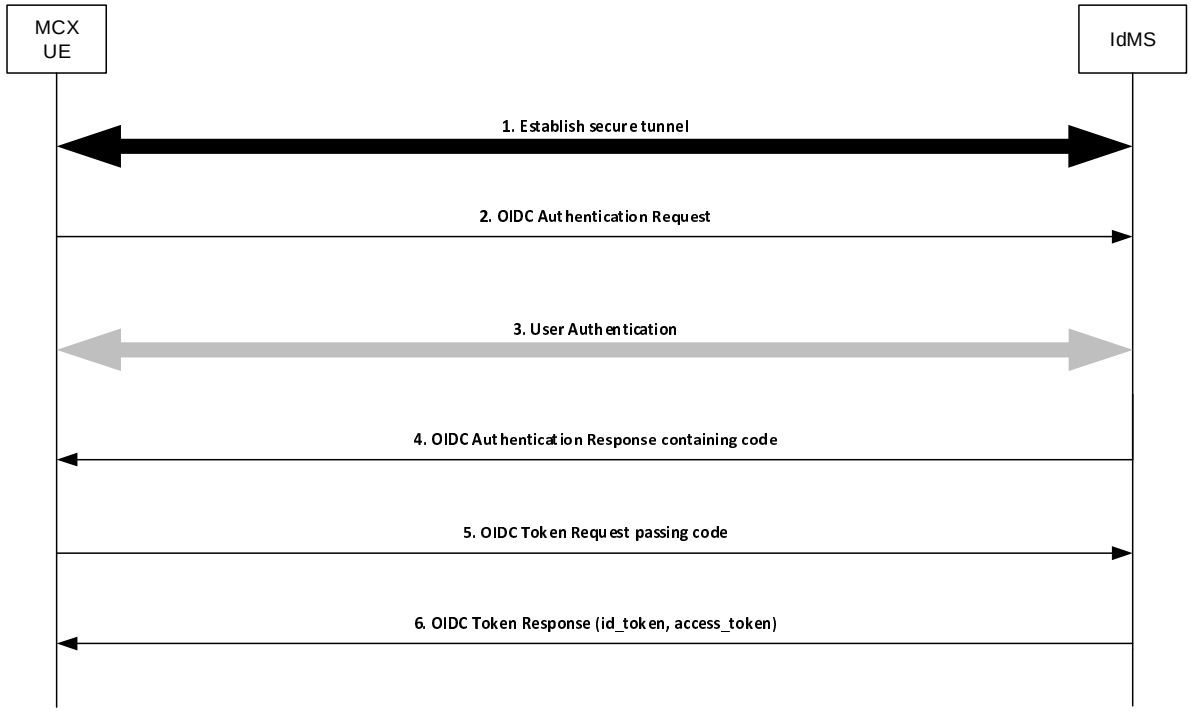

Figure 5.1.2.3.1-1: OpenID Connect (OIDC) flow supporting MCX user authentication

Step 1: UE establishes a secure tunnel with the Identity Management server (IdMS).

Step 2: UE sends an OpenID Connect Authentication Request to the IdMS. The request may contain an indication of authentication methods supported by the UE.

Step 3: User Authentication is performed.

NOTE: The primary credentials for user authentication (e.g. biometrics, secureID, OTP, username/password) are based on MCX service provider policy. The method chosen by the MCX service provider is neither defined nor limited by the present document.

Step 4: IdMS sends an OpenID Connect Authentication Response to the UE containing an authorization code.

Step 5: UE sends an OpenID Connect Token Request to the IdMS, passing the authorization code.

Step 6: IdMS sends an OpenID Connect Token Response to the UE containing an ID token and an access token (each which uniquely identify the user of the MCX service). The ID token is consumed by the UE to personalize the MCX client for the MCX user, and the access token is used by the UE to communicate the identity of the MCX user to the MCX server(s).

5.1.2.3.2 User authentication example using username/password

Figure 5.1.2.3.2-1 shows the OIDC flow when Username/Password is used as the user authentication method.

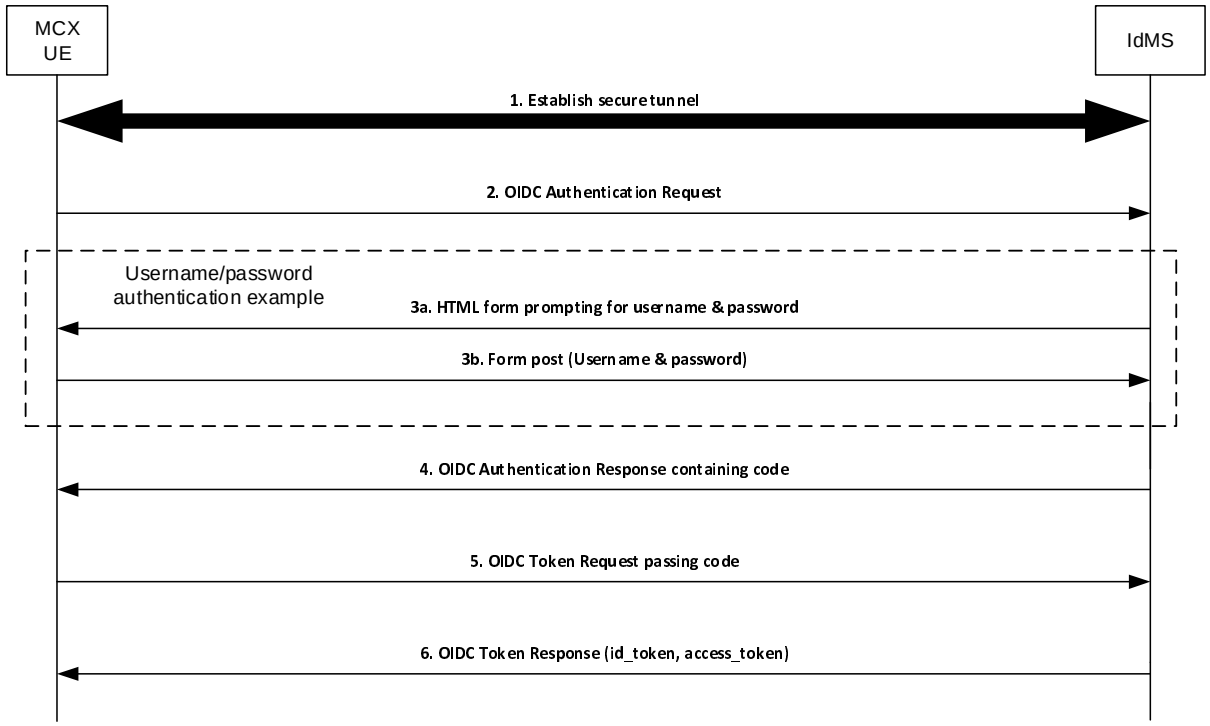

Figure 5.1.2.3.2-1: OpenID Connect (OIDC) Example Using Username/Password

Step 1: UE establishes a secure tunnel with the Identity Management server (IdMS).

Step 2: UE sends an OpenID Connect Authentication Request to the IdMS. The request may contain an indication of authentication methods supported by the UE.

Step 3a: IdMS sends an HTML form to UE prompting the user for their username & password.

Step 3b: UE sends the username & password (as provided by the user) to the IdMS.

Step 4: IdMS sends an OpenID Connect Authentication Response to the UE containing an authorization code.

Step 5: UE sends an OpenID Connect Token Request to the IdMS, passing the authorization code.

Step 6: IdMS sends an OpenID Connect Token Response to the UE containing an ID token and an access token (each which uniquely identify the user of the MCX service). The ID token is consumed by the UE to personalize the MCX client for the MCX user, and the access token is used by the UE to communicate the identity of the MCX user to the MCX server(s).

5.1.3 MCX user service authorisation

5.1.3.1 General

This clause expands on the MCX user service authorization step shown in figure 5.1.1-1 step C.

MCX User Service Authorization is the function that validates whether or not a MCX user has the authority to access certain MCX services. In order to gain access to MCX services, the MCX client in the UE presents an access token

(acquired during user authentication as described in subclause 5.1.2) to each service of interest (i.e. Key Management, MCX server, Configuration Management, Group Management, etc.). If the access token is valid, then the user is granted the use of that service. Figure 5.1.3.1-1 shows the flow for user authorization which covers key management authorization, MCX user service authorization, configuration management authorization, and group management authorization.

NOTE: All HTTP traffic between the UE and HTTP proxy, and all HTTP traffic between the UE and KMS (if not going through the HTTP proxy) is protected using HTTPS.

For key management authorization, the KM client in the UE presents an access token to the KMS over HTTP. The access token shall be scoped for key management services as defined in annex B.4.2.2. The KMS validates the access token and if successful, provides one or more sets of user specific key material back to the UE KM client based on the MC service ID(s) present in the access token (MCPTT ID, MCVideo ID and/or MCData ID). User specific key material includes identity based key information for media and signalling protection. If an interworking key management record (InterKMRec) exists and is associated to the requesting MC service ID (see clause 11.2.3), the KMS shall also provide the InterKMRec. This key management authorisation may be repeated for each KM service the user is authorised to use (MCPTT, MCVideo, MCData). In order to secure the transfer of user specific key material from the KMS to the KM client when using the TrK and InK, the KM client includes the TrK-ID and the InK-ID in the key management authorization request.

For MCPTT user service authorization, the MCPTT client in the UE presents an access token to the MCPTT server over SIP. The access token shall be scoped for MCPTT services as defined in annex B.4.2.2. The MCPTT server validates the access token and if successful, authorizes the user for full MCPTT services and sends an acknowledgement back to the MCPTT client. The MCPTT server then maps and maintains the IMPU to MCPTT ID association. The MCPTT ID to IMPU association shall only be known to the application layer. The SIP message used to convey the access token from the MCPTT client to the MCPTT server may be either a SIP REGISTER or SIP PUBLISH message.

For MCVideo service authorization, the MCVideo client in the UE presents an access token to the MCVideo server over SIP. The access token shall be scoped for MCVideo services as defined in annex B.4.2.2. The MCVideo server validates the access token and if successful, authorizes the user for full MCVideo services and sends an acknowledgement back to the MCVideo client. The MCVideo server then maps and maintains the IMPU to MCVideo ID association. The MCVideo ID to IMPU association shall only be known to the application layer. The SIP message used to convey the access token from the MCVideo client to the MCVideo server may be either a SIP REGISTER or SIP PUBLISH message.

For MCData user service authorization, the MCData client in the UE presents an access token to the MCData server over SIP. The access token shall be scoped for MCData services as defined in annex B.4.2.2. The MCData server validates the access token and if successful, authorizes the user for full MCData services and sends an acknowledgement back to the MCData client. The MCData server then maps and maintains the IMPU to MCData ID association. The MCData ID to IMPU association shall only be known to the application layer. The SIP message used to convey the access token from the MCData client to the MCData server may be either a SIP REGISTER or SIP PUBLISH message.

The UE can now perform configuration management authorization and download the user profile for the service(s) (MCPTT, MCVideo, MCData). Following the flow described in subclause 10.1.4.3 of 3GPP TS 23.280 [36] "MC service user obtains the MC service user profile(s) from the network", the Configuration Management (CM) client in the UE sends an access token in the user profile query to the Configuration Management server over HTTP. The access token shall be scoped for configuration management services as defined in annex B.4.2.2. The CM server receives the request and validates the access token, and if valid, the CM server uses the identity from the access token (MCPTT ID, MCVideo ID, MCData ID) to obtain the user profile from the MCX user database. The CM server then sends the user profile back to the CM client over HTTP. This configuration management authorisation may be repeated for each CM service the user is authorised to use (MCPTT, MCVideo, MCData).

Upon receiving each user profile, the Group Management (GM) client in the UE can now perform group management authorization. The GM client obtains the user's group membership information from the user profile, and following the flow shown in clause 10.1.5.2 of 3GPP TS 23.280 [36] "Retrieve group configurations at the group management client", the Group Management (GM) client in the UE sends an access token in the Get group configuration request to the host GM server of the group membership over HTTP. The access token shall be scoped for group management services as defined in annex B.4.2.2. The GM server validates the access token, and if valid, completes the flow. As part of group management authorization, group key information is provided as per subclause 5.7 of the present document. This group management authorisation may be repeated for each GM service the user is authorised to use (MCPTT, MCVideo, MCData).

For MC UEs that support mission critical location services, authorization is accomplished by including an access token in each location message (i.e. location information report, location reporting trigger, etc.) sent by the location management client to the location management server. The access token shall be scoped for location management services as defined in annex B.4.2.2. The location management server validates the access token and (if successful) processes the message (e.g. accepts and stores the location information report). If an access token cannot be validated, local policy may dictate an action to be taken within the location management server with regards to the received location message (e.g. the local policy may require storage of the location information report as an emergency provision).

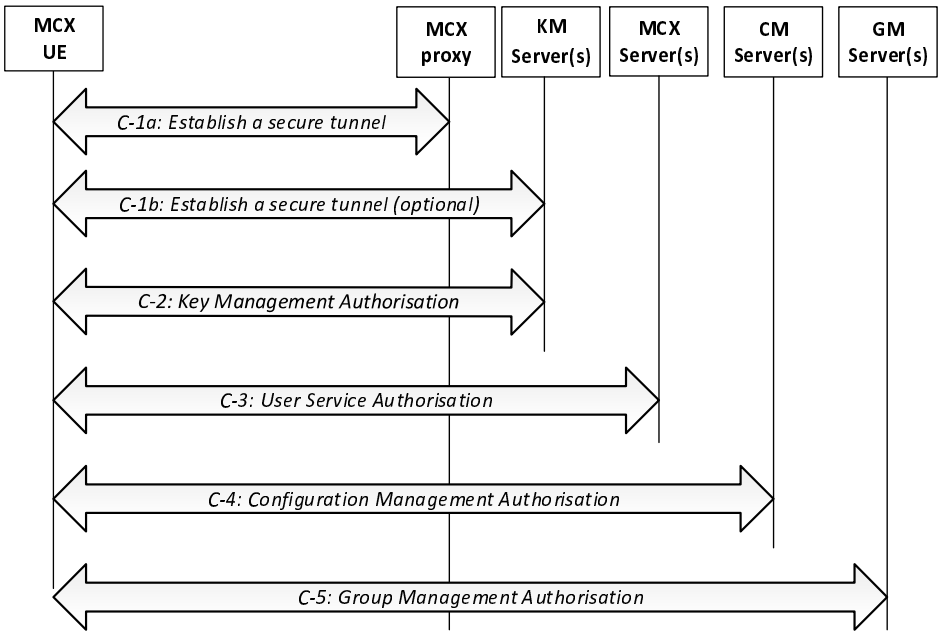

Figure 5.1.3.1-1: MCX user service authorization

The user authorization procedure in Step C of Figure 5.1.1-1 is further detailed into 5 sub steps that comprise the MCX user service authorization process:

Step C-1a:If not already done, establish a secure HTTP tunnel using HTTPS between the MCX UE and MCX proxy server. Subsequent HTTP messaging makes use of this tunnel .

Step C-1b: When required by the MCX system, establish a secure HTTP tunnel using HTTPS between the MCX KM client and the KMS. When supported, subsequent HTTP messaging between the MCX KM client and the KMS makes use of this tunnel in lieu of the tunnel set up in Step C-1a.

Step C-2: The KM client in the MCX UE presents an access token to the KMS over HTTP. The KMS authorizes the user for key management services based upon the MC service ID(s) provided and replies to the client with identity specific key information. This step may be repeated to authorise the user with additional KM services (MCPTT, MCVideo, MCData) as necessary.

Step C-3: The MCX client in the UE presents an access token to the MCX server over SIP as defined in clause 5.1.3.2 of the present document. This step may be repeated to authorise the user with additional MCX services (MCPTT, MCVideo, MCData) as necessary.

Step C-4: The CM client in the UE follows the "MCX user obtains the user profile (UE initiated)" flow from clause 10.1.4.3 of 3GPP TS 23.280 [36], presenting an access token in the Get MCX user profile request over HTTP. If the token is valid, then the CM server authorizes the user for configuration management services. Completion of this step results in the CM server providing the user's profile to the CM client. This step may be repeated as necessary to obtain the user profile for additional services (MCPTT, MCVideo, or MCData).

Step C-5: The GM client in the UE follows the "Retrieve group configurations at the group management client" flow as shown in clause 10.1.5.2 of 3GPP TS 23.280 [36], presenting an access token in the Get group configuration request over HTTP. If the token is valid, the GMS authorizes the user for group management services. Completion of this step results in the GMS sending the user's group policy information and group key information to the GM client. This step may be repeated to authorise the user for additional group services (MCPTT, MCVideo, MCData) as necessary.

5.1.3.2 MCX user service authorization with MCX Server

5.1.3.2.1 General

Depending on implementation, MCX user service authorization may be performed by sending the access token to the MCX server over the SIP-1 and SIP-2 reference points using either a SIP REGISTER message or a SIP PUBLISH message. Clause 5.1.3.2.2 describes how to use the SIP REGISTER message to transport the access token to the MCX server and clause 5.1.3.2.3 describes how to use the SIP PUBLISH message to transport the access token to the MCX server.

During initial SIP registration, the SIP REGISTER message shall not be delayed for lack of an access token. If an access token is not available then SIP registration shall proceed without the inclusion of the access token and the access token shall be transmitted to the MCX server as per Step C-3 in figure 5.1.3.1-1.

If an access token is available before SIP registration, or if the UE becomes de-registered and a SIP re-registration is required, the SIP REGISTER message may include the access token without requiring the user to re-authenticate.

The access token may be sent over SIP to the MCX server to re-bind an IMPU and MC service ID (MCPTT ID, MCVideo ID or MCData ID) if either have changed (e.g. IMPU is different due to SIP deregistration/SIP re-registration, or user logs out and another user logs onto the same UE).

5.1.3.2.2 Using SIP REGISTER

The use of a SIP REGISTER message to provide the access token to the MCX server is shown in figure 5.1.3.2.2-1. The inclusion of an access token in any particular SIP REGISTER message is optional.

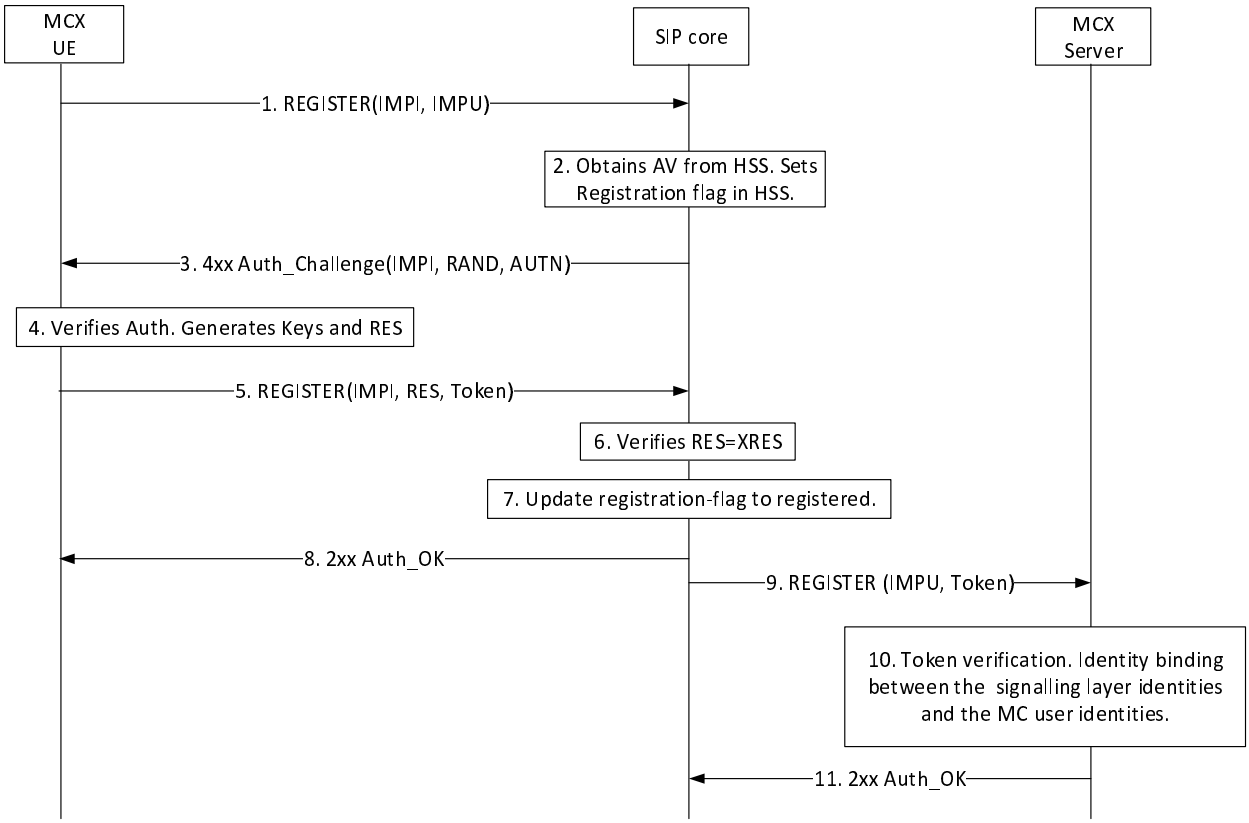

Figure 5.1.3.2.2-1: MCX User Service Authorization using SIP REGISTER message

Step 5 of figure 5.1.3.2.2-1 shows the access token message passed to the SIP core in a SIP REGISTER. Upon successful SIP authentication, the SIP core forwards the access token to the MCX server in the third part registration request message (Step 9).

In Steps 9 through 11, the MCX server receives the third part registration request message, validates the access token, binds the IMPU and MC service ID (MCPTT ID, MCVideo ID or MCData ID) if the access token is valid, and responds to the 3rd party registration message.

5.1.3.2.3 Using SIP PUBLISH

The use of a SIP PUBLISH message to provide the access token to the MCX server is shown in figure 5.1.3.2.3-1. The inclusion of an access token in any particular SIP PUBLISH message is optional.

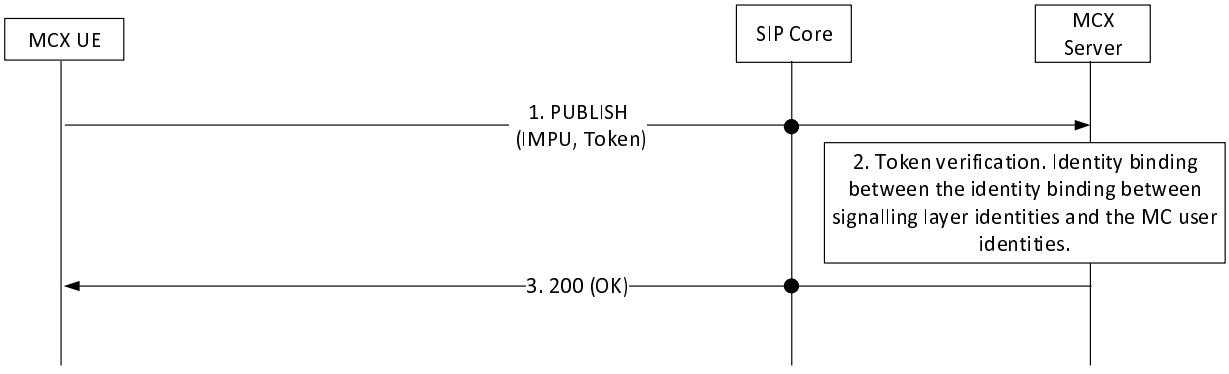

Figure 5.1.3.2.3-1: MCX User Service Authorization using SIP PUBLISH message

As shown in Step 1 of figure 5.1.3.2.3-1, the SIP PUBLISH message carries the access token through the SIP core to the MCX server.

In Steps 2 and 3, the MCX server receives the SIP PUBLISH message, validates the access token, binds the IMPU and MC service ID (MCPTT ID, MCVideo ID or MCData ID) if the access token is valid, and responds to the SIP PUBLISH message.

5.1.4 Inter-domain MC user service authorization

5.1.4.1 General

When a MC User requires service authorisation to a service that is located in a different Identity Management Domain, coordination between the identity management services of the primary Identity Management Domain and the partner Identity Management Domain is required. For example, a MC User from Identity Management Domain A may be a member of a group that is home to Identity Management Domain B within the same system or an MC user may migrate from their primary MC domain to a partner MC domain.

While inter-domain user service authorisation is not used for authorising users to services across interconnected MC systems (MC clients always connect directly to MC servers in their primary system with interconnection services provided via MC server to MC server communications), inter-domain user service authorisation shall be used for authorising migration of MC users.

This sub-clause shall be used for authenticating and authorizing a user that is home to Identity Management Domain A with a group service that is located in Identity Management Domain B or when a user from Identity Management Domain A migrates to a MC domain within Identity Management Domain B..

5.1.4.2 Inter-domain identity management functional model

The inter-domain identity management functional model is shown in Figure 5.1.4.2-1.

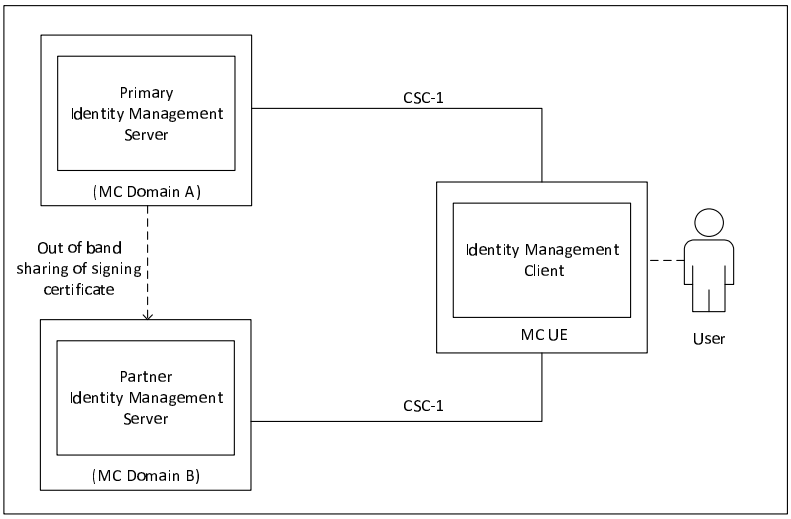

Figure 5.1.4.2-1: Functional Model for Inter-Domain Identity Management

In Figure 5.1.4.2-1, the IdMS located in the primary Identity Management Domain (MC Domain A) is the home identity management server for the user. The partner IdMS is located in a second Identity Management Domain (MC Domain B) and providesidentity mangement services for the primary user when authorising to partner group services or when the MC user is attempting to migrate.

The CSC-1 reference point between the UE IdM client and the partner IdM server endpoints shall be a direct connection and shall be protected with HTTPS (TLS).

The primary IdMS certificate(s) used to validate the user credentials at the partner IdMS are provisioned into the partner IdMS using an out of band mechanism beyond the scope of this document.

As defined in Clause 5.1.2 an access token is required for user service authorisation. The same principle applies for inter-domain user service authorisation, in that the MC client must present a valid access token issued from the partner IdMS in MC Domain B for authorisation to services located in MC Domain B.

The inter-domain identity management procedure shall be triggered when an MC client , after performing user service authorisation within the primary Identity Management Domain, determines that the user is a member of a group service that is located in a partner IdMS domain (as indicated in the user profile) .

Additionally, the inter-domain identity management procedure shall be triggered when a user attempts to migrate from their primary MC system to a partner MC system..

In order for the MC client to obtain the MC Domain B authorisation access token(s), the token exchange procedure with the primary IdM service (MC Domain A) shall be used to obtain a security token that identifies the user to the partner IdM service. This security token shall be specific to the partner IdM service and signed by the primary IdM service per IETF RFC 7515 [35]. Upon validation of the security token, the partner IdM service shall provide the access token(s) to the MC client specifically scoped for that user. The access token(s) shall provide the user with authorisation to the service(s) in the partner Identity Management Domain (MC Domain B) which may include services related to migration.

Figure 5.1.4.2-2 shows the token exchange and authentication procedure.

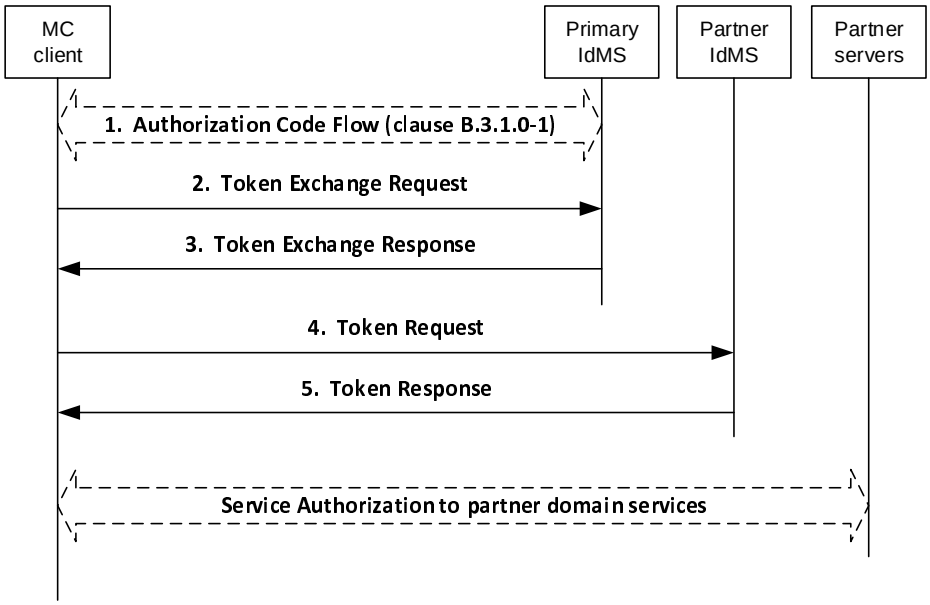

Figure 5.1.4.2-2: Token exchange procedure

The token exchange profile for accessing the partner identity management service (steps 1-5 in Figure 5.1.4.2-2) shall consist of [45] and [46] and shall be profiled as defined in Annex B.7.

NOTE: A specific and independent security token is required for each partner identity management domain.

Within a single MC System with interconnected MC domains, once the MC client obtains the access token specific to the partner group service(s) (step 5 in Figure 5.1.4.2-2), the MC client shall follow the user service authorisation procedure defined in clause 5.1.3 to access the group service(s) within the partner domain.

For migration of an MC user from their primary MC domain to a partner MC domain, once the MC client obtains the access token specific to the partner MC system (step 5 in Figure 5.1.4.2-2), the MC client shall follow the user service authorisation procedure defined in clause 5.1.5.

The token exchange procedure shall be repeated for each partner identity management domain where the MC client requires access and authorisation to group service(s) within that partner MC domain or when the user migrates from their primary MC system to a partner MC system.

Annex C.2 shows the detailed flow for inter-domain MC user service authorization using the OAuth 2.0 token exchange procedure.

5.1.5 MC user migration service authentication and authorisation

When an MC user migrates from their primary MC domain to a partner MC domain, MC user migration service authentication and MC user migration service authorisation shall be carried out prior to the migrated MC user receiving services at the partner MC domain.

Figure 5.1.5-1 shows the MC user migration service authentication and authorisation procedure.

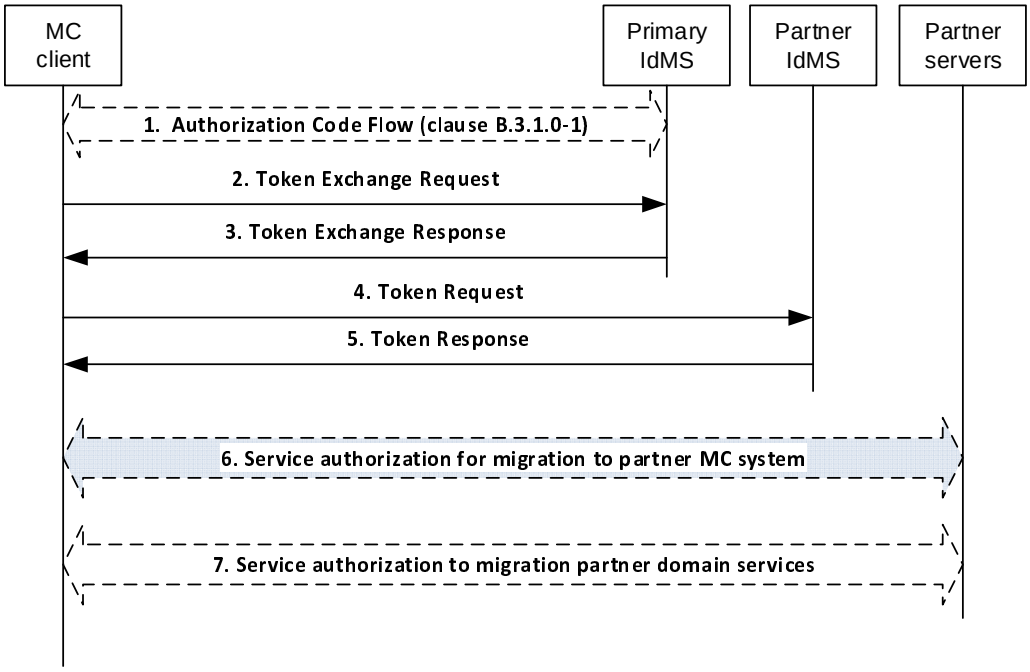

Figure 5.1.5-1 Service authorization for migration to partner MC system

1-5. MC user migration service authentication shall be the inter-domain identity management steps 1-5 in Figure 5.1.4.2-2 of clause 5.1.4.2.

6. Upon receiving a successful Token Response message, the MC client shall initiate the 'Service authorisation for migrating to a partner MC system' procedure as shown in Figure 5.1.5-2.

7. Following successful execution of step 6, service authorisation to services in the migration partner MC system shall be performed as defined in clause 5.1.3.

Figure 5.1.5-2 shows the 'Service authorisation for migrating to a partner MC system' procedure. Details of this procedure can be found in clause 10.6.3 of 23.280 [36].

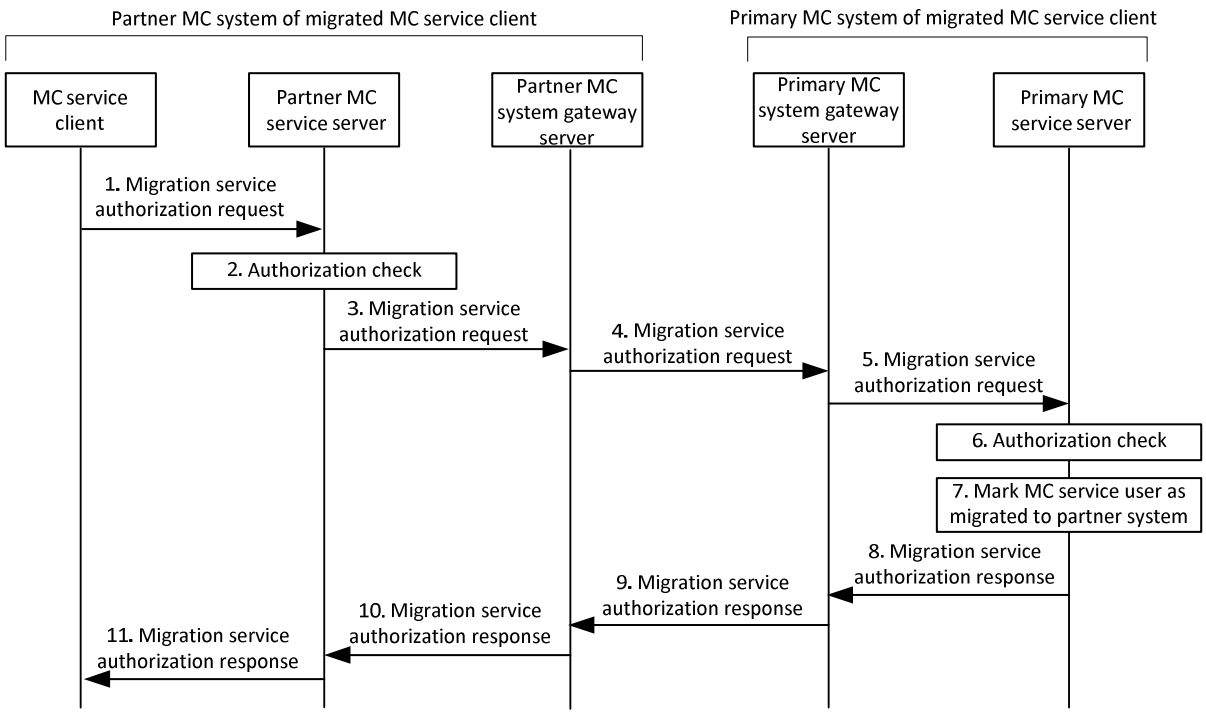

Figure 5.1.5-2 Service authorization for migration to partner MC system

1. The 'Migration service authorization request' message is sent by the MC service client to the partner MC service server and includes the access token obtained in step 5 of Figure 5.1.5-1.

2. The partner MC service server performs an initial authorization check to verify that the MC service user is permitted to migrate to the partner MC system. This step includes validation of the access token received in step 1 and shall be performed as defined in Annex B.11.

3-11. These steps are as defined in clause 10.6.3 of 23.280 [36].

NOTE: An access token is neither required nor provided in steps 3-11.

5.2 Key management common elements

5.2.1 Overview of key management

This clause details the key management procedures for MCX users. It allows entities in MCX systems to establish a security association to support future communications.

The primary purpose of these procedures is to allow MCX entities to communicate with each other using end-to-end security. End-to-end security provides assurance to MCX users that no unauthorized access to communications is taking place within the MCX network. End-to-end communication security may be applied to media when operating on-network and media, floor control, transmission control, and media control when operating off-network.

A security domain is managed by a Key Management Server (KMS). The KMS is a component of the Common Services Core within the MCX system architecture. For any end-point to use or access end-to-end secure communications, it needs to be provisioned with key material associated to its identity by the KMS. Through the use of the KMS, MC administrators are able to manage the use of, and access to, secure communications within the MCX network.

Key provisioning for groups is performed by a Group Management Server (GMS), authorized and provisioned by the KMS. The Group Management Server is responsible for distributing the key material to MCX users within the group. This establishes a group security context. With the group security context established, MCX users can communicate using end-to-end security.

Prior to protecting group communications during off-network operation, the UE shall acquire the necessary group key material either while operating on-network or through off-network provisioning.

NOTE: Void

Key provisioning for private communications is performed by the initiating UE as the communication is setup. This creates an end-to-end security context that is unique to the pair of users involved in the call. With a security context established, it may be used to encrypt media when on-network and, when off-network, media, floor control, transmission control, and media control traffic between the end-points.

Prior to protecting private calls during off-network operation, the UE shall acquire the necessary individual key material either while operating on-network or through off-network provisioning.

The key provisioning procedures described in this specification use common security methodologies for key distribution.

5.2.2 Common key distribution

The security mechanism described in this clause allows a key, K, to be distributed from an initiating party to a receiving party. It provides confidentiality of the key, and integrity and authenticity of the payload. It is used within a number of different security procedures in this specification.

The key, K, is distributed encrypted specifically to the receiving entity and signed by the initiating entity. Prior to call commencement, both MCX UEs shall be provisioned by the KMS with time-limited key material associated with the MCX entity's URI. The key is distributed with a 32-bit Key Identifier (K-ID). This payload is a MIKEY-SAKKE I_MESSAGE, as defined in IETF RFC 6509 [11], which ensures the confidentiality of the key, plus integrity and authenticity of the payload.

The key is encrypted to the user identity (UID) associated to the receiving MCX entity using the security domain parameters provided in the public values in the certificate received from the KMS. The UID used to encrypt the data is derived from the receiving entity's URI (e.g. sip:user.002@mcptt.example.org) and a time-related parameter (e.g. the current year and month). The terminating entity's URI is added to the recipient field (IDRr) of the message.

The payload includes the encrypted key and the key identifier (K-ID). The key is unique within the MC domain. On creating the key, the initiator generates a 32-bit key identifier (K-ID). The 4 most significant bits of the K-ID shall indicate the purpose of the key, the other 28-bits shall be randomly generated. The key identifier (K-ID) is stored in the CSB-ID field of the MIKEY I_MESSAGE.

The payload is signed using (the KMS-provisioned key associated to) the identity of the initiating entity. The UID used to sign the data is derived from the initiating entity's URI (e.g. sip:user.001@mcptt.example.org) and a time-related parameter (e.g. the current year and month). The initiating entity's URI is added to the initiator field (IDRi) of the message.

NOTE: This solution is for the end-to-end protection of keys and does not protect the identities transmitted. Identities may be masked by transmitting the UID within the MIKEY ID fields as described in Annex E.7.

The security processes are summarized in figure 5.2.2-1.

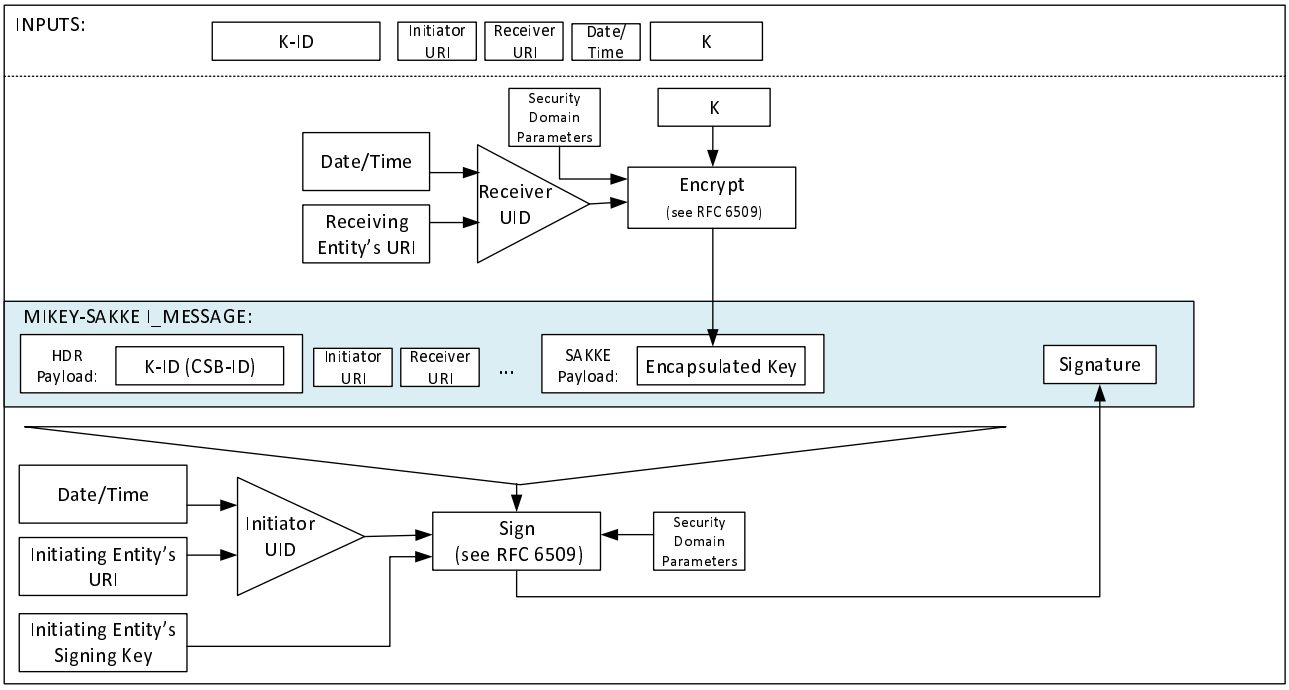

Figure 5.2.2-1: Common key distribution mechanism

Via this mechanism, the key distribution is confidentiality protected, authenticated and integrity protected.

It is possible that the key has been distributed using an unacceptable KMS, either for the initiator's KMS or for the receiver's KMS. This is particularly likely for communications being sent across multiple MC Systems (where KMS information may not have been shared prior to beginning the key distribution procedure). In this case, a KMS Redirect Response (KRR) may be sent back to the initiator. The KRR provides the initiator with information about which KMS may be acceptable. KRR procedures are described in clause 5.2.8.

Assuming that acceptable KMS(s) have been used, the I_MESSAGE will be processed by the receiving entity. The initiating entity's URI is extracted from the initiator field (IDRi) of the message. This is converted to a UID and used to check the signature on the MIKEY-SAKKE I_MESSAGE. If valid, the UE extracts and decrypts the encapsulated key, K, using the (KMS-provisioned) entity's UID key. The MCX entity also extracts the K-ID. This process is shown in figure 5.2.2-2.

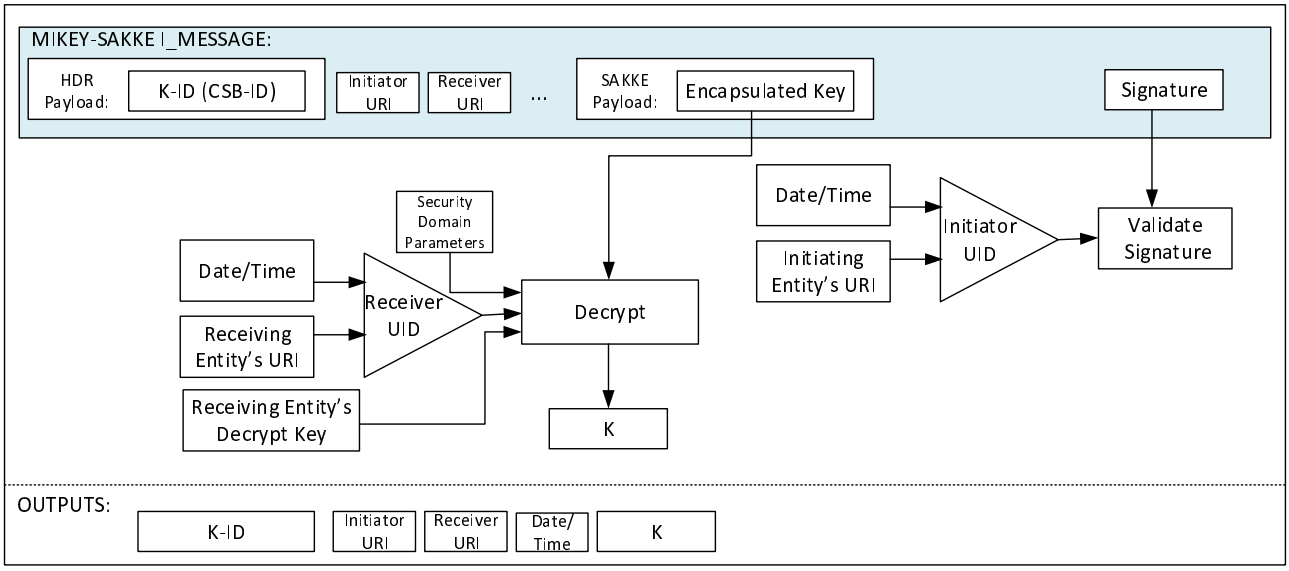

Figure 5.2.2-2: Common key extraction mechanism

With the key successfully shared between the two MCX entities, the entities are able to use the shared security context to protect communications.

5.2.3 Key distribution with end-point diversity

The security mechanism described in this clause extends that defined in clause 5.2.2 to provide end-point key diversity. The mechanism is identical to that described in clause 5.2.2, except for the distribution of K-ID. Contrary to clause 5.2.2, the key is distributed with an end-point-specific key identity (UK-ID) (e.g. a GUK-ID) derived from the key id (K-ID). This allows the receiving entity of the key distribution to diversify the shared key for end-point-specific use.

Specific types of key require use of end-point key diversity. The type of key is defined by the 'purpose tag' within the key identifier stored in the CSB-ID field of the MIKEY payload. Hence on receipt of a key, the contents of the CSB-ID field instruct the receiving entity whether end-point diversity should be applied to the key.

The key, K, is distributed encrypted specifically to the receiving entity and signed by the initiating entity as described in clause 5.2.2. The key is distributed with a 32-bit entity-specific Key Identifier (UK-ID) derived from a common key id (K-ID) and a salt (which is derived from the receiving entity's MCX URI). The security domain parameters are provided in the public values in the certificate received from the KMS.

The payload includes the entity-specific Key Identifier (UK-ID) within the CSB-ID field. The key, K, is identified by a Key Identifier (K-ID) from which the UK-ID is derived. On creating the key, K, the initiating entity generates a K-ID as follows. The 4 most significant bits of the K-ID is the 'purpose tag' which defines the purpose of the key. The 28 least significant bits of the K-ID is a 28-bit randomly-generated value.

For each receiving entity, the initiating entity creates a 28-bit Salt by hashing the receiving entity's URI through a KDF using the key, K, as the key (as defined in Annex F.1.3). The Salt is xor'd with the 28 least-significant bits of the K-ID to create the 32-bit UK-ID.

NOTE: Knowledge of the UK-ID, K-ID and Salt does not reveal the receiving entity URI to those without the key K.

The process for generating the UK-ID is summarized in figure 5.2.3-1.

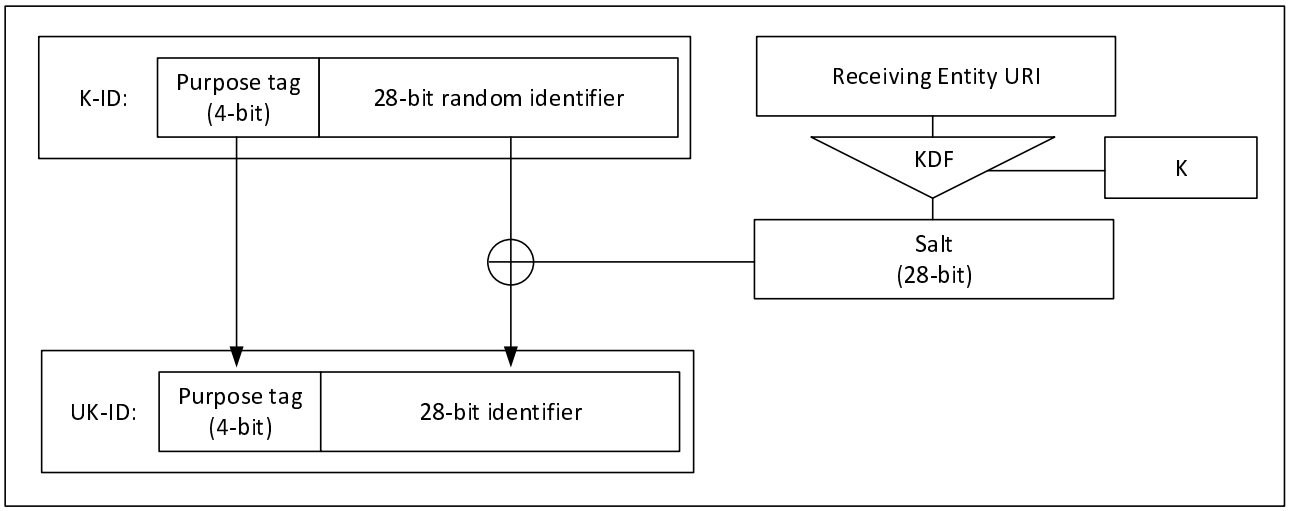

Figure 5.2.3-1: Generating the UK-ID

The UK-ID is placed in the CSB ID field within the header of the I_MESSAGE. The security processes are summarized in figure 5.2.3-2.

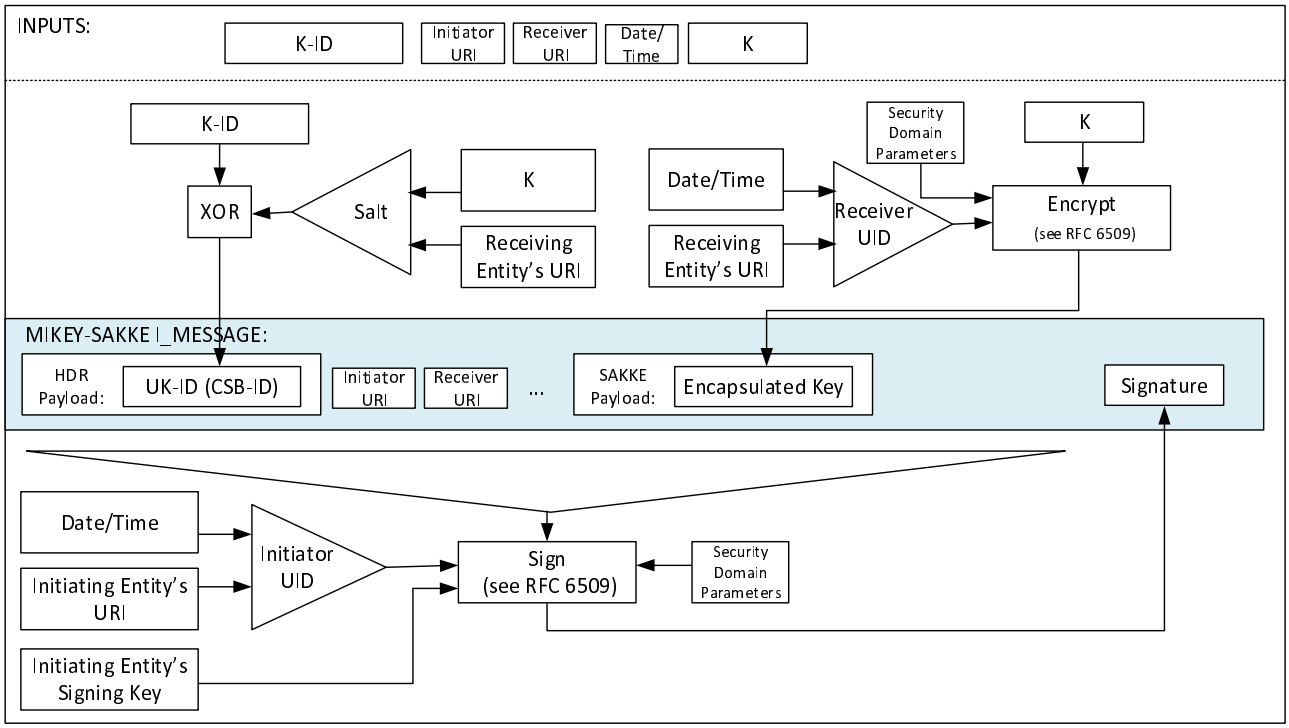

Figure 5.2.3-2: Common key distribution mechanism with end-point diversity

At the receiving MCX entity, the initiating entity's URI is extracted from the initiator field (IDRi) of the message. Along with the time, this is used to check the signature on the payload. If valid, the receiving entity extracts and decrypts the encapsulated key, K, using the (KMS-provisioned) entity's UID key.

The receiving MCX entity also extracts UK-ID from the CSB-ID field of the I_MESSAGE. If the 'purpose tag' of the UK-ID indicates that end-point diversity is applied, the receiving entity generates the Salt using its URI and the decrypted key, K. The receiving entity xors the UK-ID and Salt together to obtain the K-ID. The K-ID and UK-ID are stored.

The extraction procedure is described in figure 5.2.3-3.

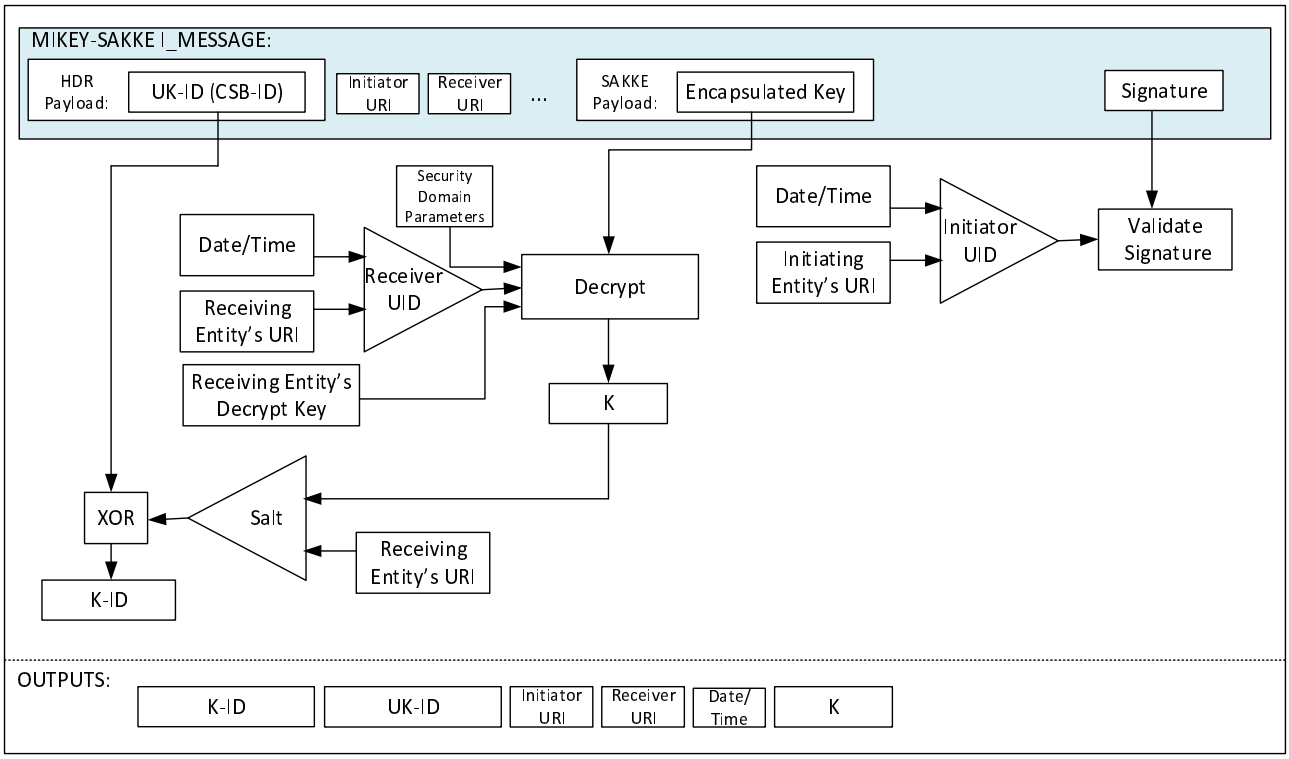

Figure 5.2.3-3: Common key extraction mechanism with end-point diversity

5.2.4 Key distribution with associated parameters

The key distribution mechanisms described in Clause 5.2.2 and clause 5.2.3 may be extended to include data associated with the key in the MIKEY I_MESSAGE. This data is stored within a format known as 'associated parameters' and defined in Annex E.6.

The associated parameters are encrypted using K, the key distributed within the MIKEY I_MESSAGE. The security mechanism is summarised in Figure 5.2.4-1.

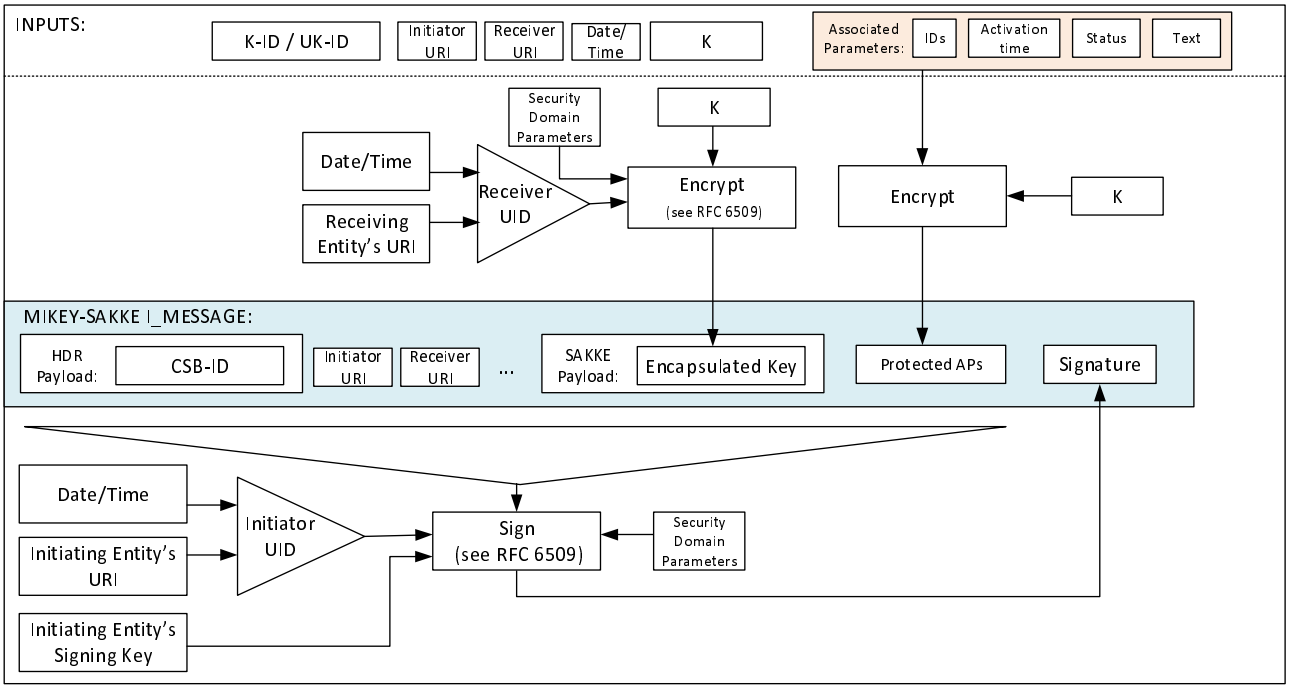

Figure 5.2.4-1: Common key distribution mechanism with associated parameters

At the receiving MCX entity, the initiating entity's URI is extracted from the initiator field (IDRi) of the message. Along with the time, this is used to check the signature on the payload. If valid, the receiving entity extracts and decrypts the encapsulated key, K, using the (KMS-provisioned) receiving entity's decryption key.

The receiving MCX entity also extracts 'associated parameters' payload from the I_MESSAGE. The receiving entity uses the decrypted key, K, to decrypt these associated parameters. The receiving entity stores these parameters with the distributed key, K. If the Status field within the ‘associated parameters' payload indicates the key has been revoked, the distributed key, K, and the K-ID shall not be used. If the decryption process for the encapsulated associated parameters fails, the key is rejected.

The security mechanism is summarised in Figure 5.2.4-2.

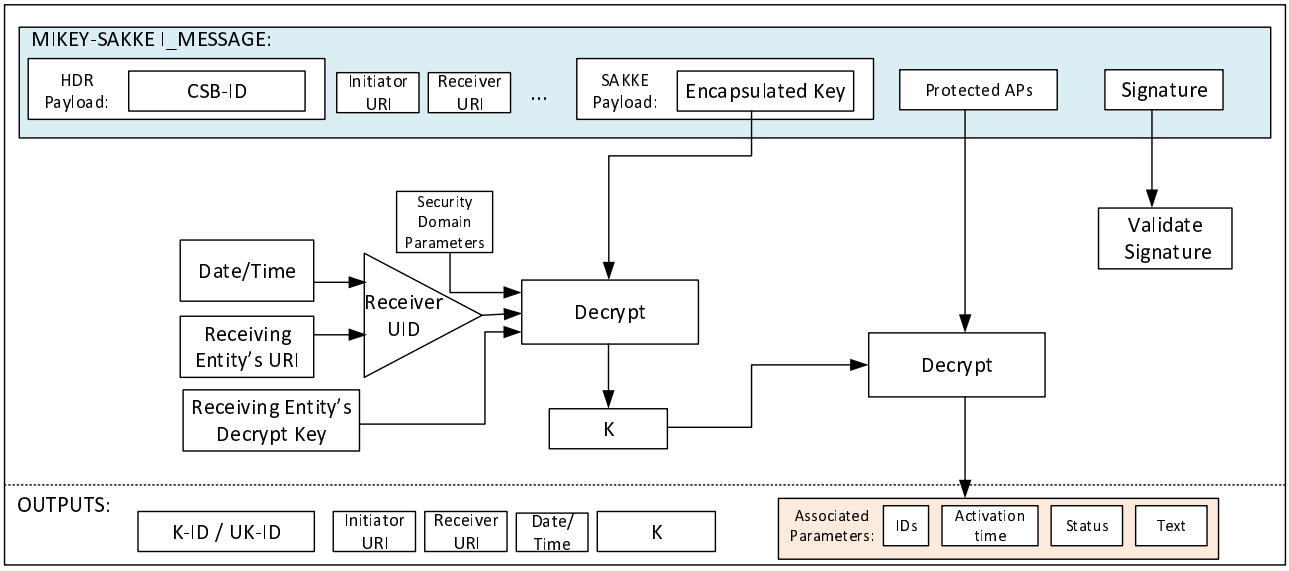

Figure 5.2.4-2: Common key extraction mechanism with associated parameters

5.2.5 Key distribution with SAKKE-to-self payload

The key distribution mechanism defined in clauses 5.2.2, 5.2.3 and 5.2.4 may be extended to allow the initiating entity to be able to decrypt the distributed key, K contained within the payload.

NOTE: Where the initiating entity is an MCX user logged into multiple devices, this extension is necessary to allow all devices to obtain the key, K and decrypt any subsequent communication.

In addition to encrypting the key, K, to the receiving entity, the key is also encrypted to the initiating entity. The UID used to encrypt the data is derived from the initiating entity's URI (e.g. sip:user.002@mcptt.example.org) and a time-related parameter (e.g. the current year and month). The encapsulated key is added to a SAKKE-to-self payload within the MIKEY I_MESSAGE. No other payloads (e.g. IDRr) are affected.

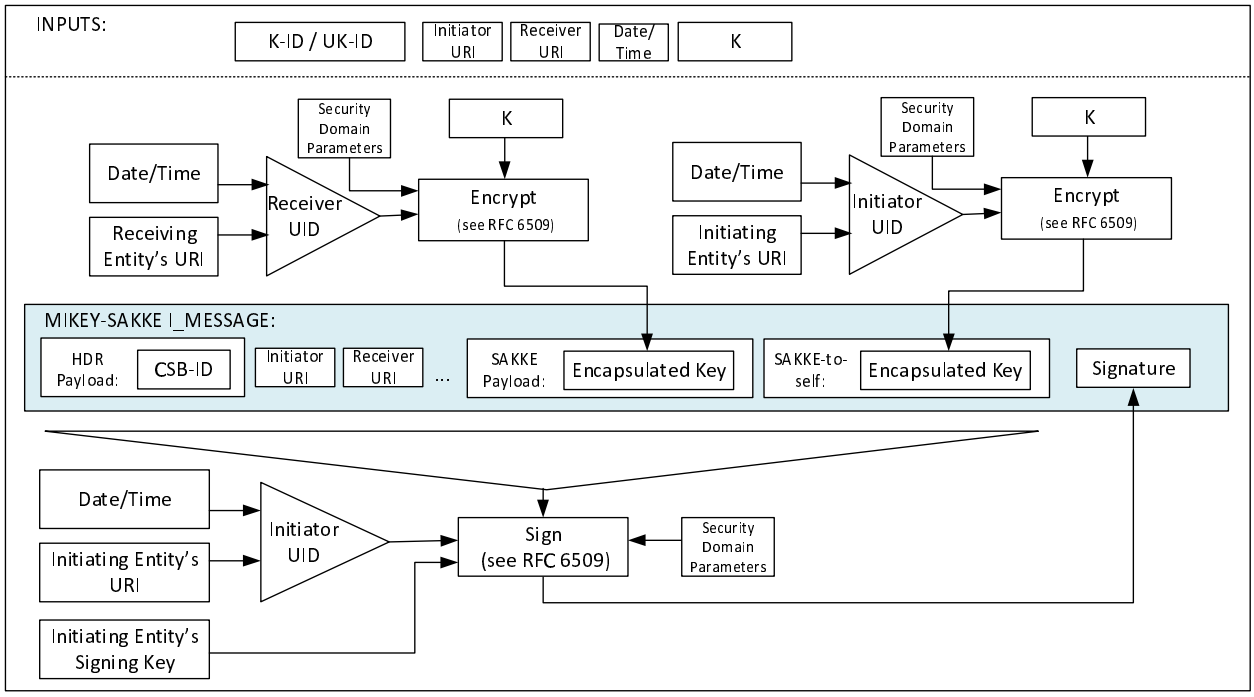

Figure 5.2.5-1: Common key distribution mechanism with SAKKE-to-self payload

5.2.6 Key distribution with identity hiding

The key distribution mechanism defined in clauses 5.2.2, 5.2.3, 5.2.4 and 5.2.5 may be extended to allow identities to be masked within the MIKEY payload. This is achieved by adding the UID, rather than the URI to the payload as described in Annex E.7 and shown in figure 5.2.6-1.

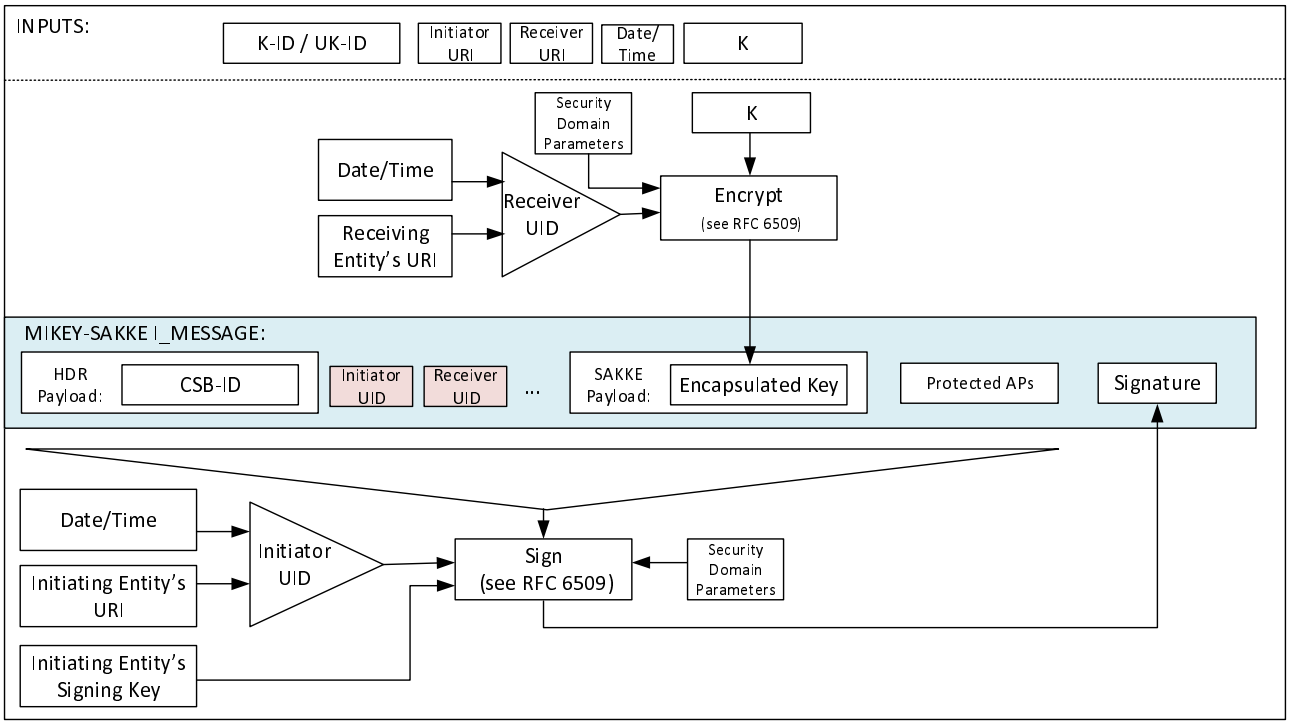

Figure 5.2.6-1: Common key distribution mechanism with identity hiding

On receipt of a MIKEY payload with identities hidden, the receiving entity should recognise the receiver UID in the packet. If not, the I_MESSAGE shall be rejected. Based on the initiator UID, the receiver checks the validity of the

I_MESSAGE signature. At this point the initiator is anonymous to the receiver. If this check fails, the I_MESSAGE shall be rejected. The receiver then extracts the key K. This may be used to decrypt other parts of the packet and extract the initiator URI. Once the initiator URI is extracted, this shall be used to generate the initiator UID and check that it is the one provided in the I_MESSAGE. If not, the I_MESSAGE shall be rejected. This procedure is shown in figure 5.2.6-2

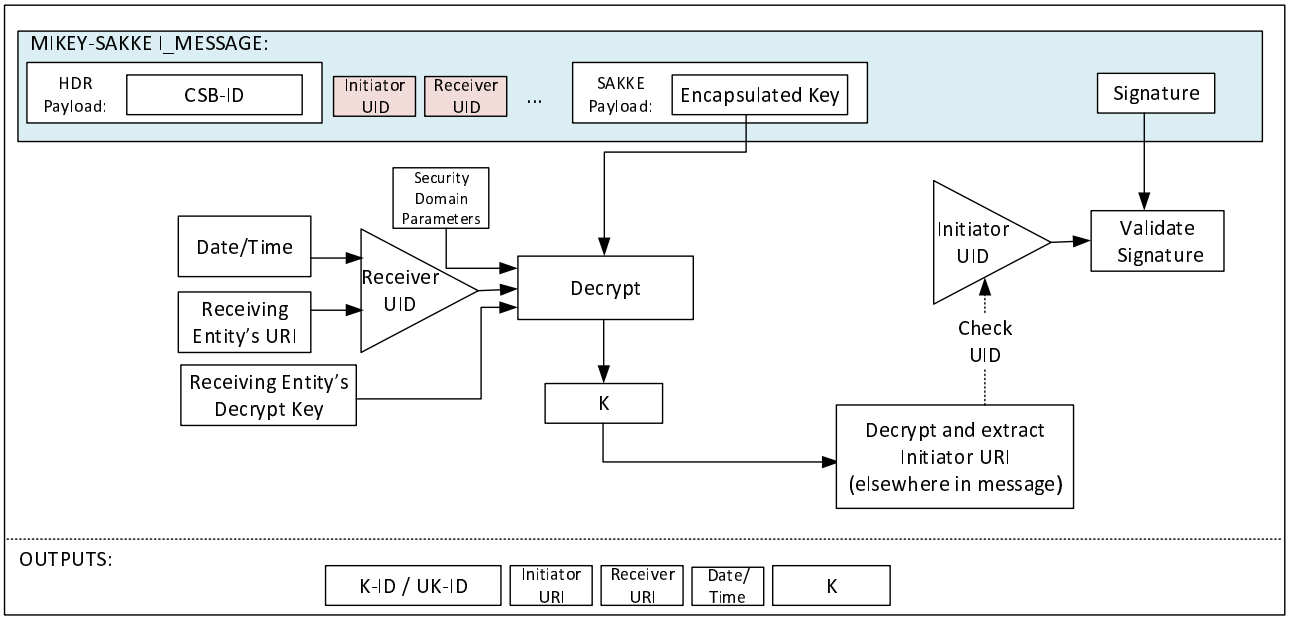

Figure 5.2.6-2: Common key extraction mechanism with identity hiding

5.2.7 Key distribution across multiple security domains

5.2.7.1 General

5.2.7.2 Identification of External Security Domains

To support multiple security domains, the security domain used by each user is recorded alongside the user's MC Service ID within configuration parameters in the MC system. Furthermore, the security domain of the GMS is recorded alongside the GMS FQDN and the security domain of the MCX Server is recorded alongside the MCX Server FQDN. Security domains are identified by a unique identifier, the 'KMSUri'. Specifically, the following describes the situations where security domain information is needed:

1) The MCX Server(s) requires knowledge of the security domain (KMSUri) of users connected to the server.

2.1) On initiating a MCPTT private call, the initiating UE requires knowledge of the security domain (KMSUri) of the receiving user.

2.2) On receiving a MCPTT private call, the receiving UE requires knowledge of the security domain (KMSUri) of the initiating user.

3.1) On initiating a MCVideo private call, the initiating UE requires knowledge of the security domain (KMSUri) of the receiving user.

3.2) On receiving a MCVideo private call, the receiving UE requires knowledge of the security domain (KMSUri) of the initiating user.

4.1) On initiating a MCData one-to-one SDS or file transfer, the initiating UE requires knowledge of the security domain (KMSUri) of the receiving user.

4.2) On receiving a MCData one-to-one SDS or file transfer, the receiving UE requires knowledge of the security domain (KMSUri) of the initiating user.

5) The Group Management Server requires knowledge of the security domain (KMSUri) of each member of the group.

6) Group members require knowledge of the security domain (KMSUri) of the group management server.

7) MC users require knowledge of the security domain (KMSUri) of the MCX Server(s) to which they connect.

NOTE: In most cases, the required security domain will be the Home security domain, meaning that the required KMSUri will be the user's Home KMSUri. It may be more space efficient to only keep a record where the KMSUri is not the Home KMSUri.

5.2.7.3 Using multiple security domains

On encrypting to an entity within the MC System using an I_MESSAGE, the client shall lookup the KMSUri from the appropriate configuration data, then lookup the appropriate KMS Certificate with that KMSUri from the certificate cache downloaded from it's home KMS. The security parameters within the KMS Certificate are used to perform encryption. The KMSUri is added to the I_MESSAGE within the IDRkmsr field.

Equivalently, when verifying a received I_MESSAGE, the receiving client shall extract the KMSUri from the I_MESSAGE (if present) and check this matches the KMSUri from the appropriate configuration data. The client shall then lookup the appropriate KMS Certificate with that KMSUri from the certificate cache downloaded from it's home KMS. The security parameters within the KMS Certificate are used to perform verification.

Should a matching certificate not be found, the client may request the certificate based on the KmsUri from it's home KMS using an appropriate KMS Cert request, as defined in Clause D.2.6.

5.2.8 KMS Redirect Responses (KRRs)

5.2.8.1 Overview of KMS Redirect Response procedure (KRR)

5.2.8.1.1 General

The purpose of KMS Redirect Response procedures is to allow key distribution where the KMS used by the receiver is not known. It also allows policy to be applied to ensure the KMS used by the receiver and initiator is acceptable along the path of the communication.

The key message is a KMS Redirect Response (KRR) which conveys KMS policy to the initator. The initiator could be a MC client or GMS. Sometimes multiple MIKEY messages and KRR exchanges will be required to establish a suitable choice of (KMS initiator, KMS receiver) pair. It is also possible that there is no acceptable choice, and as a result of the process the communication fails.

The partner (external) security domains (KMS URIs) and certificates are typically provisioned to the UE by the user's Home KMS (see Annex D). The KRR procedure does not provision KMS certificates, but shares information about which KMS may be used with the target key management client.

The following scenarios may trigger a KRR procedure in order to communicate KMS information to the initiating entity:

- The KMS URI (IDRkmsr) used in the MIKEY message may be incorrect for the target; or

- While the specified KMS URI may be correct for the receiver, the primary or partner application server may for various reasons still disallow communications with the target entity using the specified receiver KMS URI (IDRkmsr); or

- While the specified KMS URI may be correct for the receiver, the primary or partner application server may for various reasons still disallow communications with the receiver using the specified initiator KMS URI (IDRkmsi);

The KRR procedure may be initiated by application servers in the signalling path or it may be initiated by the terminating MCX entity. Client shall support receipt of KRRs, and may process or ignore the KRR based on local policy.

5.2.8.1.2 KMSs and KMS URIs

The KMS URI is the URI used to identify a logical KMS. This represents a security domain of users with shared trust. A logical KMS supports exactly one security domain, hence there is a one-to-one correspondence between KMS URIs, security domains and logical KMSs.

The types and uses of KMSs are described in Clause 5.3.

5.2.8.2 Use of KRRs

5.2.8.2.1 Content of KRRs

The KMS Redirect Response (KRR) contains a list of KMS URIs for both the initiator and the receiver. Both the initiator list and receiver list is an ordered list, with the preferred KMS URIs first. The KMS URI list can also be 'Any' meaning that any KMS is acceptable.

The content of a KRR is:

- An identifier for this type of response.
- The date and time.
- The identity of the KRR creator.
- The MIKEY initiating identity used within the MIKEY message (IDRi).
- The MIKEY initiator's KMS URI used within the MIKEY message (IDRkmsi).
- The MIKEY receiving identity used within the MIKEY message (IDRr).
- The MIKEY receiving's KMS URI used within the MIKEY message (IDRkmsr).
- The initiator list containing a list of acceptable KMS URIs (List of IDRkmsi options).
- The receiver list containing a list of acceptable KMS URIs (List of IDRkmsr options).
- An embedded received KRR (if this KRR is generated as a result of a received KRR).
- A signature (using the originating identity) over the entire message (optional, but recommended).

All fields, except for the signature, are required content.

5.2.8.2.2 KRR creation procedure by a receiver

The KRR initiator and receiver lists (as defined in clause 5.2.8.2.1) are populated based on the received MIKEY message from the initiator. The message contains an initiating KMS URI (IDRkmsi) and receiving KMS URI (IDRkmsr).

1) The KRR initiator list is populated as follows:

   a) If this is the first received MIKEY message from the initiator, the receiver may respond with a preferred list of KMS URIs based on local policy. If IDRkmsi corresponds to one of the receiver's External KMSs, the initiator list shall contain, at minimum, the IDRkmsi.

   b) Otherwise, the IDRkmsi does not correspond to one of the receiver's External KMSs, and a list of KMS URIs corresponding to External KMSs is provided based on local policy (not all KMS URIs need be included).

2) The KRR receiver list is populated as follows:

   a) If this is the first received MIKEY message from the initiator, the receiver may respond with a preferred list of KMS URIs based on local policy. If the IDRkmsr corresponds to one of the receiver's Home KMS or a provisioned Migration KMS, the receiver list shall include, at a minimum, the IDRkmsr.

b) Otherwise, the IDRkmsr does not correspond to one of the receiver's Home KMS or a provisioned Migation KMS, and a list of KMS URIs corresponding to the Home KMS and Migration KMSs is provided based upon local policy (not all KMS URIs need be included).

5.2.8.2.3 KRR creation procedure by a MCX server or signalling proxy

A MCX Server or Signalling proxy can create a KRR on receipt of a MIKEY message from the initiator en route to the receiver. The message contains an initiating KMS URI (IDRkmsi) and receiving KMS URI (IDRkmsr). A KRR is created under the following conditions.

Case A: For MIKEY messages entering from a MC client (inbound CS Proxy), a KRR is created if the IDRkmsi is not acceptable. This could be either that the KMS is not supported within the domain, or that the KMS is not supported for the user given the user's current state, location or juristriction. In this case:

1. the initiator KMS URI list contains a list of acceptable KMS URIs supported by the domain for the user based on the user's current state.

2. the receiver KMS URI list shall be 'ANY'.

Case B: For MIKEY messages entering/leaving a domain (IS Proxy), if the initiating user (IDRi) relates to this domain and the IDRkmsi is not acceptable then:

1. the initiator KMS URI list contains a list of acceptable Home and Migration KMS URIs used by the IDRi for this domain..

2. the receiver KMS URI list is 'ANY'.

NOTE 1: This case is primarily used where the initiator has migrated out of the domain, meaning that the user's traffic is transiting the domain, but ultimately enters/exits the domain via an IS Proxy.

Case C: For MIKEY messages entering/leaving a domain (IS Proxy), if the receiving user (IDRr) relates to this domain and the IDRkmsr is not acceptable then:

1. the initiator KMS URI list is 'ANY'.

2. the receiver KMS URI list contains a list of acceptable Home and Migration KMS URIs used by the IDRr for this domain.

NOTE 2: This case is primarily used where the receiver has migrated out of the domain, meaning that the user's traffic is transiting the domain, but ultimately enters/exits the domain via an IS Proxy.

Case D: For MIKEY messages exiting towards a MC client (outbound CS Proxy), a KRR is created if the IDRkmsr is not acceptable. This could be as the KMS is not supported given the user's current state, location or juristriction. In this case:

- the initiator KMS URI list shall be 'ANY'

- the receiver KMS URI list contains a list of acceptable KMS URIs supported by the domain based on the user's (IDRr) current state.

Should any network entity create a KRR, the network entity shall drop the received MIKEY message. Entities in the path receiving a KRR shall forward the KRR towards the initiating IDRi.

5.2.8.2.4 Processing a KRR at a MCX server or signalling proxy

A MCX Server or Signalling proxy can create a new KRR on receipt of a KRR. The content of the KRR is based on local policy of which KMSs are supported within the domain. A new KRR is created under the following conditions:

Case A: For KRRs entering the domain from a MC client (inbound CS Proxy), a new KRR is created if the contents of the receiver KMS URI list contains a KMS URI that is not acceptable. This could be as the KMS is not supported given the user's current state, location or juristriction. Within the new KRR in this case:

1. the initiator list is unchanged.

2. the receiver list is reduced to remove the unacceptable KMS URIs. If the list is empty, an empty list is returned within the KRR (and consequently, the communication will fail).

Case B: For KRRs entering/existing the domain towards another domain (IS Proxy), a new KRR is created if the receiving user (IDRr) relates to this domain and the receiver list contains a KMS that is not acceptable then within the new KRR:

1. the initiator list is unchanged.

2. the receiver list is reduced to remove the unacceptable KMS URIs. If the list is empty, an empty list is returned within the KRR (and consequently, the communication will fail).

Case C: For KRRs entering/exiting the domain from another domain (inbound IS Proxy), a new KRR is created if the initiating user (IDRi) relates to this domain and the initiator list contains a KMS that is not acceptable then within the new KRR:

1. the initiator list is reduced to remove the unacceptable KMS URIs. If the list is empty, an empty list is returned within the KRR (and consequently, the communication will fail).

2. the receiver list is unchanged.

Case D: For KRRs exiting the domain towards a MC client domain (outbound CS Proxy), a new KRR is created if the contents of the initiator KMS URI list contains a KMS URI that is not acceptable. This could be as the KMS is not supported given the user's current state, location or juristriction. Within the new KRR in this case:

1. the initiator list is reduced to remove the unacceptable KMS URIs. If the list is empty, an empty list is returned within the KRR (and consequently, the communication will fail).

2. the receiver list is unchanged.

Should the network entity create a new KRR, the received KRR is dropped and the new KRR is forwarded to the initiator. Entities in the path receiving a KRR shall forward the KRR towards the initiating IDRi.

The new KRR contains the received KRR. Consequently, the KRR could contain multiple sub-KRRs. It is recommended that a maximum of 5 sub-KRRs are supported.

5.2.8.2.5 KMS Selection at the initiator

Where the initiator is distributing a key to a receiver (e.g. at the beginning of a private call) it is possible that a KMS selection procedure needs to be performed by the initiator. The KMS selection procedure results in the choice of an initiator and receiver KMS (IDRkmsi and IDRkmsr) for the MIKEY message.

The KMS selection procedure is only required in two situations:

- Initial distribution of a key where the receiver's KMS is not known (e.g. the receiver's KMS is not listed in the user profile, the group document, or known due to previous communication).

- Upon receipt of a KRR due to a previous attempt to distribute a key.

In the first case, (ANY, ANY) is used as the initiator KMS list and receiver KMS list pair. Otherwise the initiator KMS list and receiver KMS list is provided within the KRR.

Using the provided initiator KMS list and receiver KMS list, the initiator shall select the initiator KMS and receiver KMS based on the following procedure:

1. For the initiator KMS list, the initiator shall:

   a. If the initiator KMS list is 'ANY' the initiator shall populate the KMS list with the Home KMS and with all provisioned Migration KMSs.

   b. If the KMS list is not empty, the initiator shall create a reduced list of all KMS URIs that do not belong to the initiator's Home KMS or to a provisioned Migration KMS. If the reduced list still contains at least one KMS URI; then:

      i. The initiator may apply local policy to select a KMS URI from the reduced list; the initiator shall use the selected KMS (to sign the MIKEY message); else

ii. If the KMS list contains the initiator's Home KMS URI; the initiator shall use the Home KMS (to sign the MIKEY message); else

ii. The initiator shall select the first KMS URI from the list. The initiator shall use the selected KMS (to sign the MIKEY message);

d. If the reduced list contains no KMS URIs, then the communication fails.

2. For the receiver KMS list, the initiator shall:

a. If the receiver KMS list is 'ANY' the initiator shall populate the receiver KMS list with the initiator's Home KMS, with all provisioned Migration KMSs and with all provisioned External KMSs.

b. If the receiver KMS list is not empty, the initiator shall create a reduced list of all KMS URIs that do not belong to the initiator's Home KMS, to a provisioned Migration KMSs or to an External KMS. If the reduced list still contains at least one KMS URI; then:

i. The initiator may apply local policy to select a KMS URI from the reduced list; the initiator shall use the selected KMS (to encrypt the MIKEY message); else

ii. If the KMS list contains the initiator's Home KMS URI; the initiator shall use the Home KMS (to encrypt the MIKEY message); else

ii. The initiator shall select the first KMS URI from the list. The initiator shall use the selected KMS (to encrypt the MIKEY message);

d. If the reduced list contains no KMS URIs, then the communication fails.

If an initiating and receiving KMS has been successfully selected, the initiator shall send a new MIKEY message using the selected KMSs. If not, the communication fails.

The purpose of KMS Discovery / Redirection procedures is to allow session key distribution where the KMS used by the receiver is not known. It also allows policy to be applied to ensure the KMS used by the receiver and initiator is acceptable along the path of the communication.

The key message is a KMS Redirect Response (KRR) which conveys KMS policy to the initator. The initiator could be a MC client or GMS. Sometimes multiple messages and KRR exchanges will be required to establish a suitable choice of (KMS initiator, KMS receiver) pair. It is also possible that there is no acceptable choice, and as a result of the process the communication fails.

5.2.8.3 Security procedures for KMS Redirection Response

The KMS Redirect Response (KRR) procedure allows for MC Services to negotiate and inform an MCX entity about which security domains (KMS URIs) are acceptable for an MCX session.

Prior to beginning this process, it is assumed that:

- The initiating user has been provisioned by its Home KMS with some information on the permitted external security domains, including the KMS certificate of External KMSs.

- The terminating user has been provisioned by its Home KMS with some information on the permitted external security domains, including the KMS certificate of External KMSs.

A user shall only communicate with its Home KMS. External KMS Certificates shall be manually loaded onto the Home KMS and distributed to the user as part of the KMS's user key management processes.

The procedure for security domain redirection is shown in Figure 5.2.8.3-1.

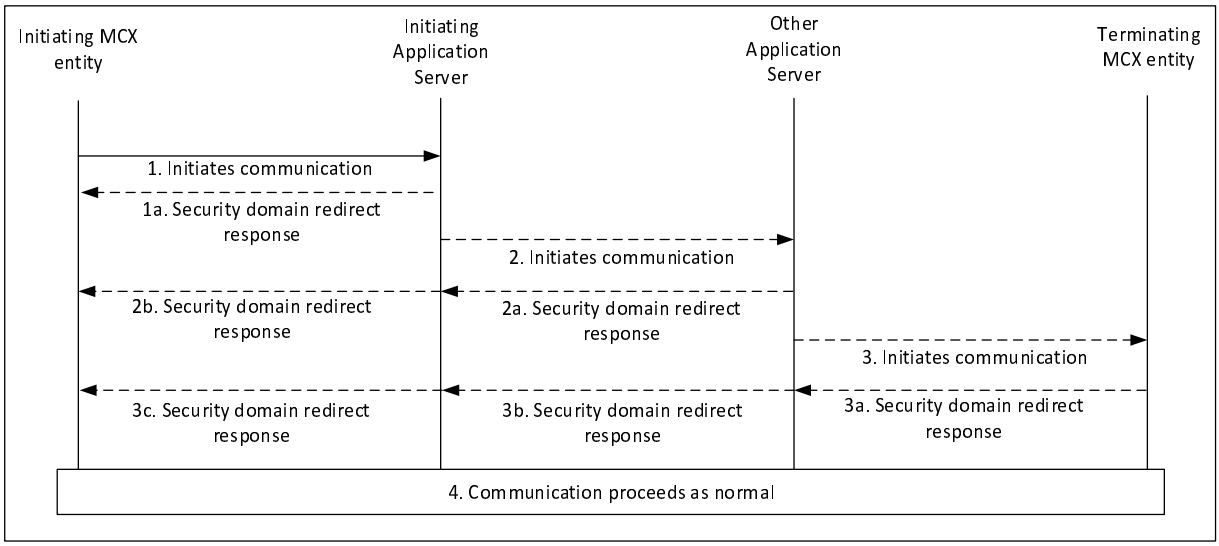

Figure 5.2.8.3-1: Security domain redirection

The procedures in Figure 5.2.8.3-1 are now described in detail. Where the security domains (KMS URIs) used by the initiating client are acceptable to the MC Service(s) and terminating client, communication proceeds as normal. However, where the initiating client uses security domain(s) (KMS URIs) that are rejected along the signalling path or by the terminating client, the following procedures take place:

The initiating client or function initiates a session with a user or function. It is assumed that the receiver's KMS is not known (not listed in the initiator's user profile or group document and there has not been a previous successful communication), hence the the client performs the procedure in clause 5.2.8.2.5 to select the KMS URIs to use in the MIKEY message.

1. The initiating client sends the communication request to the initiating application server. Under normal conditions the server routes the request on the normal signalling path.

   1a. Should the incorrect security domain(s) (KMS URIs) be used (based on local policy or the policies of the terminating security domain), the server will not forward on the request and may send a KRR message back to the client using the procedures in clause in 5.2.8.2.3.

2. If the communication request is forwarded on the normal signalling route, the other application server should receive the request.

   2a. Should an unacceptable security domain(s) be used (based on local policy or the policies of the terminating security domain), the other application server shall not forward on the request and may send a KRR message to the initiating application server using the procedures in clause in 5.2.8.2.3.

   2b. Upon receiving a KRR message from the other application server, the application server may replace the 'security domain redirect response' message with another KRR message using the procedures in clause 5.2.8.2.4, before forwarding the message down the normal signalling path.

3. Should the request be forwarded on the normal signalling route to the terminating client or function, the terminating MCX entity should receive the request.

   3a. The terminating client may determine that the security domains used by the initiating client are not permitted. In this case, the terminating client may send a KRR message containing permitted security domains back to the initiating client using the procedures in clause 5.2.8.2.2.

   3b. Upon receiving a KRR message and based on local policy, the other application server may replace the KRR message using the procedures in clause 5.2.8.2.4, before forwarding the message down the normal signalling path to the initial application server.

   3c. Upon receiving a KRR message and based on local policy, the initial application server may replace the KRR using the procedures in clause 5.2.8.2.4, before forwarding the message down the normal signalling path to the client.

4. On receiving a KRR, the initiator will perform the procedures in clause 5.2.8.2.5, and may repeat the above procedure from step 1. Upon next connection to the Home KMS, the initiating client should upload the received KRR message to allow fraud detection.

A MC client shall only accept external security domains that have been permitted by the home security domain and provisioned by the Home KMS. The Home KMS may also provision policy around the use of external security domains, see clause 5.2.8.5.

NOTE 1: It is possible that the MC client receives a KRR either unsigned or signed using a KMS URI that is not recognised/provisioned. In this case, it is subject to policy (determined by the Home KMS) whether the redirect is accepted, see clause 5.2.8.5.

NOTE 2: Under the most stringent policy, the KMS a policy may be implemented that requires the client to hold the communication until the Home KMS has responded with notification that the redirect is acceptable, see clause 5.2.8.5.

5.2.8.4 Security Procedures for reporting external security domain use

Domain administrators should only allow users to communicate with trusted external security domains. Should an external security domain be misused, it is possible that users could be impersonated within the MCX system (in the same way that misuse of a CA compromises communications that trust that CA). To allow such misuse to be detected, and the associated KMS certificates to be revoked, clients should report the use of external security domains to the Home KMS.

5.2.8.5 Policy around use of external security domains

The following are policies that the Home KMS may apply around the use of external security domains.

- Allow KRRs (yes/no). If no, all KRRs shall be ignored.
- Report KRRs to the Home KMS (yes/no).
- Require signed KRRs (yes/no). If no, all unsigned KRRs shall be ignored.
- Request unknown KMS certificates (yes/no). If yes, should an unknown external KMS certificate be in the list of receiving KMS URI(s) in the KRR, the client shall request this certificate from the Home KMS as defined in Annex D.
- Hold communication until KMS acceptance (yes/no). If yes, the client will not act upon any KRRs until the Home KMS has provided a notification that the redirect is acceptable (or otherwise), as defined in Annex D.

5.3 User key management

5.3.1 Key Management Server (KMS)

5.3.1.1 General

To be able to be involved in end-to-end communication security the MC user requires key material to be provisioned from their Home Key Management Server (KMS). In addition, management entities which setup or control the end-to-end communication, such as the MCX Server and Group Management Server, will also require provisioning of key material.

NOTE: For clarity, an MC KMS provides different functionality to a MIKEY-TICKET KMS defined in 3GPP TS 33.328 [8].

In this clause, the 'user' could be a GMS, MCX Server, Signalling Proxy or any other entity containing a Key Management (KM) client.

From the perspective of a user (KM client), there are three types of KMS:

- The Home KMS.

- Migration KMSs.
- External KMSs.

A user has exactly one Home KMS, zero or more Migration KMSs and zero or more External KMSs. The relationship between the KMSs is shown in Figure 5.3.1.1-1.

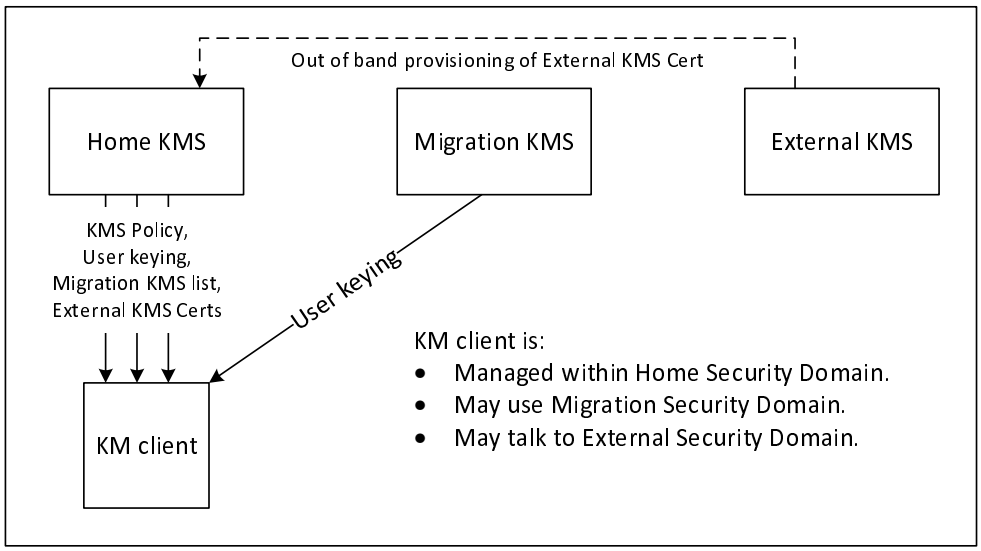

Figure 5.3.1.1-1: Types of KMS

5.3.1.2 Home KMS

The Home KMS is the KMS trusted by the user (KM client) to manage the user's primary security domain. The Home KMS controls the use of media security for users (KM clients) within the primary security domain and is the source of KMS certificates for users or groups home to partner (i.e. external) MC domains.

The Home KMS shall provide the following to KM clients:

- The Home KMS Certificate.
- Policy around the use of provisioned key material.
- User key material for the authenticated user's identity.
- A list of permitted Migration KMSs and their addresses.
- A list of permitted External KMSs, and their KMS certificates.

KM clients may provide the following to their Home KMS:

- Received KMS Redirect Responses (KRRs).

5.3.1.3 Migration KMS

A Migration KMS is the KMS of a migration MC domain that controls media security of users (KM clients) while those users are authorised members of that migration MC domain.

NOTE 1: A Home KMS is able to continue to support a user with primary KMS certificates and key material while the user is migrated.

A Migration KMS may provide the following to KM clients:

- The Migation KMS Certificate.
- User key material for the authenticated user's identity while a member of the migration MC domain.

NOTE 2: A Migration KMS may also support revocation of key material issued by a Home KMS (e.g. due to compromise), while still allowing communication using (uncompromised) key material provisioned by the Migration KMS.

KM clients may provide the following to their Migration KMS:

- KMS Redirect Responses (KRRs).

5.3.1.4 External KMS

External KMSs serve partner security domains with which the user is able to communicate. To communicate with the partner security domain, the user (KM client) requires the External KMS certificate. External KMS certificates are provided by the user's Home KMS. In this way, the Home KMS maintains control of external communications.

The user (KM client) never connects to an External KMS.

NOTE 1: Without provisioning the KMS certificate of an External KMS, secure communication with users and groups home to the corresponding partner security domain is not possible.

5.3.2 Functional model for key management

Within the mission critical architecture, the Key Management Server (KMS) provisions key material associated with a specific MC identity (e.g. MCPTT ID). The KMS has interfaces with the key management clients. A key management client is responsible for making requests for identity-specific key material. Key provisioning clients are located in the MC UE, in the MCX Server(s) and in the Group Management Server(s).

The reference points for the KMS are shown in figure 5.3.2-1.

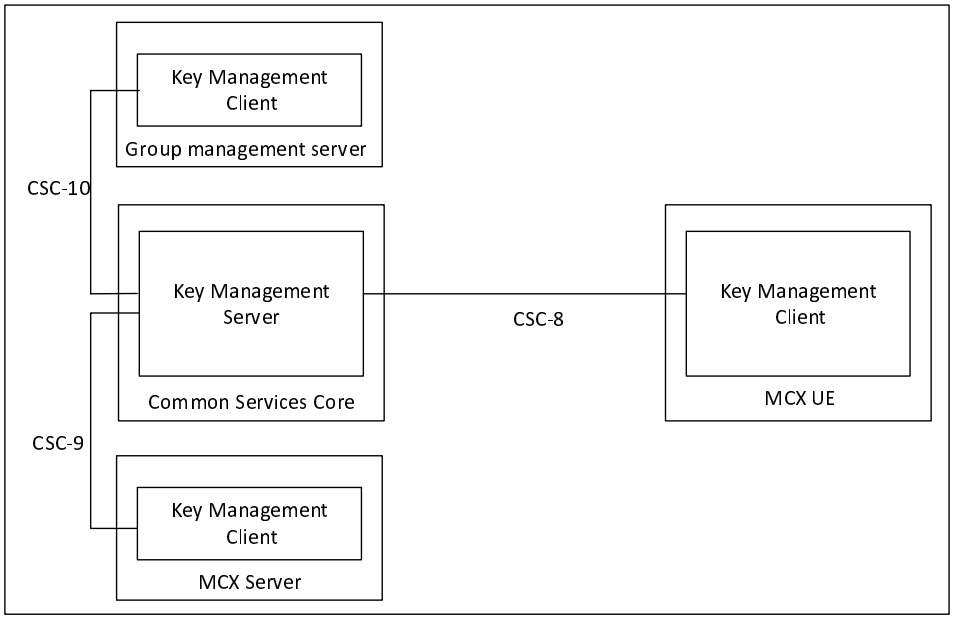

Figure 5.3.2-1: Reference Points for Key Management Server

Figure 5.3.2.1-1 shows the CSC-8, CSC-9 and CSC-10 reference points for the Key Management Server within a MC domain.

The KMS may or may not be located within the Common Services Core (CSC) of the MC domain and may or may not make use of the HTTP proxy.

If the KMS does not make use of the HTTP proxy, then a secure HTTP connection (HTTPS) shall be established directly between the KMS and the KM client. In this case, each of CSC-8, CSC-9 and CSC-10 is a direct HTTP connection between the KMS and KM client in the MC UE, MCX Server or GMS (resp). The use of the TrK as defined

in clause 9.3.3 may be used to protect the key material content in this configuration, and the InK may be used to integrity protect the key material content.

If the KMS does connect to and employ the use of the HTTP proxy, then for public safety users the TrK shall be used as defined in clause 9.3.3 to protect the key material content and the InK should be used for integrity protection. In this case, each of CSC-8, CSC-9 and CSC-10 uses HTTP-1 and HTTP-2 between the KMS and KM client in the MC UE, MCX Server or GMS (resp).

When a TrK is used to protect the transfer of key material between the KM client and KMS, the MC UE TrK identifier (TrK-ID) shall be provided by the KM client to the KMS during user KM authorization. If the InK-ID is not provided in the same user authorization request, then the TrK shall be used for integrity protection. If the InK-ID is provided in the same user authorization request, then the InK shall be used for integrity protection.

5.3.3 Security procedures for key management

The procedure for the provision of identity-specific key material when the HTTP proxy is supported between the KMS and the KM client is described in figure 5.3.3-1. The procedure is the same whether the key management client in the MC UE, an MCX Server or a Group Management Server is making the request.

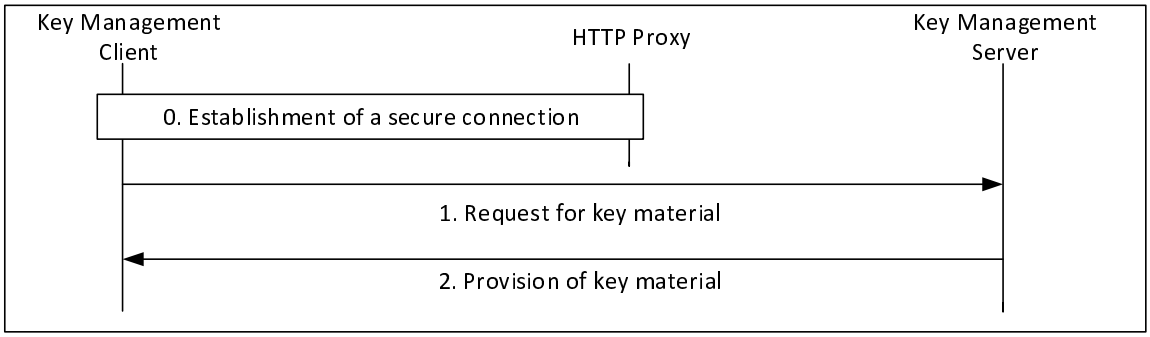

Figure 5.3.3-1: Provisioning of key material via the HTTP proxy

The procedure in figure 5.3.3-1 is now described step-by-step.

0) The key management client establishes a connection to the KMS. As with other elements in the Common Services Core, the connection is routed via, and secured by, the HTTP Proxy. The message flow below is within this secure connection.

NOTE: Additionally, the connection between the KMS and the HTTP Proxy is secured according to clause 6.1.

1) The key management client makes a request for user key material from the KMS. The request contains an access token to authenticate the user as defined in clause 5.1. The request shall also contain the TrK-ID and may contain the InK-ID. There are the followingtypes of request (as defined in Annex D):

   a) KMSInit Request. This request is the first request sent to the KMS to setup the user. This type of request is permitted to the Home KMS or to Migration KMSs.

   b) KMSKeyProv Request: This request is to obtain new key material from the KMS. The request may contain details of a specific identity (e.g. MCPTT ID) required for key management, and may contain a specific time for which the key material is required. This type of request is permitted to the Home KMS or to Migration KMSs.

   c) KMSCertCache Request: This request is to obtain external KMS certificates associated with external security domains (managed by another KMS). The request may contain details of the latest version of the cache received by the client. This type of request shall only be made to the Home KMS.

   d) KMS Redirect Response (KRR) upload: This procedure uploads a KRR received by the client to the Home KMS. This type of message shall only be sent to the Home KMS.

   e) KMS Cert Request: This request is to obtain a single KMS certificate based on a provided KMS URI.

   f) KMS Lookup: This message is to lookup the external KMS that should be used for a provided SIP URI.

g) KMS Redirect Upload: This message is to upload a received discovery request response to the KMS for audit purposes.

2) The KMS provides a response based upon the authenticated user and the user's request. For public safety use, the key material itself shall be encrypted using a 256-bit transport key (TrK). The response may also be signed by the TrK or the InK. The TrK and InK are initially distributed via an out-of-band mechanism along with their 32-bit identifiers, the TrK-ID and InK-ID, respectively. The responses are:

a) KMSInit Response. This response contains domain parameters and optionally, a new TrK and/or a new InK.

b) KMSKeyProv Response: This response provides new key material to the user and optionally, a new TrK and/or a new InK.

c) KMSCertCache Response: This response contains new or updated home KMS certificates and/or external KMS certificates required by the user for communications with external security domains.

d) KMS Cert Response: This response is a KMSCertCache Response containing a single KMS Certificate (or an error message).

e) KMS Lookup Response: This response is a to provide the client with information related to a Discovery Lookup request. Either the KMS URI that should be used for a user is provided, or permission is provided to use a specific External KMS.

The procedure for the provisioning of identity-specific key material when the HTTP proxy is not used between the KMS and the KM client is as described in Figure 5.3.3-2.

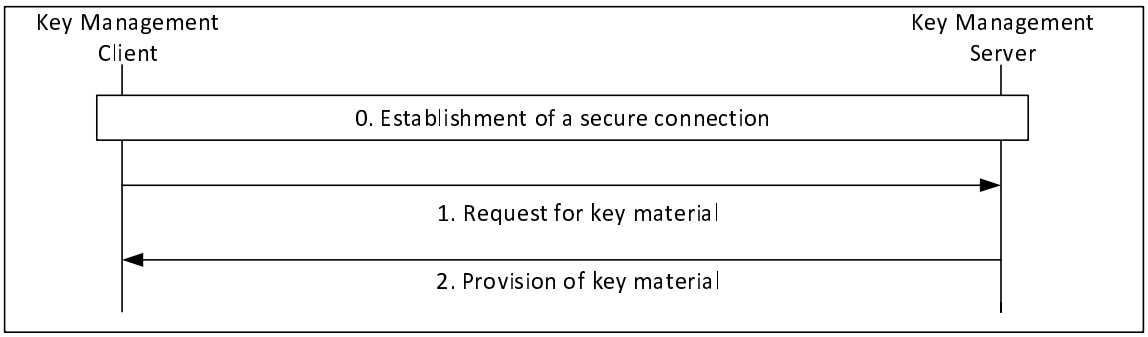

Figure 5.3.3-2: Provisioning of key material without a proxy

The procedure in Figure 5.3.3-2 is now described step-by-step:

0) The key management client establishes a direct HTTPS connection to the KMS. The following message flow is within this secure connection.

1) The key management client makes a request to the KMS. The request may contain the TrK-ID and may contain the InK-ID. The same requests can be made as defined above with a proxy.

2) The KMS provides a response based upon the authenticated user and the user's request. Optionally, the key material itself may also be encrypted using a 256-bit transport key (TrK). The response may also be signed using the TrK or the InK. The TrK and InK are initially distributed via an out-of-band mechanism along with their 32-bit identifiers (TrK-ID and InK-ID respectively).

As a result of this procedure, the key management client has securely obtained key material for use within the MC system.

5.3.4 Provisioned key material to support end-to-end communication security

End-to-end communication security for either group or private calls requires the provisioning of key material from the KMS. The key material provisioned to each user is listed below:

- A KMSInit Response contains the KMS Certificate (domain specific key material associated to the KMS), and may contain:
  - An updated TrK for the user (to replace the off-network-provisioned, bootstrap TrK).
  - Policy around the use of KMS key material (Home KMS only)
  - Address to which ‘KMSCertCache' requests should be sent.
  - Address to which ‘KRRupload' messages should be sent.
- A KMSKeyProv Response contains zero, or more, KMSKeySets and may contain:
  - An updated TrK for the user (to replace existing TrK).
- A KMSCertCache Response may contain:
  - The KMS's Certificate(s) (current, updated or future).
  - Migration KMSs (KMS URIs, access addresses, provisional TrKs).
  - External KMS Certificates. This is domain specific key material associated with other KMSs. It is required to enable secure communications across security domains.
- A KMSCert Response may contain:
  - An External KMS Certificate. This is domain specific key material associated with the requested KMS URI. It is required to enable secure communications across security domains.
- A KMS Lookup Response does not contain key material, but may contain KMS URIs.

5.3.5 KMS Certificate

A KMS Certificate is defined in Annex D.3.2. A KMS Certificate contains the following:

- A Role of 'Home' or 'External', depending on whether the certificate is the issuing KMS's or is provided by another external KMS.
- The KMS Public Authentication Key (KPAK in IETF RFC 6507 [9]).
- The KMS Public Confidentiality Key (Z_T in IETF RFC 6508 [10]).
- The UID conversion (as described below).
- Choice of cryptographic domain parameters (such as those listed in IETF RFC 6509 [8]).
- The time period for which this information is valid.

Certificates are identified by the KMS (KMSUri) and a unique identifier (CertUri). A (logical) KMS should only have a single KMS certificate active at any one time (based upon the KMSUri). Certificates may be updated using the CertURI. Should a client receive a certificate with a CertURI of an existing certificate, the client shall replace this existing certificate with the newly provisioned certificate.

The UID conversion mechanism defines how UIDs are generated. Using this information a MC client can take a user identifier (e.g. an MCPTT ID), and the current time, (e.g. the year and month) and convert these to a UID.

EXAMPLE: UID = Hash (MCPTT ID, KMS URI, validity period info).

As a consequence, there is a one-to-one correspondence between MC Service IDs and UIDs during each time period.

5.3.6 KMS provisioned Key Set

KMSKeySet(s) are defined in Annex D.3.3.2 and contain the following:

- A user signing key for each UID for the current time period (SSK and PVT in IETF RFC 6507 [9]).
- A user decryption key for each UID for the current time period (RSK in IETF RFC 6508 [10]).
- The key period number associated with the current keys.
- Optionally, the time period, for which the user key material is valid (e.g. month).

5.4 Key management from MC client to MC server (CSK upload)

The key (CSK) is distributed from the MCX client to the MCX Server(s) using the ‘CSK upload' procedure. The procedure shall use the common key distribution mechanism described in clause 5.2.2, transported over the SIP bearer. Identity hiding may be supported as defined in clause 5.2.6. The MCX Server may respond with a KMS Redirect Response (KRR) as described in clause 5.2.8 (e.g. if the MC client has migrated or is roaming).

The initiating entity of the CSK upload procedure shall be the MCX UE and the receiving entity shall be the MCX Server. With respect to the common key distribution procedure, the initiating entity URI shall be the MCX Service user ID of the user andthe receiving entity URI shall be the MCX Server Domain Security Identifier (MDSI). The MDSI is added to the recipient field (IDRr) of the message. The distributed key, K, shall be the CSK and the distributed identifier K-ID shall be the CSK-ID.

Clause E.4 provides MIKEY message structure for CSK distribution.

Before the CSK upload procedure can be used by the client to securely share the encryption key, the MC user shall first be authorized by KMS for key management services. Once the MC user is authorized, the KMS distributes the user's key material to the client as specified in clause 5.3.3.

The server receives the SIP message with the protected CSK and retrieves it from the message. It associates the MC User's SIP Core identity (IMPU), MC Service user ID (e.g. MCPTT ID) and the received CSK. Identity binding is used to uniquely identify the CSK used in protection of the SIP payload in subsequent SIP messages sent by both the client and the servers within a MC domain.

5.5 Key management between MCX servers (SPK)

Floor control, transmission control, and media control between MCX servers may need to be protected. Additionally, certain values and identifiers transferred in the signalling plane between servers within an MC domain, or between MC domains, may be treated as sensitive by public safety users and therefore may also require protection.

To protect information from all other entities outside of the MC domain(s), a shared 128-bit Signalling Protection Key (SPK) needs to be established between the servers. The SPK is provided along with a 32-bit identifier, the SPK-ID and 128-bit random value SPK-RAND. The most significant four bits of the identifier (the Purpose Tag) of the SPK-ID shall be '3' to denote the purpose of the SPK is for signalling protection, as described in Annex G.

The SPK and associated values shall be directly provisioned into the communicating servers, along with the SPK-ID. With the SPK provisioned, RTCP and XML content (within SIP) may be protected.

5.6 Key management for one-to-one (private) communications (PCK)

The purpose of this procedure is to allow two MCP UEs to create an end-to-end security context to protect an MCX private communication. To create the security context, the initiating MCX UE generates a Private Communication Key (PCK) and securely transfers this key, along with a key identifier (PCK-ID), to the terminating MCX UE. Prior to key distribution, both MCX UE shall be provisioned by the Key Management Server (KMS) with time-limited key material associated with the MCX user as described in clause 5.3.

The PCK is distributed between the MCX clients using the security mechanism described in clause 5.2.2, transported over the SIP bearer within the SDP content of a SIP INVITE (or within the SDP content of a SIP MESSAGE message when used for MCData SDS). The SAKKE-to-self extension may be included as defined in clause 5.2.5. Identity hiding may be supported as defined in clause 5.2.6. The receiving MCX client and any MCX Server through which the SIP INVITE is routed, may respond with a KMS Redirect Response (KRR) as described in clause 5.2.8.

The initiating entity shall be the initiating MCX user. The initiating entity URI shall be the MCX service ID of the initiating user. The receiving entity shall be the terminating MCX user. The receiving entity URI shall be the MCX service ID of the terminating user. The distributed key, K, shall be the PCK and the distributed identifier K-ID shall be the PCK-ID.

Clause E.2 provides MIKEY message structure for PCK distribution.

5.7 Key management for group communications (GMK)

5.7.1 General

To create the group's security association, a Group Master Key (GMK) and associated identifier (GMK-ID) is distributed to MCX UEs by a Group Management Server (GMS). The GMK is distributed encrypted specifically to a user and signed using an identity representing the Group Management Server. Prior to group key distribution, each MCX UE within the group shall be provisioned by the MCX Key Management Server (KMS) with time-limited key material associated with the MCX user as described in clause 5.3. The Group Management Server shall also be provisioned by the MCX KMS with key material for the GMS's identity (the GMS Server URI).

The GMK is distributed from the GMS to a MCX client using the security mechanism described in clause 5.2.2, transported over the SIP bearer. For GMKs, end-point diversity is required and hence the extension in clause 5.2.3 is applied. Additional parameters may be included as defined in clause 5.2.4. The SAKKE-to-self extension may be included as defined in clause 5.2.5. Identity hiding may be supported as defined in clause 5.2.6. The receiving MCX client and any MCX Server through which the SIP INVITE is routed, may respond with a KMS Redirect Response (KRR) as described in clause 5.2.8.

GMKs may be managed individually per Group ID or the same GMK may be assigned to multiple MC Group IDs (using the MIKEY general extension payload defined in Clause E.6). This means that each specified MC Group ID in the MIKEY general extension payload shall use this GMK for group communications. Assigned MC Group IDs may include any combination of MCPTT Group IDs, MCData Group IDs or MCVideo Group IDs. Assigning the same GMK to multiple Group IDs does not prevent individual key management at a later time or vice versa.

An MC client may have multiple active GMKs associated with a Group ID. When this occurs, the MC client shall use the active GMK with the most recent Activation Time (as defined in Clause E.6.4) when encrypting group media.

The initiating entity shall be the initiating GMS. The initiating entity URI shall be the URI of the GMS (e.g. sip: gp.manager@mcptt.example.org). The receiving entity shall be the terminating MCX user. The receiving entity URI shall be the MCX service ID of the terminating user. The distributed key, K, shall be the GMK, the key identifier K-ID shall be the GMK-ID and the end-point-specific key identifier, UK-ID shall be the GUK-ID.

Clause E.3 provides MIKEY message structure for GMK distribution.

5.7.2 Security procedures for GMK provisioning

This procedure uses a MIKEY payload to distribute a GMK from the GMS to the MC UEs within the group. The payload is transported as part of the 'Notify group configuration request' message defined in clause 10.1.2.7 of 3GPP TS 23.280 [36].

Figure 5.7.2-1 shows the security procedures for creating a security association for a group.

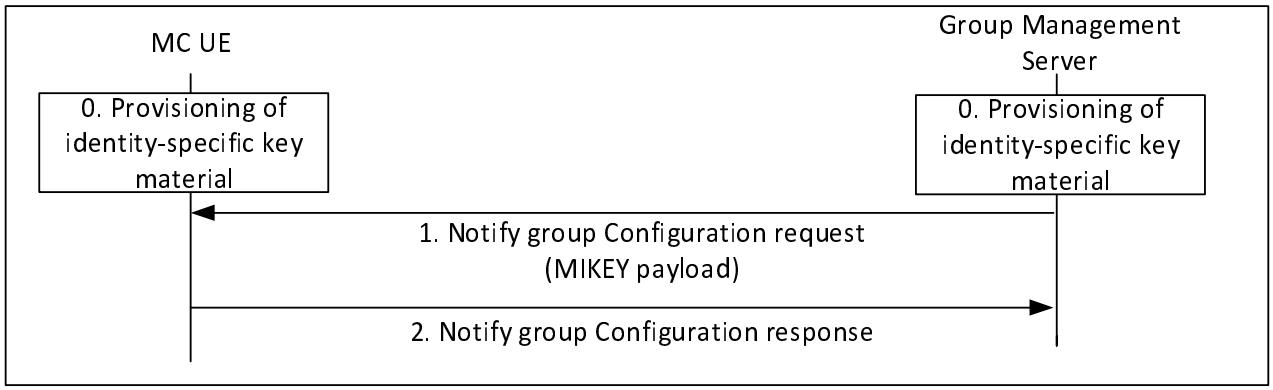

Figure 5.7.2-1: Security configuration for groups

A description of the procedures depicted in figure 5.7.2-1 follows. For clarity, figure 10.1.5.3-2 in clause 10.1.5.3 of 3GPP TS 23.280 [36] is referenced.

0) Prior to beginning this procedure the MC client shall be provisioned with identity-specific key material by a MC KMS as described in clause 5.3. The GMS shall also be securely provisioned with identity-specific key material for the GMS's Server URI.

1) The GMS shall send a MIKEY payload to MC clients within the group within a 'Notify group configuration request' message. The message shall encapsulate a GMK for the group. The payload shall be encrypted to the user identity (MCX service user ID) associated to the MC client and shall be signed by the GMS. The message shall also provide the GUK-ID. Parameters associated with the GMK shall be encrypted using the GMK, and sent in the MIKEY payload together with the encapsulated GMK. This process is shown in Figure 5.2.4-1.

   a) If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the GMS providing KMS information. In this case, the GMS may re-attempt the above procedures.

2) On receipt of a MIKEY message, the MC client shall check the signature on the payload, extract the GMK, GUK-ID and GMK-ID and check that the GMK-ID is not a duplicate for an existing GMK. The MC client shall also extract the group identity, activation time and text from the encapsulated associated parameters in the payload using the GMK, and check that decryption is successful. The MC client shall lookup the Group ID in its user profile data and verify that the GMS identity corresponds with the Server URI for that Group ID. This process is shown in Figure 5.2.4-2. Should any of these checks fail, an error shall be returned to the GMS. Upon successful receipt and processing, the MC UE shall store the GMK, GMK-ID, GUK-ID and associated parameters, and respond to the GMS with a 'Notify group configuration response' message.

It is recommended that a mechanism is used to ensure that no two Group IDs from different servers collide.

To revoke a security context, the group management server repeats the above steps with the Status field of the GMK parameters indicating that the GMK has been revoked.

It is possible that an MC user in the group may be represented by an MC Security Gateway (as defined in Annex L), rather than using full end-to-end security. In this case, the user's KMS Certificate will have the ‘IsSecurityGateway' attribute set to ‘true' (see clause D.3.2.2). Should any client in the group be represented by an MC Security Gateway, the GMS shall indicate to all users that the GMK is shared with an MC Security Gateway. This is achieved by setting the ‘Security Gateway' bit in the ‘Status' field of the GMK's key parameters (see clause E.6.9).

Should an MC client receive a GMK with the ‘Security Gateway' bit set, the initiating MC client shall warn the MC user that an MC Security Gateway is in use during the group's comunications.

5.7.3 Group member GMK management

In some situations, the membership of a group may be modified whereby an MC user may be added to or removed from an MCX group. Users are alerted to these changes by a user profile update from the CMS for which they are subscribed. The updated user configuration profile indicates the group ID to which the group membership change is associated.

When users are added to a new or existing group they may be implicitly affiliated to that group in which case the user is automatically subscribed to group configuration updates from the GMS. The user shall be authorised for group management services to the GMS before the GMS provides the associated group management records and the GMK. Once the user is authorised, the GMS sends the group management record as well as the GMK to the UE. The user may join in on the group communication immediately after receiving the group update and GMK.

When the user configuration record indicates the user has been added to a new or existing group but is required to explicitly affiliate to the group, the user shall be authorised for group management services to the GMS followed by a subscription to group updates from the GMS. The user shall be authorised for group management services and the subscription shall be validated before the GMS provides group management records and the GMK. Once the user is authorised and the subscription processed by the GMS, the GMS sends the group management record and the GMK to the UE. The user may then join in on the group communication immediately after receiving the group update and GMK.

When a user is removed from a group, the UE receives a user profile update from the CMS indicating the user is no longer a member of the specified group ID(s). Upon receiving the user profile update, ending of any group communication(s) associated with that group, and if the GMK associated with the group ID is not associated with another group that the user remains a member, the UE shall immediately and securely delete the GMK associated with that group ID. If the group ID is associated to more than one service (i.e. MCPTT, MCData and/or MCVideo) then upon the ending of any group communication(s) associated with that group ID, and if the GMKs associated with that group ID is not associated with another group that the user remains a member, the GMKs associated with that group ID shall be immediately and securely deleted.

When a user is removed from the group, the Group Management Server may choose to update the GMK associated with the group ID (depending on the security profile of the group).

5.8 Key management from MC server to MC client (Key download)

5.8.1 General

The 'key download' procedure is used to send keys from the MCX server to the MC client. It is used to distribute Multicast Signalling Keys (MuSiKs) to the MC clients, and it is used to update both the CSKs and MuSiKs.

Within the 'key download' procedure, keys (CSK or MuSiKs) are encrypted specifically to the MC user and signed using an identity representing the MC Server. Prior to group key distribution, each MC client shall be provisioned by the KMS with time-limited key material associated with the MC User as described clause 5.3. The MC Server shall also be provisioned by the MC KMS with key material for an identity which is authorised to act as an MCX Server.

The key (CSK or MuSiK) is distributed from the MCX Server to a MC client using the security mechanism described in clause 5.2.2, transported over the SIP bearer. End-point diversity is not required as end-points do not encrypt data, hence the extension in clause 5.2.3 is not applied. Additional parameters may be included as defined in clause 5.2.4. The SAKKE-to-self extension may be included as defined in clause 5.2.5. Identity hiding may be supported as defined in clause 5.2.6.

The initiating entity shall be the initiating MCX Server and the receiving entity shall be the terminating MC user. The initiating entity URI shall be the FQDN of the MCX Server (e.g. MDSI of the MC Domain) and the receiving entity URI shall be the MC Service ID of the terminating user. The distributed key, K, shall be the CSK or MuSiK and the key identifier K-ID shall be the CSK-ID or MuSiK-ID (respectively).

As a result of this 'key download' mechanism, the MC clients receive a new signalling key, CSK or MuSiK, identified by the 4 most significant bits of the key ID.

The MCData Service server may use the Key Download procedure to indicate or modify the algorithm used to protect the MCData signalling fields (i.e. MCData signaling parameters, Data signaling payload and End to end security parameters) by including a ‘signalling algorithm’ parameter. The ‘signalling algorithm’ parameter is described in clause 8.5.4.1. The available algorithms shall be as defined in clause 8.5.4.2.

5.8.2 ‘Key download' procedure

The procedure for key download is described in figure 5.8.2-1:

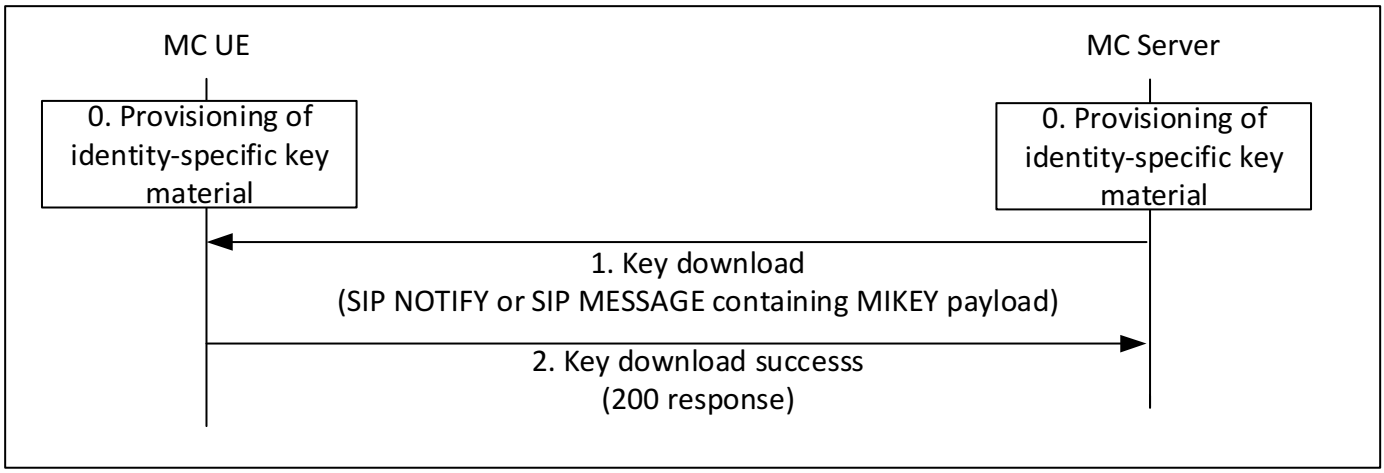

Figure 5.8.2-1: Procedures for key download

0. The MCX UE has been provisioned by a KMS with key material associated with the MC user. The MC UE has also registered with an MCX Server. As a consequence of this registration, the MC UE is subscribed to key download notifications from the MCX Server.

1. The MCX Server sends a key download message (SIP NOTIFY or SIP MESSAGE) to the MC UE. The MC UE extracts the signalling key from the key download message.

2. Upon successful extraction of the signalling key, the MC UE returns a key download success message (200 OK response) to the MCX Server. Upon receipt of a notification of success, the MCX Server is able to begin to use the key for protection of signalling traffic.

5.9 Key management during MBMS bearer announcement

The MBMS bearer announcement message is used to distribute a MSCCK as described in Annex H.

The security procedures for key distribution via an MBMS bearer announcement message are identical to those used for 'key download' messages, described in clause 5.8.

5.10 Void

5.10.1 Void

5.10.2 Void

5.10.3 Void

5.10.3.1 Void

5.10.3.2 Void

5.10.3.3 Void

5.10.3.4 Void

5.10.3.5 Void

5.10.4 Void

5.10.4.1 Void

5.10.4.2 Void

5.11 UE key storage and key persistence

5.11.1 Key storage

To prevent the exposure of mission critical keys and key material, the following guidelines shall be followed to ensure that mission critical keys and key material are protected within the UE. Persistence of keys and key material through transistional states of the UE are defined in clause 5.11.2.

All long term keys and private certificates used to establish secure communications with MC domain servers such as the IdMS, KMS, and MC domain proxies (e.g. CS proxies and MC domain proxies) shall be stored in a protected state within the UE while not actively in use. The method used to protect these keys and certificates is out of scope of this document. These long term keys and key material include, but are not limited to the TrK, InK, and TLS private certificates.

CSK(s), GMK(s) and MuSiK(s) shall be stored in a protected state within the UE while not actively in use. The method used to protect these keys is out of scope of this document.

Identity based key material, e.g. KMS Key Set(s), shall be stored in a protected state within the UE while not actively in use. The method used to protect these keys is out of scope of this document.

Identity based tokens, such as the ID token, access token(s), refresh token(s), and security token(s) shall be stored in a protected state within the UE while not actively in use. The method used to protect these tokens is out of scope of this document.

Dynamic keys used for MCPTT, MCData and MCVideo signalling and media protection such as the MKFC, MSCCK and PCK and any derived keys (e.g. DPCK, SRTP Master Keys, XML keys, AES keys) should be securely stored as dictated by the operational needs of these keys and shall be securely deleted when these keys are no longer operationally needed.

5.11.2 Key persistence

Static and semi-static keys and key material such as CSK(s), GMK(s), TrK, InK, TLS private certificates, IPsec private certificates, KMS Key Sets, ID token, access token(s), refresh token(s), and security token(s) shall be securely stored and maintained through UE power cycles. These static and semi-static keys and key material shall also be maintained through user log-off and log-on cycles. This ensures that the UE maintains the credentials, keys and key material for the user through various UE transitional states including on-network to off-network transitions. This is especially critical for identity based key material and GMK(s) that are used for off-network communications.

When the current user logs off and a different user logs onto the UE, all keys and key material associated to the previous user shall either be securely deleted from the UE or alternatively, a method to securely partition user's keys and key material from other users shall be implemented. Keys and key material that remain in the UE through log-off and log-on cycles and during usage by mulitiple users shall be securely stored and accessible only to the user to which the keys and key material are associated. The method used to securely partition user's keys and key material is out of scope of this document.

Dynamic keys for MCPTT, MCData and MCVideo media and signalling protection such as the MKFC, MSCCK, and PCK and any derived keys (e.g. DPCK, SRTP Master Keys, XML keys, AES keys) shall be securely deleted from the UE at UE power down, log off of the current user, expiry of any associated access tokens (for technical or authentication reasons), or as dictated by the operational needs of these keys. Dynamic keys and tokens may be renegotiated during establishment of follow-on communications.

When an access token expires because the IdMS cannot or will not refresh the existing access token for technical or authentication reasons, the following mission critical static, semi-static and dynamic keys shall be securely deleted from the UE; CSK(s), GMK(s), MuSiK, MKFC, MSCCK, KFC, DPPK, PCK, and identity based tokens (i.e. access tokens, refresh tokens, and security tokens). Expired access tokens, refresh tokens or security tokens may require re-authentication of the user with the IdM services and MC domain.

6 Supporting security mechanisms

6.1 HTTP

6.1.1 Authentication for HTTP-1 interface

For authentication of the HTTP-1 reference point, one of the following authentication mechanisms shall be performed between the HTTP client in the MC UE and the HTTP server endpoint (HTTP proxy, IdM server or KMS):

- one-way authentication of the HTTP server endpoint based on the server certificate;
- mutual authentication based on client and server certificates;
- mutual authentication based on pre-shared key.

Certificate based authentication shall follow the profiles given in 3GPP TS 33.310 [5], clauses 6.1.3a and 6.1.4a. The structure of the PKI used for the certificate is out of scope of the present document. Guidance on certificate based mutual authentication is provided in 3GPP TS 33.222 [16], annex B.

The usage of Pre-Shared Key Ciphersuites for Transport Layer Security (TLS-PSK) is specified in the TLS profile given in 3GPP TS 33.310 [5], annex E.

6.1.2 HTTP-1 interface security

The support of Transport Layer Security (TLS) on HTTP-1 is mandatory. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [5], annex E.

If the PSK TLS based authentication mechanism is supported, the HTTP client in the MC UE and the HTTP Proxy shall support the TLS version, PSK ciphersuites and TLS Extensions as specified in the TLS profile given in 3GPP TS 33.310 [5], annex E. The usage of pre-shared key ciphersuites for TLS is specified in the TLS profile given in 3GPP TS 33.310 [5], annex E.

6.1.3 HTTP-3 interface security

The support of Transport Layer Security (TLS) on HTTP-3 is recommended between HTTP proxies. Where used, the profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [5], annex E.

6.2 SIP

6.2.1 Authentication for SIP core access

This clause specifies the mutual authentication between the UE and the SIP core.

IMS AKA authentication shall be performed as specified in 3GPP TS 33.203 [6] for SIP core access. IMS AKA authentication mechanism as specified in 3GPP TS 33.203 [6] shall be performed irrespective of whether SIP core architecture is compliant with 3GPP TS 23.228 [15] or not.

Authentication related information shall be provided by SIP database that may be part of the HSS or may be part of the MC service provider's SIP database depending on the SIP core deployment scenarios specified in 3GPP TS 23.379 [2].

Implementation options and requirements on the ISIM or USIM application to support SIP core access security are specified in 3GPP TS 33.203 [6].

6.2.2 SIP-1 interface security

The security mechanisms as specified in 3GPP TS 33.203 [6] for Gm interface shall be used to provide confidentiality and integrity of signalling on SIP-1 interface.

6.3 Network domain security

6.3.1 LTE access authentication and security

An MC UE shall perform the authentication and security mechanisms as specified in 3GPP TS 33.401 [14] for LTE network access security.

6.3.2 Inter/Intra domain interface security

To ensure security of the interfaces between network elements within a trusted domain and between trusted domains, namely HTTP-2, HTTP-3, SIP-2 and SIP-3:

- 3GPP TS 33.210 [4] shall be applied to secure signalling messages on the reference points unless specified otherwise; and
- 3GPP TS 33.310 [5] may be applied regarding the use of certificates with the security mechanisms of 3GPP TS 33.210 [4] unless specified otherwise in the present document.

NOTE: For the case of an interface between two network elements in the same trusted domain, 3GPP TS 33.210 [4] does not mandate the protection of the interface by means of IPsec. However, it is up to the domain administrator's policy to also protect interfaces within the same trusted domain.

SEG as specified in 3GPP TS 33.210 [4] may be used in the trusted domain to terminate the IPsec tunnel.

7 MCPTT and MCVideo

7.1 General

This clause described the security procedures for both MCPTT and MCVideo.

7.2 Private communications

7.2.1 Key management

7.2.2 Security procedures (on-network)

The following private communication security procedures provide a mechanism for establishing a security context as part of the Private Call Request sent from the initiating UE to the terminating UE.

3GPP TS 23.379 [2] describes manual and automatic commencement for private MCPTT communications in both a single MC system and across multiple MC systems, while 3GPP TS 23.281 [37] describes manual and automatic commencement for private MCVideo communications within a single MC system.

Securing of on-network private MCPTT or MCVideo communications is summarized in the following sub clauses and applies to the aforementioned MCPTT and MCVideo private call use cases.

The private call setup message used to establish these security procedures may be pre-generated to increase the efficiency of the communication. Additionally, the MC UE may attach a second SAKKE component which encrypts the PCK to the initiating user (in addition to the terminating user) for use in the ‘SAKKE-to-self' procedure.

The security procedure for an on-network MCPTT or MCVideo private call within a single MC system is summarized in figure 7.2.2-1, The security procedure for securing an on-network MCPTT private call between multiple MC systems is summarized in figure 7.2.2-2. The intent of these on-network security procedures is to transfer a PCK and PCK-ID to the terminating UE in order to provide end-to-end security of the media.

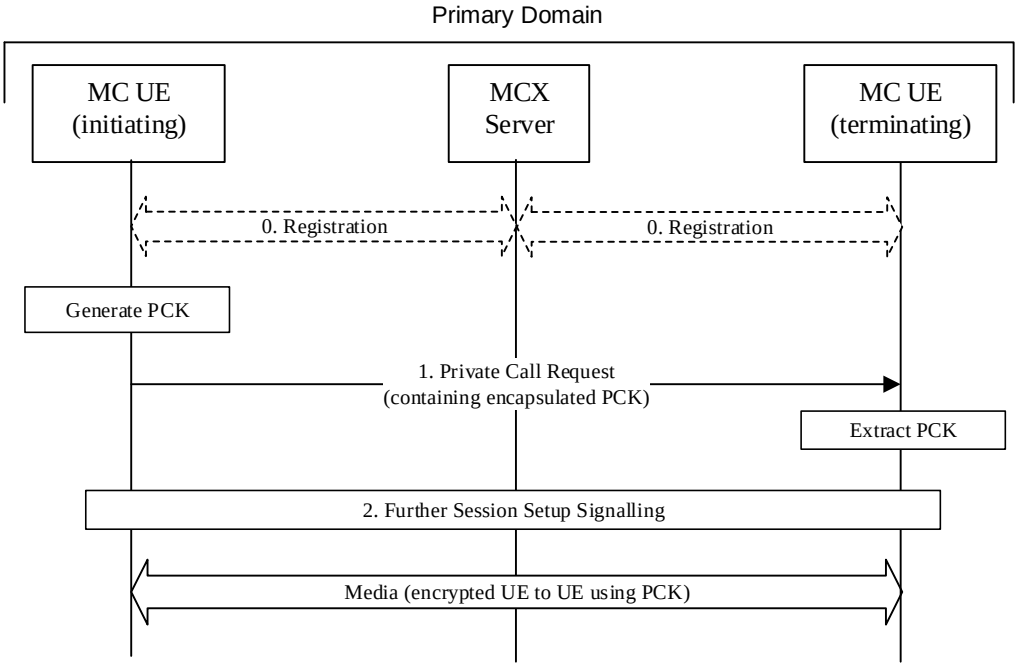

Figure 7.2.2-1: Private call security procedure for on-network PCK distribution for single domain

The procedure in figure 7.2.2-1 is now described step-by-step.

0. Prior to beginning this procedure it is assumed that the MC UEs have an authenticated MC user and that the MC UEs have been provisioned with key material associated with a user's MC service ID by a KMS as described in clause 5.3.

1. The initiating MC UE generates the PCK and sends a private call request to the terminating MC UE. The message is sent to the primary MC server of the initiating UE where it is forwarded to the intended receipient UE. Within this message includes an SDP offer which contains a MIKEY-SAKKE I_MESSAGEs as defined in IETF RFC 6509 [11]. The I_MESSAGE encapsulates the PCK for the terminating MC user, encrypting the key to the UID of the terminating user (derived from the user's URI). The I_MESSAGE also contains an identifier for the PCK (PCK-ID). The I_MESSAGE is signed using (the key associated with) the initiating user's UID.

   a) If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the initiating client providing KMS information. In this case, the initiating client may re-attempt the above procedures.

2. Further session signalling occurs as defined in 3GPP TS 23.379 [2] for MCPTT and 3GPP TS 23.281 [39] for MCVideo.

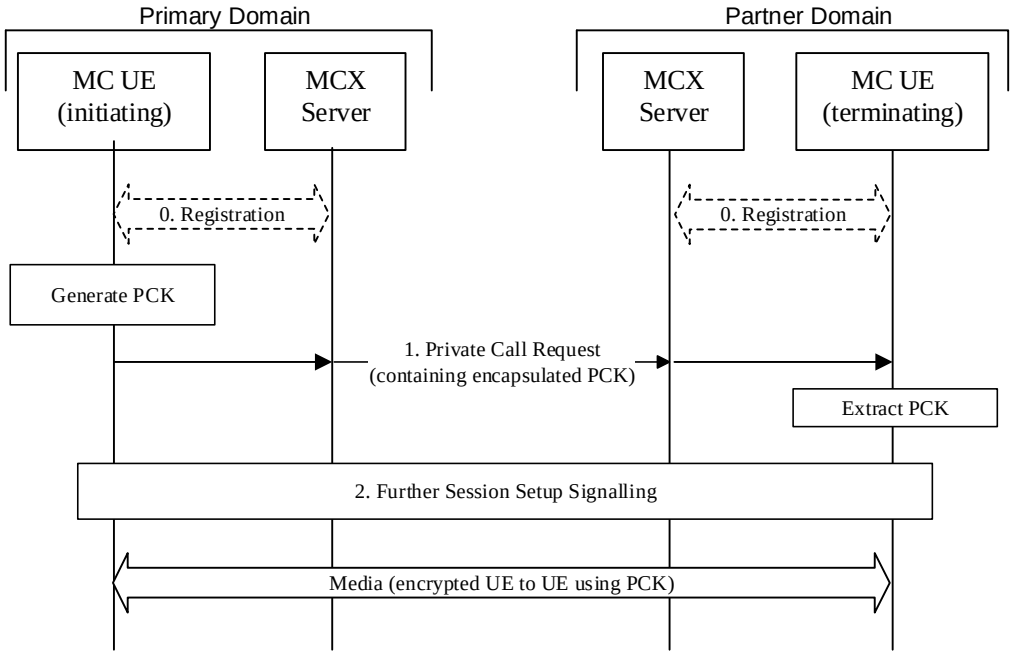

Figure 7.2.2-2: Private call security procedure for on-network PCK distribution between multiple domains

The procedure in figure 7.2.2-2 is now described step-by-step.

0. Prior to beginning this procedure it is assumed that the MC UEs have an authenticated MC user and that the MC UEs have been provisioned with key material associated with a user's MC service ID by a KMS as described in clause 5.3.

1. The initiating MC UE generates the PCK and sends a private call request addressed to the terminating MC UE. The message is first routed to the primary MC server of the initiating UE. The primary MC server routes the private call request to the partner server (home of the intended receipient UE), which is then routed to the receipient UE. The private call request message includes an SDP offer which contains a MIKEY-SAKKE I_MESSAGE as defined in IETF RFC 6509 [11]. The I_MESSAGE encapsulates the PCK for the terminating MC user, encrypting the key to the UID of the terminating user (derived from the user's URI). The I_MESSAGE also contains an identifier for the PCK (PCK-ID). The I_MESSAGE is signed using (the key associated with) the initiating user's UID.

   a) If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the initiating client providing KMS information. In this case, the initiating client may re-attempt the above procedures.

2. Further session signalling occurs as defined in 3GPP TS 23.379 [2].

It is possible that the terminating MC client may be represented by an MC Security Gateway (as defined in Annex L), rather than using full end-to-end security. In this case, the user's KMS Certificate will have the ‘IsSecurityGateway' attribute set to ‘true' (see clause D.3.2.2). Should the terminating client be represented by an MC Security Gateway, the initiating MC client shall warn the MC user that an MC Security Gateway is in use during the private call.

With the PCK and PCK-ID shared between the initiating and terminating users, the media communicated between the UEs may be end-to-end protected using the PCK.

7.2.3 Security procedures (off-network)

3GPP TS 23.379 [2] describes manual and automatic commencement for private off-network MCPTT communications, while 3GPP TS 23.281 [37] describes manual and automatic commencement for private off-network MCVideo communications.

Securing off-network private MCPTT or MCVideo communications is summarized in the following sub clauses and applies to the aforementioned MCPTT and MCVideo off-network private call use cases.

The private call setup message used to establish these security procedures may be pre-generated to increase the efficiency of the communication. Additionally, the MC UE may attach a second SAKKE component which encrypts the PCK to the initiating user (in addition to the terminating user) for use in the 'SAKKE-to-self' procedure.

The security procedure for securing an off-network private call is summarized in figure 7.2.3-1. As part of this process, the PCK and PCK-ID are securely transferred to the terminating UE.

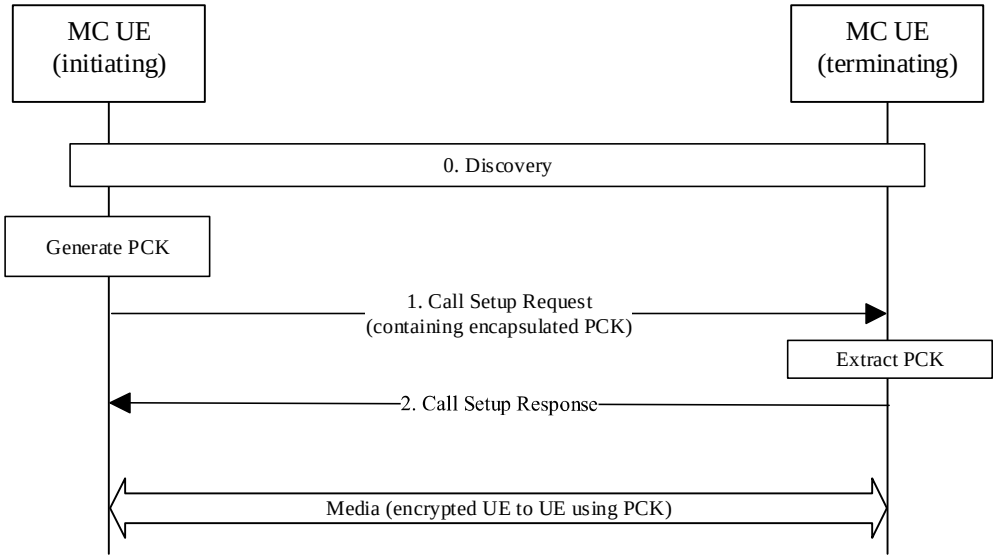

Figure 7.2.3-1: Private Call security procedure for off-network PCK distribution

The procedure in figure 7.2.3-1 is now described step-by-step.

0. Prior to beginning this procedure the MC UEs may have performed a discovery procedure. Additionally, the MC UEs have been provisioned with key material associated with a user's MC Service user IDs by a KMS as described in clause 5.3.

1. The initiating UE generates the PCK and sends a Call Setup Request to the terminating UE. Within this message, the initiating UE includes a MIKEY-SAKKE I_MESSAGEs as defined in IETF RFC 6509 [11]. The I_MESSAGE encapsulates the PCK for the terminating UE, encrypting the key to the UID of the terminating user (derived from the user's URI). The I_MESSAGE also contains an identifier for the PCK (PCK-ID). The I_MESSAGE is signed using (the key associated with) the initiating user's UID.

2. A Call Setup Response is returned to the initiating UE as defined in 3GPP TS 23.379 [2] for MCPTT and as defined in TS 23.281 [37] for MCVideo.

With the PCK and PCK-ID shared between the initiating and terminating users, the media communicated between the UEs may be protected using the PCK.

7.2.4 First-to-answer security and key management

7.2.4.1 Overview

A 'first-to-answer' call as defined in clause 10.15 of TS 23.379 [2], is a call request sent to multiple users inviting them into a private call, and where the first user to answer the request is brought into the private call with the initiator while the rest of the invited users are subsequently rejected. Consequently, a specific key management solution is required.

The following defines a method for performing key distribution for a first-to-answer call. From a security point-of-view, the approach is to perform a private call key distribution from the answering client to the initiating client of the call.

The first-to-answer messages are routed over the signalling reference points. Consequently, the security mechanisms for protecting signalling between the MC Domain and the MC UE are applied to these messages. This includes the security mechanisms defined in clause 6. Where application signalling security is supported, the security mechanisms defined in clause 5.3 are used, ensuring that the user identities (MCPTT IDs) are confidentiality protected with the CSK or SPK as per clause 5.3.

7.2.4.2 First-to-answer request and response

The first-to-answer request (containing the list of target MCPTT IDs) is sent by an initiating UE to the MCPTT server. No key material is provided in the first-to-answer request.

The first-to-answer response is sent by a target UE in response to a first-to-answer request. The first-to-answer response contains both an encapsulatedPCK for the private call and a pair of MCPTT IDs corresponding to the participants (intiator and target) of the private call.

The PCK is encapsulated as defined in clause 5.6. In this case, the 'initiating entity' shall be the MC user who provides the first-to-answer response. The initiating entity URI shall be the MC Service user ID of the user who made the first-to-answer response. The receiving entity shall be the MC user who made the first-to-answer request. The receiving entity URI shall be the MC Service user ID of the user who made the first-to-answer request.

7.2.4.3 First-to-answer call setup with security

Figure 7.2.4.3-1 below illustrates the first-to-answer call setup procedure with security.

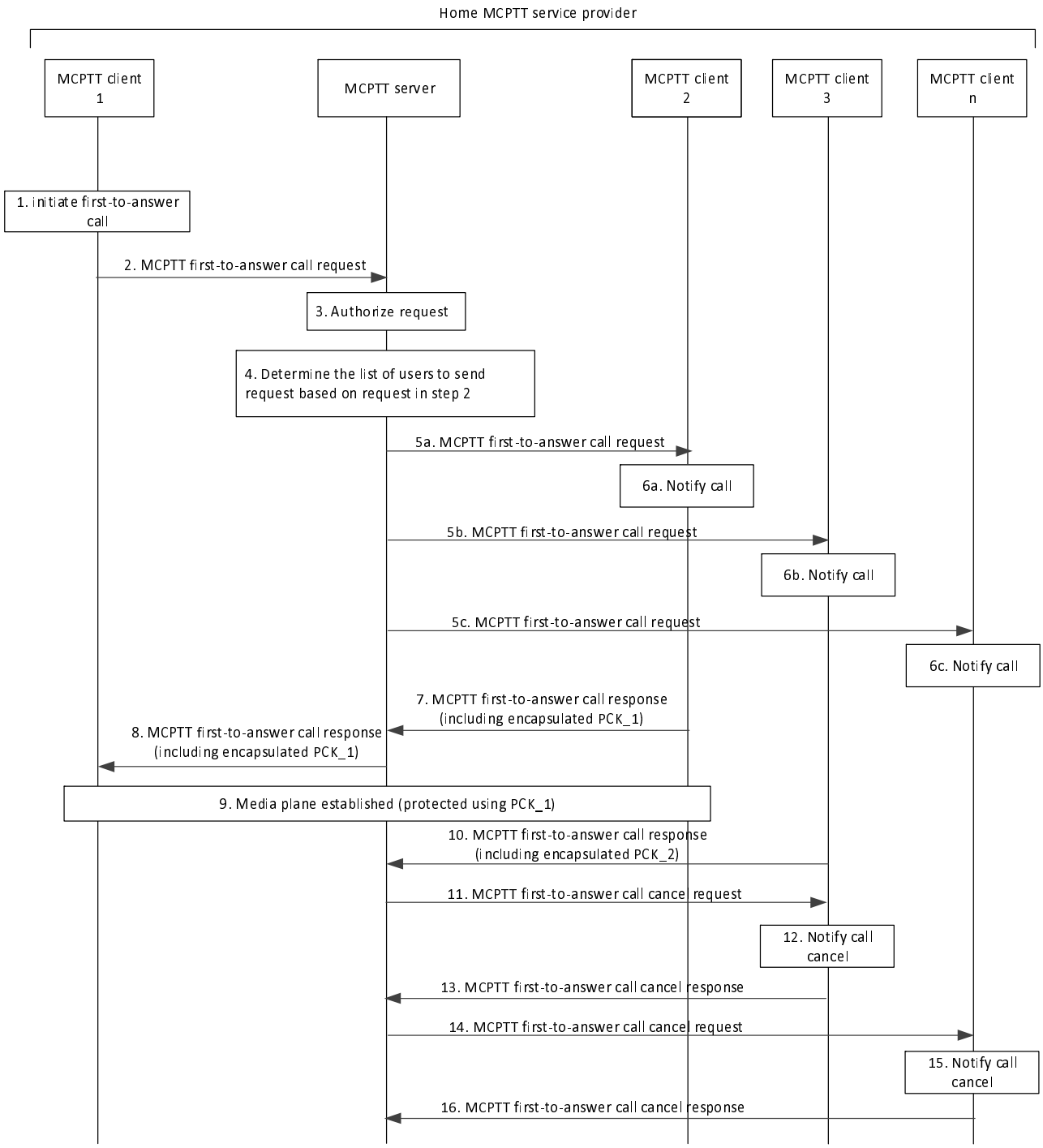

Figure 7.2.4.3-1: First-to-answer call setup and key management

1 to 6. First-to-answer call signalling occurs as defined in TS 23.379 [2]. These messages do not contain security-related key material.

7. MCPTT user at MCPTT client 2 accepts the call, which causes the MCPTT client 2 to send a first-to-answer call response to the MCPTT server. Included in the response, is the PCK (PCK_1) encapsulated to the user associated with the initiating client, MCPTT client 1. The PCK is then included in the SDP content of the response.

NOTE 1: If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the responding client providing KMS information. In this case, the responding client may re-attempt the above procedures.

8. The MCPTT server forwards the first-to-answer call response to MCPTT client 1 indicating that the MCPTT user at MCPTT client 2 has accepted the call. MCPTT client 1 extracts the PCK from the message.

9. The media plane for communication is now established and protected with the shared PCK.

10. MCPTT user at MCPTT client 3 accepts the call and sends a first-to-answer call response to the MCPTT server. MCPTT client 3 also includes an encapsulated PCK (PCK_2) in the response.

11. Since the first-to-answer call response from MCPTT client 2 has already been accepted, the MCPTT server sends a MCPTT first-to-answer call cancel request to MCPTT client 3. The encapsulated PCK provided by MCPTT client 3 (PCK_2) is discarded.

12-16. First-to-answer call signalling occurs as defined in TS 23.379 [2]. These messages do not contain security-related key material.

7.2.4.4 First-to-answer media protection

The first-to-answer media plane shall be protected as for a private call. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the first-to-answer media.

7.2.5 Ambient listening call

Ambient listening is a required feature for public safety users. Where the MC client may be used by non-public safety users, the feature shall not be implemented on the MC client and it shall not be possible to enable its use.

Ambient listening is described in clause 10.14 of 23.379 [2] and allows an authorised user to establish a “listening” private voice call with a target user without an indication that the communication is taking place. There are two types of ambient listening; the first type consists of the authorised user “listening” to a target user and the second type consists of the authorised user transmitting to a target user. Both types are intiated by the authorised user.

The MCPTT server provides the control and authorisation verification associated with an ambient listening call.

The security for an ambient listening call is established similar to that of a secure private call, i.e. a PCK is created for the session and provided securely in the ambient listening call request from the authorised user to the target user as per clause 7.2.2 for on-network private calls and clause 7.2.3 for off-network private calls.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCPTT IDs) shall be protected as described in clause 9.3.

The media plane for ambient listening shall be protected as for a private call using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the ambient listening media.

Floor control signalling for an ambient listening call shall be protected as described in clause 9.4.

7.2.6 Ambient viewing call

Ambient viewing is a required feature for public safety users. Where the MC client may be used by non-public safety users, the feature shall not be implemented on the MC client and it shall not be possible to enable its use.

Ambient viewing is described in clause 7.6 of 23.281 [37] and allows an authorised user to establish a “viewing” private video call with a target user without an indication that the communication is taking place. There are two types of ambient viewing; the first type consists of the authorised user “viewing” to a target user and the second type consists of the authorised user transmitting or “viewing to” a target user. Both types are intiated by the authorised user.

The MCVideo server provides the control and authorisation verification associated with an ambient viewing call.

The security for an ambient vewing communication is established similar to that of a secure private video communication, i.e. a PCK is created for the session and provided securely in the ambient viewing call request from the authorised user to the target user as per clause 7.2.2 for on-network private video communications and clause 7.2.3 for off-network video private communications.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCPTT IDs) shall be protected as described in clause 9.3.

The media plane for ambient viewing shall be protected as for a private video communication using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the ambient viewing media.

Transmission control signalling for an ambient viewing communication shall be protected as described in clause 9.4.

7.2.7 Private video pull

7.2.7.1 One-to-one video pull

One-to-one video pull is described in clause 7.3.2.3 of 23.281 [37] and consists of a private unidirectional video transmission from the called party to the calling party. Off-network video pull (video pull to self) is described in clause 7.3.3 of 23.281 [37].

The security for a one-to-one video pull communication is established similar to that of a secure private video call, i.e. a PCK is created for the session and provided securely in the Private call request from the calling user to the called user as per clause 7.2.2 for on-network private calls and clause 7.2.3 for off-network (video pull to self) calls.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCVideo IDs) shall be protected as described in clause 9.3.

The media plane for one-to-one video pull shall be protected as for a private video call using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the video media.

Trnasmission control signalling for a one-to-one video pull communication shall be protected as described in clause 9.4, while transmission control signalling for off-network video pull to self communications shall be protected as described in clause 9.4.4.

7.2.7.2 One-from-server video pull

One-from-server video pull is described in clause 7.3.2.4 of 23.281 [37] and consists of a private unidirectional video transmission pulled by the calling party from the MCVideo server. Figure 7.2.7.2-1 shows the messaging for a one-from-server video pull.

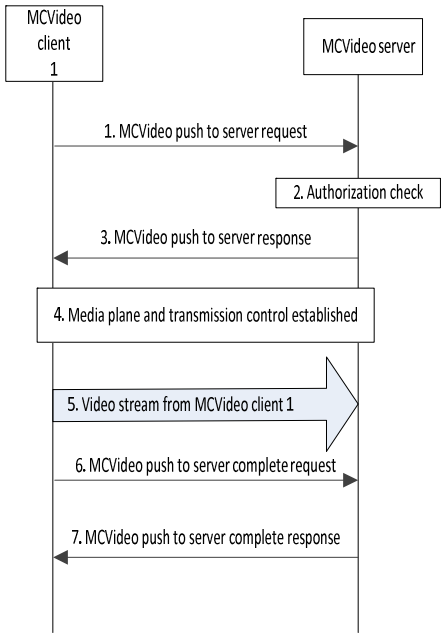

Figure 7.2.7.2-1 – One-from-server video pull

The security for a one-from-server video pull communication is established similar to that of a secure private video call, i.e. a PCK is created for the session and provided securely in the MCVideo pull from server request sent from the MCVideo client to the MCVideo server. Clause 7.2.2 applies for on-network one-from-server video commnications. Note that the PCK shall be encrypted to the identity of the MCVideo server rather than to that of another MCVideo user.

Off-network operation is not supported for one-from-server video pull communications.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCVideo IDs) shall be protected as described in clause 9.3.

The media plane for one-to-one video pull shall be protected as for a private call using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the video media.

Transmission control signalling for a one-from-server video pull communication shall be protected as described in clause 9.4.

7.2.8 Private video push

7.2.8.1 One-to-one video push

One-to-one video push is described in clause 7.4.2.3 of 23.281 [37] and consists of a private unidirectional video transmission from the calling party to the called party. Off-network video push to another MCVideo user is described in clause 7.4.3.3 of 23.281 [37].

The security for a one-to-one video push communication is established similar to that of a secure private video call, i.e. a PCK is created for the session and provided securely in the Private call request from the calling user to the called user as per clause 7.2.2 for on-network private calls and clause 7.2.3 for off-network private calls.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCVideo IDs) shall be protected as described in clause 9.3.

The media plane for one-to-one video pull shall be protected as for a private video call using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the video media.

Transmission control signalling for a one-to-one video push communication shall be protected as described in clause 9.4, while transmission control signalling for off-network video push to another MCVideo user communications shall be protected as described in clause 9.4.4.

7.2.8.2 One-to-server video push

One-to-server video push is described in clause 7.4.2.4 of 23.281 [37] and consists of a private unidirectional video transmission pushed from the calling party to the MCVideo server. Figure 7.2.8.2-1 shows the messaging for a one-from-server video push.

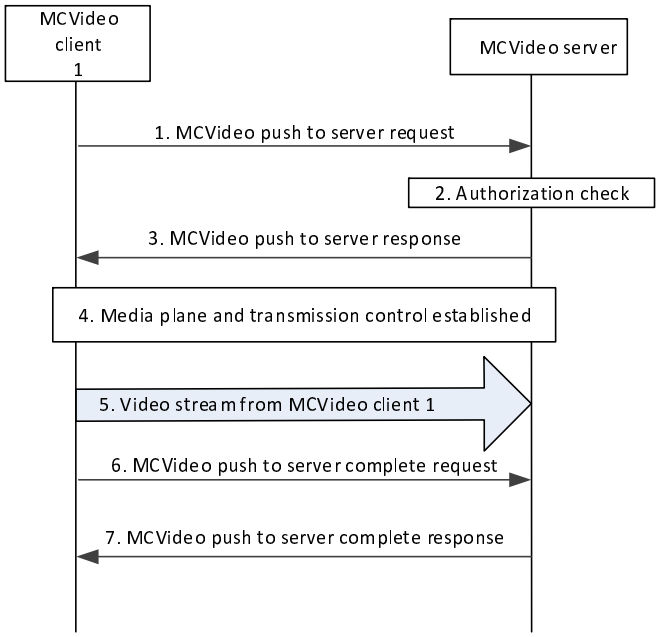

Figure 7.2.8.2-1 – One-to-server video push

The security for a one-to-server video push communication is established similar to that of a secure private video call (i.e. a PCK is created for the session and provided securely in the MCVideo push to server request sent from the MCVideo client to the MCVideo server). Clause 7.2.2 applies for on-network one-to-server video communications. Note this requires that the PCK shall be encrypted to the identity of the MCVideo server rather than to that of another MCVideo user.

Off-network operation is not supported for one-to-server video push communications.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCVideo IDs) shall be protected as described in clause 9.3.

The media plane for one-to-server video push shall be protected as for a private call using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the video media.

Transmission control signalling for a one-to-server video push communication shall be protected as described in clause 9.4.

7.2.8.3 Remotely initiated video push

On-network remotely intiated video push is described in clause 7.4.2.5 of 23.281 [37] and consists of an authorised MCVideo user initiating a private unidirectional video transmission from a source MCVideo user and a destination MCVideo user.

Off-network remotely initiated video push is described in clause 7.4.3.4 of 23.281 [37] and consists of an authorised MCVideo user initiating a private unidirectional video transmission from a source MCVideo user to a destination MCVideo user without the MCVideo serving the call setup and media transmission.

Figure 7.2.8.3-1 shows the messaging for an on-network remotely initiated video push communication.

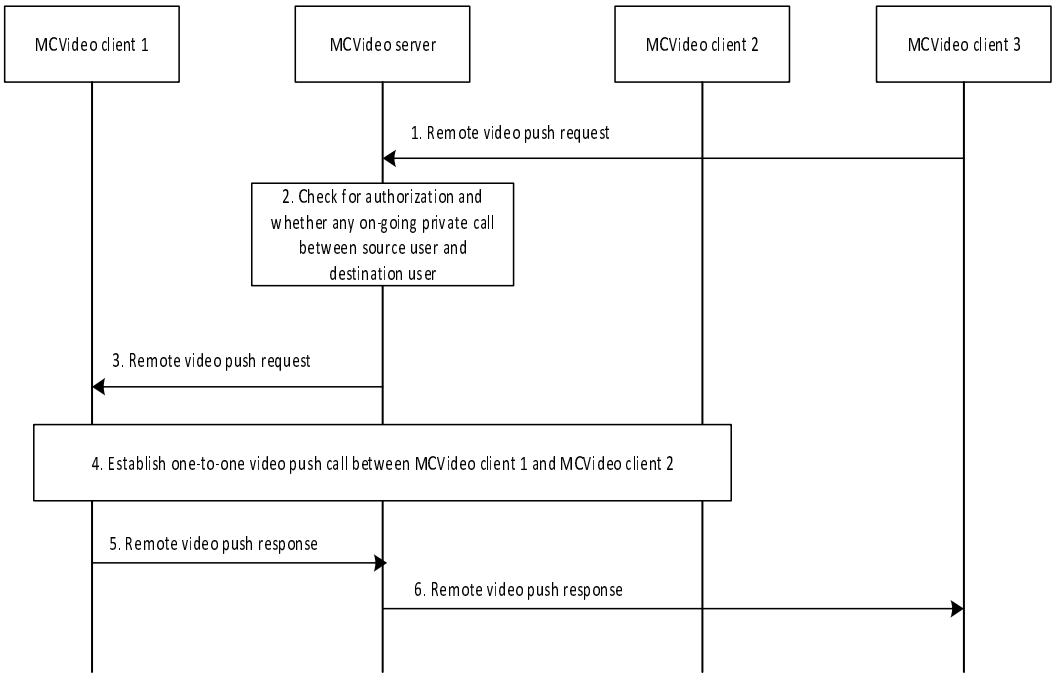

Figure 7.2.8.3-1: On-network remotely initiated video push

The security context for an on-network remotely initiated video push communication is established during step 4 of figure 7.2.8.3-1 between MCVideo client 1 and MCVideo client 2 and is similar to that of a secure private video call, i.e. a PCK is created for the session and provided securely in the Private communication request from MCVideo client 1 to MCVideo client 2 as described in clause 7.2.2.

In figure 7.2.8.3-2, the remote video push request message from MCVideo client 3 does not establish a security context for the call; however it does provide the MCVideo IDs of participating MCVideo client 1 and MCVideo client 2.

Figure 7.2.8.3-2 shows the messaging for an off-network remotely initiated video push communication.

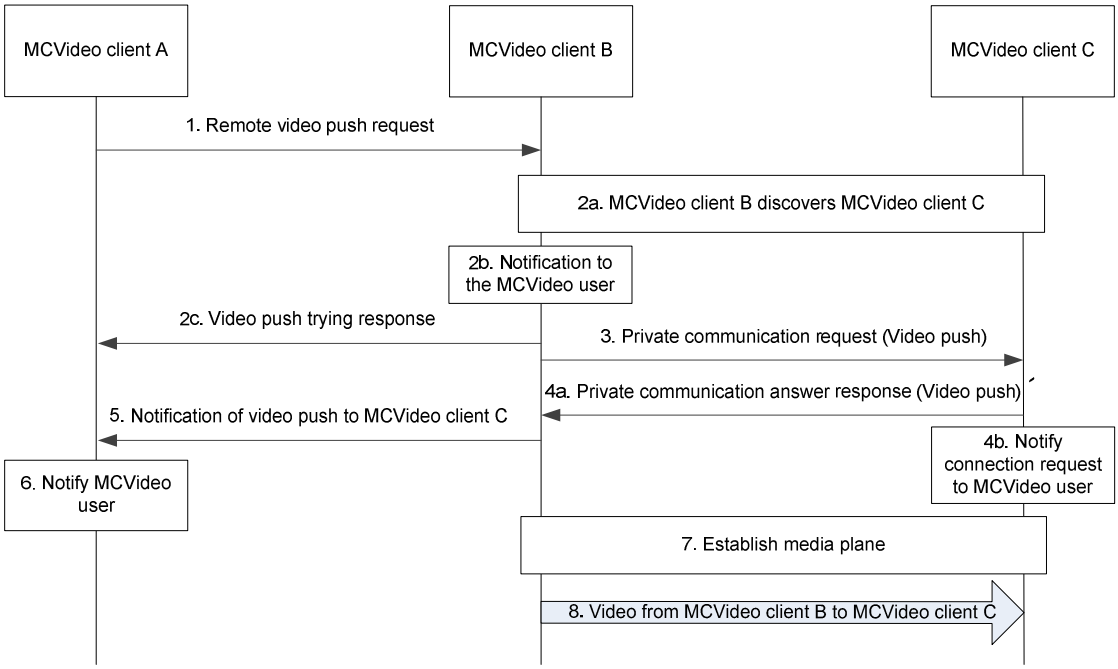

Figure 7.2.8.3-2: Off-network remotely initiated video push

The security context for an off-network remotely initiated video push communication is established during step 3 of figure 7.2.8.3-2 between MCVideo client B and MCVideo client C and is similar to that of a off-network secure private video call, i.e a PCK is created for the session and provided securely as described in clause 7.2.3.

In figure 7.2.8.3-2, the remote video push request message from MCVideo client A does not establish a security context for the call; however it does provide the MCVideo IDs of participating MCVideo client B and MCVideo client C.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCVideo IDs) shall be protected as described in clause 9.3 for both on-network and off-network operation.

For both on-network and off-network, the media plane for a remotely initiated video push communication shall be protected as for a one-to-one video push communication using a PCK. Clause 7.4.1 is applied to convert the PCK into the SRTP Master Key/Salt, and clause 7.5 is applied for the protection of the video media.

Transmission control signalling for a remotely initiated video push communication shall be protected as for a one-to-one video push communication, as described in clause 9.4, while transmission control signalling for off-network remotely initiated video push communications shall be protected as described in clause 9.4.4.

7.3 Group communications

7.3.1 General

To support MCPTT and MCVideo group communications, group security procedures are used to establish and distribute keys to the members of predefined or dynamically defined groups.

Key material (GMK and GMK-ID) for a predefined group is created and distributed by the GMS to the members of the group via the common key distribution mechanisms defined in clause 5.7.

Key material for dynamically created groups is created and distributed by the GMS to the members of the group via the security mechanisms defined in the following sub clauses.

NOTE: Void

7.3.2 Group creation security procedure

The group creation procedure is described in clause 10.2.3 of 3GPP TS 23.280 [36] and applies to the MCPTT scenario of normal group creation by an MC administrator and user regrouping operations by an authorized user/dispatcher. To establish the security context for the group, the GMS follows the procedures in clause 5.7 to create a new GMK and GMK-ID.

The encapsulated GMK and GUK-ID is sent to group members by the GMS within a notification message (step 4 in clause 10.2.3 of 3GPP TS 23.280 [36]). The procedure is equivalent to that described in clause 5.7 of this specification.

7.3.3 Dynamic group keying

7.3.3.1 General

In the GMK distribution procedures described in this clause, the GMS shall be provisioned with the same information as any MC UE by the KMS as described in clause 5.3; the only distinguishing feature is that the GMS's identity is a Server URI rather than an MC Service ID.

NOTE 1: Void.

In addition to authorisation, the only information the GMS requires to create the group are the MCPTT IDs of the group members. These two features combined allow groups to be created and keyed at any time, by any authorized entity.

Such flexibility is required to support a number of MCPTT group procedures within 3GPP TS 23.280 [36].

NOTE 2: The dynamic group keying mechanisms may not support off-network scenarios.

7.3.3.2 Group regrouping security procedure (within a single MC domain)

Group Regroup procedures for the MC system are described in clause 10.2.4.1 of 3GPP TS 23.280 [36]. To create the security context for the temporary group, the GMS follows the procedures in clause 5.7, creating a new GMK and GMK-ID for the temporary group.

An encapsulated GMK and GUK-ID is sent to the temporary group members by the GMS within a notification message (step 5 in clause 10.2.4.1 of 3GPP TS 23.280 [36]). The procedure is equivalent to that described in clause 5.7.

7.3.3.3 Group regrouping security procedure (involving multiple MC domains)

The MCPTT group regroup security procedure (shown below in figure 7.3.3.3-1) involves multiple MC users from multiple MC domains and is an integrated component of the MCPTT group regrouping procedure described in clause 10.2.4.2 of 3GPP TS 23.280 [36].

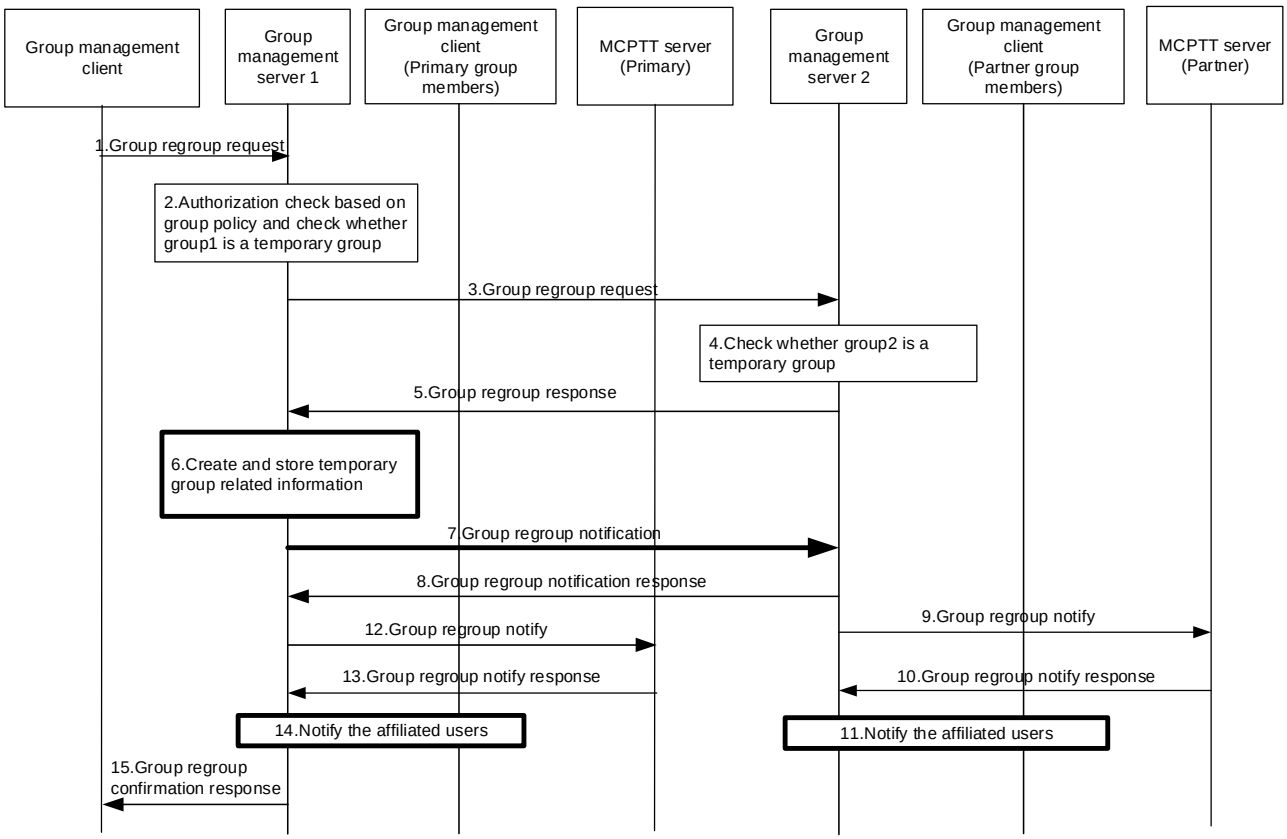

Figure 7.3.3.3-1: Group regroup security procedure (multiple MC domains)

Prior to beginning the procedure, the MC UEs, primary GMS and partner GMS are provisioned by a KMS as described in clause 5.3.

1-5: These steps are as defined in clause 10.2.4.2 of 3GPP TS 23.280 [36].

6: To create the security context for the temporary group, the primary GMS creates a new GMK and GMK-ID for the temporary group along with other group related information.

7,8: The primary GMS notifies the partner GMS of the group regroup operation. The primary GMS includes a Group Key Transport payload following the procedures in clause 5.7, treating the partner GMS as another user within the group. Accordingly, the payload encrypts the new GMK to the identity of the partner GMS and is signed using the identity of the primary GMS. The GUK-ID is derived using the User Salt generated from the partner GMS's URI.

NOTE 1: If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the primary GMS providing KMS information. In this case, the primary GMS may re-attempt the above procedures.

9,10: These steps are as defined in clause 10.2.4.2 of 3GPP TS 23.280 [36].

11: The partner GMS extracts the GMK and GMK-ID from the notification. The partner GMS then notifies the affiliated users within the partner MC domain. The partner GMS re-encrypts the GMK to the identity of the affiliated users in the partner system, generates new GUK-IDs for each user and signs using its identity (the identity of the partner GMS) following the procedure in clause 5.7.

12,13: These steps are as defined in clause 10.2.4.2 of 3GPP TS 23.280 [36].

14: The primary GMS notifies the affiliated users within its own MC domain. The primary GMS includes a Group Key Transport payload including a GMK and GUK-ID following the procedures in clause 5.7. The GMK is encrypted to the identity of the MC user and is signed using the identity of the primary GMS.

15: This step is as defined in clause 10.2.4.2 of 3GPP TS 23.280 [36].

It is possible that a partner GMS may be represented by an MC Security Gateway (as defined in Annex L), rather than using full end-to-end security. In this case, the partner GMS's KMS Certificate will have the ‘IsSecurityGateway' attribute set to ‘true' (see clause D.3.2.2). Should a partner GMS be represented by an MC Security Gateway, the primary GMS shall indicate to all group users and partner GMS(s) that the GMK is shared with an MC Security Gateway. This is achieved by setting the ‘Security Gateway' bit in the ‘Status' field of the GMK's key parameters (see clause E.6.9).

7.3.4 Broadcast group call

Broadcast group call is described in clause 10.6.2.5.2 of 23.379 [2] and consists of a group transmission where the initiating user expects no response from the other group members. When the initiating user's transmission is complete, the broadcast group communication ends.

The security context for a broadcast group communication is established similar to that of a secure group communication where the GMK associated to the broadcast group shall be converted into the SRTP Master Key/Salt per clause 7.4.2. Clause 7.5 shall be applied to establish protection of the broadcast group media.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCX Service User IDs and MCX Group IDs) shall be protected as described in clause 9.3.

Floor control signalling for on-network broadcast group communications shall be protected as described in clause 9.4.6 while floor control signalling for off-network broadcast group communications shall be protected as described in clause 7.4.2.

7.3.5 Group-broadcast group call

Group-broadcast group call is described in clause 10.6.2.5.2.1 of 23.379 [2] and consists of a group transmission to a set of groups rather than to a set of users. Like a broadcast group communication, the initiating user expects no response from the target groups. When the initiating user's transmission is complete, the group-broadcast group communication ends.

The security context for a group-broadcast group communication is similar to that of a secure group communication, i.e. the group-broadcast group ID is predefined and assigned a GMK. The GMK associated to the group-broadcast group shall be converted into the SRTP Master Key/Salt per clause 7.4.2. Clause 7.5 shall be applied to establish protection of the group-broadcast group media.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCX Service User IDs and MCX Group IDs) shall be protected as described in clause 9.3.

Floor control signalling for on-network group-broadcast group communications shall be protected as described in clause 9.4.6 while floor control signalling for off-network group-broadcast group communications shall be protected as described in clause 7.4.2.

7.3.6 Emergency group call

An emergency group call is described in clause 10.6.2.6.1 of 23.379 [2] and consists of a group communication where the priority of the transmission or group is set to an emergency status. An existing group call may be elevated to an emergency status or a separate designated emergency group may be used.

When an existing group call is elevated to emergency status, there should be no change to the ongoing security context for that group. Media protection, floor control protection, and application signalling protection continue to use the existing keys and mechanisms that were in place prior to elevating the group to emergency status. However, the call may be downgraded to clear to ensure the emergency group call is setup by the MCX system.

When a designated emergency group is used and the user intiates an emergency call, the emergency group is established and a new security context set up shall be attempted (assuming the emergency group is not already active). If the security context setup is successful the emergency group call shall proceed with security, otherwise based on MCX operator policy the call may default to unencrypted.

For either existing or designated call types, a secured emergency group call uses group communication mechanisms where a GMK associated with the emergency group is distributed to the affiliated members per clause 5.7. The GMK is used to encrypt the media for on-network and off-network, unicast or multicast, emergency group communications. The GMK shall be converted into the SRTP master key and master salt as described in clause 7.4.2 and the emergency group media shall be encrypted using the SRTP master key and master salt as defined in clause 7.5.1.

For either existing or designated call types, floor control signalling for a on-network and off-network secured emergency group communication shall be protected as described in clause 9.4.

For either existing or designated call types, when required by the MCX service provider, sensitive application signalling parameters (e.g. MCX Service User IDs and MCX Group IDs) shall be protected as described in clause 9.3.

7.3.7 Imminent peril group call

An imminent peril group call is described in clause 10.6.2.6.2 of 23.379 [2] and consists of a group communication where the priority of the transmission is elevated to imminent peril status. The imminent peril transmission may be sent within an existing group call or alternatively a separate designated imminent peril group may be used.

When an imminent peril transmission is sent on an existing group call, there should be no change to the ongoing security context for that group. Media protection, floor control protection, and application signalling protection continue to use the existing keys and mechanisms that were in place prior to the imminent peril transmission. However, the call may be downgraded to clear to ensure the emergency alert is setup by the MCX system.

When a designated imminent peril group is used and the user initiates an imminent peril transmission, the imminent peril group is established and a new security context set up shall be attempted (assuming the imminent peril group is not already active). If the security context setup is successful the imminent peril group call shall proceed with security, otherwise based on MCX operator policy the call may be downgraded to unencrypted.

For either existing or designated call types, a secured imminent peril group call uses group communication mechanisms where a GMK associated with the existing or imminent peril group is distributed to the affiliated members per clause 5.7. The GMK is subsequently used to encrypt the media for on-network and off-network, unicast or multicast, imminent peril group communications. The GMK shall be converted into the SRTP master key and master salt as described in clause 7.4.2 and the imminent peril group media shall be encrypted using the SRTP master key and master salt as defined in clause 7.5.1.

For either existing or designated call types, floor control signalling for an on-network and off-network secured imminent peril group communication shall be protected as described in clause 9.4.

For either existing or designated call types, when required by the MCX service provider, sensitive application signalling parameters (e.g. MCX Service User IDs and MCX Group IDs) shall be protected as described in clause 9.3.

7.3.8 Emergency Alert

An emergency alert is described in clause 10.6.2.6.3 of 23.379 [2] and consists of a group communication where at least one user has issued an emergency alert indication, elevating that user to an emergency state. A transmission by a user while in the emergency state has elevated priority and may be sent within an existing group call or alternatively a separate designated emergency group.When an existing group call is used as the emergency group, there should be no

change to the ongoing security context for that group. Media protection, floor control protection, and application signalling protection continue to use the existing keys and mechanisms that were in place prior to the emergency alert. However, the call may be downgraded to clear to ensure the emergency alert is setup by the MCX system

When a designated emergency group is used and the user intiates an emergency alert and transmission, the assigned emergency group is established and a new security context set up is attempted (assuming the emergency group is not already active). If the security context setup is successful, the imminent peril group call shall proceed with security, otherwise based on MCX operator policy the call may default to unencrypted.

For either existing or designated call types, a secured emergency alert issued emergency group call uses group communication mechanisms where a GMK associated with the emergency group is distributed to the affiliated members per clause 5.7. The GMK is used to encrypt the media for on-network and off-network, unicast or multicast, emergency group communications. The GMK shall be converted into the SRTP master key and master salt as described in clause 7.4.2 and the emergency group media shall be encrypted using the SRTP master key and master salt as defined in clause 7.5.1.

For either existing or designated call types, floor control signalling for an on-network and off-network secured emergency group communication shall be protected as described in clause 9.4.

For either existing or designated call types, when required by the MCX service provider, sensitive application signalling parameters (e.g. MCX Service User IDs and MCX Group IDs) shall be protected as described in clause 9.3.

7.3.9 Remotely initiated video push to group

On-network remotely intiated video push to group is described in clause 7.4.2.6 of 23.281 [37] and consists of an authorised MCVideo user initiating a broadcast video transmission sourced from a second MCVideo user.

Off-network remotely initiated video push to a group is described in clause 7.4.3.5 of 23.281 [37] and consists of an authorised MCVideo user initiating a broadcast video transmission sourced from a second MCVideo user without the MCVideo serving the call setup and media transmission.

Figure 7.3.11.1-1 shows the messaging for on-network remotely initiated video push to group communication.

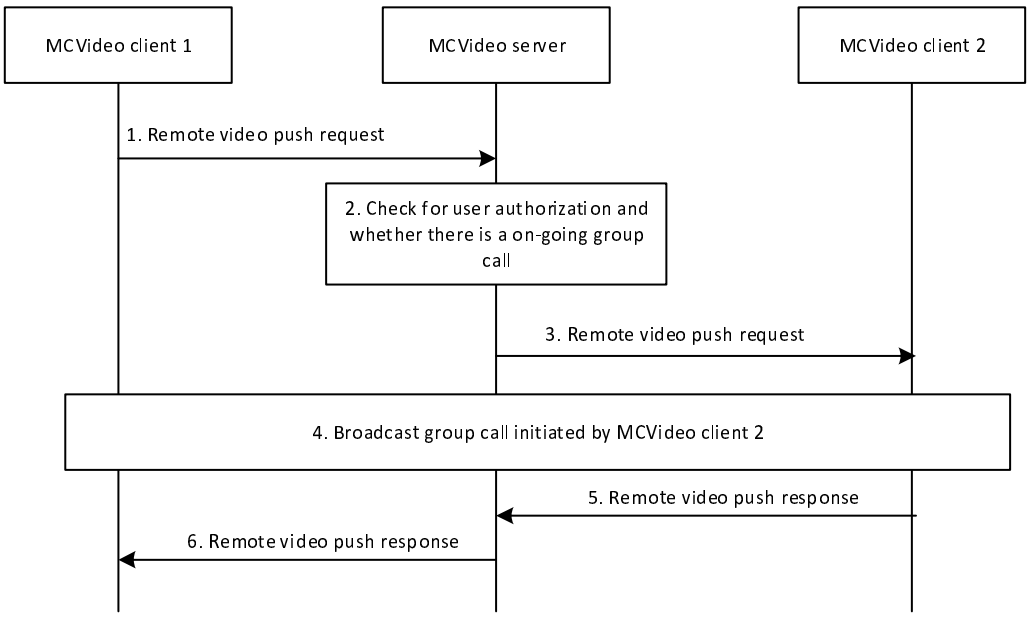

Figure 7.3.11.1-1: On-network remotely initiated video push to group

The security context for an on-network remotely initiated video push to group communication is established in step 4 of figure 7.3.11.1-1 by the target MCVideo user (MCVideo client 2) similar to that of a secure group broadcast communication where a GMK associated to the broadcast group shall be converted into the SRTP Master Key/Salt per clause 7.4.2. Clause 7.5 shall be applied to establish protection of the broadcast group media.

In figure 7.3.11.1-1, the remote video push request message does not establish a security context for the broadcast call but does however provide the group ID for the broadcast group.

Figure 7.3.11.1-2 shows the messaging for an off-network remotely initiated video push to group communication.

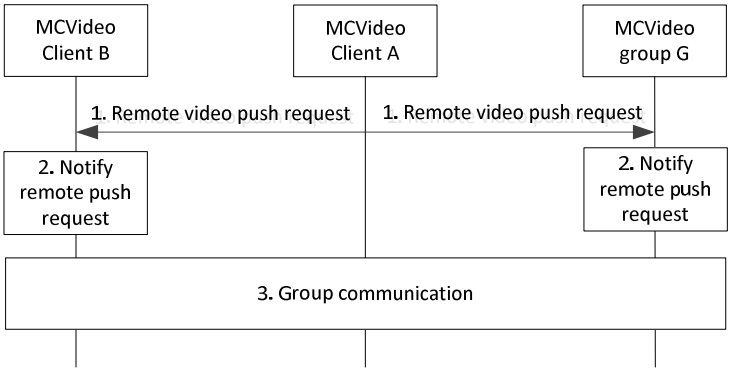

Figure 7.3.11.1-2: Off-network remotely initiated video push to group

The security context for an off-network remotely initiated video push to group communication is established in step 3 of figure 7.3.11.1-2 by the target MCVideo user (MCVideo client B) similar to that of a secure group broadcast communication where a GMK associated to the broadcast group shall be converted into the SRTP Master Key/Salt per clause 7.4.2. Clause 7.5 shall be applied to establish protection of the broadcast group media.

In figure 7.3.11.1-2, the remote video push request message from MCVideo client A does not establish a security context for the broadcast call but does however provide the group ID for the broadcast group.

When required by the MCX operator, sensitive application signalling parameters (e.g. MCX Group IDs) shall be protected as described in clause 9.3 for both on-network and off-network operation.

Transmission control signalling for on-network remotely initiated video push to group communications shall be protected as described in clause 9.4.6 while transmission control signalling for off-network remotely initiated video push to group communications shall be protected as described in clause 7.4.2.

7.3.10 Multi-talker configured MCPTT group

The requirements of the multi-talker feature for mission critical communications are defined in 22.179 [3] and may occur as part of multi-talker configured MCPTT group communications as decribed in 23.379 [2]. In a multi-talker configured MCPTT group communication, more than one media stream may be mixed together and simultaneously heard by a call participant.

When media protection is applied, the GMK assigned to the multi-talker configured MCPTT group is used to protect the group media as described in clauses 7.4.2 and 7.5, however because the controlling MCPTT server cannot decrypt the media streams (as it does not possess the GMK of the group), the controlling MCPTT server is therefore unable to mix media streams together prior to dissemination of the media to the members of the group. In this case, all protected and active media streams for the multi-talker configured group call shall be sent by the controlling MCPTT server to the participating MC UEs where each participating MC UE shall decrypt the received media streams. Once the media streams have been decrypted by the MC UE, the media may be mixed and presented to the user.

If media protection is not applied then the multi-talker configured group media shall be mixed and provided to the participating MC UEs as indicated in 23.379 [2].

When MC signaling protection is enabled, unicast delivery of signalling messages applicable to multi-talker configured MCPTT group communications shall be protected as defined in clause 9.4.2 and multicast delivery of these signaling messages shall be protected as defined in clause 9.4.3.

7.4 Key derivation for media

7.4.1 Derivation of SRTP master keys for private call

As a result of this mechanism, the private call members share a PCK and PCK-ID. The PCK shall be used as the MIKEY Traffic Generating Key (TGK), the PCK-ID shall be used as the MIKEY CSB ID. The MIKEY RAND shall be the MIKEY RAND value transmitted together with the PCK provision. The CS-ID value is defined in Table E.1.3-1. These shall be used to generate the SRTP Master Key and SRTP Master Salt as specified in IETF RFC 3830 [22]. The key derivation function defined in section 4.1.4 of RFC 3830 [22] using the PRF-HMAC-SHA-256 Pseudo-Random Function as described in IETF RFC 6043 [25], section 6.1 shall be supported for generating the SRTP Master Key and Salt.

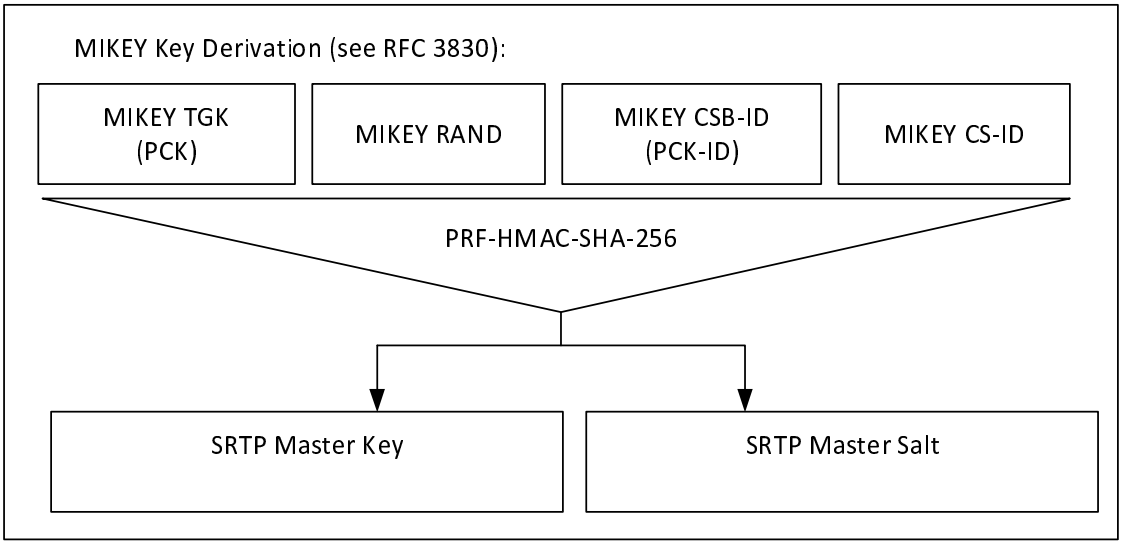

Figure 7.4.1-1: Key Derivation for media stream protection

To identify the security context from the media stream a SRTP Master Key Identifier (MKI) is required. The MKI shall be the 32-bit PCK-ID which has a purpose tag of '1'.

When the MC client is operating off-network, the PCK is used to derive keys for floor control, transmission control,and media control (SRTCP). Thus, the Master Key and Master Salt used for SRTCP is the same with the Master Key and Master Salt used for SRTP, so is the MKI.

See clause 9.4.6 for key derivation procedures for private communication floor, transmission, and media control (SRTCP) when the MC client is operating on-network.

7.4.2 Derivation of SRTP master keys for group media

As a result of this mechanism, the group members share a GMK and GUK-ID. The GMK shall be used as the MIKEY Traffic Generating Key (TGK), the GUK-ID shall be used as the MIKEY CSB ID. The MIKEY RAND shall be the MIKEY RAND value transmitted in the MIKEY message used to distribute the GMK. The CS-ID value is defined in Table E.1.3-1. These shall be used to generate the SRTP Master Key and SRTP Master Salt as specified in IETF RFC 3830 [22]. The key derivation function defined in section 4.1.4 of IETF RFC 3830 [22] using the PRF-HMAC-SHA-256 Pseudo-Random Function as described in IETF RFC 6043 [25], section 6.1 shall be supported for generating the SRTP Master Key and Salt.

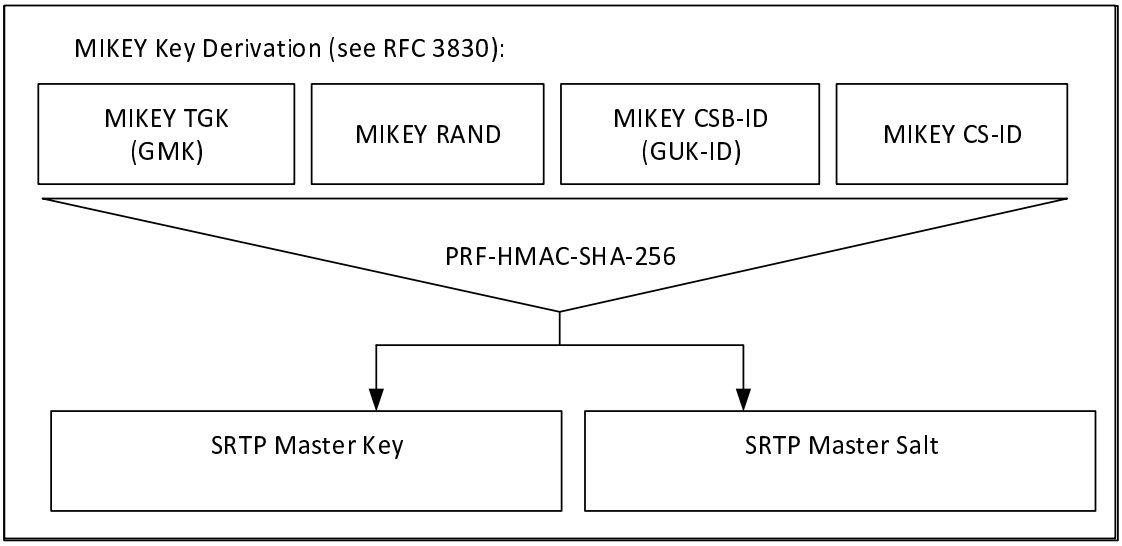

Figure 7.4.2-1: Key Derivation for media stream protection

To identify the security context from the media stream a SRTP Master Key Identifier (MKI) is required. The MKI should be a 64-bit value formed by concatenating the GMK-ID with the GUK-ID (GMK-ID || GUK-ID). The GMK-ID shall have a purpose-tag of '0'. The GUK-ID is derived as specified in Annex F.1.3, using the MC service user ID of the transmitting user.

Where the transmitting user is known through other means, the MKI may be solely the 32-bit GMK-ID. In this case the terminating user extracts the GUK-ID by calculating the User Salt and xor'ing this value with the GMK-ID.

When the MC client is operating off-network, the GMK is used to derive keys for floor control, transmission control,and media control (SRTCP). Thus, the Master Key and Master Salt used for SRTCP is the same with the Master Key and Master Salt used for SRTP, so is the MKI.

See clause 9.4.6 for key derivation procedures for group communication floor, transmission, and media control (SRTCP) when the MC client is operating on-network.

7.5 Media protection profile

7.5.1 General

The following mechanism shall be used to protect MCPTT and MCVideo communications which use the Real-Time Transport Protocol (RTP), cf. IETF RFC 3550 [12]. The integrity and confidentiality protection for MCPTT and MCVideo communications using RTP shall be achieved by using the Secure Real-Time Transport Protocol (SRTP), IETF RFC 3711 [13].

The key management mechanism for SRTP is described elsewhere. As a result of this mechanism, those communicating will have shared the following:

1) A SRTP Master Key.

2) A SRTP Master Salt.

3) A SRTP Master Key Identifier (MKI) referencing the above two values.

The mechanism described in IETF RFC 3711 [13] is used to encrypt the RTP payload. A diagram of the key derivation mechanism (as described in IETF RFC 3711) is shown in figure 7.5.1-1.

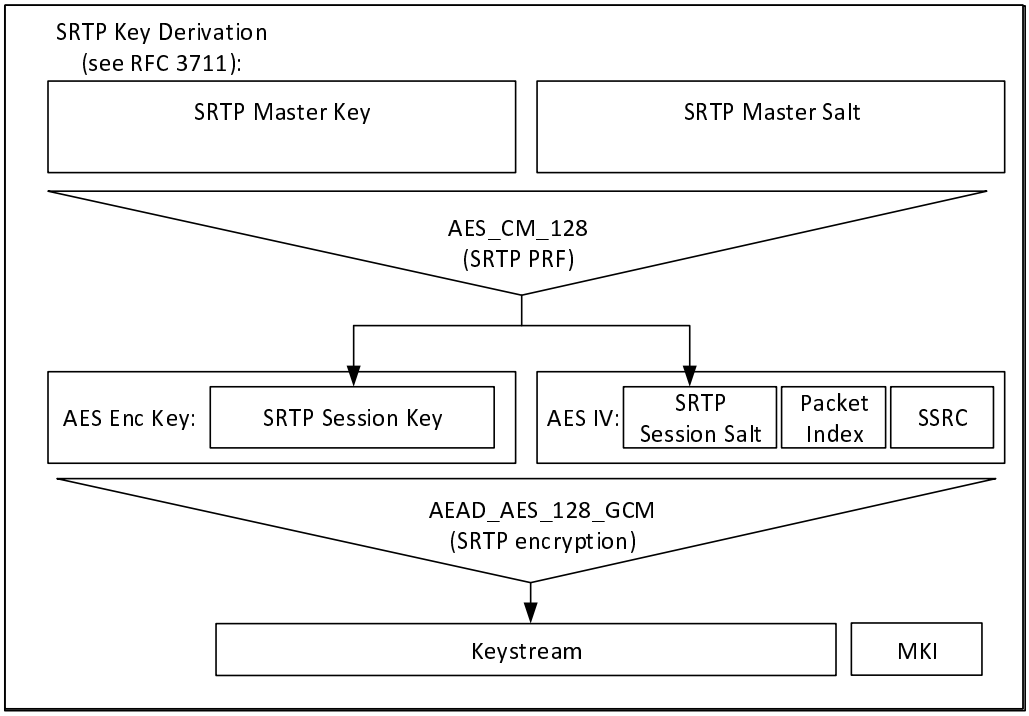

Figure 7.5.1-1: Security mechanism for media stream protection

The AES-CM-128 algorithm as defined in IETF RFC 3711 [13] shall be supported as the SRTP PRF (which is used to derive the SRTP session key and salt). A SRTP key derivation rate of 0 shall be used to indicate that session keys and salts shall not be refreshed. The AEAD_AES_128_GCM algorithm as defined in IETF RFC 7714 [26] shall be supported for providing confidentiality and data authentication of SRTP packets. The AEAD_AES_128_GCM algorithm requires that the SRTP session key is 16 octets in length, and the SRTP session salt is 12 octets in length.

Unless provided in the MIKEY message used to distribute the SRTP Master Key, the SSRC shall be randomly generated for each session. For group communications, the GMS shall not provide SSRCs, and hence the SSRC shall be randomly generated.

Care should be taken to avoid SSRC repetition when a user uses the SRTP Master Key for more than one session. In particular, SSRCs shall not be generated in a way which could cause collisions (e.g. from the same seed).

The SRTP authentication tag may be appended to every 'rth' packet as defined in IETF RFC 4771 [24] to provide the SRTP ROC counter to MC UEs performing a late-entry to the communication. A 'mode 3' integrity transform (RCCm3) shall be supported for transmitting the ROC within a 4 octet SRTP authentication tag.

NOTE: The ROC and MKI fields of the SRTP packet are not authenticated as part of the AEAD_AES_128_GCM algorithm. However, modification of these fields would cause a failure to validate the AEAD authentication tag which would cause the packet to be correctly rejected by the receiver. If an unauthenticated SRTP encryption mode be used, there will be a security impact of a malicious modification of the ROC or MKI packets.

7.5.2 Security procedures for media stream protection

Media stream protection does not require any signalling mechanism to convey information. The information is provided within each SRTP Packet.

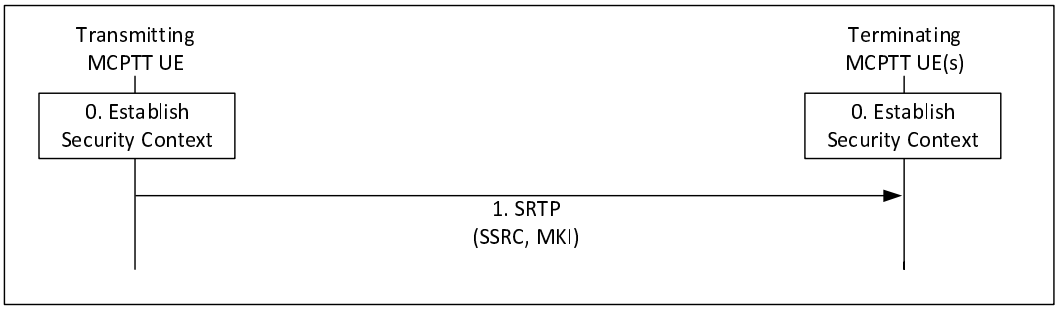

Figure 7.5.2-1: Security procedure for media stream protection

Figure 7.5.2-1 shows the security mechanism.

0) Prior to beginning this procedure the MC UEs involved in the communication shall have established a security context (SRTP Master Key, SRTP Master Salt, MKI).

1) Transmitting UEs shall send SRTP packets using the format described in IETF RFC 3711 [13]. The packet shall include a Master Key Identifier (MKI) field which contains the information required to locate the SRTP Master Key and Master Salt, and may include the SRTP ROC as defined in IETF RFC 4771 [24]. On receipt of a SRTP packet, a terminating UE shall use the contents of the MKI to look up the appropriate SRTP Master Key and salt and generate the appropriate SRTP session key and salt if it satisfies the key derivation rate criteria as specified in IETF RFC3711. If it appears in the SRTP packet, the terminating UE shall use the contents of the SRTP authentication tag to establish the SRTP ROC as defined in IETF RFC 4771 [24].

NOTE 1: Assuming members of the group have been keyed/pre-provisioned at some point in the past, this security mechanism is entirely stateless.

NOTE 2: The receiver does not need to generate an appropriate SRTP session key and salt every time when it receives a SRTP packet. The key derivation rate defined in IETF RFC3711 [13] determines the session key generation frequency. Refer to RFC3711 for more information.

NOTE 3: As the SRTP synchronization source identifier (SSRC) is used for encryption and decryption, the SSRC value in the SRTP packet needs to be maintained from the transmitting UE to the receiving UE. This includes the uplink and the downlink, over unicast or multicast.

A diagram of the SRTP packet format is within figure 7.5.2-2.

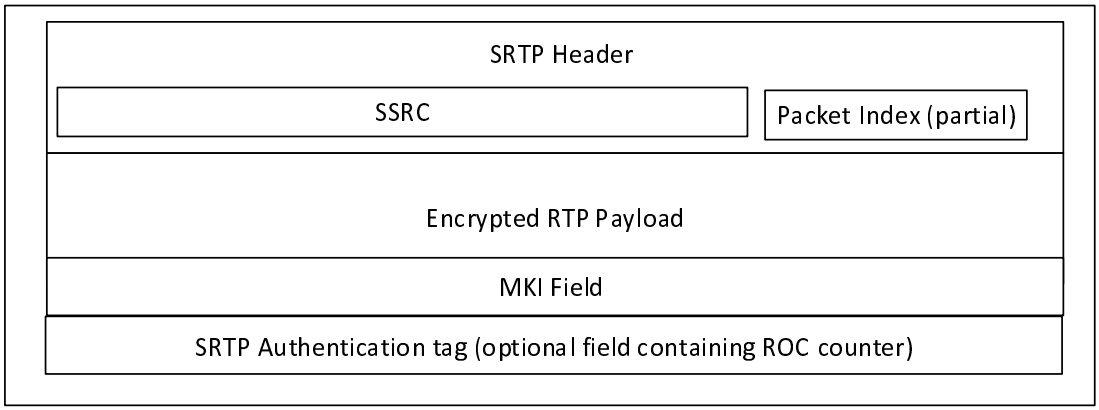

Figure 7.5.2-2: SRTP packet format showing security parameters

The length of the MKI field is defined by the key distribution procedure used to create the original security context.

8 MCData

8.1 Overview

MCData SDS allows transmission of short data messages (SDS), either private or group, over both the signalling plane (reference point MCData-SDS-1) and media plane (reference point MCData-SDS-2).

MCData File Distribution (FD) also allows for transmission of files, either private or group, over the media plane or using HTTP. When distributed using HTTP, binary data representing the file is uploaded and downloaded using HTTP POST and HTTP GET.

MCData signalling parameters for SDS and File Distribution are routed within SIP messages. Protection for these signalling messages and files when distributed using HTTP, use the same key material as for MCPTT and MCVideo.

The MCData SDS or FD messages may also contain a MCData Data signalling payload or a MCData Data payload or both. These payloads may be within a SIP message should the signalling plane be used, or within a MSRP message should the media plane be used. The MCData Data payload may be end-to-end confidentiality and integrity protected according to an end to end security context payload.

The file when distributed using HTTP may be end-to-end confidentiality and integrity protected according to an end to end security context payload before being uploaded.

Components of MCData messages:

- **MCData signalling parameters**: generic Mission Critical Services signalling elements e.g. MCData Group ID, MCData user ID. These parameters are confidentiality protected between the MCData Client and the MCData server with signalling plane security mechanisms.
- **MCData Data signalling payload**: information elements necessary for identification and management of the MCData messages e.g. conversation identifiers, session identifiers, transaction identifiers, disposition requests, etc. This payload is confidentiality protected between the MCData Client and the MCData server with signalling plane security mechanisms.
- **End to end security parameters**: information specifying the cryptographic elements used to protect the data payload).
- **MCData Data payload**: the actual user payload for MCData user or application consumption. This payload is end to end confidentiality and integrity protected.

Components of the MCData message (MCData signalling parameters and MCData Data signalling payload) are integrity protected between the MCData Client and the MCData server with the signalling plane security mechanisms.

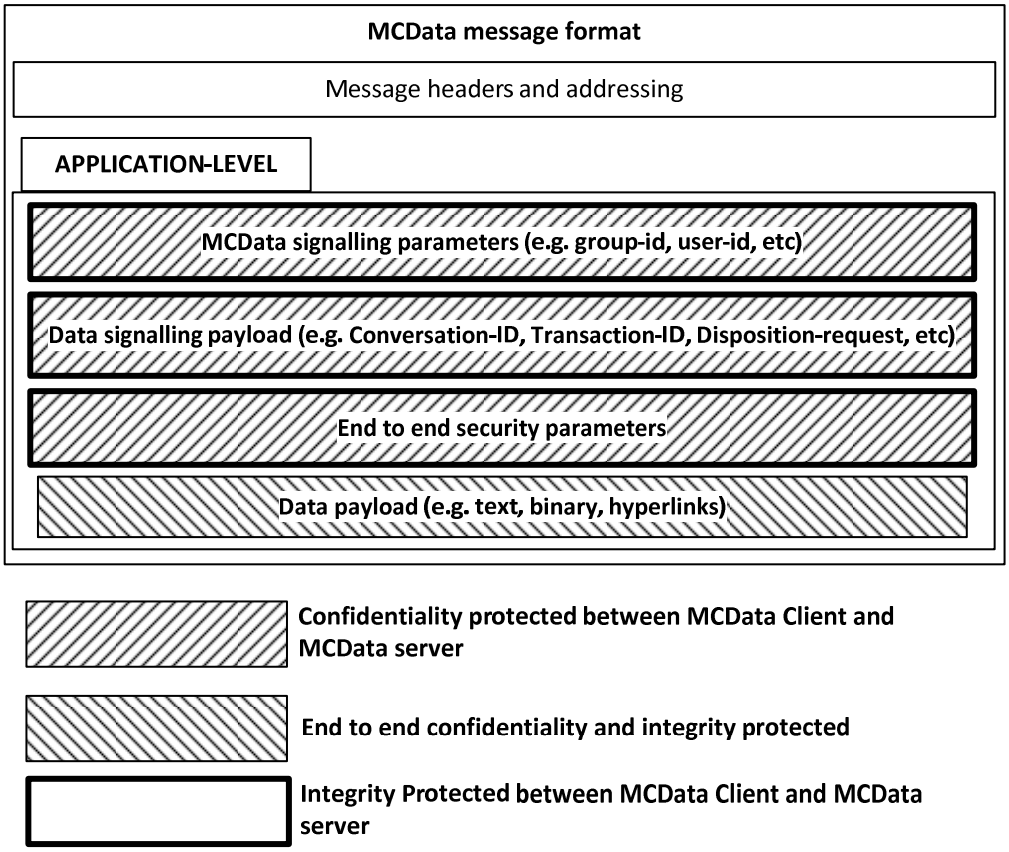

Figure 8.1-1: MCData message components

For one-to-one communications the PCK is used to protect the MCData data payload or the file when distributed using HTTP. For group communications, the GMK is used to protect the MCData data payload or the file when distributed using HTTP. The data payload or the file when distributed using HTTP may also be authenticated by the initiator.

Distribution of the PCK is within the signalling channel setup for the MCData private message (either SDS or FD). Distribution of the GMK is as defined in clause 5.7.

8.2 Key Management

Key management for MCData follows the same model as MCVideo and MCPTT. Where a key is used for protection of MCData or MCVideo data, the same type of key shall be used in the same circumstance for MCData. Each key used for protection of MCData payloads is known as the MCData Payload Protection Key (DPPK).

MCData signalling parameters and Data signaling payloads are protected as follows:

- Unicast MCData signalling parameters and Data signaling payloads between client and server are protected using the CSK (e.g. the DPPK is the CSK).

- Multicast MCData signalling parameters and Data signaling payloads from server to client are protected using a MuSiK (e.g. the DPPK is a MuSiK).

- MCData signalling parameters and Data signaling payloads between servers are protected using the SPK (e.g. the DPPK is the SPK).

- MCData signalling parameters and Data signaling payloads between two off-network clients are protected using a PCK (e.g. the DPPK is the PCK).

- MCData signalling parameters and Data signaling payloads between a group of off-network clients are protected using a GMK (e.g. the DPPK is the GMK).

MCData Data payloads are protected as follows:

- MCData Data payloads end-to-end protected between two online clients are protected using a PCK (e.g. the DPPK is the PCK).

- MCData Data payloads end-to-end protected between two off-network clients are protected using a PCK (e.g. the DPPK is the PCK).

- MCData Data payloads end-to-end protected between a group of online clients are protected using a GMK distributed by a GMS (e.g. the DPPK is the GMK).

- MCData Data payloads end-to-end protected between a group of off-network clients are protected using a GMK distributed by a GMS (e.g. the DPPK is the GMK).

- MCData Data payloads are end-to-end authenticated based on SSK, PVT and KPAK distributed by a KMS.

Files when distributed using HTTP are protected as follows:

- Files end-to-end protected between two online clients when distributed using HTTP are protected using a PCK (e.g. the DPPK is the PCK).

- Files end-to-end protected between a group of online clients when distributed using HTTP are protected using a GMK distributed by a GMS (e.g. the DPPK is the GMK).

NOTE: The DPPK is not a new type of key, it describes how the MC system's existing key types are used to protect MCData. Consequently, there will be multiple DPPKs in the MC System depending on the communication channel. Furthermore, while a PCK and a GMK may both be used as a DPPK to protect MCData in different channels, the PCK and the GMK are not the same key.

8.3 One-to-one communications

The purpose of key management is to establish a MCData Payload Protection Key (DPPK) for the one-to-one communication channel between the pair of communicating clients. In the case of a one-to-one communication, the DPPK shall be the PCK. The PCK is used for end-to-end protection of one-to-one (private) SDS or FD data payloads.

The PCK and PCK-ID are distributed within the SIP message used to initiate the session.

The PCK and PCK-ID is distributed using service-specific signalling. For all MCData services, SIP signalling is used to establish or send the MCData communication. The PCK and PCK-ID is distributed within a MIKEY payload contained within the SDP offer sent from the initiator to the receiver in the same way as for MCPTT and MCVideo. The procedures for PCK distribution are defined within clause 5.6.

For off-network MCData operations, an MCData payload containing a MIKEY_SAKKE I-MESSAGE (clause 8.5.4.1) is used to distribute an MCData DPPK (PCK) from the initiating MCX client to the terminating MCX client.

This key distribution mechanism applies to the following messages defined in TS 23.282 [38]:

- MCData standalone data request

- MCData session data request

- MCData FD request

When required by the MCData service provider, protection shall be applied to the MCData Data payloads using the PCK. Payload authentication may also be applied. The mechanisms used to secure these payloads are described in clause 8.5.

Once the PCK is established between the source and destination, SDS and FD exchanges between this same source and destination may continue to use the same PCK for subsequent MCData communications by simply providing the PCK-ID in every SDS message.

8.4 Group communications

The purpose of key management is to establish a MCData Payload Protection Key (DPPK) for the group communication between the group of communicating clients. In the case of group communication, the DPPK shall be

the GMK. The GMK is distributed in the same way as for MCPTT and MCVideo group communications, as defined in clause 5.7.

When required by the MCData service provider, protection shall be applied to the MCData Data payloads using the GMK. Payload authentication may also be applied. The mechanisms used to secure these payloads are described in clause 8.5.

8.5 MCData payload protection

8.5.1 General

The protection applied to the MCData payload is indicated by the 'Message Type' of the MCData payload. If the payload is protected (encrypted and integrity protected), Bit '7' of the Message Type shall be '1' (otherwise it shall be '0'), if the payload is authenticated, Bit '8' of the Message Type shall be '1' (otherwise it shall be '0'). See Clause 15.2.2 of TS 24.282 [50].

The following protected (encrypted and integrity protected) payloads are defined for MCData SDS and file distribution:

- Protected SDS Signalling Payload.
- Protected FD Signalling Payload.
- Protected Data Payload.
- Protected SDS notification message.
- Protected FD notification message.
- Protected FD network notification message.
- Protected Communication release message.
- Protected binary data representing the file.

The following authenticated payloads are defined for MCData SDS and file distribution:

- Authenticated Data Payload.

The following authenticated and protected (encrypted and integrity protected) payloads are defined for MCData SDS and file distribution:

- Authenticated and Protected Data Payload.

In this case both the procedures for protecting a payload and authenticating a payload are applied

8.5.2 Prequisites

8.5.2.1 Prequisites for protected payloads

The prequisites for encryption and integrity protection of a protected payload is that the MC client(s) or MC server(s) have a shared MCData Payload Protection Key (DPPK). This shall be the CSK, SPK, MuSiK, GMK or PCK depending on the payload that will be protected. The DPPK will also have a shared key identifier, the DPPK-ID. This shall be the CSK-ID, MuSiK-ID, SPK-ID, GMK-ID or PCK-ID respectively, based upon the type of key used.

8.5.2.2 Prequisites for authenticated payloads

The prequisites for authentication of an authenticated payload is that the MC client will have been keyed (SSK, PVT and KPAK) by a KMS as defined in clause 5.3.

8.5.3 Key derivation for protected payloads

Before protecting an MCData payload, the DPPK is hashed through a KDF (similar to the process used for XML protection for application signalling), to produce a MCData Payload Cipher Key (DPCK). The KDF is defined in Annex F.1.5.

8.5.4 Payload protection

8.5.4.1 Format of protected payloads

All protected payloads shall have the format defined in table 8.5.4.1-1:

Table 8.5.4.1-1: MCData Protected Payload message content

Information Element	Type/Reference	Presence	Format	Length
Message Type	Message type	M	V	1
Date and Time	Date and Time of creation of protected payload message.	M	V	5
Payload ID	The identifier for the payload.	M	V	4
Payload sequence number	The sequence number of the protected payload.	M	V	1
Payload algorithm	See 8.5.4.2	M (NOTE 5)	V	1
Signalling algorithm	See 8.5.4.2	O (NOTE 6)	V	1
IV	Initialisation vector (or nonce) for message	M	V	16
DPPK-ID	Key identifier	M	V	4
Payload	Protected Payload (Ciphertext)	M	TLV-E	x
MIKEY_SAKKE I-MESSAGE	DPPK(PCK) encapsulated within MIKEY_SAKKE I-MESSAGE	O (NOTE 4)	TLV-E	x

Where ‘Payload' will be the encrypted and integrity-protected payload encoded in a binary format.

NOTE 1: Date and Time is included as plaintext to allow the MCData server to order end-to-end protected messages and assess whether end-to-end protected messages may have expired.

NOTE 2: Payload ID and Payload sequence number allow protected payloads to be split over multiple SIP messages.

NOTE 3: When file is distributed using HTTP, MCData Protected Payload message is distributed as part of protected FD Signalling Payload and the protected binary data representing the file is uploaded using HTTP.

NOTE 4 This information element applies only to off-network communications. It is optionally included, for example, when the originating client does not have an active PCK for the terminating client.

NOTE 5 This field applies to the protection of the data payload field.

NOTE 6 This field applies to the protection of the MCData signalling parameters field, Data signalling payload field, and End to end security parameters fields. This field defaults to DP_AES_128_GCM (as defined in clause 8.5.4.2) if not present.

8.5.4.2 Encryption of protected payloads

Protection of payloads shall support the following algorithms (cipher suites):

Table 8.5.4.2-1: DP_AES_128_GCM algorithm parameters

Parameter	Value/Reference
Algorithm ID	DP_AES_128_GCM
Cipher	AEAD_AES_128_GCM (as defined in RFC 5116 [43])
DPCK Key length	128 bits
IV length	128 bits
AEAD authentication tag length	128 bits

Table 8.5.4.2-2: DP_AES_256_GCM algorithm parameters

Parameter	Value/Reference
Algorithm ID	DP_AES_256_GCM
Cipher	AEAD_AES_256_GCM (as defined in RFC 5116 [43])
DPCK Key length	256 bits
IV length	256 bits
AEAD authentication tag length	128 bits

In using the above cipher suites as defined in RFC 5116 [43], the plaintext, P, shall be the full original plaintext payload. The associate data (AD) shall be the Message Type, Date and Time, Payload ID, Payload sequence number, Algorithm, IV, and DPPK-ID fields within the MCData Protected Payload message content defined in clause 8.5.4.1.

8.5.5 Payload authentication

Authenticated payloads shall have the format defined in table 8.5.5-1:

Table 8.5.5-1: MCData Authenticated Payload message content

Information Element	Type/Reference	Presence	Format	Length
Original Payload	Original Payload (unchanged)	M	TLV-E	x
Algorithm	Algorithm used to sign the message	M	V	1
Signing Data	Signature data	M	TLV-E	x
Signature	Based on algorithm	M	TLV-E	x

The signature shall be on the entire payload excluding the value of the signature element. However, the type and length of the signature element shall be included in the signature. The signature value shall be encoded in binary format.

The ECCSI signature algorithm as defined in RFC 6507 [9] shall be supported by MC clients.

The contents of the Signing Data field is determined by the signature algorithm. For ECCSI, the signing data shall be as defined in table 8.5.5-2:

Table 8.5.5-2: ECCSI Signing Data content

Information Element	Type/Reference	Presence	Format	Length
Signing UID	Signers UID as defined in Annex F.2.1	M	TLV	x
Signing KMS	Signer's KMS URI	M	TLV	x

After signature vertification, the verifier shall extract the sender's URI from elsewhere in the message and check that this corresponds to the UID contained in the Signing UID field above. If not, signature verification shall have failed.

9 Signalling protection

9.1 General

Signalling between entities in the MC System are defined as:

- RTCP signalling (e.g. floor control),
- XML signalling (within SIP messages) , or
- MCData Data signalling (withing SIP or MSRP messages).

To allow this signalling to be protected, key distribution mechanisms are required to distribute the associated keys.

For protecting signalling between the client and the server, there are two key distribution mechanisms:

- 'CSK upload' procedure (as defined in clause 5.4).
- 'Key download' procedure (as defined in clause 5.8).

For protecting signalling between MCX Servers, there is one key distribution mechanism:

- manual SPK configuration (as defined in clause 5.5).

9.2 Key distribution for signalling protection

9.2.1 Client-Server Key (CSK)

9.2.1.1 General

A Client-Server Key is required to protect unicast RTCP signalling between the MC client and the MCX Server. The use of the CSK in this context is defined in clause 9.4.

Additionally, the MC Service provider may require that MC identities, access tokens and other sensitive information transferred between clients and MC domain on the SIP-1 and SIP-2 interfaces be protected at the application layer from any viewing, including protection from viewing at the SIP signalling layer. Symmetric key based protection of SIP payload using CSK may be used to satisfy this requirement. The use of CSK in this context is defined in clause 9.3.

The uses of the CSK are shown in Figure 9.2.1-1.

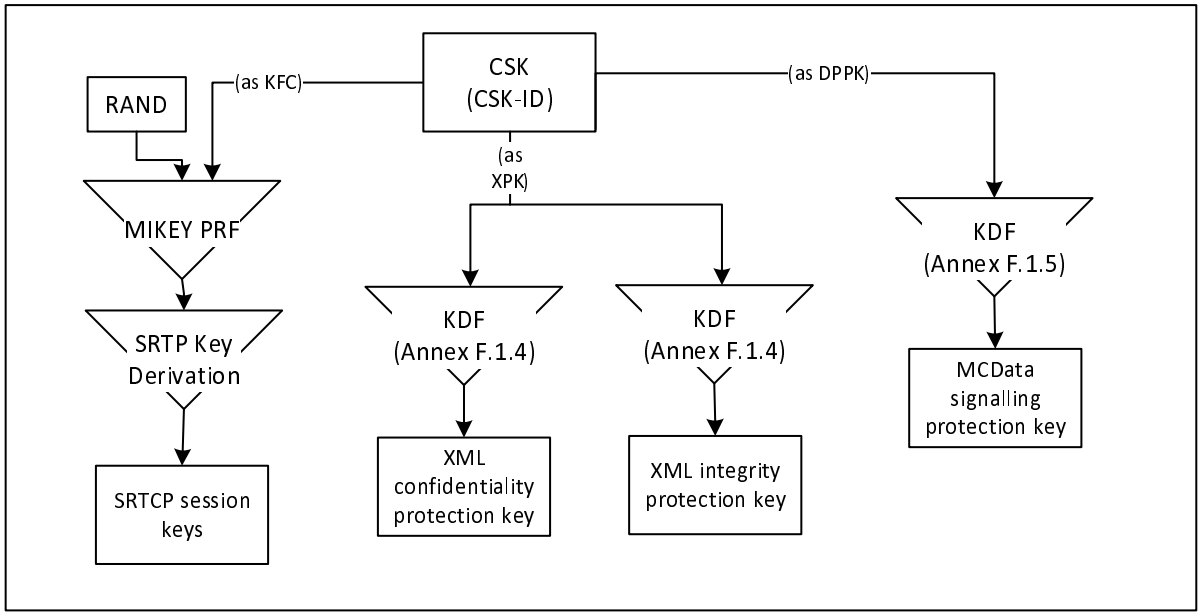

Figure 9.2.1-1: Uses of the Client-Server Key

9.2.1.2 Creation of the CSK

The 128-bit CSK is initially generated by the client and provided encrypted to the server through the SIP interface along with the CSK-ID identifying the CSK.

The key remains in use until: a new CSK is required, the SIP session is torn down, the MC user logs off, or some other indication. If during the active SIP session an update of the CSK is required, the server generates a CSK and provides it to the client using the mechanism defined in clause 5.8.

9.2.1.3 Initial 'CSK Upload' Procedure

The CSK is initially distributed via the 'CSK upload' procedure as defined in clause 5.4. The 'CSK upload' procedure creates a security association between the MC client and the MCX Server and occurs during the client's initial connection with the MC Server.

The following steps describe how the client obtains the user specific key material and securely transfers the CSK to a server within the MC domain.

Prior to beginning of this procedure, the client would have obtained user-specific key material from the KMS.

1) The client randomly generates the CSK and encapsulates the CSK as described in clause 5.4.

2) The client includes the encapsulated CSK in its initial SIP REGISTER or in a SIP PUBLISH message that is used to perform the MC user authorization procedure, and sends the SIP message addressed to the PSI of the server.

An illustration is provided below as an example of how this message in included in the body of the SIP REGISTER message. The MIME media type "application/mikey" IETF RFC 3830 [22] is used in this example to insert a MIKEY I_MESSAGE in the SIP payload:

EXAMPLE:

```
REGISTER sip:MCPTT_Server_PSI SIP/2.0
Via: SIP/2.0/UDP den3.level3.com
Max-Forwards:70
From: MCPTT client IMPU
To:
Call-ID: <>
```

```
CSeq:  1 REGISTER
Contact: <URI>
Content-Type: multipart/mixed;boundary="boundary1"
Content-Length: 619

--boundary1
Content-Type: application/mikey
MIKEY I_MESSAGE
--boundary1
Content-Type: application/...
Encrypted Access token, MCPTT ID
--boundary1—
```

The following steps describe how the MCX Server retrieves the CSK from the SIP message:

1) The server receives the SIP message and decrypts the encapsulated the CSK as described in clause 5.4.

2) Once the CSK has been extracted, MC user specific information (e.g. the access token) protected in the SIP message as defined in clause 9.3.4, may be decrypted.

9.2.1.4 CSK update via 'key download'

The MCX Server may decide to update an existing CSK at any time. This may be due to CSK revocation or expiry.

The CSK shall be updated by the MCX Server using the 'key download' procedure, defined in clause 5.8. Upon receipt of a CSK via a 'key download' procedure, the MC client shall identify the type of key as a CSK via the 4 most significant bits of the CSK-ID. The MC client shall:

- discard any previous CSKs associated with the MC Server FQDN, and
- use the new CSK for uplink signaling with the MC Server.

9.2.2 Multicast Signalling Key (MuSiK)

The Multicast Signalling Key (MuSiK) is required to protect multicast RTCP signalling from the MCX Server to the MC client. This includes MBMS floor control, media control and transmission control messages.

The MuSiK shall be distributed using the ‘key download' procedure.

A ‘key download' procedure is described in clause 5.8.

The use of the MuSiK is shown in Figure 9.2.2-1.

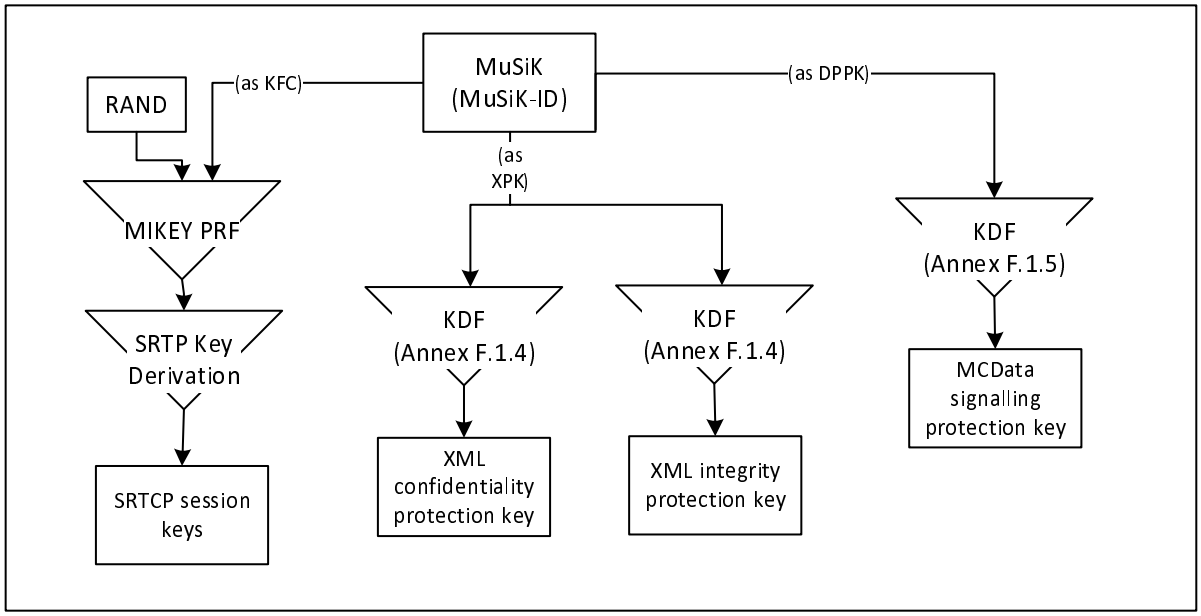

Figure 9.2.2-1: Uses of the Multicast Signalling Key (MuSiK)

The MCX Server distributes the Multicast Signalling Key (MuSiK) to a client when:

- The MCX Server requires protected signalling over the MBMS bearer to the MC client. In this case, an initial MuSiK ($MuSiK_{All}$) is distributed to the client using the key download procedure. By default, this MuSiK is used to protect all multicast signalling excluding bearer announcement messages.

- The MCX Server requires the transmission of group-related signalling (e.g. media control or floor control) over an MBMS bearer to the MC client, and the group configuration indicates that cryptographic protection is required for multicast group signalling. In this case, a new MuSiK ($MuSiK_{GRP}$) is created, assigned to the group and distributed to Group clients using the key download procedure. - The MCX Server requires an existing MuSiK to be replaced. This may be due to revocation or expiry.

- A participating UE (MC client) of the multicast group roams into the MBMS bearer coverage area.

NOTE: The participating MCX Server could use a single MuSiK for its MCX Groups and MBMS bearers. Where a MCX Group or MBMS bearer has privacy requirements, these procedures allow a new MuSiK to be distributed specifically for that purpose. A new MuSiK may not need to be distributed before each new bearer is established.

Upon receipt of a MuSiK the MC client shall store the MuSiK and MuSiK-ID. Should the MuSiK be rejected by the MC client, the MCX Server shall only use a unicast bearer when distributing signalling to the MC client.

Upon receipt of group-related signalling (e.g. media control or floor control) in the form of multicast SRTCP, the MC client shall inspect the MKI of the SRTCP packet which shall contain the MuSiK-ID. The MuSiK-ID shall be used to lookup the correct MuSiK for decrypting the SRTCP packet.Upon receipt of multicast MCData Data Signalling payloads, the MC client shall inspect the DPPK-ID element of the payload and extract the MuSiK-ID. The MuSiK-ID shall be used to lookup the correct MuSiK for decrypting the payload.

9.2.3 Signalling Protection Key (SPK)

The SPK is used to protect communications between MCX Servers. The SPK is distributed as defined in clause 5.5. The uses of the SPK for inter-server protection are shown in Figure 9.2.3-1.

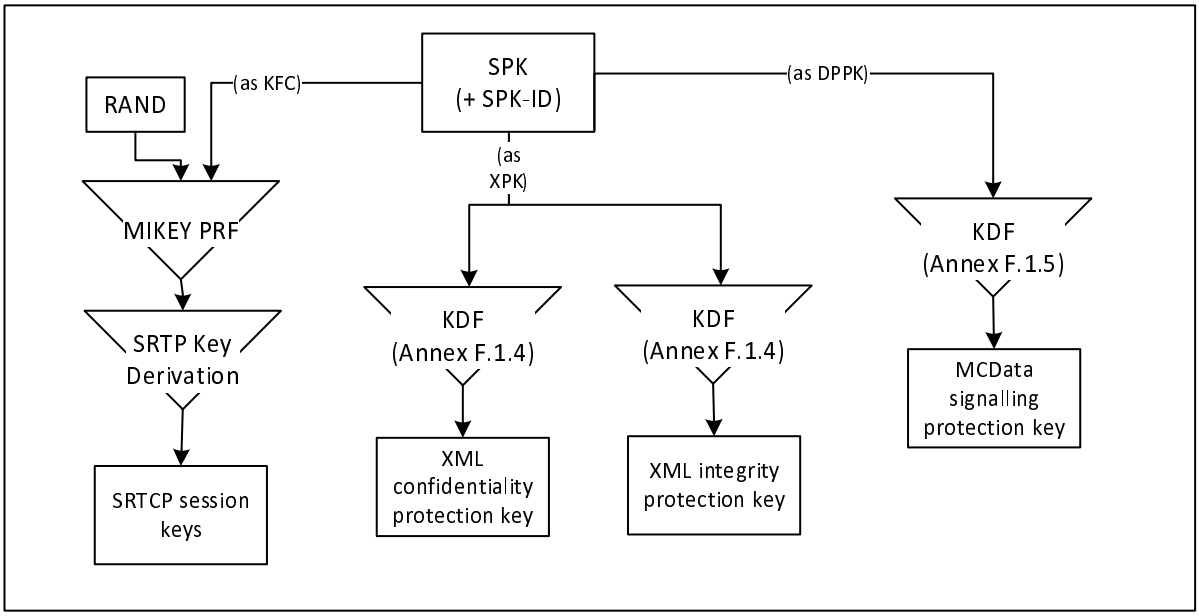

Figure 9.2.3-1: Uses of the Signalling Protection Key

9.3 Application signalling security (XML protection)

9.3.1 General

Certain values, keys and identifiers transferred in XML between a server in the MC domain and client may be treated as sensitive by public safety users and may require protection. To protect these values from all other entities outside of the MC Domain, this clause defines an optional mechanism to provide confidentiality protection on these values using XML encryption. Additionally, as some public safety users may require integrity protection on transmitted content, this clause defines an optional mechanism to provide integrity protection using XML signatures.

NOTE 1: The protection mechanism specified in this clause is for public-safety use only.

NOTE 2: The introduction of XML security mechanisms increases the size of the XML document. Consideration should be given to the impact of this size increase.

Editor's Note: It needs to be confirmed that the virtual proxy techniques being studied in SA3-LI (LIV8 S8HR study) can be extended to control use of MCPTT encryption in VPLMN roaming scenarios.

9.3.2 Protected content

Confidentiality protection may be applied to the entire XML document or to the following individual identifiers and values:

- MCX service user ID (e.g. MCPTT ID, MCData ID, MCVideo ID).
- MCX Group ID.
- User location information.
- Alerts.
- Access token.
- KMS provisioned key material.
- Functional aliases.

NOTE 1: The use of functional aliases for mission critical communications is defined in clause 5.9a of 22.280 [47] and may be included as part of MCPTT communications call setup and signaling as described in 23.379 [2].

Where confidentiality protection is applied to the entire XML document, the 'type' of message shall be clearly stated within the EncryptedData payload. The name shall reflect the names used in the message flows defined in TS 23.379 [2], TS 23.280 [36], TS 23.281 [37] and TS 23.282 [38]. This will allow the serving network to understand how their network is being used.

NOTE 2: Where the MCPTT Server is supporting legacy clients, these clients may not support confidentiality protection of the entire XML document. In this case, only individual identifiers and values should be confidentiality protected.

Integrity protection may be applied to the entire XML document, and to individual KMS certificates.

9.3.3 Key agreement

The confidentiality and integrity protection mechanisms defined in F.1.4 rely on a shared XML Protection Key (XPK) to be able to encrypt and sign XML.

For connections between the client and the MC Domain, the XPK shall be the 128-bit shared Client-Server Key (CSK) established as defined in clause 9.2.1. The XPK-ID shall be the CSK-ID.

For connections between servers inside and across MC Domains the XPK shall be the 128-bit manually provisioned Signalling Protection Key (SPK) established as defined in clause 9.2.3. The XPK-ID shall be the SPK-ID

For connections between the KMS and the MC KM client (as described in clause 5.3.3), confidentiality protection shall use the 256-bit TrK as the XPK and the TrK-ID as the XPK-ID. Integrity protection shall use the InK as the XPK and the XPK-ID shall be the InK-ID.

The integrity key and confidentiality key for application data protection shall be derived from the XPK as defined in annex F.1.4. The XPK-ID may be listed in the XML to aid decryption.

9.3.4 Confidentiality protection using XML encryption (xmlenc)

9.3.4.1 General

This clause defines an optional mechanism to allow specific XML content within the XML elements and XML URI attributes to be encrypted between the client and the server.

NOTE: Only encryption of XML simple content within XML elements and XML URI attributes is supported. Encryption of XML tags is not supported.

9.3.4.2 XML content encryption

XML content within XML elements is encrypted as defined by XML Encryption Syntax, Version 1.1 [27].

To encrypt content within a specific XML element, the content shall be replaced with the <EncryptedData> element. The <EncryptedData> element shall contain a <CipherData> element, containing a <CipherValue> element containing the encrypted content. Encryption shall be performed as defined in [27] using the CSK as the cipher key.

Where protecting content, the <EncryptedData> element may:

- Use the 'Type' attribute specifying that content is encrypted ('http://www.w3.org/2001/04/xmlenc#Content').
- Contain <KeyData><KeyInfo> element containing the base64 encoded XPK-ID.
- Contain <EncryptionMethod> element listing the encryption algorithm used for encrypting the XML content. The AES-128-GCM algorithm shall be supported, as identified by the algorithm identifier: 'http://www.w3.org/2009/xmlenc11#aes128-gcm'.

Where protecting key material, the <EncryptedData> element may:

- Use the 'Type' attribute specifying that content is encrypted ('http://www.w3.org/2001/04/xmlenc# EncryptedKey').
- Contain <KeyData><KeyInfo> element containing the base64 encoded XPK-ID.
- Contain <EncryptionMethod> element listing the encryption algorithm used for encrypting the XML key material. The AES-256 key wrap algorithm as defined in RFC 3394 [34] shall be supported, as identified by the algorithm identifier 'http://www.w3.org/2001/04/xmlenc#kw-aes256'.

Where these elements do not occur, the information they contain shall be known to both the client and server in the MC domain through other means.

The following is an example of unprotected XML content:

EXAMPLE:

```
<ExampleTag xsd:type="Normal">
  sensitive.data@example.org
</ExampleTag>
```

When XML encryption is applied, the following is an example of the encrypted content:

EXAMPLE:

```
<ExampleTag xsd:type="Encrypted">
  <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
   Type='http://www.w3.org/2001/04/xmlenc#Content'>
    <EncryptionMethod Algorithm="http://www.w3.org/2009/xmlenc11#aes128-gcm"/>
    <ds:KeyInfo>
      <ds:KeyName>base64XpkId</KeyName>
    </ds:KeyInfo>
    <CipherData>
      <CipherValue>A23B45C56</CipherValue>
    </CipherData>
  </EncryptedData>
</ExampleTag>
```

9.3.4.3 XML URI attribute encryption

XML attribute encryption shall be performed by encrypting the URI and embeddeding the encrypted ciphertext within a new URI. The appended domain name of the new URI identifies the attribute as having confidentiality protection. Encryption shall be performed using the AES-128-GCM [42], as the encryption algorithm, XPK as the key, and the use of a 96 bit randomly selected IV.

The output URI is structured to contain:

- the base64 encoded encrypted URI;
- the string ";iv=" followed by the base64 encoded 96-bit random initialisation vector (IV) which is used by the AES-128 encryption algorithm (as described in TS 33.203 subclause 6.4).
- the string ";key-id=" followed by the base64 encoded encryption key identifier (XPK-ID);
- the string ";alg=" followed by the encryption algorithm identifier (128-bit encryption algorithm "128-AES-GCM");
- the appended domain name of the new URI e.g. “@mc1-encryption.3gppnetwork.org”.

An example of the resultant sip-uri after encryption is:

sip:98yudFG45tx_89TYGedb4ujF;iv=FGD567kjhfH7d4-D;key-id=eV9kl7;alg=128-aes-gcm@mc1-encryption.3gppnetwork.org

The following is an example of unprotected XML URI content within XML attributes:

EXAMPLE:

```
Content-Type: application/pidf+xml
<?xml version="1.0" encoding="UTF-8"?>
```

```
<presence entity="sip:somebody@mcptt.org">
  <tuple id="acD4rhU87bK">
    <status>
      <affiliation group="sip:thegroup@mcptt.org" />
    </status>
  </tuple>
</presence>
```

When XML URI attribute encryption is applied, the following is an example of encrypted URIs within XML attributes:

EXAMPLE:

```
Content-Type: application/pidf+xml
<?xml version="1.0" encoding="UTF-8"?>
<presence entity="sip:c4Hrt45XG8IohRFT67vfdr3V;iv=45RtfVgHY23k8Ihy;key-id=b7UJv9;alg=128-aes-
gcm@mc1-encryption.3gppnetwork.org">
  <tuple id="acD4rhU87bK">
    <status>
      <affiliation group="sip:98yudFG45tx_89TYGedb4ujF ;iv=FGD567kjhfH7d4-D;key-id=eV9kl7;alg=128-
aes-gcm@mc1-encryption.3gppnetwork.org " />
    </status>
  </tuple>
</presence>
```

9.3.5 Integrity protection using XML signature (xmlsig)

Where integrity protection is required, an XML HMAC signature may be applied using the XPK to key the HMAC to a whole XML MIME body.

The XML HMAC signature mechanism is specified by W3C [28]. The HMAC-SHA256 signature method shall be supported.

When integrity protection is enabled, all XML MIME bodies transported in SIP requests and responses are integrity protected. If one or more of the XML MIME bodies are included in a SIP request or SIP response, then a MIME body is included in the SIP request or SIP response containing one or more signatures pointing to those XML MIME bodies as illustrated in the figure 9.3.5-1.

In order to integrity protect the XML MIME bodies in SIP requests and SIP responses, the MC client and MCX server shall for each MIME body, include the Content-ID header field as specified in IETF RFC 2045 [40] containing a Content-ID ("cid") Uniform Resource Locator (URL) as specified in IETF RFC 2392 [41].

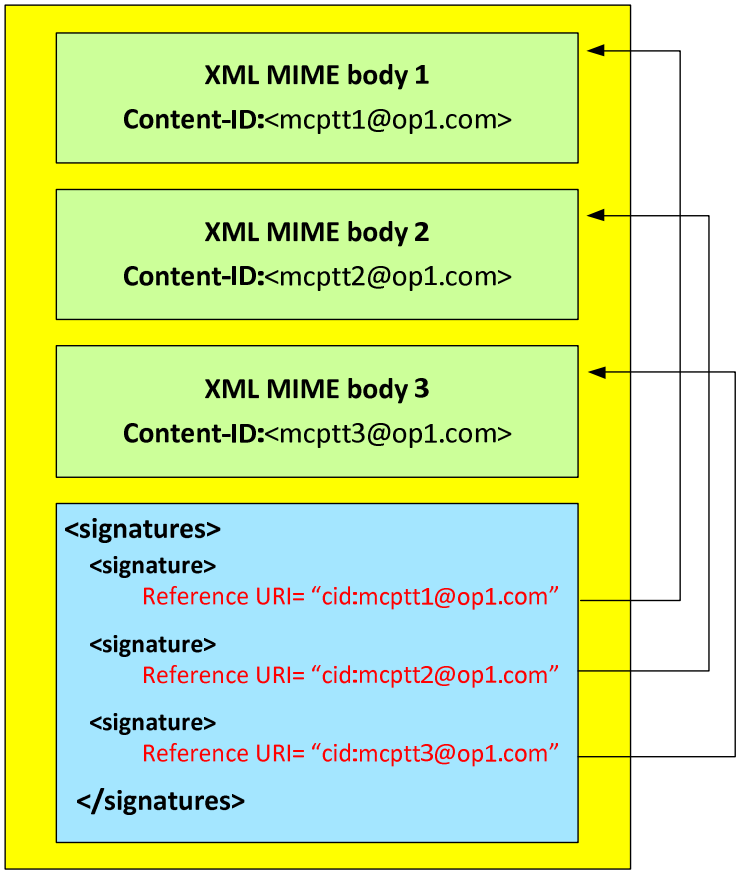

Figure 9.3.5-1: Integrity Protection of XML MIME bodies in SIP requests and SIP responses

Each MIME body that is integrity protected is assigned a unique signature contained in a <Signature> element.

The <Signature> element shall contain the following child element:

- <SignatureValue> HMAC signature of the content

The <Signature> element may contain the following child elements:

- <CanonicalizationMethod> element listing an appropriate algorithm.
- <SignatureMethod> element listing an appropriate algorithm. HMAC-SHA256 shall be supported for signatures.
- <KeyInfo><KeyName> element containing the base64 encoded XPK-ID.
- <Reference> element containing a URI identifying the content to be signed and the method for hashing the content. SHA-256 shall be supported for hashing content.

Where these elements do not occur, the information they contain shall be known to both the client and server in the MC domain through other means.

9.4 RTCP signalling protection (SRTCP)

9.4.1 General

RTCP encryption is required between the MC UE and MCX Server and between a pair of MCX Servers. RTCP is protected hop-by-hop, meaning that RTCP is always decrypted by the MCX server and then re-encrypted to its destination.

The following signalling uses RTCP and is protected using the procedures in this clause:

- MCPTT floor control signalling (MBCP or TBCP).

    - Unicast uplink and downlink (online), multicast downlink (online) and off-network transmission.
- MCVideo transmission control (online/off-network).
- Unicast uplink and downlink (online), multicast downlink (online) and off-network transmission.
- MCPTT/MCVideo media signalling.
- Unicast uplink and downlink (online), multicast downlink (online) and off-network transmission.
- MBMS subchannel control signalling (from MCX Server to MC UE).
    - multicast downlink (online).

All RTCP (floor control, media control and MBMS subchannel control signalling) is protected in the same way. RTCP is protected using SRTCP. The master key for SRTCP is derived from a Key For Control signalling (KFC). The KFC is shared between the transmitter and receiver(s) prior to distribution of the SRTCP packets. A 32-bit identifier for the key (KFC-ID) and a 128-bit random value (KFC-RAND) is also established.

There are a number of key distribution mechanisms for establishing the KFC based on the interface over which RTCP is being transmitted.

9.4.2 Unicast RTCP protection between client and server

In Clause 9.2.1, a Client-Server Key (CSK) is generated and shared between the MC client and MCX Server along with the CSK identifier (CSK-ID). For floor and media control, the KFC shall be the CSK and the KFC-ID shall be the CSK-ID. KFC-RAND shall be the MIKEY RAND value transmitted in the MIKEY message used to distribute the CSK.

9.4.3 Multicast RTCP protection between client and server

In clause 9.2.2, a Multicast Signalling Key (MuSiK) is generated and shared from the MCX Server to the MC client, along with the MuSiK identifier (MuSiK-ID). For the protection of multicast floor and media control, the KFC shall be the MuSiK and the KFC-ID shall be the MuSiK-ID. KFC-RAND shall be the MIKEY RAND value transmitted in the MIKEY message used to distribute the MuSiK.

To support multicast signalling protection, the MSCCK and the legacy MKFCs may also be used for this purpose as defined in Annex H.

9.4.4 Off-network floor and transmission control protection

Off-network, the KFC is the PCK (for private communications) or the GMK (for group communications) as described in clause 7.3.4, and the KFC-ID is the PCK-ID or GMK-ID (respectively).

9.4.5 RTCP protection between servers

In Clause 9.2.3, a Signalling Protection Key (SPK) is shared between MCX Servers along with a SPK-ID. For floor and media control signalling transferred between MCX Servers, the KFC shall be the SPK, the KFC-ID shall be the SPK-ID and the KFC-RAND shall be the SPK-RAND.

9.4.6 Key derivation for SRTCP

As a result of the key agreement process, the entities (MCX client and server, or MCX servers) shall share a KFC, a KFC-ID and a KFC-RAND. The KFC shall be used as the MIKEY Traffic Generating Key (TGK), the KFC-ID shall be used as the MIKEY CSB ID and the KFC-RAND shall be used as the MIKEY RAND value. The MIKEY CS-ID shall be set as defined in table E.1.3-1. These shall be used to generate the SRTCP Master Key and SRTCP Master Salt as specified in IETF RFC 3830 [22]. The key derivation function defined in section 4.1.4 of IETF RFC 3830 [22] using the PRF-HMAC-SHA-256 Pseudo-Random Function described in section 6.1 of IETF RFC 6043 [25], shall be supported for generating the SRTCP Master Key and Salt. SRTCP session keys are generated from the SRTCP Master Key and Salt as defined in clause 9.4.8.

NOTE: Within RFC 3830 [22], the SRTCP Master Key and SRTCP Master Salt are referred to as the SRTP Master Key and the SRTP Master salt respectively.

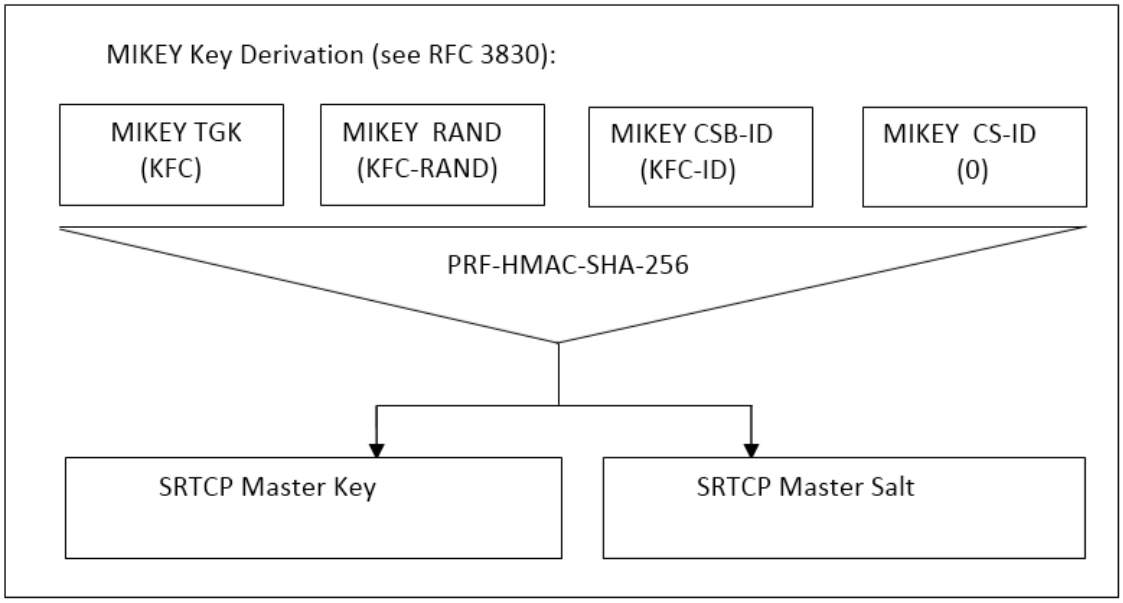

Figure 9.4.6-1: Key derivation for on-network floor and media control protection

To identify the security context from the SRTCP stream a SRTCP Master Key Identifier (MKI) is required. The MKI shall be the 32-bit KFC-ID.

9.4.7 Security procedures for transmission of RTCP content

After key establishment, RTCP protection does not require any signalling mechanism to convey information. The RTCP is protected within an SRTCP packet. The information necessary for decryption is provided within each SRTCP Packet.

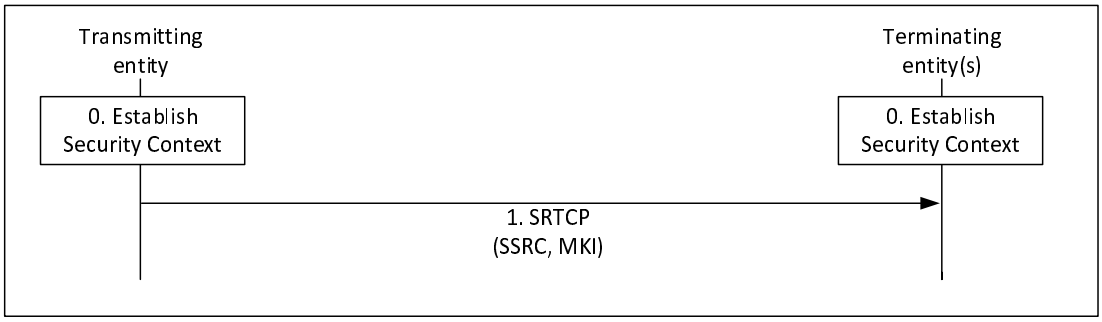

Figure 9.4.7-1: Security procedure for media stream protection

Figure 9.4.7-1 shows the security mechanism.

0) Prior to beginning this procedure the MC entities (MC UEs and/or MCX Server) involved in the communication shall have established a security context for SRTCP (Master Key, Master Salt, MKI).

1) The transmitting entity (MC UE or MCX Server) shall send SRTCP packets using the format described in IETF RFC 3711 [13]. The packet shall include a Master Key Identifier (MKI) field which contains the information required to locate the Master Key and Master Salt. On receipt of a SRTCP packet, a terminating entity (MC UE or MCX Server) shall use the contents of the MKI to look up the appropriate Master Key and Salt and generate the appropriate SRTCP session key and salt if it satisfies the key derivation rate criteria as specified in IETF RFC 3711 [13].

NOTE 1: Assuming entities have been keyed/pre-provisioned at some point in the past, this security mechanism is entirely stateless.

NOTE 2: The receiver does not need to generate an appropriate SRTCP session key and salt each time it receives a SRTCP packet. The key derivation rate defined in IETF RFC 3711 [13] determines the session key generation frequency. Refer to RFC 3711 [13] for more information.

A diagram of the SRTCP packet format is within figure 9.4.7-2.

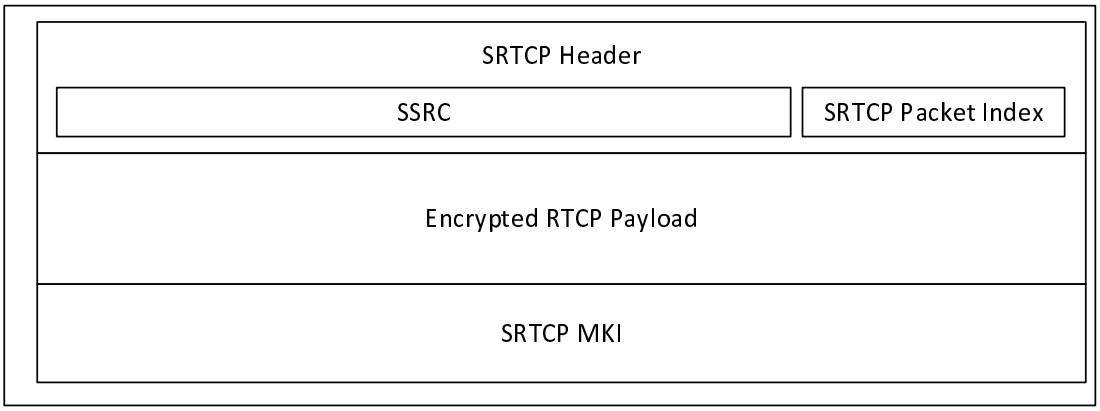

Figure 9.4.7-2: SRTCP packet format showing security parameters

The length of the MKI is determined by the key distribution mechanism.

9.4.8 RTCP protection profile

Integrity and confidentiality protection for communications using RTCP for floor control, transmission control, and media control is achieved using SRTCP, as defined in IETF RFC 3711 [13]. The mechanism described in IETF RFC 3711 [13] is used to encrypt the RTCP payload. A diagram of the key derivation mechanism (as described in IETF RFC 3711 [13]) is shown in figure 9.4.8-1.

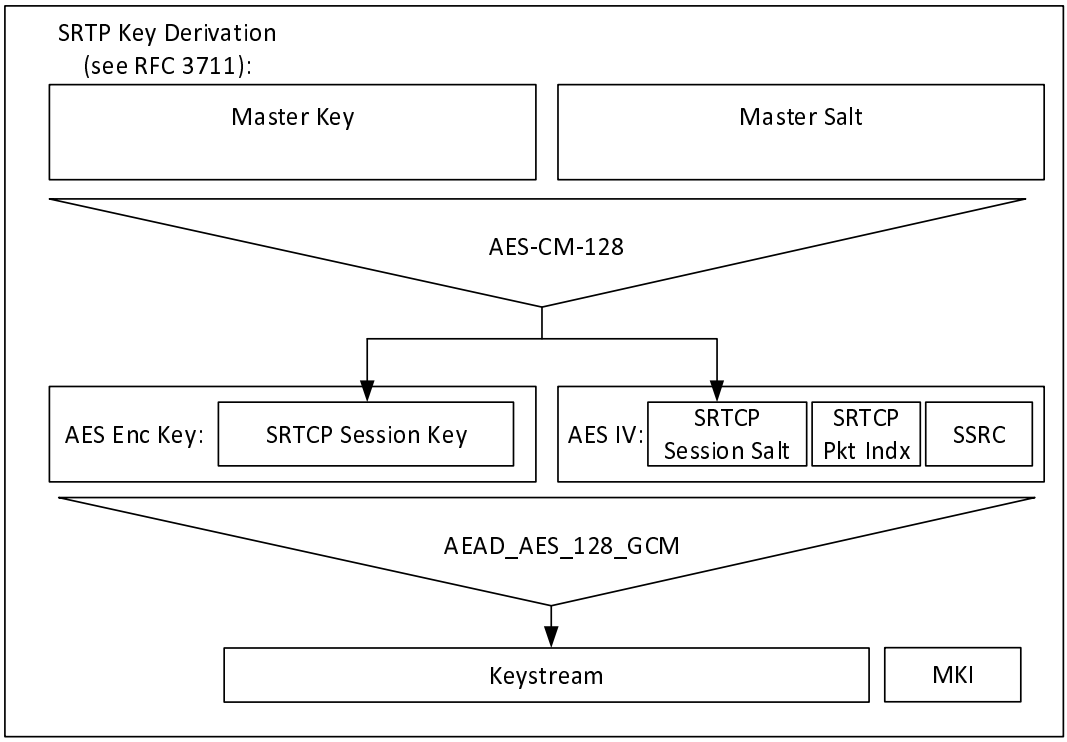

Figure 9.4.8-1: Security mechanism for floor control, transmission control, and media control protection

The AES-CM-128 algorithm as defined in IETF RFC 3711 [13] shall be supported as the SRTCP PRF (which is used to derive the SRTCP session key and salt). A SRTP key derivation rate of 0 shall be used to indicate that session keys and

salts shall not be refreshed. The AEAD_AES_128_GCM algorithm as defined in IETF RFC 7714 [26] shall be supported for providing confidentiality and data authentication of SRTCP packets. The AEAD_AES_128_GCM algorithm requires that the SRTCP session key is 16 octets in length and the session salt is 12 octets in length.

NOTE: Some SRTCP implementations are not compliant with RFC 3711 due to the size of the SRTCP index, as discussed in RFC 3711 Errata ID 3712 [51].

9.5 MCData signalling protection

9.5.1 Key distribution for signalling protection

Where MCData signalling parameters or MCData Data signalling payload protection is required, key distribution and key use for MCData signalling is equivalent to MCPTT and MCVideo. MCData signalling parameters or MCData Data signalling payload protection is defined in subclause 8.2.

The procedures for CSK distribution are defined in clause 9.2.1. The procudures for MuSiK distribution are defined in clause 9.2.2. The procedures for SPK distribution are defined in clause 9.2.3.

9.5.2 Protection of MCData application signalling payloads (XML)

Protection of MCData application signalling payloads, specifically XML content within SIP messages, is defined in clause 9.3. For the protection of MCData signalling, the XPK shall be the DPPK.

9.5.3 Protection of MCData signalling payloads

Protection of MCData Data signalling payloads is defined in clause 8.5.

9.6 Message origin authentication and authorisation

9.6.1 General

The MC System allows authorised service requests where the 'requester' may cause modification to the operation of the MC Domain service or initiate an action on a target MC client, potentially without that client's permission.

Clauses 9.3 and 9.5 describe how on-network application signalling is protected hop-by-hop within the MC System. In an MC Domain which is interconnected or supporting migration, application signalling may pass through multiple MCX Servers from the requester to the target client. Using hop-by-hop signalling security alone, the target's MCX Server and the target's client would be unable to authenticate the identity of the requester, or whether the requester has permission to perform the action. This leaves the MC domain and MC client open to attack via misuse of signalling requests along the signalling path.

For example:

a) A 'Group Affiliation Status Update' could originate from a MCX Server in a different domain to the Group Management Server. In this case the GMS requires information to ensure that the MCX Server has permission to modify the affiliation status of the requested user.

b) An 'Ambient Listening Request' could originate from a different domain to the target user. The target's MCPTT Server requires information to ensure that the originator is permitted to make the request.

c) An off-network 'Call Setup Request'. The target client requires information to ensure that the originator is permitted to make the request prior to responding to the request.

The subsequent clauses define an optional Element for Authenticating Requests (EAR), which is a signed element which may be attached to signalling requests across the MC System, both on and off-network. Where authorisation information is required to support the request, the requesting MC entity may use an Authorised Identity to sign the request.

The EAR allows an MC client or MC network entity to verify the origin, target and purpose of a signalling request, and that the origin is authorised to make such a request. In the network, authorisation is given by the profiles within the Configuration Management Server (CMS) and enforced by the MCX Server. The EAR allows authorisation to also be verified at the client, based on the contents of the EAR. EAR authorisations are provided by the the IdM to MC user (via the KMS).

With the EAR mechanism enabled, permission to use privileged signalling (e.g. Ambient Listening) needs to be enabled by both the IdM and the CMS. This allows a duel-check approach where the requesting user's permission is verified by both the network (based on the user's profile in the CMS) and the target client (based on the requesting user's Authorised Identity).

9.6.2 Origin authentication and authorisation in the MC System

9.6.2.1 Types of signalling

The purpose of authentication is to provide evidence to the receiver on the identity of the requester. The purpose of authorisation is to convey to the receiver that the requester has permission to take an action. In the MC System, MC client authentication and authorisation is primarily managed by the client's MCX Server. The MCX Server uses the access token and CSK to authenticate the user's MC client, and the user configuration to authorise the user's MC client. However, local authentication and authorisation may be insufficient where the user or signalling is moving across domains.

The additional authentication and authorisation mechanisms defined in Clause 9.6 may be attached to any signalling messages in the MC system, but in some specific cases, the mechanisms should be used. The following situations describe where the mechanisms defined in Clause 9.6 should be used in the MC system:

Case 1: Privileged signalling sent within the MC System (e.g. Ambient listening request): Authentication should be provided by an EAR using an authorised user identity (MC Service ID). This allows the target's MCX Server and MC client to assess whether the request is authorised.

NOTE 1: Privileged signalling is signalling which allows one client to remotely cause an intrusive action on a target client without the target user's permission.

Case 2: Signalling between network entities in separate MC domains (e.g. group call request from partner MCPTT server to primary MCPTT server). Authentication should be provided by an EAR using an authorised server/domain identity. This allows the receiving MC domain to confirm the requesting entity is a MCPTT server from a known partner MC domain.

Case 3: Signalling between a group client attached to a partner MC Domain and the Group Management Server. Authentication from the client to the server should be provided by an EAR using the user's identity (MC Service ID). Authentication from the server to the client should be provided by an EAR using an authorised server/domain identity.

NOTE 2: An authorised user identity is not required in this case as authorisation of the MC client is provided by the user configuration document.

Case 4: Signalling between the home network and a home client who has migrated to a partner MC Domain. Authentication from the client to the server should be provided by an EAR using the user's identity (MC Service ID). Authentication from the server to the client should be provided by an EAR using an authorised server/domain identity.

NOTE 3: An authorised user identity is not required in this case as authorisation of the MC client is provided by the user configuration document.

Case 5: Off-network signalling between MC clients. Authentication should be provided by an EAR using an authorised user identity (MC Service ID).

Where the request, containing a EAR, is routed via a MCX Server, the MCX Server should copy the EAR from the received request to the out-going request. Multiple EARs can be attached to a signalling message. For example, one EAR may be attached to authenticate the user making the request, and another may be attached to authenticate the domain sending the request on behalf of the user.

9.6.2.2 Privileged Signalling

All Privileged Signalling sent within the network should be authenticated and authorised using a EAR signed using an Authorised Identity (MC Service ID). The following are privileged signalling requests which should be explicitly authenticated and authorised:

- MCPTT Private call request in automatic commencement mode (TS 23.379).
- MCPTT Ambient listening call request (TS 23.379).
- MCPTT Remotely initiated MCPTT call request, in unnotified mode (TS 23.379).
- MCVideo Private call request (including private call, video pull and video push) in automatic commencement mode (TS 23.281).
- MCVideo Remote video push request in automatic commencement mode (TS 23.281).
- MCVideo Ambient viewing call request (TS 23.281).
- MCData standalone data request for application consumption (TS 23.282).
- MCData standalone session data request for application consumption (TS 23.282).
- MCData session data request for application consumption (TS 23.282).
- MCData group standalone data request for application consumption (TS 23.282).
- MCData group data request for application consumption (TS 23.282).
- MCData FD request with mandatory indication (TS 23.282).
- MCData group standalone FD request with mandatory indication (TS 23.282).

9.6.2.3 Signalling between network entities across domains

Where signalling is sent across domains, the servers used to send the signalling should be authenticated and authorised using a EAR signed using an Authorised Identity, indicating the server's role. within the MC domain. This ensures that only Group Management Servers are authorised to send group notifications, and only MCX servers are authorised to send key download messages. The following roles are used:

- MCPTT Server
- MCVideo Server
- MCData Server
- CS Proxy
- IS Proxy
- Group Management Server

The IS Proxy is authorised to authenticate the functions of a MCPTT Server, MCVideo Server or MCData Server towards another MC domain. The CS proxy is authorised to authenticate the functions of a MCPTT Server, MCVideo Server or MCData Server towards a MC client. Consequently, the addition of EARs may be performed at the edge of the MC domain.

9.6.2.4 Signalling between the GMS and the GMC

Where signalling is sent between the group management server and the group management client, the entity at each end should be authenticated and authorised using a EAR. The EAR from the GMS should be signed using an Authorised Identity, indicating the server is able to perform GMS functions. The GM client is not required to use an Authorised Identity. The following Group management messages should be explicitly authenticated and authorised:

- Subscribe Group Configuration Request

- Subscribe Group Configuration Response
- Notify Group Configuration Request
- Notify Group Configuration Response
- MC Group affliation request
- Group affliation status update

9.6.2.5 Signalling between the MC domain and a migrated user

Where signalling is sent out of the domain towards a migrated MC client, the servers used to send the outbound signalling should be authenticated and authorised using a EAR signed using an Authorised Identity, indicating the server's role. The server roles defined in Clause 9.6.2.3 shall be used.

The inbound signalling from the migrated client shall be authenticated and authorised using an EAR. The migrated MC client is not required to use an Authorised Identity.

This clause is applicable to all signalling sent to or from the migrated user.

9.6.2.6 Off-network signalling

All off-network signalling requests should be authenticated and authorised using a EAR signed using an Authorised Identity (MC Service ID). The client's role should be explicitly authorised. The following roles are used:

- MCPTT client
- MCVideo client
- MCData client

The client should be authorised to use any off-network functionality. Furthermore, the following are the off-network signalling requests which the client should be authorised to use:

- MCPTT Group call announcement (TS 23.379).
- MCPTT emergency alert announcement (TS 23.379).
- MCPTT Call setup request (TS 23.379).
- MCVideo Group communication announcement (TS 23.281).
- MCVideo emergency alert announcement (TS 23.281).
- MCVideo Private communication request (TS 23.281).
- MCVideo Capability request (TS 23.281).
- MCVideo Activity request (TS 23.281).
- MCData standalone data request (Clause 7.4.3.3.2, TS 23.282).
- MCData group standalone data request (Clause 7.4.3.4.2, TS 23.282).

9.6.3 Authorised Identities

9.6.3.1 Format of an Authorised Identity

Authorisation is conveyed using the MC entity's identity (e.g. MC Service ID) that is used to sign the EAR. This is known as the user/entity's Authorised Identity.

Within an Authorised Identity, authorisation information is contained within a SIP URI Header (known as an MC authorisation field). Authorisation fields are added to the entity's identity to provide information on the MC entity's

authorisations. Authorisation fields are name, value pairs. The value of the authorisation field provides a set of authorisations, formatted as a hexidecimal string. Authorisation fields are defined in Annex J.3.

9.6.3.2 Obtaining an Authorised Identity

Authorisation is originally requested and provided by the IdM. If authorisation is granted, the IdM provides the authorisation information within the scope of an access token. The scope values contained within the access token are defined in Clause J.3.3.

The access token is provided to the KMS as defined in Clause 5.1.

The KMS provisions the entity's keys to the entity as defined in Annex D. As part of the key provisioning process, the KMS may provision the entity keys for multiple SIP URIs that are associated with the entity.

If supported, where the scope of an access token contains one of the values defined in Clause J.3.3, the KMS provides multiple authorised identities which includes authorisation fields as part of the identity. Specifically, for each SIP URI associated with the entity, the KMS provisions:

- Key material for the identity: the entity's identity (e.g. MC Service ID).
- Key material for the Authorised Identity: the entity's identity with the applicable authorisation fields included.

The keyed entity may use either identity when signing within the MC System, (depending on whether it is configured to disclose its authorisations).

9.6.4 Element for Authenticating Requests (EARs)

9.6.4.1 Overview

When sending a sensitive signalling message, the requester creates the message as normal, then constructs and attaches an EAR to the message. The EAR contains the purpose and constraints of the request (type of request, restrictions on request, origin, destination) and is signed by the requester using the private signing key for the authorised identity. When used correctly, the EAR should provide a definitive record of the ‘request' that was made, including evidence that the request was not disproportionate.

9.6.4.2 The EAR information element

The EAR payload shall contain the following information elements:

Table 9.6.4.2-1: Element for Authenticating Requests Payload

Information element	Status	Description
Date/Time	M	The date/time of the request
EAR ID	M	A unique identifier for the request
Origin ID	M	The identity associated with the requester. This will be the MC Service ID of the user or the Server's URI. The identity may be an authorised identity, as defined in Clause 9.6.4.3.
Target ID	M	The identity associated with the intended receipient. This will be the MC Service ID of the user or the Server's URI.
Request Type	M	The type of request (e.g. Ambient Listening), including limitations to the request (e.g. maximum session length). Details of the Request Type IE are provided in Annex J.2.

The EAR payload shall be signed using the approach defined in Clause 8.5.5. The signed payload is attached to signalling messages as an EAR.

9.6.4.3 EAR authorisation

The contents of an EAR are defined in Clause 9.6.4.2.

To add authorisation information to the EAR, the identity associated with the requester shall be an Authorised Identity. The identity-based signature used to sign the EAR will be an Authorised Identity. The receiver processes the Authorised Identity to extract the requester's authorisations, and confirms that the type of request within the EAR is permitted given the requester's authorisations.

NOTE: The only entity that could create the signature, and hence the whole EAR, is an entity granted the means to sign using Authorised Identity. The KMS provides the means to sign using a specific identity. By doing so, the KMS has provided authorisation to create the EAR signature and authorise the specific action.

9.6.5 Security procedures for origin authentication

9.6.5.1 General

Signalling in the MC System may use Elements for Authenticating Requests (EARs) for communicating origin authentication to the receiving entities.

9.6.5.2 SIP signalling

9.6.5.2.1 General

A sender or processor of a SIP message may add an EAR to any SIP message to authenticate the origin of the message.

To add origin authentication to a SIP message, an application/vnd.3gpp.mc-signalling-ear MIME body is added to tbe message containing an EAR message as defined in Clause 9.6.4.2. Such requests are known as a "Origin authenticated SIP message".

A SIP message may contain multiple application/vnd.3gpp.mc-signalling-ear MIME bodies.

9.6.5.2.2 Group affiliation and deaffiliation signalling

Group affiliation signalling requires the EAR to be transferred by the MC Server from the client's request to the status update towards the GMS. The procedure is shown in Figure 9.6.5.2.2-1.

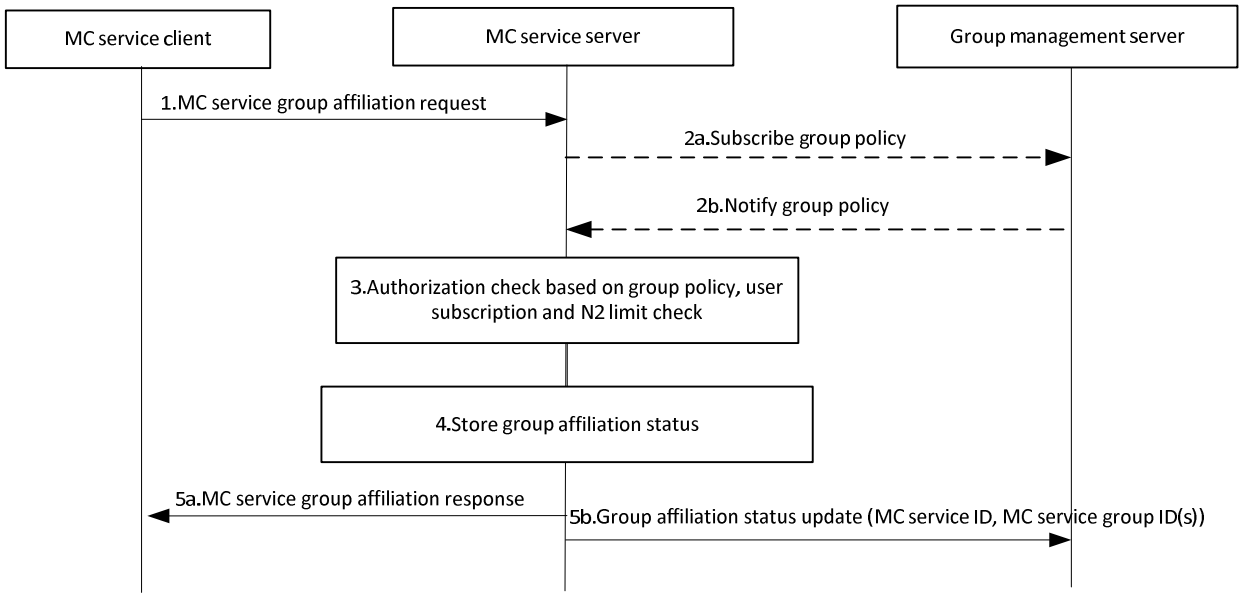

Figure 9.6.5.2.2-1: MC service group affiliation procedure

In this procedure, an EAR should be attached to Step 1 and passed to the MC service server. The MC service server should copy the client's EAR into the Group affiliation status update message, transmitted by the MC service server to the GMS in Step 5b. The MC service server may also add an additional EAR authenticating the MC service server itself

towards the GMS. On receipt of the EAR(s), the GMS has the necessary information to authorise the request to modify group affiliations.

The same procedures apply to the de-affiliation process. In this case, client EAR should be copied into the Group de-affiliation status update.

9.6.5.3 Off-network signalling

An EAR may be attached to any MONP signalling message to authenticate the origin of the message. The message then becomes an authenticated MONP message. Table 9.6.5.3-1 defines an authenticated MONP message.

Table 9.6.5.3-1: Authenticated MONP message

IEI	Information Element	Type/Reference	Presence	Format	Length
	Original MONP message	See Clause 15 in TS 24.379.	M	x	x
	Element for Authenticating Requests	EAR Annex J.1.2	O	TLV-E	3-x

9.6.5.4 Processing a received EAR

EARs may be processed by the receiver or routing network equipment to support an authentication and authorisation check on the signalling message. Clause 9.6.2 defines the types of requests where the EAR should be processed.

If supported by the receiver, upon receipt of a signalling message containing an EAR the receiver should:

1) Validate the signature on the EAR based on the provided UID.
2) Validate that the Target ID is associated with an apprpropriate user.
3) Check the date/time is within a recent window (e.g. 300 seconds) and the that a message with the EAR ID has not been already processed within that window.
4) Validate that the Origin ID of the EAR produces the UID.
5) If the request requires authorisation, extract the authorisation fields from the Origin ID. Validate that the Request Type is authorised by the authorisation fields, and that the KMS URI in the signature is permitted to authorise this type of request.
6) Verify that the Request Type corresponds to the SIP signalling message.
7) Verify that the Request Type parameters are not exceeded by the SIP signalling message.
8) Process the SIP message as normal.

If EARs are not supported by the receiver, the EAR shall be ignored.

10 Logging, Audit and Discreet Monitoring

10.1 Logging and audit of service metadata

10.1.1 Overview

The MC system should generate service metadata. This may include system events, management events, security events and user events. The full range of events that may be logged by the MC system is out-of-scope of the current specification. Furthermore, the mechanism that is used to audit MC system metadata is also out-of-scope of the current specification.

This clause defines the security and communications-related data associated to user events that are required to enable the audit of MC user actions within the MC system. User event logs are required to support discreet monitoring or audit.

To ensure the privacy of MC users' data, where this information is collected it shall be protected as defined in Clause 10.1.2.4.

10.1.2 User events

10.1.2.1 Types of events

When user events are collected within the MC System, the following events (based on the deployed MC services) are recorded

Class A: Common signalling events. Sending or receiving a common signalling message as defined in TS 23.280 [36].

Class B: MCPTT signalling events. Sending or receiving an MCPTT signalling message as defined in TS 23.379 [2].

Class C: MCVideo signalling events. Sending or receiving an MCVideo signalling message as defined in TS 23.281 [37].

Class D: MCData signalling events. Sending or receiving an MCData signalling message as defined in TS 23.282 [38].

Class E: Interworking signalling events. Sending or receiving an Interworking signalling message as defined in TS 23.283 [48].

10.1.2.2 Location of recording function

SIP events may be recorded at the CS and IS Proxies and/or MCX Servers. Depending on the configuration of the MC system, the SIP events may also be recorded within the SIP core.

HTTP events may be recorded at the HTTP Proxy(s).

10.1.2.3 Security content within user event logs

When a user event occurs, the following security-related information is required:

- The class and type of a user event
- IP addresses (source and destination)
- Signalling layer identifiers
  - SIP URI (source and destination).
  - HTTP target URL
- The initiating user or server
- The receiving user, group, server or [set of multicast users]
- Security parameters (if present in signalling):
  - MIKEY message
  - Access or Security Token
- Identifiers for related media bearers (if applicable).

NOTE: These logs are required to support disceet monitoring or audit of user content.

10.1.2.4 Protection of user event logs

User event logs need to be protected as they contain information that impacts the user's privacy. User event logs shall be encrypted and integrity protected while stored. Access to user event logs shall only be granted to authorised persons and such access shall be logged.

10.2 Audit and Discreet Monitoring of user content

10.2.1 Overview

Disceet Monitoring is access to user content at a network element within the MC Domain. Where Discreet Monitoring is used to access to user voice communications, it is known as Discreet Listening. Discreet Monitoring includes access to voice, video and data communications. For the purposes of this document, discreet monitoring and audit are equivalent processes. For discreet monitoring the access to media is in real-time. For audit, the access to media is at some point after the recording. The systems which support audit or discreet listening are out of scope of this document.

Disceet Monitoring and Audit are required functions of a public-safety network. For non-public safety services, these functions shall not be implemented in the network without explicit consent from all users of the MC system.

10.2.2 Collection of user media

It is assumed that collected Mission Critical media is held in its encrypted form within mass data storage. The storage solution is out-of-scope of this document.

User media is collected from the media paths within the MC Domain. It is expected that the encrypted media shall be collected at the media gateway into the MC system or by the MCX server .

Where SDS messages are routed within a signalling path, media will need to be extracted from within MCData signalling paths by the MC Domain. It is expected that the encrypted media routed over the signalling path shall be recorded by the CS and IS Proxies or by the MCData server.

To identify and process the collected and encrypted user media, user event logs associated with the media are required, as defined in Clause 10.1.2.

10.2.3 Storing of user media

User media in the MC System is end-to-end encrypted by default. Consequently, media can be recorded without modificiation and without additional protection. Media should be recorded alongside:

- a unique identifier for the collected media
- a unique identifier for a user event (with which the media is associated).

NOTE: Collected associated user event metadata may or may not be stored with the media.

10.2.4 Decryption of user media

To decrypt a specific target user's media for audit or discreet listening at a specific time 'T', the following process should be used for decryption of user media. The controlling entity shall be either the KMS or a secured logging device.

1. The auditor obtains the target user's key material and KMS certificate that was active at time 'T'. This could be performed, for example, using an Audit Client. An Audit Client is a Key Management Client with the special access privilege to request previously-issued user key material for existing clients The provision of user key material by the KMS grants the auditor access to the user’s communication content..
    a. Any request made by the auditor shall be controlled and logged(to allow the audit action to be audited) .
    b. It is recommended that each release of key material be authorised by a secondary auditor (e.g. a double-lock mechanism).

c. It is recommended that the number of requests from each auditor within a time-period be limited. It is also recommended that the total number of requests from all auditors within a time-period be limited.

NOTE: The release of key material for a user at time 'T' only allows the auditor access to content for the defined 'key period' associated to time ‘T’. By limiting the total number of requests (e.g. to 0.1% of users), this limits the auditor’s access to communications. These controls help to ensure that the granted access to user content is time-limited and proportionate.

2. The auditor extracts the user events associated with the user at time 'T'.

3. The auditor extracts the MIKEY messages from the signalling events and use the audited user's KMS-supplied key material to decrypt the media encryption keys held within the MIKEY messages.

4. The auditor is now able to associate media with user events and use the media encryption keys extracted from MIKEY message to decrypt the media.

11 Interconnection, interworking and migration security

11.1 Interconnection

11.1.1 Overview

MC Systems may interconnect as described in TS 23.379 [2], TS 23.280 [36] and TS 23.281 [37]. This allows inter-system communications to occur.

To ensure interconnection is secure, MC clients only connect to MC Servers within their own system (unless migrating). When information is required by a MC client from another interconnected system, the information is first transferred from the interconnected partner system to the interconnected primary system via MCX server to MCX server communications followed by the distribution of that information to the MC client. For example, group management information is transferred between Group Management Servers in Clause 10.2.7 of TS 23.280 [36], prior to distribution to MC clients.

MC systems should protect themselves at the system border from external attackers. During interconnection, the MC system should use an HTTP proxy and an MC gateway containing an IS proxy as described in clause 11.1.3 to enforce policies and apply security functions (such as topology hiding). Among the security functions that can be performed at both proxies are preventing any direct MC client connection over this interface. Cross-system authentication of interconnection signalling requests may be implicit or explicit, subject to the policy of each MC system. Where authentication is implicit, the HTTP Proxy and IS Proxy should prevent messages that do not have an external MC service ID in the source of the request. MC servers should enforce policy to limit the information provided to a signalling requests from external MC service IDs.

Where authentication is explicit, the signalling request shall contain an Element for Authenticating Requests, (EAR), as defined in Clause 9.6. It is recommended that an authorised identity should be used within the EAR, to convey the source's authorisation to make the request.

11.1.2 Security procedures for interconnection

11.1.2.1 General

This clause defines security procedures that are used to support interconnection between MC systems

11.1.2.2 GMK transfer between MC systems

To allow a group to span two, or more, MC Systems, the GMK and GMK-ID needs to be transferred between the GMSs in different MC systems. The procedure follows an equivalent security procedure to that defined in Clause 7.3.3 for group regrouping. In this case, the GMK is transported within a ‘group information notify request' as defined in Clause 10.2.7.5 of TS 23.280 [36].

Pre-conditions:

- Both the primary and partner GMS have been keyed by their KMS.

The notify group configuration procedure is shown in Figure 11.1.2.2-1.

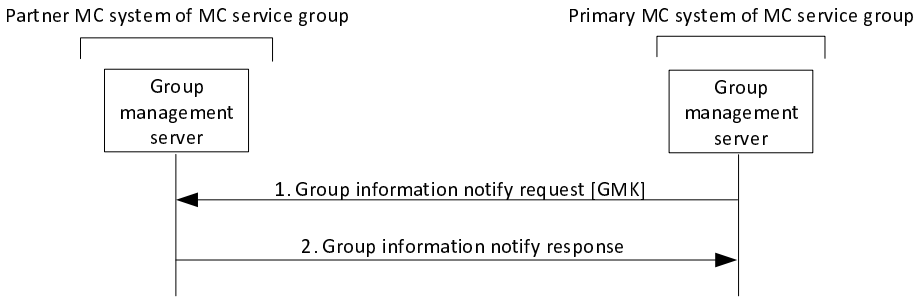

Figure 11.1.2.2-1: Inter-system GMK transfer

1. The GMS in the primary MC system of the MC service group provides the notification to the GMS in the partner MC system of the MC service group. The primary GMS includes a Group Key Transport payload following the procedures in Clause 5.7, treating the partner GMS as another user within the group. Accordingly, the payload encrypts the GMK to the identity of the partner GMS and is signed using the identity of the primary GMS. The GUK-ID is derived using the User Salt generated from the partner GMS's URI.

NOTE 1: If the choice of initiator KMS (IDRkmsi) or receiver KMS (IDRkmsr) within the MIKEY message is unacceptable, a KMS Redirect Response may be returned to the primary GMS providing KMS information. In this case, the primary GMS may re-attempt the above procedures.

NOTE 2: In this case, the partner GMS may discard the GUK-ID once the GMK-ID has been extracted.

2. Further signalling occurs as defined in TS 23.280 [36].

Upon receipt of the GMK, the partner GMS shall distribute the GMK and GMK-ID on to group MC clients as described in Clause 5.7.

11.1.3 Interconnection security with MC gateway server

A MC gateway server is part of the mission critical architecture for interconnection as defined in 3GPP TS 23.280 [36]. The MC gateway server includes an IS Proxy for inter-domain security as defined in Annex I. The IS Proxy provides protection of the SIP-3 interface (i.e. SIP payload and RTCP protection using a SPK as defined in clause 9 and clause 6.3.2). The SIP-3 interface is covered as part of the interconnection MCX-1 reference point.

Figure 11.1.3-1 shows an interconnection architecture between two MC domains (MC domain A and MC Domain B) each with the MC gateway server which contains the IS proxy for interconnection security. The MC gateway provides the necessary topology hiding and address translation along with signalling protection via the IS proxy. HTTP communications for interconnection over the HTTP-3 reference point are provided for via the HTTP proxy as described in 3GPP TS 23.280 [36] and protected as defined in clause 6.1.3 of this specification.

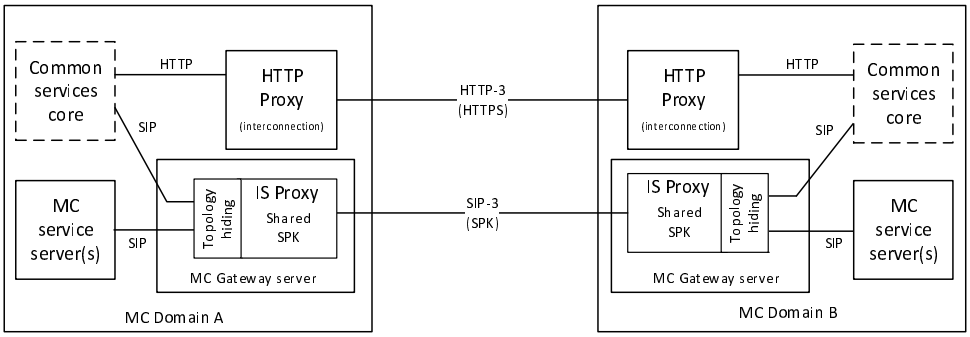

Figure 11.1.3-1: Interconnection security using MC gateway with HTTP and IS proxies

In Figure 11.1.3-1, the interface between the MC domains shall be protected hop-by-hop as defined in Clause 6.3.2. The SIP-3 interface between IS Proxies may be protected at the application layer using a shared SPK as defined in Clause 9 and the HTTP-3 interface between HTTP Proxies may be protected using TLS as defined in Clause 6.1.3.For interconnection communications with an MC gateway server (e.g. MC domain A to MC domain B in this example), HTTP and SIP messages are sent by an MC service server or a server in the common services core within the MC domain, towards the MC gateway server or HTTP proxy for processing, protection, and external routing to a partner MC domain.

For HTTP messages, the HTTP proxy applies topology hiding by replacing the internal to/from addresses in the HTTP message with the associated external HTTP routing addresses. The HTTP proxy determines the target HTTP proxy for MC domain B and choses the certificates appropriate for that TLS tunnel. The HTTP message is protected and sent towards MC domain B on the HTTP-3 interface. The HTTP proxy in MC domain B receives the HTTP message where it is decrypted from the external TLS tunnel. The HTTP proxy in MC domain B then replaces any external HTTP routing addresses with internal HTTP addresses applicable to MC domain B and forwards the message to the appropriate server within MC domain B.

For SIP messages, the MC gateway server in MC domain A applies topology hiding by replacing the internal to/from SIP addresses (e.g. Public Service Identities) in the SIP header with the associated external SIP routing addresses and passes the SIP message to the MC gateway IS proxy. The IS proxy removes any internal SIP payload encryption, then based on the target MC domain (MC domain B) selects the appropriate inter-domain SPK to re-encrypt the SIP payload(s). The SIP message is then sent towards the MC gateway server in MC domain B over the SIP-3 interface where the MC gateway IS proxy in MC domain B receives the SIP message and decrypts it using the inter-domain SPK it has in common with MC domain A. The IS proxy in MC domain B may then re-encrypt the SIP payload(s) with an internal MC domain B SPK. The topology hiding function of the MC gateway server in MC domain B then replaces the external SIP routing addresses with internal SIP addresses applicable to MC domain B and forwards the message to the appropriate server within MC domain B.

11.2 Interworking

11.2.1 General

The 3GPP security architecture supports the transfer of interworking signalling and media.

For media sent towards the 3GPP system, the IWF shall apply 3GPP security prior to sending the media to the 3GPP system. This is performed using MC Security Gateway functionality as defined in Annex L.

For media sent from the 3GPP system, the IWF shall remove 3GPP security prior to performing any further processing of the media. This is performed using MC Security Gateway functionality as defined in Annex L.

Interworking media may be end-to-end protected using LMR mechanisms that are out-of-scope of this specification. 3GPP MC application security shall be applied, regardless of whether the LMR security mechanism is used. For further details of LMR end-to-end security mechanisms see Annex K.

When signalling protection is used by the 3GPP MC system, the IWF shall apply the applicable 3GPP signalling protection mechanisms to the signalling packets sent towards the 3GPP system and shall remove the applicable 3GPP signalling protection mechanisms for signalling packets received from the 3GPP system. This is performed using MC Security Gateway functionality as defined in Annex L.

When signalling protection is not used by the 3GPP MC system, the signaling packets sent towards the 3GPP system shall be forwarded by the IWF without signalling protection.

11.2.2 Transport of non-3GPP interworking security data (InterSD)

To support the exchange of end-to-end interworking security data (a.k.a. Key Management Messages) between 3GPP MC UEs and the non-3GPP system when the interworking keys are home to the non-3GPP system, transport of the interworking security data is carried out using an Interworking Security Data (InterSD) message as defined in 23.283 [48]. An InterSD message may be generated by either the IWF or the 3GPP interworking MC UE.

The formatting and content of non-3GPP security data payloads are defined by the non-3GPP system or the non-3GPP layer of the 3GPP interworking MC UE and are out of scope for this document. The InterSD message shall support the transfer of non-3GPP security data payloads regardless of the security data payload content, format, or the architecture of the non-3GPP system beyond the IWF.

Signalling protection may be applied to the InterSD message as defined in clause 9.

An interworking key management record as defined in clause 11.2.3 may be required to enable secure InterSD messaging between a MC UE and a non-3GPP system.

11.2.3 Interworking key management enablement

To support interworking key management with a non-3GPP system (i.e. InterSD messaging), an interworking MC UE may require provisioning of an interworking key management record (InterKMRec) that supports the secure transfer of InterSD messages. Generally speaking, an InterKMRec provides initial key management parameters needed to send, receive, address, protect, or otherwise interpret InterSD messages passed between an interworking 3GPP MC UE and the non-3GPP interworking system (e.g. interworking key management addressing, interworking key management identifiers, interworking key management root keys, or other interworking key management related parameters).

The InterKMRec is provided from the MC KMS to the interworking MC UE during MC user key management authorization.

The format of an InterKMRec is shown in figure 11.2.3-1 and consists of a Primary InterKMRec ID, a Secondary InterKMRec ID, and the InterKMRec Payload.

IEI	Information Element	Presence	Format	Length
	Primary InterKMRec ID	M	TLV	varies
	Secondary InterKMRec ID	M	TLV	varies
	InterKMRec Payload	M	LV-E	varies

Figure 11.2.3-1 Interworking key management record (InterKMRec) structure

The Primary InterKMRec ID and Secondary InterKMRec ID are used in combination to identify and manage interworking MC UE clients for a single MC user (i.e. the same MC Service ID such as a MCPTT ID) when that MC user might log onto multiple interworking MC UEs at the same time. The MC service ID of a particular interworking MC user shall be coupled to the Primary InterKMRec ID and the Secondary InterKMRec ID shall be used to distinguish between the multiple MC clients in use by that MC service ID (e.g. the client ID).

For example, when an interworking MC user performs key management authorization, the MC service ID of the user is used to identify the set of InterKMRecs associated to that MC service ID. The KMS selects one of the associated InterKMRecs (assuming at least one record exists) and then further makes a dynamic association between the client ID making the request and the Secondary InterKMRec ID, this way uniquely identifying the interworking MC user and each interworking MC client the MC user is using.

The exact format and contents of the InterKMRec Payload is defined by the non-3GPP system and is out of scope for this document. The method used to provision an InterKMRec into the KMS is out of scope for this document. The method used to associate an MC service ID to a Primary InterKMRec ID and the method used to associate an MC UE client to a Secondary InterKMRec ID is also out of scope for this document.

When an interworking MC user performs key management authorization at the KMS and the access token has been validated, the KMS shall check to see if the MC service ID provided in the access token has an associated InterKMRec. If more than one InterKMRec exists for the MC service ID, the KMS shall check to see if the Secondary InterKMRec ID is also already associated to a specific client for that MC Service ID. If the client cannot be matched, then the KMS shall select one of the InterKMRecs and associate the client ID to the Secondary InterKMRec ID of that InterKMRec. The KMS shall then deliver the selected InterKMRec to the interworking MC UE during MCX user key management authorization as defined in clause 5.1.3.1.

It is out of scope of this document as to when and how the KMS disassociates a client from a particular InterKMRec ID.

When required by the MC system, the InterKMRec shall be transported from the KMS to the interworking MC UE encrypted on the TrK of the interworking MC UE. The InterKMRec may be signed and if so, shall be signed by either the Ink if the InK is present, or signed by the TrK if the InK is not present.

Annex A (normative): Security requirements

A.1 Introduction

Stage 1 requirements pertaining to MCX security are found in 3GPP TS 22.179 [3] and 3GPP TS 22.280 [47]. Stage 2 Architectural requirements pertaining to MCX security are found in 3GPP TS 23.179 [2], 3GPP TS 23.280 [36], 3GPP TS 23.281 [37], and 3GPP TS 23.282 [38]. The following are MCX derived security requirements:

A.2 Configuration & service access

[33.180 MCX-A.2-001] The MC UE and the network entity providing the MCX configuration data, shall mutually authenticate each other prior to MC UE configuration to use the MCX service.

[33.180 MCX-A.2-002] The MC User and the MCX Service shall mutually authenticate each other prior to providing the MC UE with the MCX Service User profile and access to user-specific services.

[33.180 MCX-A.2-003] The transmission of configuration data and user profile data between an authorized MCX server in the network and the MC UE shall be confidentiality protected, integrity protected and protected from replays.

A.3 Group key management

[33.180 MCX-A.3-001] Group key material shall be integrity and confidentiality protected for a specific MC User during distribution from the MCX service to MC UEs.

[33.180 MCX-A.3-002] Group key material shall be authenticated as coming from a valid, authorized source. The authorized source may be an MC Administrator or may be another authorized entity (e.g. an authorized MCX User or Dispatcher).

[33.180 MCX-A.3-003] It shall be possible for authorized entities to dynamically create and distribute a new group security context at any time. This may be as part of a group creation process, be due to a periodic update to maintain key freshness, or due to compromise of group key material.

[33.180 MCX-A.3-004] The creation of a new group security context (e.g. via User-Regroup operation) shall not change or compromise an existing group security context.

[33.180 MCX-A.3-005] It shall be possible for an authorized, authenticated entity to revoke and update a group security context from use.

A.4 On-network operation

[33.180 MCX-A.4-001] All users of the MCX service shall be authenticated to prevent an adversary impersonating a user for the purpose of denial of service.

[33.180 MCX-A.4-002] The MCX service should take measures to detect and mitigate DoS attacks to minimize the impact on the network and on MC users.

[33.180 MCX-A.4-003] The MC user shall be authenticated by the MCX application.

[33.180 MCX-A.4-004] A mechanism shall exist that allows the MCX application to be authenticated by the MCX user.

[33.180 MCX-A.4-005] The MC UE and MCX service should enforce the result of the authentication for the duration of communications (e.g. by integrity protection or implicit authentication by encryption with a key that is derived from the authentication and is unknown to the adversary).

[33.180 MCX-A.4-006] The security solution should minimize the impact of a compromised MC UE on other MC UEs.

[33.180 MCX-A.4-007] The MCX Service shall provide a means to ensure integrity of all MCX user signalling at the application layer.

[33.180 MCX-A.4-008] The MCX Service shall protect the administrative and security management parameters from manipulation by individuals who are not explicitly authorized by the Mission Critical Organization.

[33.180 MCX-A.4-009] The MCX service shall provide a means to support confidentiality of MCX user identities from all entities outside the MCX service.

[33.180 MCX-A.4-010] The MCX service shall provide a means to support confidentiality of MCX signalling from all entities outside the MCX service.

[33.180 MCX-A.4-011] The MCX Service shall provide a means to support end-to-end confidentiality and integrity protection for all media traffic transmitted between MC UEs.

[33.180 MCX-A.4-012] The MCX Service shall provide a means to support the confidentiality and integrity protection of location information transmitted from the MC UE to the MCX application server.

A.5 Ambient listening

[33.180 MCX-A.5-001] Specific roles in the organization and shall be identified to authorize and activate Ambient Listening and privileges shall be assigned to these roles to activate and register the use of ambient listening.

[33.180 MCX-A.5-002] The activation of the Ambient Listening functionality shall be automatically registered by the system and will be stored as an 'event' by the system.

[33.180 MCX-A.5-003] Any decision to activate Ambient Listening, or review of such a decision, may also be recorded in a suitable incident log unless to do so would interfere with the purpose for which the functionality is being used i.e. an investigation tool for evidence gathering in cases of suspected gross misconduct of staff or evidence gathering in criminal cases. If this is the case the authorization needs to be recorded elsewhere as appropriate.

[33.180 MCX-A.5-004] A radio user should be told as soon as possible that they are, or have been, subject to Ambient Listening and the reason why the functionality was activated. The fact they have been informed, by whom and when, should be recorded in a suitable log.

A.6 Data communication between MCX network entities

[33.180 MCX-A.6-001] A security mechanism shall exist that allows transmission of data between 3GPP MCX network entities to be authenticated, confidentiality protected, integrity protected and protected from replays.

NOTE: UE-to-UE and UE-to-network relays are not considered to be 'network entities'.

A.7 Key stream re-use

[33.180 MCX-A.7-001] The MCX system shall ensure that key streams are not reused.

A.8 Late entry to group communication

[33.180 MCX-A.8-001] An authorized MCX User shall be able to obtain the information necessary to derive the group security context for the MCX Group while an MCX Group communication is on-going. As a result, the MC User shall be able to listen to the group communication within 350ms. This requirement applies for both on-network and off-network MCX operations.

A.9 Private call confidentiality

[33.180 MCX-A.9-001] It shall be possible to establish a unique Private Call security context between any pair of authorized MCX users within the MCX system. The security context shall not be available to other MCX users, except, where necessary, authorized MCX monitoring functions (e.g. LI, Discreet Listening). If the security context is made

available to monitoring functions, appropriate controls and logging shall exist. This requirement applies when MCX UEs are operating both on-network and off-network.

[33.180 MCX-A.9-002] The Private Call security context shall provide a means to provide confidentiality and integrity protection of user traffic, and authenticate the MCX users involved in the Private Call.

A.10 Off-network operation

[33.180 MCX-A.10-001] The MCX service should take measures to detect and mitigate DoS attacks to minimize the impact to relays and to off-network MCX users.

[33.180 MCX-A.10-002] The MCX Service shall provide a means to support end-to-end security for all media traffic transmitted between MCPTT UEs, including where relays are used.

[33.180 MCX-A.10-003] The MCX Service shall provide a means to support the confidentiality and integrity protection of location information transmitted from the MCX UE to the MCX application server, including where relays are used.

[33.180 MCX-A.10-004] MCX off-network UEs shall be explicitly or implicitly authenticated to each other.

[33.180 MCX-A.10-005] MCX off-network UEs and MC relays shall be explicitly or implicitly authenticated to each other.

[33.180 MCX-A.10-006] The security solution should minimize the impact of a compromised MCX UE on other MCX UEs.

[33.180 MCX-A.10-007] The MCX Service shall provide a means to ensure integrity of all MCX user signalling at the MCX application layer.

[33.180 MCX-A.10-008] The MCX service shall provide a means to support confidentiality of MCX service user identities from all entities outside the MCX service.

[33.180 MCX-A.10-009] The MCX service shall provide a means to support confidentiality of MCX signalling from all entities outside the MCX service.

A.11 Privacy of MCX service identities

[33.180 MCX-A.11-001] The MCX service user identities of each plane shall be used within the corresponding plane and concealed to other planes.

[33.180 MCX-A.11-002] When required by the MCX Service provider, MCX application services layer identities (such as the Mission Critical user identity, MCPTT ID, MCVideo ID, MCData ID and MCX Group IDs) and other application services sensitive information (as further described in 3GPP TS 23.179 [2], clause 8.2), shall be contained within the application plane and shall provide a means to support confidentiality and integrity of the application plane from the SIP signaling plane.

[33.180 MCX-A.11-003] When protection of identities and other sensitive MCX application information is NOT required by the MCX Service provider, the MCX application services layer identities (such as the Mission Critical user identity, MCPTT ID, MCVideo ID, MCData ID and MCX Group IDs) and other application services sensitive information (as further described in 3GPP TS 23.179 [2], clause 8.2), shall remain contained within the application plane. While confidentiality protection is not required, integrity protection may be applied.

A.12 User authentication and authorization

[33.180 MCX-A.12-001] User authentication and authorization interoperability between different networks and different manufacturers' clients and servers shall satisfy the requirements for mission critical roaming and migration.

[33.180 MCX-A.12-002] User authentication and authorization shall support all deployment models listed in 3GPP TS 23.179 [2].

[33.180 MCX-A.12-003] User authentication and authorization shall support interchangeable MC user authentication solutions, allowing implementations to use different means to authenticate the user, e.g. Web SSO, SIP digest, GBA, biometric identifiers, username+password.

[33.180 MCX-A.12-004] User authentication and authorization shall support scalability (number of users), providing efficient support for small MCX systems with few users, to large MCX systems with hundreds of thousands of users.

[33.180 MCX-A.12-005] User authentication and authorization shall support extensibility, providing authorization for additional mission critical services including group aware services, additional interfaces, etc.

[33.180 MCX-A.12-006] All users of the MCX Service shall be authenticated to prevent an adversary impersonating a user for the purpose of denial of service.

A.13 Inter-domain

[33.180 MCX-A.13-001] An MCX Service shall provide mechanisms to allow an MCX User to operate in a Partner MCX Service System, subject to authorization from both the Partner and the Primary MCX Service Systems of the MCX User (R-6.17.2-001 [47]).

[33.180 MCX-A.13-002] The authentication of an MCX User with an MCX Service in a Partner MCX Service System shall be based on security parameters obtained from the Primary MCX Service System of the MCX User (R-6.17.2-002 [47]).

NOTE 1: This is an application layer authentication and not 3GPP network authentication.

[33.180 MCX-A.13-003] An MCX Service shall provide mechanisms to allow an MCX User on the Primary MCX Service System to affiliate to an MCX Service Group from a Partner MCX Service System, subject to authorization from the Primary MCX Service System and the Partner MCX Service System where the MCX Service Group is defined (R-6.17.2-004 [47]).

[33.180 MCX-A.13-004] An MCX Service shall provide mechanisms to allow a roaming MCX User to affiliate to an MCX Service Group from the Partner MCX Service System, subject to authorization from the Partner MCX Service System where the MCX Service Group is defined (R-6.17.2-005 [47]).

[33.180 MCX-A.13-005] An MCX Service shall provide mechanisms to allow an MCX User that receives service from a Partner MCX Service System to affiliate to an MCX Service Group from another Partner MCX Service System, subject to authorization from the Partner MCX Service System where the MCX Service Group is defined (R-6.17.2-006 [47]).

NOTE 2: It is assumed that once affiliation from a User to a Group is successful, subsequent communication within that Group are available to the User.

[33.180 MCX-A.13-006] End to end security of an MCX Service Group communication (including in Partner MCX Service Systems) shall be based on parameters obtained from the MCX Service system where the MCX Service Group is defined (R-6.17.2-007 [47]).

[33.180 MCX-A.13-007] All Mission Critical Users shall be authenticated with their home identity management service prior to authentication or authorisation with a partner domain.

[33.180 MCX-A.13-008] A user requiring services at a partner domain shall first acquire a verifiable credential from the user's primary identity management service.

[33.180 MCX-A.13-009] An identity management service shall authenticate a visiting user based on a verifiable credential from the user's primary identity management service prior to authorising that user for local service(s).

[33.180 MCX-A.13-010] A visiting user shall be authorised with the local server(s) at the partner MCX System before being granted local services.

[33.180 MCX-A.13-011] The partner identity management service shall have full and overruling authorisation control of all visiting users requesting services in the partner MCX System.

[33.180 MCX-A.13-012] When using external security domains, the Home Security Domain shall apply policies which ensure that only trusted external security domains are used.

[33.180 MCX-A.13-013] Use of external security domains shall be logged to detect impersonation and misuse.

[33.180 MCX-A.13-014] MCX Services shall be able to permit/deny the use of security domains over their service.

A.14 MCData

[33.180 MCX-A.14-001] The MCData Service shall provide a means to support end-to-end confidentiality and integrity protection for messaging transmitted between MCX UEs in both media and signalling streams.

[33.180 MCX-A.14-002] The MCData Service shall provide a means to authenticate messages in both media and signalling streams.

A.15 Multimedia Broadcast/Multicast Service

[33.180 MCX-A.15-001] The security of signalling transmitted between the MCX client and MCX server shall be controlled by the MCX server. As a consequence of this requirement, the MCX Server shall not require key material from external MC Domains to enable the use of MBMS.

[33.180 MCX-A.15-002] The MCX Service shall provide means to support confidentiality and integrity protection for the MBMS subchannel control messages.

Annex B (normative): OpenID connect profile for MCX

B.1 General

The information in this annex provides a normative description of the MCX Connect Authentication and Authorization framework based on the OpenID Connect 1.0 standard. Characterization of the ID token, access token, how to obtain tokens, how to validate tokens, and how to use the refresh token is explained.

The OpenID Connect 1.0 standard provides the source of the information contained in this annex. MCX Connect profiles the OpenID Connect standard and includes the service IDs in the ID token and the access token, as well as the definition of MCX specific scopes for key management, MCX services, configuration management, and group management. This profile is compliant with OpenID Connect.

B.2 MCX tokens

B.2.1 ID token

B.2.1.1 General

The ID Token shall be a JSON Web Token (JWT) and contain the following standard and MCX token claims. Token claims provide information pertaining to the authentication of the MCX user by the IdM server as well as additional claims. This clause profiles the required standard and MC claims for the MCX Connect profile.

B.2.1.2 Standard claims

These standard claims are defined by the OpenID Connect 1.0 specification and are REQUIRED for MCX implementation. Other claims defined by OpenID Connect are optional. The standards-based claims for an MCX Connect ID token are shown in table B.2.1.2-1.

Table B.2.1.2-1: ID token standard claims

Parameter	Description
iss	REQUIRED. The URL of the IdM server.
Sub	REQUIRED. A case-sensitive, never reassigned string (not to exceed 255 bytes), which uniquely identifies the MCX user within the MCX server provider's domain.
Aud	REQUIRED. The Oauth 2.0 client_id of the MCX client
exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds)
iat	REQUIRED. Time at which the ID Token was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

B.2.1.3 MCX claims

The MCX Connect profile extends the OpenID Connect standard claims with the additional claims shown in table B.2.1.3-1.

Table B.2.1.3-1: ID token MCX claims

Parameter	Description
mcptt_id	REQUIRED for MCPTT. The MCPTT ID of the current MCPTT user of the MCPTT client.
mcvideo_id	REQUIRED for MCVideo. The MCVideo ID of the current MCVideo user of the MCVideo client.
mcdata_id	REQUIRED for MCData. The MCData ID of the current MCData user of the MCData client.

B.2.2 Access token

B.2.2.1 Introduction

The access token is opaque to MCX clients and is consumed by the MCX resource servers (i.e. KMS, MCPTT server, MCVideo server, MCData server, etc). The access token shall be encoded as a JSON Web Token as defined in IETF RFC 7519 [32]. The access token shall include the JSON web digital signature profile as defined in IETF RFC 7515 [35].

B.2.2.2 Standard claims

MC access tokens shall convey the following standards-based claims as defined in IETF RFC 7662 [33].

Table B.2.2.2-1: Access token standard claims

Parameter	Description
exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds).
scope	REQUIRED. A JSON string containing a space-separated list of the MCX authorization scopes associated with this token. The scope(s) contained here reflect the requested scope(s) from the Authentication Request (clause B.4.2.2).
client_id	REQUIRED. The identifier of the MCX client making the API request as previously registered with the IdM server.

B.2.2.3 MCX claims

The MCX Connect profile extends the standard claims defined in IETF RFC 7662 [33] with the additional claims shown in table B.2.2.3-1.

Table B.2.2.3-1: Access token MCX claims

Parameter	Description
mcptt_id	REQUIRED for MCPTT. The MCPTT ID of the current MCPTT user of the MCPTT client.
mcvideo_id	REQUIRED for MCVideo. The MCVideo ID of the current MCVideo user of the MCVideo client.
mcdata_id	REQUIRED for MCData. The MCData ID of the current MCData user of the MCData client.

B.3 MCX client registration

Before an MCX 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 MCX 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 MCX client.
- Registration of the client's redirect URIs.

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

B.4 Obtaining tokens

B.4.1 General

Once an MCX client has been successfully registered with the IdM server of the MCX service provider, the MCX 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 MCX 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 an MCX client requests the access token depends upon the client profile. The MCX client profiles, along with steps required from them to obtain OAuth access tokens, are explained in technical detail below.

B.4.2 Native MCX client

B.4.2.1 General

This conforms to the Native Application profile of OAuth 2.0 as per IETF RFC 6749 [19].

MCX clients fitting the Native application profile utilize the authorization code grant type with the IETF RFC 7636 [53] PKCE extension for enhanced security as shown in figure B.4.2.1-1.

Unless indicated otherwise in this document, the use of HTTP Basic authentication shall be as specified in IETF RFC 6749 [19] and IETF RFC 6750 [20].

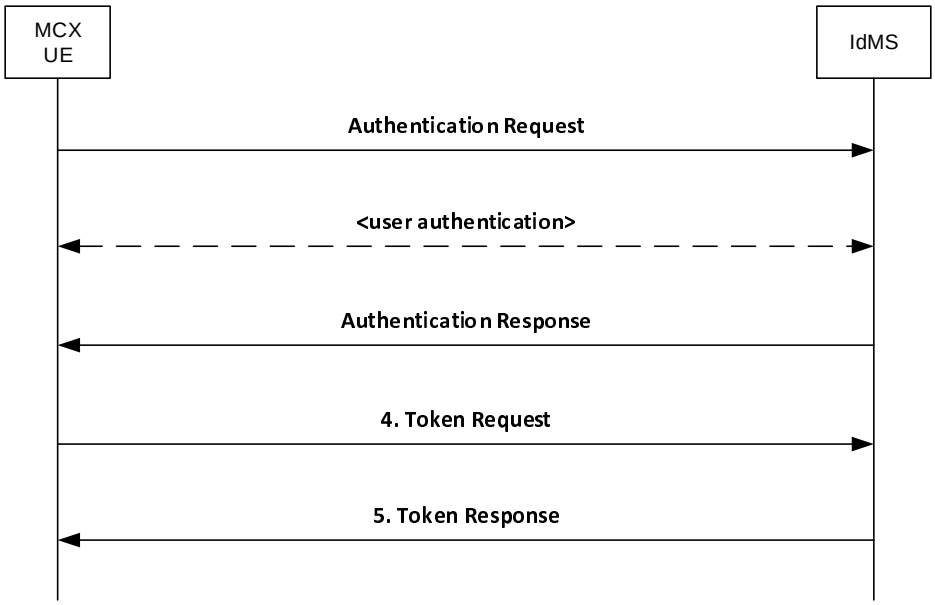

Figure B.4.2.1-1: Authorization Code flow

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 MCX clients the value shall be set to "code".
client_id	REQUIRED. The identifier of the MCX 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 MCX 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 MCX client to which the IdM server will redirect the MCX client's user agent in order to return the authorization code to the MCX client. 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 MCX client to maintain state between the authorization request and authorization response. The IdM server includes this value in its authorization response back to the MCX client.
acr_values	REQUIRED. Space-separated string that specifies the acr values that the IdM server is being requested to use for processing this authorization 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 that is sent in the authorization request, 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 IdM client, the IdM server performs user authentication. Note that user authentication is completely opaque to the IdM client (which never sees any of it, as it is run directly between the IdM server and the IdM client on the UE).

B.4.2.3 Authentication response

The authorization endpoint running on the IdM server issues an authorization code and delivers it to the MCX client. The authorization code is used by the MCX 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 MCX client via the authorization response.
state	REQUIRED. The value shall match the exact value used in the authorization request. If the state does not match exactly, then the NGMI API client is under a Cross-site request forgery attack and shallreject 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?
code=SplxlOBeZQQYbYS6WxSbIA
  &state=abc123
```

B.4.2.4 Access token request

In order to exchange the authorization code for an ID token, access token and refresh token, the MCX 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 authorization 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 authorization request.
code_verifier	REQUIRED. A cryptographically random string that is used to correlate the authorization 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
```

B.4.2.5 Access token response

If the access token request is valid and authorized, the IdM server returns an ID token, access token and refresh token to the MCX client in an access token response message; otherwise it will return an error.

The access token response standard parameters are shown in table B.4.2.5-1.

Table B.4.2.5-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be “bearer”
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

An example of a successful response might look like:

EXAMPLE:

```
HTTP/1.1 200 OK
    Content-Type: application/json;charset=UTF-8
    Cache-Control: no-store
    Pragma: no-cache
    {

"access_token":"eyJhbGciOiJSUzI1NiJ9.eyJtY3B0dF9pZCI6ImFsaWNlQG9yZy5jb20iLCJleHAiOjE0NTM1MDYxMjEsInN
jb3BlIjpbIm9wZW5pZCIsIjNncHA6bWNwdHQ6cHR0X3NlcnZlciJdLCJjbGllbnRfaWQiOiJtY3B0dF9jbGllbnQifQ.XYIqai4Y
KSZCKRNMLipGC_5nV4BE79IJpvjexWjIqqcqiEx6AmHHIRo0mhcxeCESrXei9krom9e8Goxr_hgF3szvgbwl8JRbFuv97XgepDLj
Eq4jL3Cbu41Q9b0WdXAdFmeEbiB8wo_xggiGwv6IDR1b3TgAAsdjkRxSK4ctIKPaOJSRmM7MKMcKhIug3BEkSC9-
aXBTSIv5fAGN-ShDbPvHycBpjzKWXBvMIR5PaCg-9fwjELXZXdRwz8C6JbRM8aqzhdt4CVhQ3-Arip-S9CKd0tu-
qhHfF2rvJDRlg8ZBiihdPH8mJs-qpTFep_1-kON3mL0_g54xVmlMwN0XQA",
  "refresh_token":"Y7NSzUJuS0Jp7G4SKpBKSOJVHIZxFbxqsqCIZhOEk9",
"id_token":"eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwiYXVkIjoibWNwdHRfY2xpZW50IiwiaXNzIjoiSWR
NUy5zZXJ2ZXIuY29tOjkwMzEiLCJpYXQiOjE0NTM0OTgxNTgsImV4cCI6MTQ1MzQ5ODQ1OCwibWNwdHRfaWQiOiJhbGljZUBvcmc
uY29tIn0.Dpn7AhIMaqMEgg12NYUUfJGSFJMPG8M2li9FLtPotDlHvwU2emBws8z5JLw81SXQnoLqZ8ZF8tIhZ1W7uuMbufF4Wsr
7PAadZixz3CnV2wxFV9qR_VA1-0ccDTPukUsRHsic0SgZ3aIbcYKd6VsehFe_GDwfqysYzD7yPwCfPZo",
"token_type": "Bearer",
"expires_in": 7199
}
```

The MCX client may now validate the user with the ID token and configure itself for the user (e.g. by extracting the MC service ID from the ID Token). The MCX client then uses the access token to make authorized requests to the MCX resource servers (MCPTT server, MCVideo server, MCData server, KMS, etc.) on behalf of the end user.

B.5 Refreshing an access token

B.5.1 General

To protect against leakage or other compromise, access token lifetimes are typically short lived (though it is ultimately a matter of security policy & configuration by the service provider). Some client types can be issued longer-lived refresh tokens, which enable them to refresh the access token and avoid having to prompt the user for authentication again when the access token expires. Refresh tokens are available only to clients utilizing the authorization code grant type (native MCX clients and web-based MCX clients). Refresh tokens are not given to clients utilizing the implicit grant type (browser-based MCX clients). Figure B.5.1-1 shows how Native MCX clients can use the refresh token as a grant type to obtain new access tokens.

Figure B.5.1-1: Requesting a new access token

B.5.2 Access token request

To obtain an access token from the IdM server using a refresh token, the MCX client makes an access token request to the token endpoint of the IdM server. The MCX client does this by adding the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. The access token request standard parameters are shown in table B.5.2-1.

Table B.5.2-1: Access token request standard required parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to "refresh_token".
scope	Space-delimited set of permissions that the MCX client requests. Note that the scopes requested using this grant type shall be of equal to or lesser than scope of the original scopes requested by the MCX client as part of the original authorization request.

An example of a token request for MCX Connect might look like:

EXAMPLE:

```
POST /as/token.oauth2 HTTP/1.1
Host: IdM.server.com:9031
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=Y7NSzUJuS0Jp7G4SKpBKSOJVHIZxFbxqsqCIZhOEk9&scope=3gpp:mcptt:p
tt_server
```

If the MCX client was provided with client credentials by the IdM server, then the client shall authenticate with the token endpoint of the IdM server utilizing the client credential (shared secret or public-private key pair) established during the client registration phase.

B.5.3 Access token response

In response to the access token request (above) the token endpoint on the IdM server will return an access token to the MCX client, and optionally another refresh token in an access token response message.

The access token response standard parameters are shown in table B.5.3-1.

Table B.5.3-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be “bearer”
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

An example of a successful response for MCX Connect might look like:

EXAMPLE:

```
HTTP/1.1 200 OK
    Content-Type: application/json;charset=UTF-8
    Cache-Control: no-store
    Pragma: no-cache
{
 "access_token":"eyJhbGciOiJSUzI1NiJ9.eyJtY3B0dF9pZCI6ImFsaWNlQG9yZy5jb20iLCJleHAiOjE0NTM1MDYxMjEsIn
Njb3BlIjpbIm9wZW5pZCIsIjNncHA6bWNwdHQ6cHR0X3NlcnZlciJdLCJjbGllbnRfaWQiOiJtY3B0dF9jbGllbnQifQ.XYIqai4
YKSZCKRNMLipGC_5nV4BE79IJpvjexWjIqqcqiEx6AmHHIRo0mhcxeCESrXei9krom9e8Goxr_hgF3szvgbwl8JRbFuv97XgepDL
jEq4jL3Cbu41Q9b0WdXAdFmeEbiB8wo_xggiGwv6IDR1b3TgAAsdjkRxSK4ctIKPaOJSRmM7MKMcKhIug3BEkSC9-
aXBTSIv5fAGN-ShDbPvHycBpjzKWXBvMIR5PaCg-9fwjELXZXdRwz8C6JbRM8aqzhdt4CVhQ3-Arip-S9CKd0tu-
qhHfF2rvJDRlg8ZBiihdPH8mJs-qpTFep_1-kON3mL0_g54xVmlMwN0XQA",
   "refresh_token": "iTxQYALqlc7uLyFGpnl8tR8Y9gkw91mFy2qC9Yywkz",
   "token_type": "Bearer",
   "expires_in": 7199
}
```

It is possible to configure the IdM server to confirm that the user account is still valid each time the refresh token is presented, and to revoke the refresh token if not. This security practice is RECOMMENDED.

B.6 MCX client registration with partner IdM service

MCX client registration with a partner IdM service shall be as described in clause B.3.

B.7 Obtaining an access token from a partner domain

B.7.1 Overview

When an MCX user requires user service authorisation for services owned and managed within a partner domain, the MCX client shall use the OAuth 2.0 token exchange extension grant type mechanism to obtain a security token for authentication with the partner IdM service. The OAuth 2.0 token exchange procedure defines a method for obtaining the security token from the primary IdMS which contains information about the user that is verifiable by the partner IdMS.

The MCX client then provides this security token to the partner IdM service in exchange for an access token that is specific to the services in the partner domain. The MCX UE then uses the access token for user service authorisation to those services within the partner domain.

The security token and acess token(s) are specific to a IdMS and partner domain and therefore the OAuth 2.0 token exchange procedure shall be repeated with each additional domain to obtain user service authorisation to partner services within those domains.

Figure B.7.1-1 shows the OAuth 2.0 token exchange procedure used to obtain a security token and acess token(s). The messages are described in the following sub-clauses.

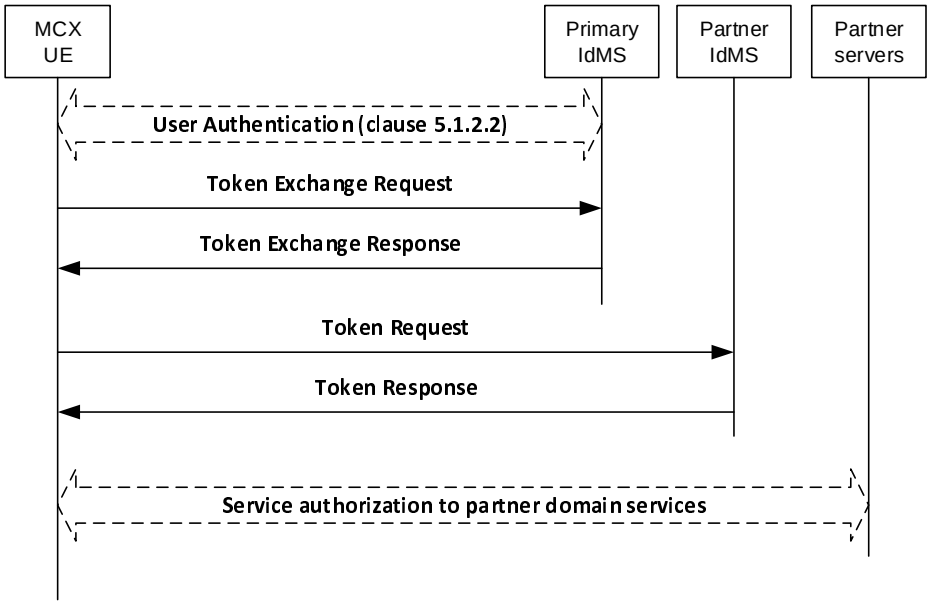

Figure B.7.1-1: Token exchange flow

B.7.2 Token Exchange Request

In order to obtain a security token, the MCX client makes a request to the primary authorization server's token endpoint by sending the following parameters using the "application/x-www-form-urlencoded" content-type and a character encoding of UTF-8 in the HTTP request entity-body. The standard parameters shown in table B.7.2-1 are required by the MCX Connect profile.

Table B.7.2-1: Token Exchange Request standard required parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to “urn:ietf:params:oauth:grant-type:token-exchange” indicating that a token exchange is being performed.
resource	REQUIRED. Indicates the physical location of the target service or resource where the client intends to use the requested security token (i.e. the address of the partner authorization server's token endpoint where the security token will be applied).
subject_token	REQUIRED. A token that represents the identity of the party on behalf of whom the request is being made. This shall be the access token previously obtained in the token response message (clause B.4.2.5) during authorisation (clause B.4).
subject_token_type	REQUIRED. An identifier that indicates the type of the security token in the subject_token parameter. The value shall be set to “urn:ietf:params:oauth:token-type:jwt” indicating the access token is a JSON Web Token.

An example of a successful token exchange request might look like:

EXAMPLE:

```
POST /as/token.oauth2 HTTP/1.1
Host: IdM.server.com:9031
Authorization: Basic cnA33hpsb25nABClY3VyZS1yYW5kb20tc2VjdnV0
Content-Type: application/x-www-form-urlencoded
grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Atoken-exchange
&resource= IdM.partner_server.com
&subject_token=baaR3jcJyb4BWCxGsndq23ScbdFMogUC5Pb233jKLTC
&subject_token_type= urn%3Aietf%3Aparams%3Aoauth%3Atoken-type%3Ajwt
```

B.7.3 Token Exchange Response

Upon successfully receiving and validating the token exchange request message from the MCX client, the IdM server shall return a token exchange response containing a security token specific to the partner IdMS.

The token exchange response standard parameters are shown in table B.7.3-1.

Table B.7.3-1: Token exchange response standard required parameters

Parameter	Values
access_token	REQUIRED. This is the security token specific to the partner IdMS.
issued_token_type	REQUIRED. This field shall be "urn:ietf:params:oauth:token-type:jwt"
token_type	REQUIRED. This field shall be "bearer"
expires_in	RECOMMENDED. The lifetime in seconds of the security token.

An example of a successful token exchange response might look like:

EXAMPLE:

```
HTTP/1.1 200 OK
   Content-Type: application/json
   Cache-Control: no-cache, no-store
  {
   "access_token":"eyJhbGciOiJFUzI1NiIsImtpZCI6IjllciJ9.eyJhdWQiOiJo
   dHRwczovL2JhY2tlbmQuZXhhbXBsZS5jb20iLCJpc3MiOiJodHRwczovL2FzLmV
   4YW1wbGUuY29tIiwiZXhwIjoxNDQxOTE3NTkzLCJpYXQiOjE0NDE5MTc1MzMsIn
   N1YiI6ImJjQGV4YW1wbGUuY29tIiwic2NwIjpbImFwaSJdfQ.MXgnpvPMo0nhce
   PwnQbunD2gw_pDyCFA-Saobl6gyLAdyPbaALFuAOyFc4XTWaPEnHV_LGmXklSTp
   z0yC7hlSQ",
   "issued_token_type":"urn:ietf:params:oauth:token-type:jwt",
   "token_type":"Bearer",
   "expires_in":600
  }
```

B.7.4 Token Request

In order to exchange the security token for an access token and (optional) refresh token, the MCX client makes a request to the partner authorization server's token endpoint by sending a token request message with 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 authentication of the security token by the partner IdMS is REQUIRED in order to exchange the security token for an access token. The security token shall be transported in the body of the token request message. The token request standard parameters are shown in table B.7.4-1.

Table B.7.4-1: Token Request standard required parameters

Parameter	Values
grant_type	REQUIRED. This value shall be set to "urn:ietf:params:oauth:grant-type:jwt-bearer" as per rfc 7523 [46].
assertion	REQUIRED. This field shall contain the security token received in annex B.7.3.
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.
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 at the partner system (e.g. MCPTT group services, MCVideo group services, MCData group services, etc.). If authorized, the requested scope values will be bound to the access token returned to the client in the token exchange response message. The scope shall include one or more of the following: - ″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″ Others may be added in the future as new MCX resource servers are introduced by 3GPP.

Examples of a successful token request might look like:

EXAMPLE 1:

```
POST /as/token.oauth2 HTTP/1.1
Host: IdM.server.com:9031
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&
assertion=eyJhbGciOiJFUzI1NiIsImtpZCI6IjllciJ9.eyJhdWQiOiJodHRwczovL2JhY2tlbmQuZXhhbXBsZS5jb20iLCJpc
3MiOiJodHRwczovL2FzLmV4YW1wbGUuY29tIiwiZXhwIjoxNDQxOTE3NTkzLCJpYXQiOjE0NDE5MTc1MzMsInN1YiI6ImJjQGV4Y
W1wbGUuY29tIiwic2NwIjpbImFwaSJdfQ.MXgnpvPMo0nhcePwnQbunD2gw_pDyCFASaobl6gyLAdyPbaALFuAOyFc4XTWaPEnHV
_LGmXklSTpz0yC7hlSQ&
client_id=myNativeApp&
scope=openid 3gpp:mc:ptt_group_management_service&
```

EXAMPLE 2:

```
POST /as/token.oauth2 HTTP/1.1
Host: IdM.server.com:9031
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded

grant_type=urn%3Aietf%3Aparams%3Aoauth%3Agrant-type%3Ajwt-bearer&
assertion=eyJhbGciOiJFUzI1NiIsImtpZCI6IjllciJ9.eyJhdWQiOiJodHRwczovL2JhY2tlbmQuZXhhbXBsZS5jb20iLCJpc
3MiOiJodHRwczovL2FzLmV4YW1wbGUuY29tIiwiZXhwIjoxNDQxOTE3NTkzLCJpYXQiOjE0NDE5MTc1MzMsInN1YiI6ImJjQGV4Y
W1wbGUuY29tIiwic2NwIjpbImFwaSJdfQ.MXgnpvPMo0nhcePwnQbunD2gwpDyCFASaobl6gyLAdyPbaALFuAOyFc4XTWaPEnHV_
LGmXklSTpz0yC7hlSQ&
client_id=myNativeApp&scope=openid 3gpp:mc:ptt_service,3gpp:mc:ptt_key_management_service,3gpp:mc:pt
t_config_management_service,3gpp:mc:ptt_group_management_service&
```

B.7.5 Token Response

If the token request is valid and authorized, the partner IdM server returns an access token to the MCX client specific to the user for the partner services and optionally a refresh token in a token response message; otherwise, it will return an error.

The token response standard parameters are shown in table B.7.5-1.

Table B.7.5-1: Token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be “bearer”
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

An example of a successful response might look like:

EXAMPLE:

```
HTTP/1.1 200 OK
    Content-Type: application/json;charset=UTF-8
    Cache-Control: no-store
    Pragma: no-cache
{
 "access_token":"eyJhbGciOiJSUzI1NiJ9.eyJtY3B0dF9pZCI6ImFsaWNlQG9yZy5jb20iLCJleHAiOjE0NTM1MDYxMjEsIn
Njb3BlIjpbIm9wZW5pZCIsIjNncHA6bWNwdHQ6cHR0X3NlcnZlciJdLCJjbGllbnRfaWQiOiJtY3B0dF9jbGllbnQifQ.XYIqai4
YKSZCKRNMLipGC_5nV4BE79IJpvjexWjIqqcqiEx6AmHHIRo0mhcxeCESrXei9krom9e8Goxr_hgF3szvgbwl8JRbFuv97XgepDL
jEq4jL3Cbu41Q9b0WdXAdFmeEbiB8wo_xggiGwv6IDR1b3TgAAsdjkRxSK4ctIKPaOJSRmM7MKMcKhIug3BEkSC9-
aXBTSIv5fAGN-ShDbPvHycBpjzKWXBvMIR5PaCg-9fwjELXZXdRwz8C6JbRM8aqzhdt4CVhQ3-Arip-S9CKd0tu-
qhHfF2rvJDRlg8ZBiihdPH8mJs-qpTFep_1-kON3mL0_g54xVmlMwN0XQA",
 "refresh_token": "iTxQYALqlc7uLyFGpnl8tR8Y9gkw91mFy2qC9Yywkz",
 "token_type": "Bearer",
 "expires_in": 7199
}
```

The MCX client then uses the access token to make authorized requests to the partner MCX resource servers (MCPTT group management service, MCVideo group management service, MCData group management service, etc) on behalf of the end user.

B.8 Security tokens

Security tokens are obtained from the primary IdMS and used for authentication with a partner IdMS.

Standard claims are REQUIRED for MCX implementation. Other claims defined by OpenID Connect are optional. The standards-based claims for an MCX Connect ID security token are shown in table B.8-1.

Table B.8-1: Security token standard claims

Parameter	Description
iss	REQUIRED. The URL of the IdM server.
Sub	REQUIRED. A case-sensitive, never reassigned string (not to exceed 255 bytes), which uniquely identifies the MCX user within the MCX server provider's domain.
Aud	REQUIRED. The Oauth 2.0 client_id of the MCX client. This field shall additionally carry the address of the target IdMS where the security token will be applied (i.e. the same value provided in the “resource” parameter from the token exchange request message).
exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds)
iat	REQUIRED. Time at which the ID Token was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

B.9 Access tokens for partner services

Access tokens obtained from a partner IdMS and used for user service authorisation to services within the partner domain shall conform to the access token requirements and format described in clause B.2.2.

B.10 Using the token to access MCX resource servers

MCX Connect shall initially support the bearer access token type. Access tokens of type "bearer" shall be communicated from the MCX client to MCX resource servers by including the access token in the HTTP Authorization Header, per IETF RFC 6750 [20].

The access token is opaque to the MCX client, meaning that the client does not have any knowledge of the access token itself. The client will be given some metadata corresponding to the access token, such as its expiration time, so that it does not send an expired access token to MCX resource servers. If the access token is presented to an MCX resource server and the scope is invalid or the token is expired or revoked, the MCX resource server should return an error message indicating such to the MCX client.

B.11 Token validation

B.11.1 ID token validation

The MCX client shall validate the ID token as per section 3.1.3.7 of the OpenID Connect 1.0 specification [21].

B.11.2 Access token validation

MCX resource servers shall validate access tokens received from the MCX client according to IETF RFC 7519 [32].

B.11.3 Security token validation

The IdM server shall validate the security token as per section 3.1.3.7 of the OpenID Connect 1.0 specification [21].

B.12 Token revocation

In order to limit the time validity of a token, the "exp" and "expires_in" parameters shall be used as a method of access token revocation.

Within the standard claims of an access token or security token, the "exp" parameter shall be used by the authorising server to determine whether or not the token is valid. If the current time is beyond the time specified by the "exp" parameter, the associated token shall no longer be considered valid and any requests made with an expired token shall be rejected by the authorising server.

Within the standard claims of an access token response, token exchange response or token response message, the "expires_in" parameter shall be used by the UE client(s) to determine validity of the associated token. If the current time is beyond the time specified by the "expires_in" parameter, the associated token shall no longer be considered valid and no client requests shall be made using the expired token.A refresh token may be used per annex B.5 to obtain a new access token.

B.12 IdMS interface security

The support of Transport Layer Security (TLS) between the IdM client in the MC UE and the IdM server is mandatory. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [5], annex E.

If PSK TLS based authentication is supported, the IdM client in the MC UE and the IdMS shall support the TLS version, PSK ciphersuites and TLS Extensions as specified in the TLS profile given in 3GPP TS 33.310 [5], annex E. The usage of pre-shared key ciphersuites for TLS is specified in the TLS profile given in 3GPP TS 33.310 [5], annex E.

Annex C (informative): OpenID connect detailed flow

C.1 Detailed flow for MC user authentication and registration using OpenID Connect

Figure C.1-1 shows the detailed flow for MC User Authentication and Registration using the OpenID Connect messages as described in annex B.

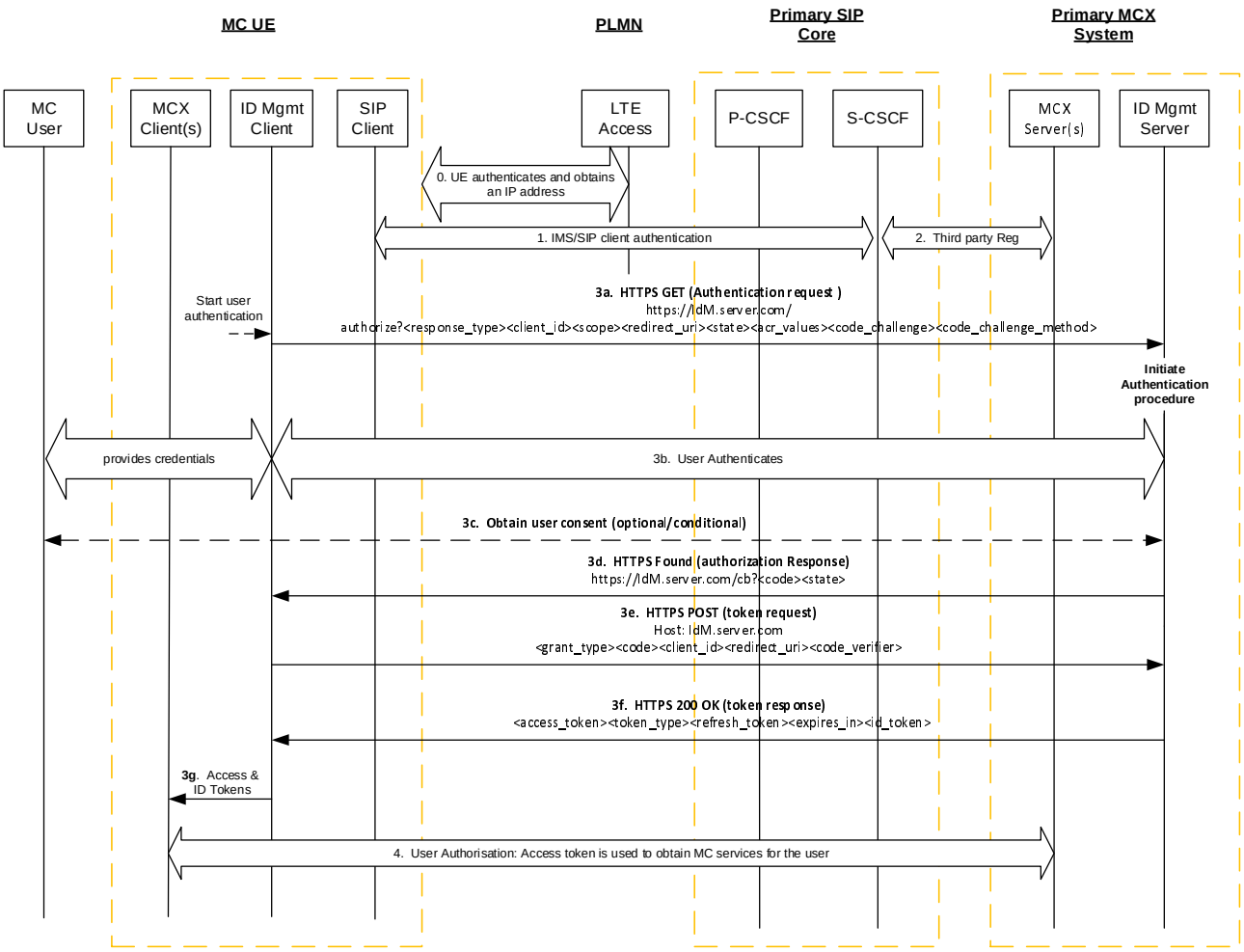

Figure C.1-1: OpenID Connect MC User Authentication and Registration

Step 0: The UE attaches to the network, establishes normal connectivity, and sets up network security as defined in 3GPP TS 33.401 [14]. Local P-CSCF in the Home IMS network is discovered at this point.

Step 1: The UE IMS/SIP Client authenticates with the primary IMS/SIP core. For IMS authentication, 3GPP TS 33.203 [9] applies.

Step 2: The SIP core sends a SIP 3rd Party Registration to the MCX application Server(s), notifying them of the MC UE SIP registration. The 3rd party REGISTER message includes the registered IMPU and S-CSCF's SIP-URI or IP Address.

Step 3a: The IdM client in the UE issues a HTTPS Authentication request to the OIDC based IdM Server in the MC network. The client includes the code_challenge value in this request.

Step 3b: The MC User Identity and associated credentials are provided to the IdM server. The credentials are successfully authenticated (and optionally authorized) by the IdM Server.

Step 3c: The IdM Server may optionally request user consent for granting the MCX client access to the MCX service in the MCX Server.

Step 3d: The IdM Server generates an authorization code that is associated with the code_challenge provided by the client. It sends a browser redirect HTTP message with the Authorization Response containing the authorization code.

Step 3e: The UE IdM Client performs a HTTP POST request to exchange the authorization code for an access token. In the request, the client includes the code-verifier string. This string is cryptographically associated with the code_challenge value provided in the Authorization Request in Step 3a.

Step 3f: The IdM Server verifies the IdM Client based on the received code-verifier string and issues a 200 OK with an access token and ID token (specific to the MC user and MCX service(s)) included in it.

NOTE: The server verifies by calculating the code challenge from the received code_verifier and comparing it with the code_challenge value provided by the client in Step 3a.

Step 3g: The access token and ID token are made available to the MCX client(s).

Step 4: The MC UE performs user service authorization.

C.2 Detailed flow for inter-domain MC user service authorization using OpenID Connect token exchange

Figure C.2-1 shows the detailed message flow for inter-domain MCX user authentication and service authorisation using the OpenID Connect token exchange method as described in Annex B.

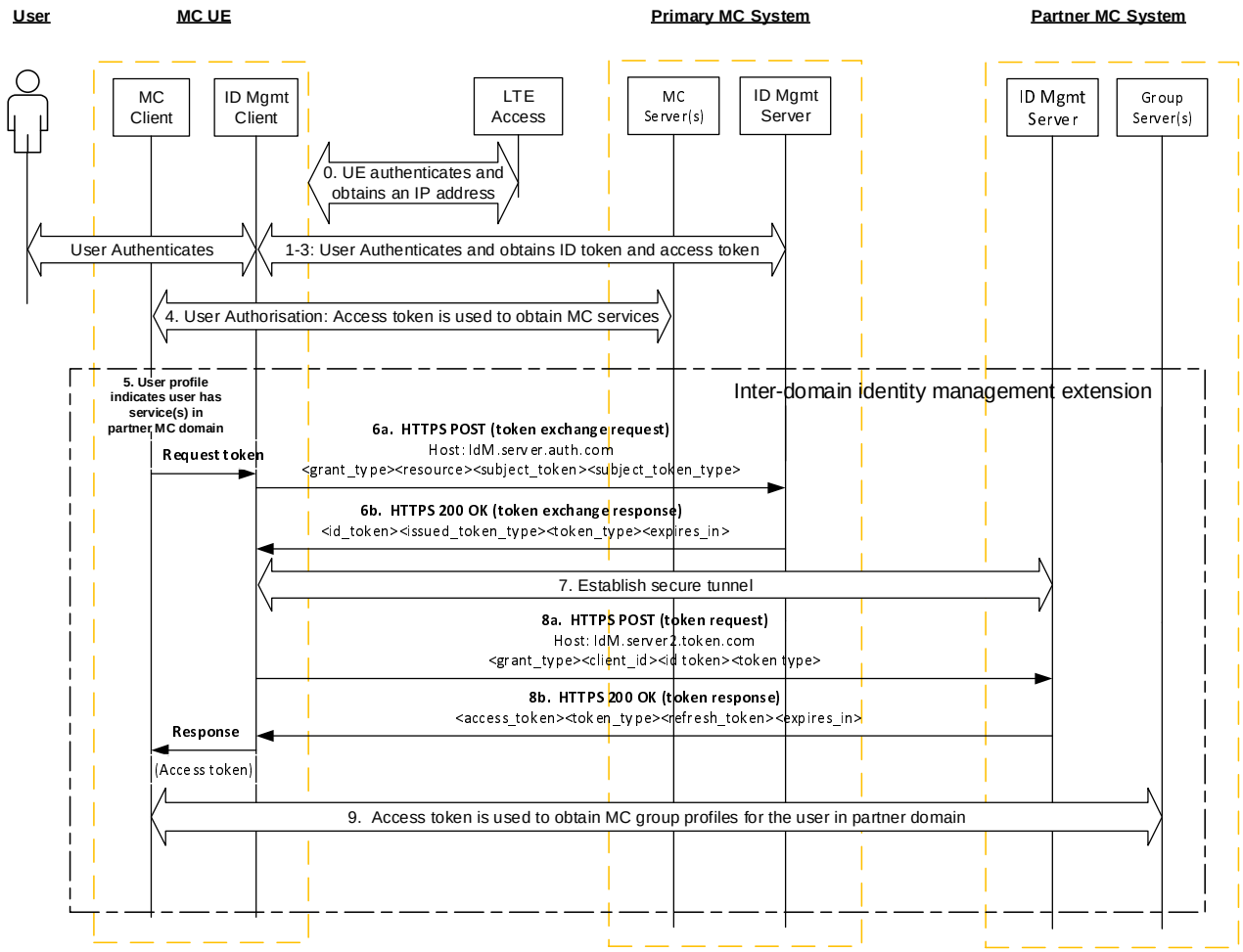

Figure C.2-1: Inter-domain user authentication and service authorisation

Steps 0-3: These steps are the same as described in steps 0-3 of Figure C.1-1, which provide the initial network access, network security, HTTPS tunnel to IdM server, user authentication, IMS authentication, and SIP registration.

Step 4: This step represents the culmination of steps C-1 through C-5 in Figure 5.1.3.1-1, which authorises the user for services in the primary domain. As part of this step the UE obtains the user's profile, which specifies both the local (primary domain) and the non-local (partner domain) group services.

Step 5: From the user's profile, the UE identifies group service(s) home to a partner domain. The user profile includes metadata of the group service(s) and information about the partner IdMS (i.e. the token endpoint host address and the "aud" parameter for use in the token exchange request).

Step 6a: Based on the OAuth token exchange procedure, the UE IdM Client performs a HTTP POST (token exchange) request to the user's primary IdM Server token endpoint. This request consists of the access token obtained in step 3 and information about the partner IdMS (i.e. the "aud" parameter obtained from the user profile group metadata).

Step 6b: The primary IdM Server token endpoint verifies the access token and returns a security token specific to the partner IdM Server.

Step 7: The UE establishes a secure HTTP tunnel with the partner IdM token endpoint using HTTPS.

Editor's Note: It is FFS how the TLS tunnel between the visiting user and the partner systems IdM server is authenticated.

Step 8a: The UE IdM Client performs a HTTP POST token request to the partner IdM token endpoint to exchange the security token for an access token. This message is defined in [19].

Step 8b: The partner IdM Server token endpoint verifies the security token and issues an access token specific to the user and the user's local MC group service(s).

NOTE 1: Additional access tokens may be requested as needed by repeating steps 8a and 8b.

Step 9: For each group service, the GM client in the UE follows the "Retrieve group configurations at the group management client" flow as shown in clause 10.1.5.2 of TS 23.280 [36], presenting an access token in the Get group configuration request over HTTP. If the access token is valid, the GMS authorises the user for the specific group management service. Completion of this step results in the GMS sending the user's group policy information and group key information to the GM client. This step is repeated for each additional group service that is home to this partner domain.

NOTE 2: Steps 5–9 are repeated for user service authorization to services in each additional partner domain.

Annex D (Normative): KMS provisioning messages

D.1 General aspects

This annex specifies the key management procedures between the KMS and the key management client that allows keys to be provisioned to the key management client based on an identity. It describes the requests and responses for the authorization following provisioning messages:

- KMS Initialize.
- KMS KeyProvision.
- KMS CertCache.
- KMS Cert.
- KMS Discovery Lookup
- KMS Discovery Upload

All KMS communications are made via HTTPS. The key management client is provisioned via XML content in the KMS's response. The XML content is designed to be extendable to allow KMS/client providers to add further information in the XML. Where the interface is extended, a different XML namespace should be used (so that may be ignored by non-compatible clients).

It is assumed that transmissions between the KMS and the key management client are secure and that the KMS has authenticated the identity of the key management client.

Additionally, to allow the transmission of key material securely between a secure element within the KMS and a secure element within the key management client, a security extension is defined which allows messages to be signed using the shared Integrity key (InK) or Transport Key (TrK) and key material to be encrypted using a shared Transport Key (TrK).

D.2 KMS requests

D.2.1 General

Requests to the KMS are made to specific resource URIs. Requests are made using a HTTP POST request to a URI. The content of the URI indicates the type of request. Resource URIs are rooted under the tree "/keymanagement/identity/v1" for a particular domain.

For example, the resource path to initialize a user within the domain "example.org" is:

EXAMPLE:

```
http://example.org/keymanagement/identity/v1/init
```

D.2.2 KMS request security

The content of the KMS Request Type XML payload is:

Table D.2.2-1: Contents of a KMS Request Type XML

Name	Description
Version	(Attribute) The version number of the key provision XML (1.1.0).
UserUri	URI of the user for which is making the request.
KmsUri	The URI of the KMS to which the request is sent.
ClientId	(Optional) A string representing the client
DeviceId	(Optional) A string representing the device
Time	Date/time that the request is made by the client.
ClientReqUrl	The resource URI to which the HTTP POST request is sent.
KrrList	(Optional) Zero or more KMS Redirect Responses (KRRs). Only used when posting to the 'redirect' subdirectory.
ClientError	(Optional) If a previous failure had occurred, this complex type can provide error information to the KMS
TrK-ID	(Optional) The ID of the TrK used for confidentiality protection of key management payloads.
Signature-ID	(Optional) The ID of the key used to sign KMS messages.

When application confidentiality is required by the MC operator, the TrK-ID of the TrK currently residing in the MC UE shall be included in the KMS request message as shown in Table D.2.2-1.

When a signature is applied to the KMS request, the Signature-ID field in Table D.2.2-1 shall be present and indicate either the InK-ID if the InK is used or the TrK-ID if the TrK is used. When a signature is applied and the InK is present, the InK shall be used. When a signature is applied and an InK is not present but a TrK is present, then the TrK shall be used.

The XML schema for the SignedKmsRequestType is provided in Clause D.3.5.1.

An optional security extension may be used to authenticate the KMS request from the client. To use the optional security extension, the POST request shall be accompanied with an XML payload MIME type containing details of the request, signed by the shared InK or TrK.

If the KMS supports authenticated requests, upon receipt of a SignedKmsRequestType attached to a KMS Request, the KMS shall verify that:

- the signature is valid, based on the UserUri and the InK or TrK used to sign the message.
- the XML is valid.
- the KmsUri is the KMS's KMS URI.
- the Time is within a recent time window (e.g. 5 seconds).
- the ClientReqUrl is the same as the resource URI to which the HTTP POST request is sent.

If so, the request is accepted and processed.

D.2.3 KMS Initialize request

To make a "KMS Initialize" request the key management client shall make a HTTP POST request to the subdirectory "init" i.e. Request-URI takes the form of:

EXAMPLE:

```
…/keymanagement/identity/v1/init
```

D.2.4 KMS KeyProvision request

To make a "KMS KeyProvision" request the key management client shall make a HTTP POST request to the subdirectory "keyprov" i.e. Request-URI takes the form of

EXAMPLE1:

```
…/keymanagement/identity/v1/keyprov
```

Optionally, the Request-URI of the POST request may contain a specific user or group URI which the key management client would like the KMS to provision. The URI shall be within a subdirectory of "keyprov". For example, the user URI "sip:user@example.org" is provisioned via a request to: "/keymanagement/identity/v1/keyprov/sip%3Auser%40example.org". Additionally, if the Request-URI contains a specific URI, the client may also request a specific time which the client would like the KMS to provision. The time URI shall be the same time as used in the MIKEY payload, a NTP-UTC 64-bit timestamp as defined in IETF RFC 5905 [29]. For example, if the user required keys specifically for 23rd Feb 2014 at 08:39:14.000 UTC, the request would be:

EXAMPLE 2:

```
…/keymanagement/identity/v1/keyprov/sip%3Auser%40example.org/D6B4323200000000
```

D.2.5 KMS CertCache request

To make a "KMS CertCache" request the key management client shall make a HTTP POST request to the subdirectory "certcache". For example, the request-URI takes the form of "/keymanagement/identity/v1/certcache". If a cache has been previously received, the request URI may optionally be directed to the subdirectory indicating the number of the client's latest version of the cache. For example, the request-URI takes the form of

EXAMPLE:

```
…/keymanagement/identity/v1/certcache/12345
```

D.2.6 KMS Cert request

"KMS Cert" requests are used to request the KMS certificate of a specific external KMS, referenced by the KMS's KMS URI.

To make a "KMS Cert" request the key management client shall make a HTTP POST request to the subdirectory "cert". Within the subdirectory "cert", the POST request shall contain a specific KMS URI of the External KMS. For example, the request-URI takes the form:

EXAMPLE:

```
…/keymanagement/identity/v1/cert/kms.example.org
```

D.2.7 KMS Lookup request

"KMS Discovery Lookup" requests are used to request the KMS certificate of a specific entity, referenced by the entity's URI.

To make a "KMS Discovery Lookup" request the key management client shall make a HTTP POST request to the subdirectory "lookup". Within the subdirectory "lookup", the POST request shall contain a specific SIP URI of the entity. For example, the request-URI takes the form:

EXAMPLE:

```
…/keymanagement/identity/v1/lookup/user%40example.org
```

The KMS responds with a list of permitted KMS URIs for the target entity.

D.2.8 KMS Redirect Upload

"KMS Redirect Upload" messages are used to upload KMS Redirect Responses (KRRs) to the KMS for audit purposes.

To send a "KMS Redirect Upload" message the key management client shall make a HTTP POST request to the subdirectory "redirect". Within the subdirectory "redirect", the POST request shall contain a specific SIP URI of the entity that resulted in the received KRR. For example, the request-URI takes the form:

EXAMPLE:

```
…/keymanagement/identity/v1/redirect/user%40example.org
```

The POST message shall be accompanied with an MIME type containing a "KMS Request Type" XML payload (as defined in Clause D.2.2). The XML payload shall contain one or more KMS Redirect Responses (KRRs). The "KMS Request Type" XML payload may also be signed as defined in Clause D.2.2.

D.3 KMS responses

D.3.1 General

This clause defines the HTTP responses made by the KMS to KMS requests. The KMS attaches XML content to the HTTP responses. The XML serves to provision the client based upon its request.

Though a "KmsResponse" message containing a "KmsMessage" Type is the general response to any request, the content of the "KmsMessage" varies depending on the exact response type (i.e. KmsInit, KmsKeyProv, KmsCertCache, KmsLookup).

The content provided within a KmsInit, KmsKeyProv, KmsCertCache or KmsLookup may include a TrK, InK, KMS URIs, (public) KMS Certificates, (private) user Key Set provisioning, or combinations thereof.

The "KmsResponse" message is shown in Table D.3.1-1.

Table D.3.1-1: Contents of a "KmsResponse" message

Name	Description
UserUri	URI of the user for which the response is intended.
KmsUri	The URI of the KMS sending the response.
KmsId	(Optional) The ID of the KMS providing the response message.
Time	Date/time that the response is sent by the KMS.
ClientReqUrl	The resource client URI from where the request originated.
KmsMessage	One of the following response types: KmsInit, KmsKeyProv, KmsCertCache, or KmsLookup.
TrK-ID	(Optional) The ID of the TrK used to confidentiality protect the KmsMessage.
Signature-ID	(Optional) The ID of the key used to sign the KmsMessage.

In response to a "KMS Initialize" request, the KMS shall respond with the KMS's own certificate (the Root KMS certificate), and may respond with a new TrK and/or a new InK. The data is returned within a "KMSInit" tag.

In response to a "KMS KeyProvision" request, the KMS shall provision appropriate user Key Sets within a "KMSKeyProv" tag, and may also respond with a new TrK and/or a new InK.

In response to a "KMS CertCache" request, the KMS shall provision a cache of KMS certificates allowing inter-domain communications within a "KMSCertCache" tag.

In response to a "KMS Cert" request, the KMS shall provision a single KMS certificate within a "KMSCertCache" tag. If the requested KMS Certificate is not available, then an error message is returned.

In response to a "KMS Lookup" request, the KMS shall provide information on the KMS URI associated with the requested SIP URI, within a "KMSLookup" tag.

The KMS does not respond to a "KMS Redirect Upload" message, unless an error occurs.

When confidentiality is applied to the KmsResponse payload (KmsMessage), the KMS shall use the TrK currently residing in the MC UE to encrypt the KmsMessage. The associated TrK-ID shall then be included in the KmsResponse message as shown in Table D.3.1-1.

When a signature is applied to the KmsResponse message, the Signature-ID field in Table D.3.1-1 shall be present and indicate either the InK-ID if the InK is used or the TrK-ID if the TrK is used. When a signature is applied and the InK is present, the InK shall be used. When a signature is applied and an InK is not present but a TrK is present, then the TrK shall be used.

The XML schema for the SignedKmsRequestType is provided in Clause D.3.5.1.

D.3.2 KMS certificates

D.3.2.1 Description

A KMS Certificate is a certificate that applies to an entire domain of users. A Certificate consists of XML containing the information required to encrypt messages to a domain of users and verify signatures from the domain of users.

A KMS has exactly one root certificate at any one time, which contains the public keys used by the KMS. The root certificate is the only certificate for which the KMS has the private keys and is able to issue user-specific key material. Should the root certificate need to be updated, a new KMS with a new KMS URI should be established with a new root certificate.

It is assumed that the user is managed by a single KMS. The root certificate for this KMS is required to encrypt messages to the user, and verify signatures from the user.

The KMS may also provision a number of 'external' KMS certificates to allow inter-domain communications.

D.3.2.2 Fields

The KMS Certificate shall be within a XML tag named "KmsCertificate". This type shall have the following subfields.

Table D.3.2.2-1: Contents of a KMS Certificate

Name	Description
Version	(Attribute) The version number of the certificate type ('1. 2.0' or '1.1.0').
Role	(Attribute) This shall indicate whether the certificate is a "Root" or "External" certificate.
CertUri	(Optional) The URI of the Certificate (this object).
KmsUri	The URI of the KMS which issued the Certificate.
Issuer	(Optional) String describing the issuing entity.
ValidFrom	(Optional) Date from which the Certificate may be used.
ValidTo	(Optional) Date at which the Certificate expires.
Revoked	(Optional) A Boolean value defining whether a Certificate has been revoked.
UserIDFormat	Shall contain the value '2', indicating that the generation mechanism defined in clause F.2.1 shall be used.
UserKeyPeriod	The number of seconds that each user key issued by this KMS should be used (e.g. '2419200').
UserKeyOffset	The offset in seconds from 0h on 1st Jan 1900 that the segmentation of key periods starts (e.g. '0').
PubEncKey	The SAKKE Public Key, "Z_T", as defined in [10]. This is an OCTET STRING encoding of an elliptic curve point.
PubAuthKey	The ECCSI Public Key, "KPAK" as defined in [9]. This is an OCTET STRING encoding of an elliptic curve point.
ParameterSet	(Optional) The choice of parameter set used for SAKKE and ECCSI (e.g. '1').
KmsDomainList	(Optional) List of domains associated with the certificate.
IsSecurityGateway	(Optional Attribute) Is 'true' if the KMS Certificate corresponds to a pseudo-KMS within a MC Security Gateway. If present, the version number of the certificate shall be '1.2.0'.

D.3.2.3 User IDs

To secure communications with a specific user, the initiator shall compose the User Identifier (UID) to which the message will be encrypted. IETF RFC 6509 [11] defines a UID generation scheme for Tel URIs, however this cannot be used with Mission Critical Services as MC Service IDs are not Tel URIs.

Clause F.2.1 defines the UID generation scheme for the Mission Critical System. This shall be identified within the KMS certificate by using the value '2' within the UserIDFormat field.

D.3.3 User Key Provision

D.3.3.1 Description

User keys are private information associated to a user's identity (UserID) which allow a user to decrypt information encrypted to that identity and sign information as that identity. User keys are provisioned as XML containing the key information required and associated metadata.

D.3.3.2 Fields

The KMS shall provision keys within an XML tag named "KmsKeySet". This shall have the following subfields.

Table D.3.3.2-1: Contents of a KMS Key Set

Name	Description
Version	(Attribute) The version number of the key provision XML (1.1.0).
KmsUri	The URI of the KMS which issued the key set.
CertUri	(Optional) The URI of the Certificate which may be used to validate the key set.
Issuer	(Optional) String describing the issuing entity.
UserUri	URI of the user for which the key set is issued.
UserID	UID corresponding to the key set.
ValidFrom	(Optional) Date and time from which the key set may be used.
ValidTo	(Optional) Date and time at which the key set expires.
KeyPeriodNo	Current Key Period No. since 1 January 1900 (e.g. 1514)
Revoked	(Optional) A Boolean value defining whether the key set has been revoked.
UserDecryptKey	The SAKKE "Receiver Secret Key" as defined in [10]. This is an OCTET STRING encoding of an elliptic curve point as defined in section 2.2 of [31].
UserSigningKeySSK	The ECCSI private Key, "SSK" as defined in [9]. This is an OCTET STRING encoding of an integer as described in section 6 of [30].
UserPubTokenPVT	The ECCSI public validation token, "PVT" as defined in [9]. This is an OCTET STRING encoding of an elliptic curve point as defined in Section 2.2 of [31].

NOTE: The key may be valid outside of its defined key period of use to enable decryption of old messages encrypted to the user.

D.3.4 Example KMS response XML

D.3.4.1 Example KMSInit XML

If the security extension is used, it is assumed that before this response is received, the secure element within the KMS and the secure element within the key management client have shared a bootstrap TrK, e.g. 'tk.11.user@example.org'.

In this example, the KMS provides the user with the KMS root certificate and a new TrK to protect future KMS communications. Keys are encrypted and the message is signed using the bootstrap TrK.

EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedKmsResponse xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                   Id="xmldoc">
  <KmsResponse Version="1.0.0">
    <UserUri>example:user@example.org</UserUri>
    <KmsUri>kms.example.org</KmsUri>     <Time>2014-01-26T10:05:52</Time>
    <KmsId>KMSProvider12345</KmsId>
    <ClientReqUrl>http://kms.example.org/keymanagement/identity/v1/init</ClientReqUrl>
    <KmsMessage>
      <KmsInit Version="1.0.0" xsi:type="KmsInitTkIkType">
        <KmsCertificate Version="1.1.0" Role="Root">
          <CertUri>cert1.kms.example.org</CertUri>
          <KmsUri>kms.example.org</KmsUri>
          <Issuer>www.example.org</Issuer>
```

```
          <ValidFrom>2000-01-26T00:00:00</ValidFrom>
          <ValidTo>2025-01-26T23:59:59</ValidTo>
          <Revoked>false</Revoked>
          <UserIdFormat>2</UserIdFormat>
          <UserKeyPeriod>2592000</UserKeyPeriod>
          <UserKeyOffset>0</UserKeyOffset>
          <PubEncKey>029A2F</PubEncKey>
          <PubAuthKey>029A2F</PubAuthKey>
          <ParameterSet>1</ParameterSet>
          <KmsDomainList>
            <KmsDomain>sec1.example.org</KmsDomain>
            <KmsDomain>sec2.example.org</KmsDomain>
          </KmsDomainList>
        </KmsCertificate>
        <NewTransportKey xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0">
          <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey">
            <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
            <ds:KeyInfo>
              <ds:KeyName>
                tk.11.user@example.org</ds:KeyName>
              </ds:KeyInfo>
            <CipherData>
              <CipherValue>DEADBEEF</CipherValue>
            </CipherData>
            <CarriedKeyName>tk.12.user@example.org</CarriedKeyName>
          </EncryptedKey>
        </NewTransportKey>
        <NewIntegrityKey xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0">
          <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey">
            <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
            <ds:KeyInfo>
              <ds:KeyName>
                tk.11.user@example.org</ds:KeyName>
              </ds:KeyInfo>
            <CipherData>
              <CipherValue>DEADBEEF</CipherValue>
            </CipherData>
            <CarriedKeyName>ink.12.user@example.org</CarriedKeyName>
          </EncryptedKey>
        </NewIntegrityKey>
      </KmsInit>
    </KmsMessage>
  </KmsResponse>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256">
        <HMACOutputLength>128</HMACOutputLength>
      </SignatureMethod>
      <Reference URI="#xmldoc">
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <DigestValue>nnnn</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>DEADBEEF</SignatureValue>
    <KeyInfo>
      <KeyName>tk.11.user@example.org</KeyName>
    </KeyInfo>
  </Signature>
</SignedKmsResponse>
```

D.3.4.2 Example KMSKeyProv XML

In this example, the user's key material is provided for two user identifiers. The key material includes the UserDecryptKey (see IETF RFC 6508 [10]) and the UserSigningKey and PVT (see IETF RFC 6507 [9]) for each identifier.

As the security extension has been used, the key material is encrypted using the shared TrK and the message signed using the shared InK. Additionally, a new TrK is provided as part of the key provision.

EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<SignedKmsResponse xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                   Id="xmldoc">
  <KmsResponse Version="1.0.0">
    <UserUri>example:user@example.org</UserUri>
    <KmsUri>kms.example.org</KmsUri>    <Time>2014-01-26T10:07:14</Time>
    <KmsId>KMSProvider12345</KmsId>
    <ClientReqUrl>http://kms.example.org/keymanagement/identity/v1/keyprov</ClientReqUrl>
    <KmsMessage>
      <KmsKeyProv Version = "1.0.0" xsi:type="KmsKeyProvTkIkType">
        <KmsKeySet Version = "1.1.0">
          <KmsUri>kms.example.org</KmsUri>
          <CertUri>cert1.kms.example.org</CertUri>
          <Issuer>www.example.org</Issuer>
          <UserUri>example:user@example.org</UserUri>
          <UserID>0123456789ABCDEF0123456789ABCDEF</UserID>
          <ValidFrom>2015-12-30T00:00:00</ValidFrom>
          <ValidTo>2016-03-29T23:59:59</ValidTo>
          <KeyPeriodNo>1514</KeyPeriodNo>
          <Revoked>false</Revoked>
          <UserDecryptKey xsi:type="EncKeyContentType">
            <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserDecryptKey>
          <UserSigningKeySSK xsi:type="EncKeyContentType">
            <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserSigningKeySSK>
          <UserPubTokenPVT xsi:type="EncKeyContentType">
            <EncryptedKey xmlns = "http://www.w3.org/2001/04/xmlenc#">
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserPubTokenPVT>
        </KmsKeySet>
        <KmsKeySet Version = "1.1.0">
          <KmsUri>kms.example.org</KmsUri>
          <CertUri>cert1.kms.example.org</CertUri>
          <Issuer>www.example.org</Issuer>
          <UserUri>example:user.pseudonym@example.org</UserUri>
          <UserID>00112233445566778899AABBCCDDEEFF</UserID>
          <ValidFrom>2015-12-30T00:00:00</ValidFrom>
          <ValidTo>2016-03-29T23:59:59</ValidTo>
          <KeyPeriodNo>1514</KeyPeriodNo>
          <Revoked>false</Revoked>
          <UserDecryptKey xsi:type="EncKeyContentType">
            <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserDecryptKey>
          <UserSigningKeySSK xsi:type="EncKeyContentType">
            <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
```

```
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserSigningKeySSK>
          <UserPubTokenPVT xsi:type="EncKeyContentType">
            <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
              <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
              <ds:KeyInfo>
                <ds:KeyName>tk.12.user@example.org</ds:KeyName>
                </ds:KeyInfo>
              <CipherData>
                <CipherValue>DEADBEEF</CipherValue>
              </CipherData>
            </EncryptedKey>
          </UserPubTokenPVT>
        </KmsKeySet>
        <NewTransportKey>
          <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey">
            <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#kw-aes256"/>
            <ds:KeyInfo>
              <ds:KeyName>tk.12.user@example.org</ds:KeyName>
              </ds:KeyInfo>
            <CipherData>
              <CipherValue>DEADBEEF</CipherValue>
            </CipherData>
            <CarriedKeyName>tk.13.user@example.org</CarriedKeyName>
          </EncryptedKey>
        </NewTransportKey>
      </KmsKeyProv>
    </KmsMessage>
  </KmsResponse>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256">
        <HMACOutputLength>128</HMACOutputLength>
      </SignatureMethod>
      <Reference URI="#xmldoc">
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <DigestValue>nnnn</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>DEADBEEF</SignatureValue>
    <KeyInfo>
      <KeyName>ink.12.user@example.org</KeyName>
    </KeyInfo>
  </Signature>
</SignedKmsResponse>
```

D.3.4.3 Example KMSCertCache XML

In this example, a number of 'external' KMS certificates are provided to the user. These allow the user to encrypt to users managed by a different KMS.

As the security extension is in use, the message is signed using the shared InK.

EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignedKmsResponse xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0"
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                   Id="xmldoc">
  <KmsResponse Version="1.0.0">
    <UserUri>example:user@example.org</UserUri>
    <KmsUri>kms.example.org</KmsUri>
    <Time>2014-01-26T10:14:12</Time>
    <KmsId>KMSProvider12345</KmsId>
    <ClientReqUrl>http://kms.example.org/keymanagement/identity/v1/certcache</ClientReqUrl>
    <KmsMessage>
      <KmsCertCache Version = "1.0.0">
```

```
      <SignedKmsCertificate Id = "cert1">
        <KmsCertificate Version = "1.1.0" Role = "External">
          <CertUri>cert2.kms.example.org</CertUri>
          <KmsUri>kms.example.org</KmsUri>
          <Issuer>www.example.org</Issuer>
          <ValidFrom>2000-01-26T00:00:00</ValidFrom>
          <ValidTo>2100-01-26T23:59:59</ValidTo>
          <Revoked>false</Revoked>
          <UserIdFormat>2</UserIdFormat>
          <UserKeyPeriod>2592000</UserKeyPeriod>
          <UserKeyOffset>0</UserKeyOffset>
          <PubEncKey>029A2F</PubEncKey>
          <PubAuthKey>029A2F</PubAuthKey>
          <ParameterSet>1</ParameterSet>
          <KmsDomainList>
            <KmsDomain>sec3.example.org</KmsDomain>
          </KmsDomainList>
        </KmsCertificate>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
          <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
            <Reference URI="#cert1">
              <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
              <DigestValue>nnnn</DigestValue>
            </Reference>
          </SignedInfo>
          <SignatureValue>DEADBEEF</SignatureValue>
          <KeyInfo>
            <KeyName>cert1.kms.example.org</KeyName>
          </KeyInfo>
        </Signature>
      </SignedKmsCertificate>
      <SignedKmsCertificate Id="cert2">
        <KmsCertificate Version="1.1.0" Role="External">
          <CertUri>cert1.kms.another.example.org</CertUri>
          <KmsUri>kms.another.example.org</KmsUri>
          <Issuer>www.another.example.org</Issuer>
          <ValidFrom>2000-01-26T00:00:00</ValidFrom>
          <ValidTo>2100-01-26T23:59:59</ValidTo>
          <Revoked>false</Revoked>
          <UserIdFormat>2</UserIdFormat>
          <UserKeyPeriod>604800</UserKeyPeriod>
          <UserKeyOffset>432000</UserKeyOffset>
          <PubEncKey>029A2F</PubEncKey>
          <PubAuthKey>029A2F</PubAuthKey>
          <ParameterSet>1</ParameterSet>
          <KmsDomainList>
            <KmsDomain>another.example.org</KmsDomain>
          </KmsDomainList>
        </KmsCertificate>
        <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
          <SignedInfo>
            <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
            <Reference URI="#cert2">
              <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
              <DigestValue>nnnn</DigestValue>
            </Reference>
          </SignedInfo>
          <SignatureValue>DEADBEEF</SignatureValue>
          <KeyInfo>
            <KeyName>cert1.kms.example.org</KeyName>
          </KeyInfo>
        </Signature>
      </SignedKmsCertificate>
    </KmsCertCache>
  </KmsMessage>
</KmsResponse>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#hmac-sha256">
      <HMACOutputLength>128</HMACOutputLength>
    </SignatureMethod>
    <Reference URI="#xmldoc">
      <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <DigestValue>nnnn</DigestValue>
```

```
      </Reference>
    </SignedInfo>
    <SignatureValue>DEADBEEF</SignatureValue>
    <KeyInfo>
      <KeyName>ink.12.user@example.org</KeyName>
    </KeyInfo>
  </Signature>
</SignedKmsResponse>
```

D.3.5 KMS response XML schema

D.3.5.1 Base XML schema

This clause contains the XML schema for KMS responses. This will validate Version '1.1.0' or '1.2.0' certificates:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
            xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
            xmlns:krr="urn:3gpp:ns:mcsecKMSKRR:1.0"
            xmlns="urn:3gpp:ns:mcsecKMSInterface:1.0"
            targetNamespace="urn:3gpp:ns:mcsecKMSInterface:1.0"
            elementFormDefault="qualified" version="1.0">

  <xsd:import namespace = "http://www.w3.org/2000/09/xmldsig#" />
  <xsd:import namespace = "http://www.w3.org/2001/04/xmlenc#" />
  <xsd:import namespace ="urn:3gpp:ns:mcsecKMSKRR:1.0"/>

  <!-- Global elements -->
  <xsd:element name="KmsRequest" type="KmsRequestType" />
  <xsd:element name="SignedKmsRequest" type="SignedKmsRequestType"/>

  <xsd:element type="KmsResponseType" name="KmsResponse"/>
  <xsd:element type="SignedKmsResponseType" name="SignedKmsResponse"/>

  <!-- KMS Request Type definitions (see clause D.2.2) -->
  <xsd:complexType name = "KmsRequestType">
    <xsd:sequence>
      <xsd:element name="UserUri" type="xsd:anyURI"/>
      <xsd:element name="KmsUri" type="xsd:anyURI"/>
      <xsd:element name="Time" type="xsd:dateTime"/>
      <xsd:element name="ClientId" type="xsd:string" minOccurs="0"/>
      <xsd:element name="DeviceId" type="xsd:string" minOccurs="0"/>
      <xsd:element name="ClientReqUrl" type="xsd:anyURI"/>
      <xsd:element name="KrrList" type="krr:KmsRedirectResponseType" minOccurs="0"></xsd:element>
      <xsd:element name="ClientError" type="ErrorType" minOccurs="0"/>
      <!-- Can extend in another namespace - for more types of communication-->
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string" fixed="1.1.0"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="SignedKmsRequestType">
    <xsd:sequence>
      <xsd:element name="KmsRequest" type="KmsRequestType"/>
      <xsd:element ref="ds:Signature"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name = "ErrorType">
    <xsd:sequence>
      <xsd:element type = "xsd:integer" name = "ErrorCode" maxOccurs = "1"/>
      <xsd:element type = "xsd:string" name = "ErrorMsg" maxOccurs = "1"/>
      <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0" maxOccurs =
"unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Id" type = "xsd:string"/>
    <xsd:attribute name = "Version" type = "xsd:string"/>
    <xsd:anyAttribute namespace = "##other" processContents = "lax"/>
  </xsd:complexType>

  <!-- KMS Response Type definitions (see clause D.2.3) -->
  <xsd:complexType name="KmsResponseType">
    <xsd:sequence>
      <xsd:element name="UserUri" type="xsd:anyURI"/>
      <xsd:element name="KmsUri" type="xsd:anyURI"/>
      <xsd:element name="Time"  type="xsd:dateTime"/>
      <xsd:element name="KmsId" type="xsd:string" minOccurs = "0"/>
      <xsd:element name="ClientReqUrl" type = "xsd:anyURI"/>
      <xsd:element name="KmsMessage" type="KMSMessage" minOccurs = "0" />
```

```
      <xsd:element name="KmsError" type="ErrorType" minOccurs = "0"/>
      <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0" maxOccurs =
"unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string" fixed="1.0.0"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>


  <xsd:complexType name="SignedKmsResponseType">
    <xsd:sequence>
      <xsd:element ref="KmsResponse"/>
      <xsd:element ref="ds:Signature" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="KMSMessage">
    <xsd:choice>
      <xsd:element name="KmsInit" type="KmsInitType"/>
      <xsd:element name="KmsKeyProv" type="KmsKeyProvType"/>
      <xsd:element name="KmsCertCache" type="KmsCertCacheType"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="KmsInitType">
    <xsd:sequence>
      <xsd:choice>
        <xsd:element name="SignedKmsCertificate" type="SignedKmsCertificateType"/>
        <xsd:element name="KmsCertificate" type="KmsCertificateType"/>
      </xsd:choice>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="KmsKeyProvType">
    <xsd:sequence>
      <xsd:element name="KmsKeySet" type="KmsKeySetType" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string" fixed="1.0.0"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="KmsCertCacheType">
    <xsd:sequence>
      <xsd:element name="SignedKmsCertificate" type="SignedKmsCertificateType" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element name="KmsCertificate" type="KmsCertificateType" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string" fixed="1.0.0"/>
    <xsd:attribute name="CacheNum" type="xsd:integer"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <!-- KmsCertificate definition - see clause D.3.2.2 -->
  <xsd:element name = "KmsCertificate" type = "KmsCertificateType"/>
  <xsd:complexType name = "KmsCertificateType">
    <xsd:sequence>
      <xsd:element name="CertUri" type="xsd:anyURI" minOccurs = "0"/>
      <xsd:element name="KmsUri" type="xsd:anyURI"/>
      <xsd:element name="Issuer" type="xsd:string" minOccurs = "0"/>
      <xsd:element name="ValidFrom" type="xsd:dateTime" minOccurs = "0"/>
      <xsd:element name="ValidTo" type="xsd:dateTime" minOccurs = "0"/>
      <xsd:element name="Revoked" type="xsd:boolean" minOccurs = "0"/>
      <xsd:element name="UserIdFormat" type="xsd:string"/>
      <xsd:element name="UserKeyPeriod" type="xsd:integer"/>
```

```
      <xsd:element name="UserKeyOffset" type="xsd:integer"/>
      <xsd:element name="PubEncKey" type="xsd:hexBinary"/>
      <xsd:element name="PubAuthKey" type="xsd:hexBinary"/>
      <xsd:element name="ParameterSet" type="xsd:integer" minOccurs = "0"/>
      <xsd:element name="KmsDomainList" minOccurs = "0">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element type = "xsd:anyURI" name = "KmsDomain" maxOccurs = "unbounded"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0" maxOccurs =
"unbounded"/>
    </xsd:sequence>
    <xsd:attribute name = "Id" type = "xsd:string"/>
    <xsd:attribute name = "Version" type = "xsd:string"/>
    <xsd:attribute name = "Role" type = "RoleType"/>
    <xsd:attribute name = "IsSecurityGateway" type = "xsd:boolean" use="optional"/>
    <xsd:anyAttribute namespace = "##other" processContents = "lax"/>
  </xsd:complexType>

  <xsd:simpleType name = "RoleType">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = "Root"/>
      <xsd:enumeration value = "External"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:element name="SignedKmsCertificate" type="SignedKmsCertificateType"/>
  <xsd:complexType name="SignedKmsCertificateType">
    <xsd:sequence>
      <xsd:element name="KmsCertificate" type="KmsCertificateType"/>
      <xsd:element ref="ds:Signature" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:element name="KmsKeySet" type="KmsKeySetType"/>

  <xsd:complexType name = "KmsKeySetType">
    <xsd:sequence>
      <xsd:element name="KmsUri" type="xsd:anyURI"/>
      <xsd:element name="CertUri" type="xsd:anyURI" minOccurs = "0"/>
      <xsd:element name="Issuer" type="xsd:string" minOccurs = "0"/>
      <xsd:element name="UserUri" type="xsd:anyURI"/>
      <xsd:element name="UserID" type="xsd:string"/>
      <xsd:element name="ValidFrom" type="xsd:dateTime" minOccurs = "0"/>
      <xsd:element name="ValidTo" type="xsd:dateTime" minOccurs = "0"/>
      <xsd:element name="KeyPeriodNo" type="xsd:integer"/>
      <xsd:element name="Revoked" type="xsd:boolean" minOccurs = "0"/>
      <xsd:element name="UserDecryptKey" type="abstractKeyContentType"/>
      <xsd:element name="UserSigningKeySSK" type="abstractKeyContentType"/>
      <xsd:element name="UserPubTokenPVT" type="abstractKeyContentType"/>
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:string"/>
    <xsd:attribute name="Version" type="xsd:string" fixed="1.1.0"/>
    <xsd:anyAttribute namespace="##other" processContents="lax"/>
  </xsd:complexType>

  <xsd:complexType name="abstractKeyContentType" abstract="true" mixed="true" />

  <xsd:complexType name = "KeyContentType">
    <xsd:simpleContent>
      <xsd:restriction base = "abstractKeyContentType">
        <xsd:simpleType>
          <xsd:restriction base="xsd:hexBinary"></xsd:restriction>
        </xsd:simpleType>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>

  <xsd:complexType name="EncKeyContentTypeMixed" mixed="false" abstract="true">
    <xsd:complexContent>
      <xsd:restriction base="abstractKeyContentType">
        <xsd:sequence>
        </xsd:sequence>
      </xsd:restriction>
```

```
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="EncKeyContentType">
    <xsd:complexContent>
      <xsd:extension base="EncKeyContentTypeMixed">
        <xsd:sequence>
          <xsd:element ref="xenc:EncryptedKey"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="KmsInitTkIkType">
    <xsd:complexContent>
      <xsd:extension base="KmsInitType">
        <xsd:sequence>
          <xsd:element type="EncKeyContentType" name="NewTransportKey" maxOccurs="unbounded"
minOccurs="0"/>
          <xsd:element type="EncKeyContentType" name="NewIntegrityKey" maxOccurs="unbounded"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name = "KmsKeyProvTkIkType">
    <xsd:complexContent>
      <xsd:extension base="KmsKeyProvType">
        <xsd:sequence>
          <xsd:element type="EncKeyContentType" name="NewTransportKey" maxOccurs="unbounded"
minOccurs="0"/>
          <xsd:element type="EncKeyContentType" name="NewIntegrityKey" maxOccurs="unbounded"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

</xsd:schema>
```

D.3.5.2 Void

D.4 KMS Redirect Response (KRR)

D.4.1 General

A KMS Redirect Response is generated by an entity within the MC system on detection that a KMS URI within a MIKEY I_MESSAGE is not acceptable in the current circumstance, or on receipt of a KRR containing KMS URIs that are not acceptable to be returned to the sender.

The entity returns a SIP 488 response 'Not Acceptable Here', attaching an XML MIME body to the response. The MIME body shall have content type: 'application/vnd.3gpp.kmsredirectresponse+xml'. The MIME body shall adhere to the XML schema in Clause D.4.4.

D.4.2 KRR XML signature profile

Signatures should be used to authenticate KRRs. Where supported, senders and processors of KRRs shall support the following XML Signature 1.1 profile as defined in [28]:

Digest: SHA256 (http://www.w3.org/2001/04/xmlenc#sha256)

Signature: ECDSAwithSHA256 (http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256)

Canonicalization: http://www.w3.org/TR/2001/REC-xml-c14n-20010315

KeyInfo: X509 certificate

Verification of the X509 certificate is out of scope of this document.

D.4.3 Example XML

In this example, 'example:initiator@example.org' has sent a MIKEY I_MESSAGE to 'example:receiver@example.org' with KMS 'kms.reject.example.org'. At some point along the message flow, the KMS used for the receiver is rejected by entity 'example:processor.1@example.org'. The rejecting entity creates and returns a KRR proposing two KMS URIs for the receiver ('kms.option1.example.org' and 'kms.option2.example.org'). The KRR is received by 'example:processor.2@example.org' who rejects one of the proposed KMS URIs and creates a new KRR, embedding the old KRR within the new KRR and updating the list of receiver KMS URIs to the one that is acceptable ('kms.option1.example.org').

```
<?xml version="1.0" encoding="UTF-8"?>
<KmsRedirectResponse xmlns= "urn:3gpp:ns:mcsecKMSKRR:1.0"
                     xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
                     xmlns:ds = "http://www.w3.org/2000/09/xmldsig#"
                     Version = "1.0.0" Id = "0123456789abcdef">
  <Time>2018-01-26T11:15:43</Time>
  <KRRCreatorUri>example:processor.2@example.org</KRRCreatorUri>
  <InitiatorUri>example:initiator@example.org</InitiatorUri>
  <InitiatorKmsUri>kms.init.example.org</InitiatorKmsUri>
  <ReceiverUri>example:receiver@example.org</ReceiverUri>
  <ReceiverKmsUri>kms.reject.example.org</ReceiverKmsUri>
  <InitiatorKmsList>
    <ANY></ANY>
  </InitiatorKmsList>
  <ReceiverKmsList>
    <KmsUri>kms.option1.example.org</KmsUri>
  </ReceiverKmsList>
  <ReceivedKmsRedirectResponse>
    <KmsRedirectResponse Version = "1.0.0" Id = "001122334455667788">
      <Time>2018-01-26T11:15:40</Time>
      <KRRCreatorUri>example:processor.1@example.org</KRRCreatorUri>
      <InitiatorUri>example:initiator@example.org</InitiatorUri>
      <InitiatorKmsUri>kms.init.example.org</InitiatorKmsUri>
      <ReceiverUri>example:receiver@example.org</ReceiverUri>
      <ReceiverKmsUri>kms.reject.example.org</ReceiverKmsUri>
      <InitiatorKmsList>
        <ANY></ANY>
      </InitiatorKmsList>
      <ReceiverKmsList>
        <KmsUri>kms.option1.example.org</KmsUri>
        <KmsUri>kms.option2.example.org</KmsUri>
      </ReceiverKmsList>
      <Signature xmlns = "http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
          <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
          <Reference URI="#001122334455667788">
            <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <DigestValue>...</DigestValue>
          </Reference>
        </SignedInfo>
        <SignatureValue>...</SignatureValue>
        <KeyInfo>
          <X509Data>
            ...
          </X509Data>
        </KeyInfo>
      </Signature>
    </KmsRedirectResponse >
  </ReceivedKmsRedirectResponse>
  <Signature>
    <SignedInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <CanonicalizationMethod Algorithm=" http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
      <Reference URI="#0123456789abcdef">
        <DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <DigestValue>...</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue xmlns="http://www.w3.org/2000/09/xmldsig#">...</SignatureValue>
```

```
    <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <X509Data>
        ...
      </X509Data>
    </KeyInfo>
  </Signature>
</KmsRedirectResponse>
```

D.4.4 Example XML schema

This clause contains the XML schema for KMS Redirect Responses:

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema" xmlns:ds =
"http://www.w3.org/2000/09/xmldsig#"
  xmlns = "urn:3gpp:ns:mcsecKMSKRR:1.0" targetNamespace = "urn:3gpp:ns:mcsecKMSKRR:1.0"
elementFormDefault = "qualified" version = "1.0">
  <xsd:import namespace = "http://www.w3.org/2000/09/xmldsig#"/>

    <!-- An identifier for this type of response -->
  <xsd:element type = "KmsRedirectResponseType" name = "KmsRedirectResponse"/>

  <xsd:complexType name = "KmsRedirectResponseType">
    <xsd:sequence>
      <!-- The date and time -->
      <xsd:element type = "xsd:dateTime" name = "Time" minOccurs = "1" maxOccurs = "1"/>
      <!-- The identity of the KRR creator. -->
      <xsd:element type = "xsd:anyURI" name = "KRRCreatorUri" minOccurs = "1" maxOccurs = "1"/>
      <!-- The MIKEY initiating identity used within the MIKEY message (IDRi). -->
      <xsd:element type = "xsd:anyURI" name = "InitiatorUri" minOccurs = "1" maxOccurs = "1"/>
      <!-- The MIKEY initiating KMS URI used within the MIKEY message (IDRkmsi). -->
      <xsd:element type = "xsd:anyURI" name = "InitiatorKmsUri" minOccurs = "1" maxOccurs = "1"/>
      <!-- The MIKEY receiving identity used within the MIKEY message (IDRr). -->
      <xsd:element type = "xsd:anyURI" name = "ReceiverUri" minOccurs = "1" maxOccurs = "1"/>
      <!-- The MIKEY receiving KMS URI used within the MIKEY message (IDRkmsr). -->
      <xsd:element type = "xsd:anyURI" name = "ReceiverKmsUri" minOccurs = "1" maxOccurs = "1"/>
      <!-- The initiator list containing a list of acceptable KMS URIs (List of IDRkmsi's). -->
      <xsd:element name = "InitiatorKmsList" type = "KmsUriListType" minOccurs = "1" maxOccurs =
"1"/>
      <!-- The receiver list containing a list of acceptable KMS URIs (List of IDRkmsr's). -->
      <xsd:element name = "ReceiverKmsList" type = "KmsUriListType" minOccurs = "1" maxOccurs =
"1"/>
      <!-- An embedded received KRR (optional, used if KRR is generated from a received KRR). -->
      <xsd:element name = "ReceivedKmsRedirectResponse" type = "ReceivedKmsRedirectResponseType"
minOccurs = "0" maxOccurs = "1"/>
      <!-- Allow extensions -->
      <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0" maxOccurs =
"unbounded"/>
      <!-- A signature (using the originating identity) over the entire message (optional, but
recommended). -->
      <xsd:element name="Signature" type = "ds:SignatureType" minOccurs = "0"/>
    </xsd:sequence>
    <xsd:attribute name = "Id" type = "xsd:string"/>
    <xsd:attribute name = "Version" type = "xsd:string" fixed="1.0.0"/>
    <xsd:anyAttribute namespace = "##other" processContents = "lax"/>
  </xsd:complexType>

  <xsd:complexType name = "KmsUriListType">
    <xsd:sequence>
      <xsd:choice maxOccurs = "1">
        <xsd:sequence>
          <xsd:element type="xsd:string" name="ANY" fixed=""/>
        </xsd:sequence>
        <xsd:sequence>
          <xsd:element type = "xsd:anyURI" name = "KmsUri" minOccurs = "0" maxOccurs = "unbounded"/>
        </xsd:sequence>
      </xsd:choice>
      <xsd:any namespace = "##other" processContents = "lax" minOccurs = "0" maxOccurs =
"unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace = "##other" processContents = "lax"/>
  </xsd:complexType>

  <xsd:complexType name = "ReceivedKmsRedirectResponseType">
    <xsd:sequence>
      <xsd:element name = "KmsRedirectResponse" type = "KmsRedirectResponseType" minOccurs = "1"
maxOccurs = "1"/>
```

```
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Annex E (normative): MIKEY message formats for media security

E.1 General aspects

E.1.1 Introduction

MIKEY-SAKKE as defined in IETF RFC 6509 [11] is used to transport Group Master Keys (GMKs) from a Group Management Server to a Group Management Client on a MC UE, Private Call Keys (PCKs) between MC UEs, Client-Server keys (CSKs) between MCX Server and MC client, and Multicast Signalling Keys (MuSiK) from MCX Servers to MC clients.

The GMK is encrypted to the UID generated from the receiving user's MC Service user ID and current time period. It is signed using the UID generated from the URI associated to the Group Management Server and current time period. Similarly, the PCK is encrypted to the UID generated from the receiving user's MC Service user ID and current time period. It is signed using the UID generated from the initiating user's MC Service user ID and current time period. When uploaded, the CSK is encrypted to the UID generated from the MCX Server's FQDN and current time period and signed using the UID of the MC user. When downloaded, the CSK and MuSiK is encrypted to the UID of the MC user and signed using the UID of the MCX Server. Details of this process are defined in IETF RFC 6508 [10] and IETF RFC 6507 [9]. The generation of the MIKEY-SAKKE UID is defined in clause F.2.1.

The GMK, PCK, CSK and MuSiK shall be 16 octets in length.

E.1.2 MIKEY common fields

All MIKEY-SAKKE messages shall include the Common Header payload (HDR), Timestamp payload (TS), RAND payload, IDRi payload, IDRr payload, IDRkmsi payload, IDRkmsr payload, SAKKE payload and a SIGN (ECCSI) payload.

Optionally, the MIKEY-SAKKE message may contain a Security Properties payload (SP), a second SAKKE payload (SAKKE-to-self specified in Annex E.5), and a key parameter payload (specified in Annex E.6)

In the MIKEY HDR, the 'data type' shall be '26' (as this is a MIKEY-SAKKE message). The 'V' bit shall be '0'. The 'PRF func' may be '1' indicating the use of 'PRF-HMAC-SHA-256' ('PRF-HMAC-SHA-256' is the only PRF algorithm that is mandatory to support). The 'CS#' may be 0 or more.

- Where the 'CS#' is '0', the 'CS ID map type' shall be '1' (empty map) and 'CS ID Map Info' shall have length '0'. This shall imply that default security policies shall be applied (as defined in further clauses).
- Where the 'CS#' is greater than '0', the 'CS ID map type' shall be '2' (GENERIC-ID as defined in RFC 6043 [25]).

Each MIKEY message contains the timestamp field (TS). The timestamp field shall be TS type NTP-UTC (TS type 0), and hence is a 64-bit UTC time.

The ID Scheme in the SAKKE payload shall be '3GPP MCX hashed UID ' to reflect the generation scheme defined in clause F.2.1.

The ID Scheme '3GPP MCX hashed UID' takes on the IANA assigned value of '2' [52].

The entire MIKEY message shall be signed by including an SIGN payload providing authentication of the origin of the message. The signature shall be of type '2' (ECCSI).

E.1.3 Crypto Session Identifiers

The MIKEY payload defines the use of Crypto Sessions. Each Crypto Session is identified by a CS-ID. To ensure that a crypto session can be assigned to a specific use within the MC System, the Crypto Session identifiers are defined in Table E.1.3-1.

Table E.1.3-1: CS-ID assignment

CS-ID	Use
0	Initiator's MCPTT Private Call
1	Receiver's MCPTT Private Call
2	Initiator's MCVideo Private Call
3	Receiver's MCVideo Private Call
4	MCPTT Group Call
5	MCVideo Group Call
6	CSK SRTCP protection for MCPTT
7	MuSiK SRTCP protection for MCPTT
8	CSK SRTCP protection for MCVideo
9	MuSiK SRTCP protection for MCVideo

In Table E.1.3-1, CS-ID '0' and '2' are used for SRTP/SRTCP streams originating from the initator of the private call. CS-ID '1' and '3' are used for SRTP/SRTCP streams originating from the receiver of the private call.

E.2 MIKEY message structure for GMK distribution

E.2.1 General

In the Common Header payload, the CSB ID field of MIKEY common header shall be the GUK-ID.

Where no crypto sessions are included in the payload, (CS# is 0), the default security profile defined in Annex E.2.2 shall be used, and no Secuirty Properties payload (SP) is required. The profile in Annex E.2.2 is mandatory to support.

Identity payloads shall be IDR payloads as defined in section 6.6 of IETF RFC 6043 [25]. The IDRi payload shall contain the MCX service identifier associated with the group management server. The IDRr payload shall contain the MC Service user ID associated to the group management client. The message shall also include IDRkmsi and IDRkmsr that contains the URI of the MC KMS used by the group management server and MC user respectively.

NOTE: In some deployments MC Service user IDs (i.e. MCPTT ID, MCVideo ID, MCData ID) within these payloads may be treated as private. In this case, these identities may be hidden using the mechanism in clause E.7.

The SAKKE payload shall encapsulate the GMK to the UID generated from the MC Service user ID of the group management client. Only one GMK key shall be transported in the SAKKE payload. The same GMK shall be encapsulated to each member of the group.

A SAKKE-to-SELF payload may be included. It is recommended that where the GMK is being transported beyond a single MC system, the message should include a SAKKE-to-SELF payload as described in clause E.5.

A 'Key Properties' payload (Annex E.6) should be included to provide details of the GMK.

The signature shall use the UID generated from the identifier associated with the group management server.

E.2.2 Default SRTP security profile for GMK use

The default security profile is used to support MCPTT and MCVideo communications. It defines the mandatory to support security settings for distribution and use of the GMK. It is the profile that should be used should no information (Crypto session information or security policies) be provided in the MIKEY message.

The CS-ID (for input into the MIKEY PRF) shall be '4' for MCPTT and '5' for MCVideo. The 'Prot Type' shall be '0' (SRTP).

The Security Policies are shown in Table E.2-1.

Table E.2.2-1: MIKEY Group call SRTP Default Profile

SRTP Type	Meaning	Value	Meaning
0	Encryption Algorithm	6	AES-GCM
1	Session encryption key length	16	16 octets
2	Authentication algorithm	4	RCCm3 (Use of unauthenticated ROC)
4	Session salt key length	12	12 octets
5	SRTP PRF	0	AES-CM
6	Key derivation rate	0	No session key refresh.
13	ROC transmission rate	1	ROC transmitted in every packet.
18	SRTP Authentication tag length	4	4 octets for transmission of ROC
19	SRTCP Authentication tag length	0	ROC need not be transmitted in SRTCP.
20	AEAD authentication tag length	16	16 octets

Should a security profile be provided by the GMS, the mapping is provided in a GENERIC-ID component of the MIKEY HDR. The CS-ID shall be '4' for MCPTT and/or '5' for MCVideo. Consequently, the CS# shall be '1' or '2'. The 'Prot Type' shall be '0' (SRTP).

In each GENERIC-ID crypto session, '#P' shall be 1 (a single security policy shall be referenced). The 'Session Data length' shall be '0' as SSRCs are not provided by the GMS. The MKI (GMK-ID || GUK-ID) may be included in the SPI field.

E.3 MIKEY message structure for PCK distribution

E.3.1 General

In the Common Header payload, the CSB ID field of MIKEY common header shall be the PCK-ID.

Where no crypto sessions are included in the payload, (CS# is 0), the default security profile defined in Annex E.3.2 shall be used, and no Secuirty Properties payload (SP) is required. The profile in Annex E.3.2 is mandatory to support.

Identity payloads shall be IDR payloads as defined in section 6.6 of IETF RFC 6043 [25]. The IDRi payload shall contain the MC Service user ID associated with the initiating user. The IDRr payload shall contain the MC Service user ID associated to the receiving user. The message shall also include IDRkmsi and IDRkmsr that contains the URI of the KMS used by the initiating user and terminating user respectively

NOTE: In some deployments MC Service user IDs (i.e. MCPTT ID, MCVideo ID, MCData ID) within these payloads may be treated as private. In this case, these identities may be hidden using the mechanism in clause E.7.

The SAKKE payload shall encapsulate the PCK to the UID generated from the MC Service user ID of the terminating user. The ID Scheme in the SAKKE payload shall be 'URI Scheme' to reflect the generation scheme defined in clause F.2.1.

A SAKKE-to-SELF payload may be included. It is recommended that where the PCK is being transported beyond a single MC system, the message should include a SAKKE-to-SELF payload as described in clause E.5.

The signature shall use the UID generated from the MC Service user ID of the initiating user.

E.3.2 Default SRTP security profile for PCK

The default security profile is used to support MCPTT and MCVideo communications. It defines the mandatory to support security settings for distribution and use of the PCK. It is the profile that should be used should no information (Crypto session information or security policies) be provided in the MIKEY message.

The CS-ID (for input into the MIKEY PRF) shall be '0' for the MCPTT session from the initiator, '1' for MCPTT session from the receiver, '2' for the MCVideo session from the initiator and '3' for the MCVideo session from the receiver.

The Security Policies are shown in Table E.3.2-1.

Table E.3.2-1: MIKEY Private call SRTP Default Profile

SRTP Type	Meaning	Value	Meaning
0	Encryption Algorithm	6	AES-GCM
1	Session encryption key length	16	16 octets
4	Session salt key length	12	12 octets
5	SRTP PRF	0	AES-CM
6	Key derivation rate	0	No session key refresh.
20	AEAD authentication tag length	16	16 octets

E.3.3 Providing a SRTP security profile for PCK use

Should a security profile be provided by the initiator, the mapping is provided in a GENERIC-ID component of the MIKEY HDR. The CS-ID shall be '0' for the MCPTT session from the initiator, '1' for MCPTT session from the receiver, '2' for the MCVideo session from the initiator and '3' for the MCVideo session from the receiver. Consequently, the CS# shall be between 1 and 4 inclusive. The 'Prot Type' shall be '0' (SRTP).

In each GENERIC-ID crypto session, '#P' shall be 1 (a single security policy shall be referenced). It is recommended that the 'Session Data length' is '0' as SSRCs do not need to be provided. The MKI (PCK-ID) may be included in the SPI field.

E.4 MIKEY message structure for CSK and MuSiK distribution

E.4.1 General

The CSK and MuSiK shall only be used to protect SRTCP payloads and shall not be used to protect SRTP payloads.

In the Common Header payload, the CSB ID field of MIKEY common header for CSK and MuSiK distribution shall be the CSK-ID or MuSiK-ID (resp).

Where no crypto sessions are included in the payload, (CS# is 0), the default security profile defined in Annex E.4.2 shall be used, and no Secuirty Properties payload (SP) is required. The profile in Annex E.4.2 is mandatory to support.

Identity payloads shall be IDR payloads as defined in section 6.6 of IETF RFC 6043 [25].

For CSK upload, the IDRi payload shall contain the MC Service user ID associated with the initiating user. The IDRr payload shall contain the MDSI of the MCX Domain. The message shall also include IDRkmsi and IDRkmsr that contains the URI of the KMS used by the initiating user and MCX Domain respectively.

For CSK and MuSiK download, the IDRi payload shall contain the MDSI of the MCX Domain. The IDRr payload shall contain the MC Service user ID associated with the initiating user. The message shall also include IDRkmsi and IDRkmsr that contains the URI of the KMS used by the MCX Domain and initiating user respectively.

NOTE: In some deployments MC Service user IDs (i.e. MCPTT ID, MCVideo ID, MCData ID) within these payloads may be treated as private. In this case, these identities may be hidden using the mechanism in clause E.7.

For CSK upload, the SAKKE payload shall encapsulate the CSK to the UID generated from the MDSI of the MCX Domain, and the current time period. For CSK or MuSiK download, the SAKKE payload shall encapsulate the key to the UID generated from the user ID associated with the initiating user and the current time period.

A 'Key Properties' payload (Annex E.6) may be included to provide details of the CSK or MuSiK.

For CSK Upload, the signature shall use the UID generated from the identifier associated with MC Service user ID associated with the initiating user. For CSK and MuSiK download, the signature shall use the UID generated from the identifier associated with MDSI of the MCX Domain.

E.4.2 Default SRTCP security profile for CSK and MuSiK

The default security profile is used to support SRTCP for MCPTT and MCVideo communications. It defines the mandatory to support security settings for distribution and use of the CSK and MuSiK. It is the profile that should be used should no information (Crypto session information or security policies) be provided in the MIKEY message.

The CS-ID (for input into the MIKEY PRF) shall be '6' for CSK use within MCPTT (floor control and media control), '7' for MuSiK use within MCPTT, '8' for CSK use within MCVideo (transmission control), and '9' for MuSiK use within MCVideo. The 'Prot Type' shall be '0' (SRTP).

The Security Policies are shown in Table E.4.2-1.

Table E.4.2-1: MIKEY Default Profile for CSK and MuSiK

SRTP Type	Meaning	Value	Meaning
0	Encryption Algorithm	6	AES-GCM
1	Session encryption key length	16	16 octets
4	Session salt key length	12	12 octets
5	SRTP PRF	0	AES-CM
6	Key derivation rate	0	No session key refresh.
20	AEAD authentication tag length	16	16 octets

E.4.3 Providing a SRTCP security profile for CSK or MuSiK

Should a security profile be provided, the mapping is provided in a GENERIC-ID component of the MIKEY HDR. For CSK transmission, the CS-ID shall be '6' for CSK use within MCPTT (floor control and media control) and '8' for CSK use within MCVideo (transmission control),. For MuSiK transmission, the CS-ID shall be '7' for MuSiK use within MCPTT and '9' for MuSiK use within MCVideo. Consequently, the CS# shall be '1' or '2' for either CSK or MuSiK transmission.

In each GENERIC-ID crypto session, '#P' shall be 1 (a single security policy shall be referenced). The MC Server may provide SSRCs for SRTCP within the Session Data. The MKI (GMK-ID || GUK-ID) may be included in the SPI field.

E.5 MIKEY general extension payload to support 'SAKKE-to-self'

In some circumstances it is useful for the initiator to be able to decrypt a MIKEY-SAKKE payload and recover the key (as well as the receiver). For example, where the initiating user is attached to the MCX service via more than one MC UE, the other MC UEs associated with the initiating user will also need the key material to be able to join the communication.

To support this scenario, an optional MIKEY General Extension Payload may be added to the MIKEY-SAKKE message. This general extension payload has type 'SAKKE-to-self'. The contents of the payload will be a full SAKKE payload as defined in IETF RFC 6509 [11]. Within the second SAKKE payload the key (GMK or PCK) shall be encapsulated to the UID generated from the MC identifier associated with the initiating user (either group management

server or private call initiator). The ID Scheme in the SAKKE payload shall be '3GPP MCX hashed UID ' to reflect the generation scheme defined in clause F.2.1.

The General Extensions Field Name 'SAKKE-to-self' type takes on the IANA assigned value of '6' [52].

EXAMPLE SAKKE-to-self payload:

```
*   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
*  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
*  ! Next payload  ! Type          ! Length                      !
*  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
*  ! Next payload  ! SAKKE params  !   ID scheme   !  SAKKE data   ~
*  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
*  ~ length (cont) !                 SAKKE data                    ~
*  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The SAKKE-to-self payload encapsulates a SAKKE payload. Consequently, the SAKKE-to-self payload will contain two 'next payload' fields. The second 'next payload' field, which corresponds to the encapsulated SAKKE payload, shall be set to zero and ignored.

E.6 MIKEY general extension payload to encapsulate parameters associated with a key

E.6.1 General

The parameters associated with the key shall be contained in the 'General extension payload' specified in IETF RFC 3830 [22] using the '3GPP key parameters ' Type value and contained within the signed envelope of the MIKEY-SAKKE I_MESSAGE specified in clause E.2. The format and cryptography of the payload are specified in this subclause.

The General Extensions Field Name '3GPP key parameters' type takes on the IANA assigned value of '7' [52].

The payload consist of a series of information elements. The standard format and encoding rules for the information elements follow that defined for the MCPTT Off-Network Protocol (MONP) as documented in Annex I of 3GPP TS 24.379 [10].

The four octets consisting of the header of the 'General extension payload' shall be formatted according to IETF RFC 3830 [22].

The contents of the 'General extension payload' shall be an MCData Protected Payload message as defined in Clause 8.5.4 with the ‘Payload' element consisting of the 'Key Parameters' payload defined in this clause. The ‘Payload ID’ and the ‘Payload sequence number' of the Protected Payload shall be set to '0' by the sender and ignored by the receiver. The DPPK-ID of the Protected Payload shall be the same as the CSB-ID of the encapsulating MIKEY payload. The key encapsulated by the MIKEY payload (e.g. GMK, MuSiK, etc) shall be used to protect the Protected Payload (the Key Parameters payload).

The 'Key Parameters' payload is a type 6 information element composing a 1 byte Key Parameters IEI, a 2 byte length of the Key Parameters payload contents, and the Key Parameters payload content itself. The Key Parameters payload content shall be of the format specified in Table E.6.1-1.

Table E.6.1-1: Key Parameters Payload content

Information Element	Type/Reference	Presence	Format	Length
Key Type	The type of key. Clause E.6.11.	M	V	1
Status	The current status of the key. Clause E.6.9.	M	V	4
Activation Time	Date and Time when the key may start to be used. Clause E.6.4.	M	V	5
Expiry Time	Date and Time when the key may no longer be used. Clause E.6.10.	M	V	5
Text	A human-readable name for the key Clause E.6.5.	M	LV-E	2-x
MC Group IDs	The MC Group IDs associated with the key (if any) Clause E.6.3.	C	LV-E	2-x
Reserved	Additional information associated with the key (if any) Clause E.6.6.	O	TLV-E	x

NOTE: The 'MC group IDs' IE is only present in the Key Parameters payload if the key type is 'GMK', 'MKFC' or 'MuSiK'.

The IEs in the Key Parameters Payload are described in the following subclauses.

E.6.2 Void

E.6.3 MC group IDs

The 'MC group IDs' IE is only present in the Key Parameters payload if the key type is 'GMK', 'MKFC' or 'MuSiK'.

The 'MC group IDs' IE shall be of the format specified in Table E.6.3-1.

Table E.6.3-1: MC Group IDs IE content

Information Element	Type/Reference	Presence	Format	Length
Number of Group IDs	The number of Group IDs in the payload, at most 255.	M	V	1
Group ID	The ID for the group associated with the key. Clause 15.2.14 of TS 24.282	O	TLV-E	3-x

NOTE: The Number of Group IDs dictates the number of Group ID information elements that are included in the payload. If the number of group IDs is zero, there will be no Group ID IEs in the payload.

The Group ID payload has the same format as the 'MCData Group ID' payload defined in clause 15.2.14 of TS 24.282.

Where the key does not correspond to a group ID, the 'MC group ID' IE shall contain a two octet 'Length' sub-element with the value '1', followed by a 'Number of Group IDs' element of value '0'. .

This field allows distribution of MC Group IDs that are associated with the current key carried in the MIKEY-SAKKE I_MESSAGE. This means that each specified MC Group ID shall use this key for group communications. Assigned MC Group IDs may include any combination of MCPTT Group IDs, MCData Group IDs or MCVideo Group IDs.

E.6.4 Activation time

The 'Activation time' element shall define the time in UTC at which the associated key is to be made active for transmission in seconds since midnight UTC of January 1, 1970 (not counting leap seconds). It shall be 5 octets in length.

A value of 0 shall imply the activation time is the timestamp of the received MIKEY I_MESSAGE.

E.6.5 Text

The 'Text' sub-element shall contain the user-readable name associated with the key.

Where there is no text, the 'Text' element shall contain a two octet 'Length' sub-element with the value 0 .

E.6.6 Reserved

The definition and encoding of the Reserved IE is outside of scope of the present document.

E.6.7 Void

E.6.8 Void

E.6.9 Status

The 'Status' element shall determine the current status of the key. It shall be 4 octets in length. The following values are defined in Table E.6.9-1.:

Table E.6.9-1: Key status bit field

Bit (LSB first)	Bit purpose	'0' meaning	'1' meaning
0	Key Revokation	Key is revoked (may not be used)	Not revoked.
1	Security Gateway	Key has not been shared with a Security Gateway	Key shared with a Security Gateway

Undefined bits shall be ignored.

E.6.10 Expiry time

The 'Expiry time' element shall define the time in UTC at which the associated key shall no longer be used in seconds since midnight UTC of January 1, 1970 (not counting leap seconds). It shall be 5 octets in length.

A value of 0 shall imply the key shall not expire.

E.6.11 Key Type

The purpose of Key Type IE is to specify the type and purpose of the key.

The value part of the Key type information element is coded as shown in Table E.6.11-1.

Table E.6.11-1: Key type

Bits 8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	GMK
0	0	0	0	0	0	0	1	PCK
0	0	0	0	0	0	1	0	CSK
0	0	0	0	0	0	1	1	SPK
0	0	0	0	0	1	0	0	MKFC
0	0	0	0	0	1	0	1	MSCCK
0	0	0	0	0	1	1	0	MuSiK
All other values are reserved.								

E.7 Hiding identities within MIKEY messages

In some public-safety use cases there is a requirement to protect MC Service user IDs in transit. To protect these identifiers in MIKEY-SAKKE messages the following approach may be taken.

The sensitive MC Service user ID in the IDRr or IDRi field is replaced with the UID generated from the MC Service user ID as defined in clause F.2.1. In the former case, the 'role' of the IDRr field is replaced with a role of IDRuidr. In the latter case, the 'role' of the IDRi field is replaced with a role of IDRuidi.

The ID Role of Hashed Initiator (IDRuidi) takes on the IANA assigned value of '8' while the ID Role of Hashed Responder (IDRuidr) takes on the IANA assigned value of '9' [52].

The processing of the MIKEY-SAKKE I_MESSAGE at the initiator stays the same. If the initiator has hidden its own MC Service user ID, it shall ensure that the SIP message containing the I_MESSAGE contains the initiator's MC Service user ID encrypted to the receiver.

As a consequence of identity hiding, the receiver of the MIKEY-SAKKE I_MESSAGE will be able to check the signature based on the initiator's UID in the IDRuidi field, but initially will be unable to confirm the MC Service user ID that has been used to generate the UID. The receiver will recognize its own UID in the IDRuidr field, and be able to extract the encapsulated key.

Using the encapsulated key or otherwise, the receiver is able to extract associated metadata in the message, including the initiator's MC Service user ID. On obtaining the initiator's MC Service user ID, the receiver is able to compute the UID and ensure this matches the UID in the IDRuidi field. By performing this check, the receiver has authenticated the I_MESSAGE.

Annex F (normative): Key derivation and hash functions

F.1 KDF interface and input parameter construction

F.1.1 General

This annex specifies the use of the Key Derivation Function (KDF) specified in 3GPP TS 33.220 [17] for the current specification. This annex specifies how to construct the input string, S, to the KDF (which is input together with the relevant key). For each of the distinct usages of the KDF, the input parameters S are specified below.

F.1.2 FC value allocations

The FC number space used is controlled by 3GPP TS 33.220 [17].

F.1.3 Calculation of the User Salt for GUK-ID generation

When calculating a User Salt using the GMK for generating the GUK-ID from the GMK-ID, the following parameters shall be used to form the input S to the KDF that is specified in annex B of 3GPP TS 33.220 [17]:

- FC = 0x50.
- P0 = MC Service user ID.
- L0 = length of above (i.e. 0x00 0x17).

The GMK and MC Service user ID follow the encoding also specified in annex B of 3GPP TS 33.220 [17]. The 28 least significant bits of the 256 bits of the KDF output shall be used as the User Salt.

F.1.4 Calculation of keys for application data protection

The two keys used to protect either signalling plane confidentiality, or signalling plane integrity are derived from the XPK, using the KDF that is specified in annex B of 3GPP TS 33.220 [17].

The following parameters shall be used to form the input S to the KDF that is specified in annex B of 3GPP TS 33.220 [27]. The key used by the KDF shall be the XPK:

- FC = 0x51, (for signalling plane confidentiality), or
- FC = 0x52 (for signalling plane integrity).
- P0 = MC Service user ID.
- L0 = length of above, expressed in number of bytes (i.e. 0x00 0x17).
- P1 = XPK-ID.
- L1 = length of above, expressed in number of bytes (i.e. 0x00 0x17).

The MC Service user ID and XPK-ID follow the encoding also specified in annex B of 3GPP TS 33.220 [17].

Where the XPK is 128-bits, the output keys shall be 128-bits and hence the 128 least significant bits of the 256 bits of the KDF output shall be used as the signalling protection key. Where the XPK is 256-bits, the output keys shall be 256-bits and hence the entire output of the KDF shall be used.

F.1.5 Calculation of keys for MCData payload protection

The following parameters shall be used to form the input S to the KDF that is specified in annex B of 3GPP TS 33.220 [27]. The key used by the KDF shall be the DPPK:

- FC = 0x53, (for MCData Payload Protection),
- P0 = DPPK-ID.
- L0 = length of above, expressed in number of bytes (i.e. 0x00 0x17).

The DPPK-ID follow the encoding also specified in annex B of 3GPP TS 33.220 [17].

Where the DPPK is 128-bits, the DPCK shall be 128-bits and hence the 128 least significant bits of the 256 bits of the KDF output shall be used as the signalling protection key. Where the DPPK is 256-bits, the output DPCK shall be 256-bits and hence the entire output of the KDF shall be used.

For MCData signalling parameters, Data signaling payload, and End to end security parameter protection between the MCData client and MCData server, the CSK is used as the DPPK. When the selected algorithm is DP_AES_128_GCM, the DPCK shall be 128-bits and hence the 128 least significant bits of the 256 bits of the KDF output shall be used as the signalling protection key. When the selected algorithm is DP_AES_256_GCM, the output DPCK shall be 256-bits and hence the entire output of the KDF shall be used.

F.2 Hash functions

F.2.1 Generation of MIKEY-SAKKE UID

F.2.1.1 Overview

Section 3.2 of IETF RFC 6509 [11] defines an identifier for use in MIKEY SAKKE, referred to as the UID in the present document. This requires a Tel-URI as the user's URI and monthly key periods. As MC Service user IDs may not be Tel-URIs, this UID format cannot be used within MC applications. This clause defines how the 256-bit MIKEY-SAKKE UID is generated using a generic identifier and generic key period.

The MIKEY-SAKKE UID is generated by hashing a fixed string, the identifier of the user, the identifier of the KMS, the key period length, the current key period number and their respective lengths. Key periods are a repeating sequence of fixed time periods, where the first key period commences at an offset in time following 0h on 1 January 1900.

The input to the hash function shall be encoded as specified in clause B.2 of 3GPP TS 33.220 [17]. The hash function shall be SHA-256 as specified in [18]. The full 256-bit output shall be used as the identifier within MIKEY-SAKKE (referred to as 'ID' in IETF RFC 6507 [9] and 'a' or 'b' within IETF RFC 6508 [10].

FC = 0x00

P0 = The fixed string: "MIKEY-SAKKE-UID"

L0 = Length of P0 value

P1 = Identifier (e.g. MCPTT ID, MCVideo ID or MCData ID)

L1 = Length of P1 value

P2 = KMS Identifier (e.g. secgroup1.kms.example.org)

L2 = Length of P2 value

P3 = Key Period length in seconds (e.g. 2592000)

L3 = Length of P3 value

P4 = Key Period offset in seconds (e.g. 0)

L4 = Length of P4 value

P5 = Current Key Period No. since 0h on 1 January 1900 (e.g. 553)

L5 = Length of P5 value

NOTE 1: The key derivation function defined in clause B.2 of 3GPP TS 33.220 [17] is not used, therefore the FC value should only be considered as a dummy value.

P0 is a fixed 15 character string encoded as described in annex B of 3GPP TS 33.220 [17]. P1 is the identifier, which for MCPTT would be the MCPTT ID. P2 is the identifier of the KMS, and uniquely identifies the public key used for encryption and signing. P3 is the integer representing the number of seconds in every key period. P4 is the offset of the start time of the first key period from 0h on 1 January 1900 and shall be less than P3. The combination of P4 and multiples of P3 set the time at which keys are changed over at the end of every key period. Both P3 and P4 are extracted from the KMS certificate (UserKeyPeriod and UserKeyOffset from table D.3.2.2-1, respectively) and encoded as integers as described in annex B of 3GPP TS 33.220 [17]. P5 is the integer representing the current key period number since 0h on 1 January 1900, which may be calculated as:

P5 = Floor ((TIME - P4) / P3)

Where TIME is a NTP timestamp, i.e., a number in seconds relative to 0h on 1 January 1900. P4 is encoded as described in annex B of 3GPP TS 33.220 [17].

NOTE 2: When used to generate a UID for encrypting using a MIKEY payload, P1 will commonly be the 'ID Data' from the IDRr payload, P2 will be the encoded 'ID Data' from the IDRkmsr payload, and TIME will be the NTP timestamp within the MIKEY payload.

NOTE 3: When used to generate a UID for signing a MIKEY payload, P1 will commonly be the 'ID Data' from the IDRi payload, P2 will commonly be the 'ID Data' from the IDRkmsi payload, and TIME will be the NTP timestamp within the MIKEY payload.

F.2.1.2 Example UID

This clause calculates an example UID demonstrating the hash defined in clause F.2.1.1.

In this example:

- The identifier, P1, is sip:user@example.org.
- The KMS identifier, P2, is kms.example.org.
- The key period is 4 weeks, hence P3 is 2592000.
- The offset, P4, is 0.
- the calculation time is: <2014:01:26T10:07:14Z>, hence TIME is 3599719634.

Based on these details:

P5 = Floor ((3599719634 – 0) / 2592000) = 1388.

Consequently, S is constructed from the concatenation of:

FC = 0x00

P0 = MIKEY-SAKKE-UID

L0 = 15

P1 = sip:user@example.org

L1 = 20

P2 = kms.example.org

L2 = 15

P3 = 2592000

L3 = 3

P4 = 0

L4 = 1

P5 = 1388

L5 = 2

Using the conversion in Clause B.2 of TS 33.220 [17]:

S = 0x00 ||

0x4d 0x49 0x4b 0x45 0x59 0x2d 0x53 0x41 0x4b 0x4b 0x45 0x2d 0x55 0x49 0x44 || 0x00 0x0f ||

0x73 0x69 0x70 0x3a 0x75 0x73 0x65 0x72 0x40 0x65 0x78 0x61 0x6d 0x70 0x6c 0x65 0x2e 0x6f 0x72 0x67 || 0x00 0x14 ||

0x6b 0x6d 0x73 0x2e 0x65 0x78 0x61 0x6d 0x70 0x6c 0x65 0x2e 0x6f 0x72 0x67 || 0x00 0x0f ||

0x27 0x8d 0x00 || 0x00 0x03 ||

0x00 || 0x00 0x01 ||

0x05 0x6c || 0x00 0x02

Consequently:

UID = SHA-256(
004d494b45592d53414b4b452d554944000f7369703a75736572406578616d706c652e6f726700146b6d732e6578616d706c652e6f7267000f278d000003000001056c0002)

= 3

a81fb14c3b1d0fe43c9c577104d55a6d81788bfd2f09743c4557746a5a0353b

Annex G (normative): Key identifiers

The 'purpose tag' within the key identifier (e.g. GMK-ID) shall be the most significant four bits of the key and shall be used to indicate the use of the key. The use of key and application of key diversity are specified in Table G-1.

Table G-1: Key usage according to purpose tag

Purpose tag value	Key type	Key usage	Application of key diversity through UK-ID
0	GMK	Protection of group communications.	Yes
1	PCK	Protection of Private Call communications.	No
2	CSK	Protection of application signalling (XML and SRTCP) between the MC client and MC domain.	No
3	SPK	Protection of application signalling (XML and SRTCP) between servers in MC domain(s).	No
4	MKFC	Used as defined in Annex H.	No
5	MSCCK	Used as defined in Annex H.	No
6	MuSiK	Protection of multicast signalling between the MCX Server and the MC Client.	No
7-15	Not defined		

In this way, the MC UE is able to identify the purpose of the key.

Annex H (normative): Support for legacy multicast key (MKFC) and for MSCCK

H.1 General

TS 33.179 [7] specified a different key distribution mechanism for the distribution of group multicast keys (MKFC). To allow MCPTT clients to operate with legacy MCPTT servers (as defined by the functionality in TS 33.179 [7]), MCPTT clients shall support the MKFC key distribution mechanisms defined in clause H.2 with the following constraints:

- The MCPTT client shall reject MKFCs received from other MC systems (based upon the GMS identity).
- The MCPTT client shall discard previously received MKFCs upon attaching to a new MC system.

MCPTT Servers shall not support MKFC distribution. The MCPTT Server shall only support transmission of signalling over a unicast bearer to a legacy MCPTT client (as defined by the functionality TS 33.179 [7]). This shall be detected by the MCPTT Server on the rejection of the MuSiK.

MCPTT Servers and MCPTT clients shall support distribution of the MSCCK. The mechanism for the distribution of the MSCCK is defined in clause H.3.

NOTE: Void.

MSCCK and MKFC are used as defined in clause H.4.

H.2 MKFC Receipt

MKFCs are distributed using the same procedures as for GMK distribution. The client receives an MKFC from the MCPTT server using the procedures in clause 7.3, with the exception that the MKFC and MKFC-ID is distributed in the place of the GMK and GMK-ID, and the user salt is zero (meaning that the GUK-ID is the MKFC-ID).

MKFCs are either distributed in their own group key distribution message (separate from the GMK distribution message), or in the same distribution message as the GMK. Distributing the MKFC in the same message as the GMK is achieved by embedding two MIKEY payloads in one distribution message.

H.3 MSCCK Distribution

MSCCK and MSCCK-ID are distributed within MBMS bearer announcement messages. The procedures are identical to those for distribution of the MuSiK, as defined in clause 5.9, with the exception that the MSCCK and MSCCK-ID are distributed instead of the MuSiK and MuSiK-ID.

H.4 Use of multicast signalling keys (MKFC and MSCCK)

For the protection of multicast floor and media control received from legacy MCPTT servers, the KFC shall be the MKFC and the KFC-ID shall be the MKFC-ID. KFC-RAND shall be the MIKEY RAND value transmitted in the MIKEY message used to distribute the KFC. The KFC is used as defined in clause 9.4.6 and 9.4.7.

For the protection of MBMS subchannel control messages, the KFC shall be the MSCCK and the KFC-ID shall be the MSCCK-ID. KFC-RAND shall be the MIKEY RAND value transmitted in the MIKEY message used to distribute the KFC. The KFC is used as defined in clause 9.4.6 and 9.4.7.

Annex I (normative): Signalling Proxies

I.1 Overview

This solution defines the role of a Signalling Proxy within the Mission Critical System. Signalling Proxies are optional elements providing a deployment option to allow security enforcement at the edge of the MC Doman. Using the functionality defined in this document, Signalling Proxies can be used transparently to MC clients. When Signalling Proxies are used, the MCX Servers within the domain may not need to support security functionality.

The primary function of the signalling proxy is to perform key management of signalling keys, and encryption/decryption of application signalling transiting the edge of the mission critical domain.

This solution defines two types of signalling proxy:

- Client Signalling proxy (CS Proxy)
- Interconnection Signalling Proxy (IS Proxy)

The Client Signalling Proxy may perform security operations towards the client on behalf of the mission critical domain. This includes:

- Topology hiding
- Resilience against signalling storm
- CSK key management (per client);
- MuSiK key management (where protection of multicast signalling is required);
- Protection of application layer signalling (XML in SIP);
- Protection of floor control signalling, transmission control signalling and media signalling (SRTCP);
- Protection of MCData signalling payloads.
- Creation of KMS Redirect Responses (KRRs).

The Interconnection Signalling Proxy may perform security operations towards other mission critical domains. This includes:

- Topology hiding
- Resilience against signalling storm
- Storage of SPK(s);
- Protection of application layer signalling (XML in SIP);
- Protection of floor control signalling, transmission control signalling and media signalling (SRTCP);
- Protection of MCData signalling payloads.
- Creation of KMS Redirect Responses (KRRs).

Signalling proxies may present one or multiple identifiers externally. A CS proxy requires keying by the Key Management Server (KMS) to receive key material associated with the identifiers that it represents externally.

NOTE: Where signalling proxies are used, MCX Servers may not require keying by the KMS as they may not perform any security functionality.

I.2 Location of a signalling proxy

I.2.1 Overview

A signalling proxy should be located at the logical edge of the MC Domain. Signalling routed via the SIP Core should be routed via the signalling proxy on entry or exit of the MC Domain. This includes RTCP signalling such as transmission and floor control.

I.2.2 Deployment with an untrusted SIP Core

Where the SIP Core is not trusted by the Mission Critical provider, the Signalling Proxy should be located between Mission Critical Functions (MCX Server, GMS, etc.) and the external SIP Core. The use of Signalling Proxies within a MC System where the SIP Core is untrusted is shown in Figure I.2.2-1.

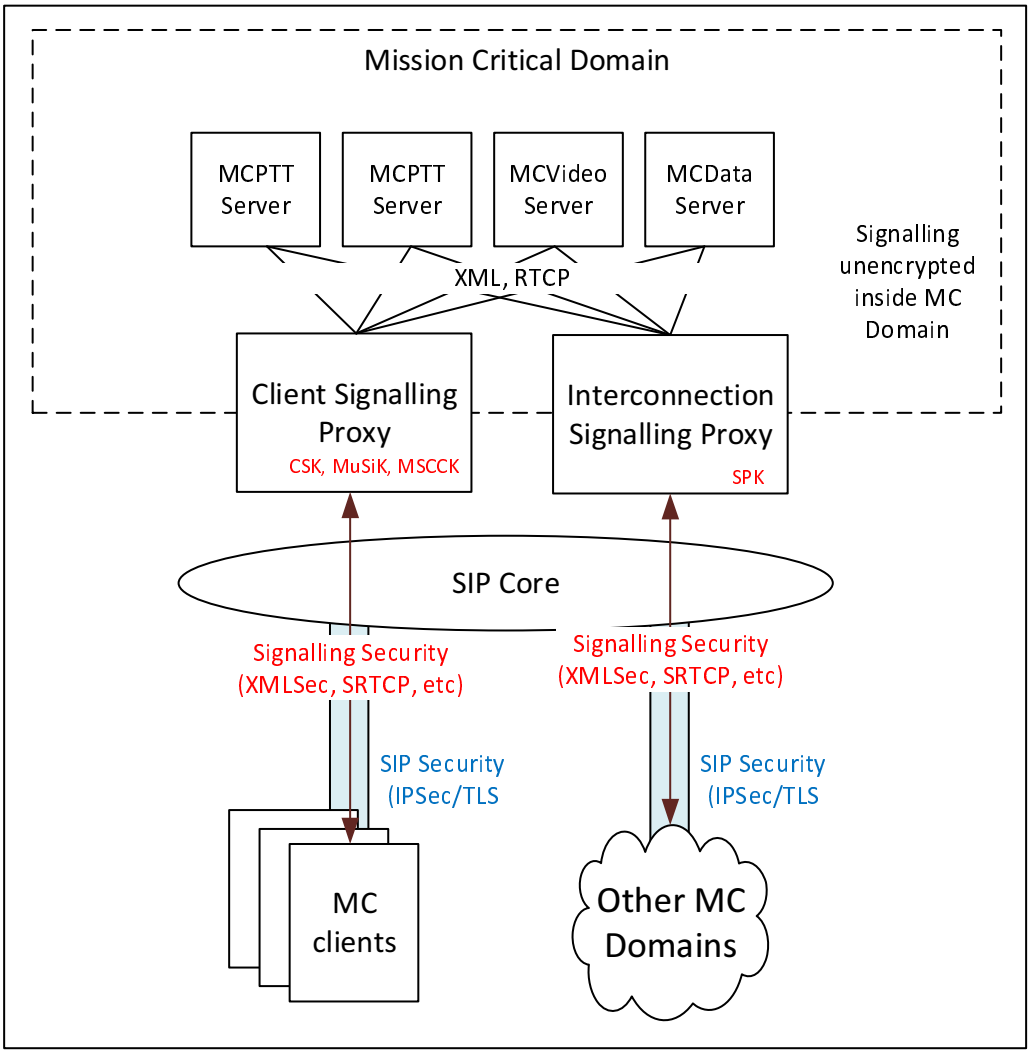

Figure I.2.2-1: Signalling proxies (with untrusted SIP Core)

Internal signalling within the MC Domain (between MCX Server(s) and GMS(s)) routes via the SIP Core. Consequently, in this scenario, the IS Proxy will route all internal signalling to/from itself via the SIP core. Each time it receives an internal signalling message, the IS Proxy should apply an SPK for protection and it should perform topology hiding towards the SIP Core.

NOTE: There may be a performance impact of locating the SIP Core outside of the MC Domain due to the increased load on the IS Proxy.

The use of the signalling proxy at the edge of the Mission Critical network does not remove the need to deploy a SIP Session Border Controller (as defined in RFC 5853 [24]), or IMS IBCF (as defined in Annex I of 3GPP TS 23.228 [23]), to protect the SIP core.

I.2.3 Deployment with a trusted SIP Core

Where the SIP Core is trusted by the Mission Critical provider, the Signalling Proxy should be located at the edge of the external SIP Core, allowing data transiting the SIP core to be unencrypted. The use of Signalling Proxies within a MC System where the SIP Core is trusted is shown in Figure I.2.3-1.

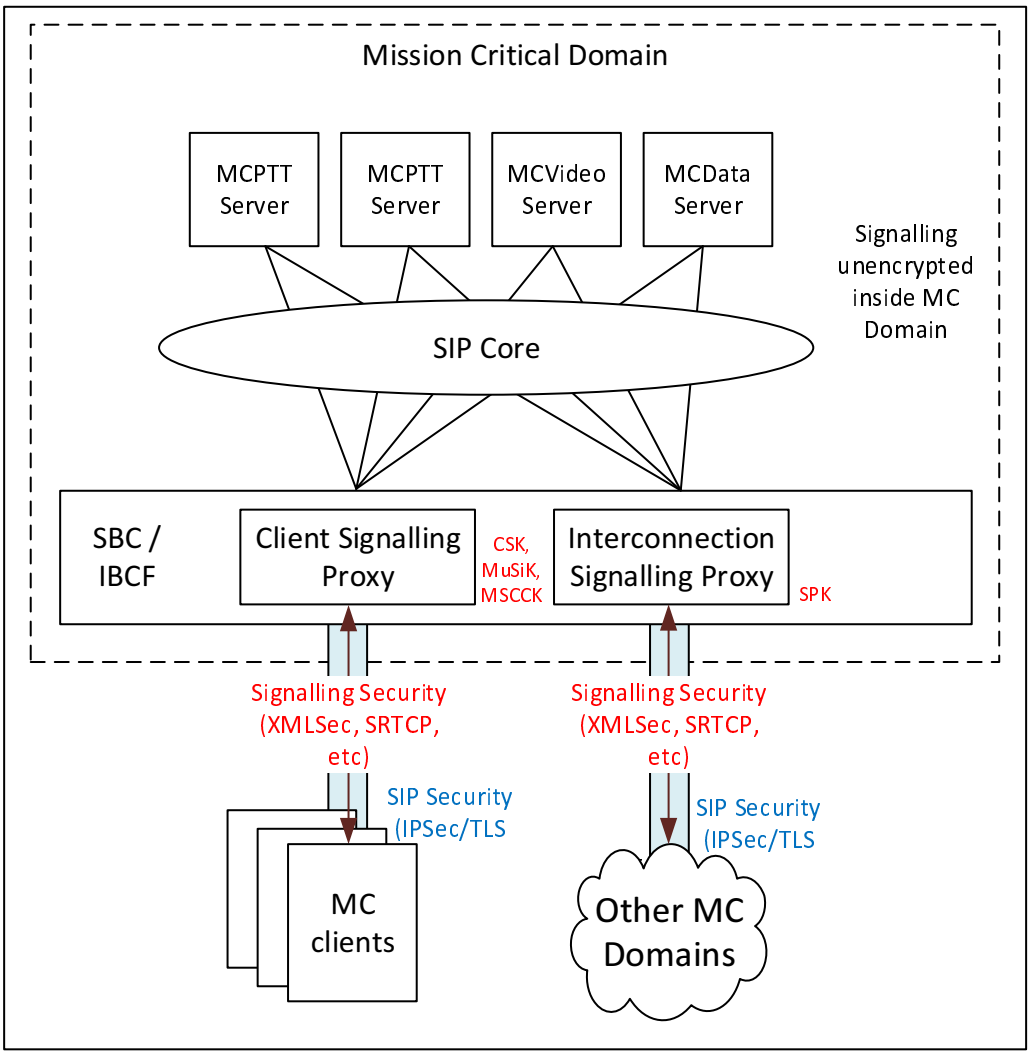

Figure I.2.3-1: Signalling proxies (with trusted SIP Core)

In this deployment scenario, the MC Signalling Proxy may be co-located with the SIP Core's Session Border Controller (as defined in RFC 5853 [24]), or IMS IBCF (as defined in Annex I of 3GPP TS 23.228 [23]). This has the security benefit that the SIP identities and the Mission Critical identities can be correlated at the edge, increasing the system's ability to detect misuse, associate signalling and media and apply system policies.

NOTE 1: In this deployment scenario, signalling security (e.g. XMLSec) and SIP security (e.g. TLS/IPSec) are performing the same function within the MC System. Consequently, the use of both signalling protection methods may not be necessary.

NOTE 2: In this deployment scenario, the IS Proxy is not involved in the routing of internal signalling.

I.3 Functions of a signalling proxy

I.3.1 Overview

A signalling proxy may perform the functions specified in this clause.

I.3.2 Identifier modification (topology hiding)

One function of a signalling proxy is to change the source and destination identifiers in signalling messages to prevent the network topology being exposed externally.

- Messages received on the external interface will be forwarded to an appropriate MC Server based on the type of message and consequently the destination identifier of the message will be changed by the proxy.
- Messages received on the internal interface will have their source identifier replaced with the proxy's identifier.

Modification of identifiers applies to all signalling handled by the proxy. Specifically:

- SIP;
- Application layer signalling (XML in SIP);
- Floor control signalling, transmission control signalling and media signalling (SRTCP);
- MCData signalling payloads.

I.3.3 Resilience against signalling storm

The signalling proxy is able to monitor the quantity and type of signalling entering the MC Domain. Signalling Proxy should be resilient to receiving a large amount of signalling, such as a high number of MCX Server registrations. The Signalling Proxy should be able to block, throttle or prioritise the signalling routed into the MC Domain to prevent overload of application signalling at the MCX Server, while maintaining the most critical MC services. Applying limits to signalling could be performed at a service-level or to a specific user's signalling.

I.3.4 Client connection to a CS Proxy

When an MC client first connects to a CS Proxy, it will provide an encapsulated CSK along with an access token as part of a SIP SUBSCRIBE or SIP PUBLISH message. The CS Proxy should extract and store the CSK and decrypt the access token. The CS Proxy may verify that the message is properly constructed and applicable to this MC Domain (e.g. verify that the access token is applicable to the current MC domain). Verification failure should cause the CS Proxy to drop the message.

The CS Proxy should then forward the SIP SUBSCRIBE or SIP PUBLISH message onto an appropriate MCX Server with an unencrypted access token and without the encapsulated CSK.

From this point onwards, signalling received from the client should be decrypted using the CSK, and signalling sent to the client should be encrypted using the CSK. This functionality is as currently defined for a MCX Server.

I.3.5 CSK key download from a CS Proxy

The CS Proxy is responsible for CSK key management. Hence, should the CSK require renewal, the CS Proxy should create and send a ‘key download' message to the MC client containing the new CSK. This functionality is as currently defined for a MCX Server.

As signalling proxies may present the one or multiple SIP URIs externally, the same client may attempt to connect to the same CS Proxy twice, using different URIs and different CSKs each time. In this scenario, the CS Proxy may remove CSK ambiguity by using the ‘CSK key download' procedure as follows:

1) The MC client connects to the CS Proxy using the URI ‘A'. The MC client provides CSK_A.
2) The CS Proxy receives CSK_A and the MC client and CS Proxy use CSK_A to protect application signalling.

3) The MC client connects to the CS Proxy using the URI ‘B'. The MC client provides CSK_B.

4) The CS Proxy observes that the same client has connected again using a different destination URI.

5) The CS Proxy performs a ‘CSK key download' to update CSK_B. The CS Proxy sets CSK_B to CSK_A.

6) The MC client and CS Proxy use CSK_A to protect application signalling (regardless of source URI).

I.3.6 MuSiK and MSCCK key download from a CS Proxy

Should multicast signalling be required, the CS Proxy should perform MuSiK key download from the CS Proxy to the MC client. To support this, the CS Proxy should perform a MuSiK key download procedure toward the MC clients that will receive multicast signalling. This functionality is as currently defined for a MCX Server.

On receipt of signalling from the MC domain towards a multicast bearer, the CS Proxy should protect the signalling with a MuSiK and forwards the message externally. This functionality is as currently defined for a MCX Server.

NOTE: As multiple MCX Servers can use the same CS Proxy for multicast signalling, this allows multiple MCX Servers to share multicast bearers.

Similarly, the CS Proxy should attach a MSCCK to MBMS Bearer Annoucement messages, and encrypt MBMS subchannel control messages with the MSCCK.

I.3.7 Signalling protection by the IS Proxy

The IS Proxy is configured with one, or more, SPKs for protection of signalling and each SPK will be associated with specific interconnection end-point(s). On receipt of signalling from the MC domain towards an interconnection end-point, the IS Proxy should encrypt the signalling using the appropriate SPK and forward the message externally. On receipt of signalling from an interconnection end-point towards the MC Domain, the IS Proxy should decrypt the signalling and forward the message internally.

I.3.8 Creation of KMS Redirect Responses (KRRs)

The Signalling Proxy may create KRRs to enforce local policy around the use of KMSs within the MC Domain. For example, should a MIKEY message be sent through the domain using a KMS that is unacceptable within the domain, the CS or IS Proxy may drop the MIKEY message, create a KRR and return the KRR to the sender of the MIKEY message.

I.3.9 Policy enforcement

As gateways to the MC domain, signalling proxies may also be appropriate locations to enforce policy within the MC domain.

Editor's Note: Defining the policies that could be enforced at the signalling proxy is FFS.

Annex J (normative): Authentication and authorisation formats

J.1 Elements for Authenticating Requests

J.1.1 General

This clause describes the functional definitions and contents for the Element for Authenticating Requests (EARs). EARs may be used to authenticate and potentially authorise signalling requests within the MC System.

Each EAR consist of a series of information elements. The standard format of an EAR and the encoding rules for each type of information element follow that defined for the MCPTT Off-Network Protocol (MONP) as documented in Annex I of 3GPP TS 24.379 [49].

J.1.2 Format of an EAR

This subclause defines the contents of an EAR message. The EAR provides details of the request and associates that request with an authenticated identity. The EAR shall be signed using the mechanism defined in Clause 8.5.5. For the contents of the EAR see Table J.1.2-1.

Message type: EAR PAYLOAD

Direction: Attached to a signalling request as defined in Clause 9.6.2.

Table J.1.2-1: EAR PAYLOAD message content

IEI	Information Element	Type/Reference	Presence	Format	Length
	EAR message ID	Message type	M	V	1
	Date and time	Date and time Clause 15.2.8 of TS 24.282 [50]	M	V	5
	EAR ID	EAR ID J.1.3	M	V	16
	Source Role	Role ID J.1.4	M	V	1
	Source ID	Entity ID J.1.5	M	LV-E	2-x
	Target Role	Role ID J.1.4	M	V	1
	Target ID	Entity ID J.1.5	M	LV-E	2-x
uu	Request	Request J.2	O	TLV-E	3-x

J.1.3 Format of an EAR ID

The EAR ID information element uniquely identifies the EAR.

The EAR ID information element is coded as shown in Figure J.1.3-1 and Table J.1.3-1.

The EAR ID information element is a type 3 information element with a length of 16 octets.

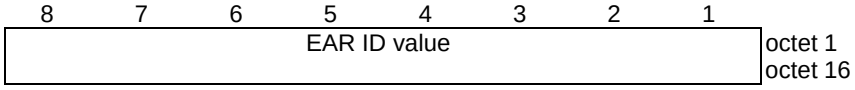

Figure J.1.3-1: EAR ID value

Table J.1.3-1: EAR ID value

EAR identifier value (octet 1 to 16) The EAR ID contains a number uniquely identifying an EAR. The value is a universally unique identifier as specified in IETF RFC 4122 [14].

J.1.4 Format of an entity's Role ID

The purpose of the Role ID information element is to identify the role of the entity and the type of entity ID used by the entity.

The value part of the Role ID information element is coded as shown in Table J.1.4-1.

The Role ID information element is a type 3 information element with a length of 1 octet.

Table J.1.4-1: Role IDs

Bits 8 7 6 5 4 3 2 1	
0 0 0 0 0 0 0 1	MC Service client
0 0 0 0 0 0 1 0	MC Service group
0 0 0 0 0 0 1 1	MC Service function/server
All other values are reserved.	

More fine-grained role identifications may be provided using an Authorised Identity (as defined in Clause 9.6.3).

J.1.5 Format of an MC Entity ID

The MC Entity ID information element is used to indicate an MC Service user ID, an MC Group ID or an FQDN associated with an MC function. The type of Entity ID is defined by the Role ID as defined Clause J.1.4.

The MC Entity ID information element is coded as shown in Figure J.1.5-1 and Table J.1.5-1.

The MC Entity ID information element is a type 6 information element.

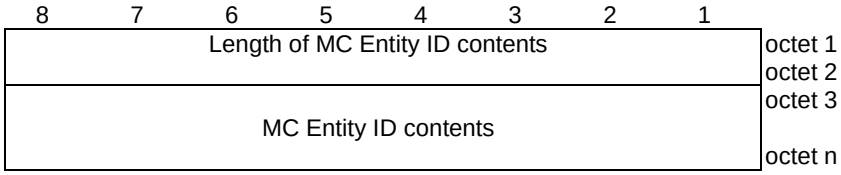

Figure J.1.5-1: MC Entity ID information element

Table J.1.5-1: MC Entity ID information element

The MC Entity ID is contained in octet 3 to octet n. The MC Entity ID may be an MC Service user ID, MC Group ID or an FQDN associated with an MC function. Max value of 65535 octets.

J.2 Request types and parameters

J.2.1 General

This clause defines the information elements that provide details of the authenticated request within an EAR payload.

J.2.2 Request Information element

This subclause defines the contents of a Request. The EAR provides details of the request and associates that request with an authenticated identity. The EAR shall be signed using the mechanism defined in Clause 8.5.5. For the contents of the EAR see Table J.1.2-1.

Message type: REQUEST PAYLOAD

Direction: Attached to a signalling request as defined in Clause 9.6.2.

Table J.2.2-1: REQUEST PAYLOAD message content

IEI	Information Element	Type/Reference	Presence	Format	Length
	Request type	Request type J.2.3	M	V	1
	Request ID	Request ID J.2.5	O	TV	3
	Request Expiry	Request Expiry J.2.4	O	TV	6

J.2.3 Request type

The purpose of the Request Type information element is to identify the type of the request.

The value part of the Request Type information element is coded as shown in Table J.2.3-1.

The Request Type information element is a type 3 information element with a length of 1 octet.

Table J.2.3-1: Request Types

Bits 8 7 6 5 4 3 2 1	
0 0 0 0 0 0 0 1	PRIVILEDGED SIGNALLING
0 0 0 0 0 0 1 0	INTERWORKING SIGNALLING
0 0 0 0 0 0 1 1	GROUP SIGNALLING
0 0 0 0 0 1 0 1	MIGRATION SIGNALLING
0 0 0 0 0 1 1 0	OFF-NETWORK SIGNALLING
All other values are reserved.	

J.2.4 Request expiry

The Request expiry information element is used to indicate the UTC time when the request shall no longer be considered valid. After this time, all events (e.g. calls) caused by the request shall be terminated.

The Request expiry information element is coded as shown in Figure J.2.4-1 and Table J.2.4-1.

The Request expiry information element is a type 3 information element with a length of 6 octets.

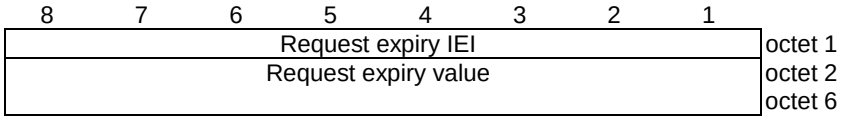

Figure J.2.4-1: Request expiry value

Table J.2.4-1: Request expiry value

Request expiry value (octet 1 to 5) The Request expiry value is an unsigned integer containing UTC time of the time when actions resulting from the request shall be terminated, in seconds since midnight UTC of January 1, 1970 (not counting leap seconds).

J.2.5 Request IDs

J.2.5.1 Format

The Request ID information element is used to indicate the exact request made by a MC entity. Only Request IDs are defined for Priviledged signalling and off-network signalling Request Types. Request ID payload shall not be used for other request types.

The Request ID information element is coded as shown in Figure J.2.5.1-1. The contents are coded as described in subsequent subclauses.

The Request expiry information element is a type 3 information element with a length of 3 octets.

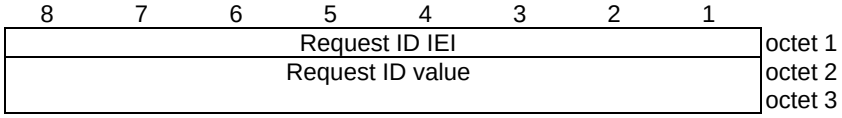

Figure J.2.5.1-1: Request ID value

J.2.5.2 Request ID values for priviledged signalling

Table J.2.5.2-1: Request ID values for priviledged signalling

Byte 1	Byte 2	Privileged signalling request
00000001	0000001	MCPTT Private call request in automatic commencement mode (TS 23.379).
00000001	0000010	MCPTT Ambient listening call request (TS 23.379).
00000001	0000011	MCPTT Remotely initiated MCPTT call request, in unnotified mode (TS 23.379).
00000001	0000100	MCVideo Private call request (including private call, video pull and video push) in automatic commencement mode (TS 23.281).
00000001	0000101	MCVideo Remote video push request in automatic commencement mode (TS 23.281).
00000001	0000110	MCVideo Ambient viewing call request (TS 23.281).
00000001	0000111	MCData standalone data request for application consumption (TS 23.282).
00000001	0001000	MCData standalone session data request for application consumption (TS 23.282).
00000001	0001001	MCData session data request for application consumption (TS 23.282).
00000001	0001010	MCData group standalone data request for application consumption (TS 23.282).
00000001	0001011	MCData group data request for application consumption (TS 23.282).
00000001	0001100	MCData FD request with mandatory indication (TS 23.282).
00000001	0001101	MCData group standalone FD request with mandatory indication (TS 23.282).

J.2.5.3 Request IDs for off-network signalling

Table J.2.5.3-1: Request ID values for off-network

Byte 1	Byte 2	Off-network signalling request
10000000	0000001	MCPTT Group call announcement (TS 23.379).
10000000	0000010	MCPTT emergency alert announcement (TS 23.379).
10000000	0000011	MCPTT Call setup request (TS 23.379).
10000000	0000100	MCVideo Group communication announcement (TS 23.281).
10000000	0000101	MCVideo emergency alert announcement (TS 23.281).
10000000	0000110	MCVideo Private communication request (TS 23.281).
10000000	0000111	MCVideo Capability request (TS 23.281).
10000000	0001000	MCVideo Activity request (TS 23.281).
10000000	0001001	MCData standalone data request (Clause 7.4.3.3.2, TS 23.282).
10000000	0001010	MCData group standalone data request (Clause 7.4.3.4.2, TS 23.282).

J.3 Authorisation fields

J.3.1 General

Authorisation fields are used to convey the entity's authorisations within the entity's identity. They are a set of name, value pairs added as SIP URI Headers.

J.3.2 Authorisation field names

MC authorisation fields are encoded using the standard SIP URI Header mechanism (RFC 3261). After the '?', the fields are encoded as ampersand separated hname = hvalue pairs. Each authorisation hvalue is a bit field denoting the entity's permissions. The bit fields are defined in Clause J.3.3. The bit field is encoded in hex within the SIP URI.

Table J.3.2-1 contains the defined SIP URI header names (hname) for the authorisation fields.

Table J.3.2-1: SIP URI Header name denoting a MC authorisation field

SIP URI Header name	Purpose	Table defining value bit-field
mc-role-client	Defines the authorised roles for the client	Table J.3.2-1
mc-role-server	Defines the authorised roles for the network function/entity	Table J.3.2-2
mc-priv-mcptt	Defines the authorised MCPTT privileged signalling.	Table J.3.3-1
mc-priv-mcvideo	Defines the authorised MCVideo privileged signalling.	Table J.3.3-2
mc-priv-mcdata	Defines the authorised MCData privileged signalling.	Table J.3.3-3
mc-offnet-mcptt	Defines the authorised MCPTT off-network signalling.	Table J.3.4-1
mc-offnet-mcvideo	Defines the authorised MCVideo off-network signalling.	Table J.3.4-2
mc-offnet-mcdata	Defines the authorised MCData off-network signalling.	Table J.3.4-3

J.3.3 Authorisation field values

J.3.3.1 General

The tables contained in this clause define the bit fields used for authorisation. In the tables, the byte ordering is left-most byte first. The bit ordering is least-signficant bit first.

The bit fields may be extended with further bytes in future specifications. Any bytes within the authorisation fields of a MC Service ID that do not correspond with a bit in a table below shall be ignored. The maximum length of a bit field shall be 1024 bits (or 256 hex characters).

J.3.3.2 Role authorisations

Table J.3.3.2-1: User role authorisations (mc-role-client)

Byte	Bit	Role authorisation	Idm scope definition
0	0	MCPTT client	"3gpp:mc:auth:role:client:ptt"
	1	MCVideo client	"3gpp:mc:auth:role:client:video"
	2	MCData client	"3gpp:mc:auth:role:client:data"

Table J.3.3.2-2: Server role authorisations (mc-role-server)

Byte	Bit	Role authorisation	Idm scope definition
0	0	Group Management Server	"3gpp:mc:auth:role:server:gms"
	1	CS Proxy	"3gpp:mc:auth:role:server:cs_proxy"
	2	IS Proxy	"3gpp:mc:auth:role:server:is_proxy"
	3	MCPTT server	"3gpp:mc:auth:role:server:mcptt"
	4	MCVideo server	"3gpp:mc:auth:role:server:mcvideo"
	5	MCData server	"3gpp:mc:auth:role:server:mcdata"

J.3.3.3 Authorisations for priviledged signalling

Table J.3.3.3-1: MCPTT privileged signalling authorisations (mc-priv-mcptt)

Byte	Bit	Privileged signalling authorisation	Idm scope definition
0	0	MCPTT Private call request in automatic commencement mode (TS 23.379).	"3gpp:mc:auth:priv:mcptt:automatic_private_call"
	1	MCPTT Ambient listening call request (TS 23.379).	"3gpp:mc:auth:priv:mcptt:ambient_listening"
	2	MCPTT Remotely initiated MCPTT call request, in unnotified mode (TS 23.379).	"3gpp:mc:auth:priv:mcptt:unnotified_remote_call"

Table J.3.3.3-2: MCVideo privileged signalling authorisations (mc-priv-mcvideo)

Byte	Bit	Privileged signalling authorisation	Idm scope definition
0	0	MCVideo Private call request (including private call, video pull and video push) in automatic commencement mode (TS 23.281).	"3gpp:mc:auth:priv:mcvideo:automatic_private_call"
	1	MCVideo Remote video push request in automatic commencement mode (TS 23.281).	"3gpp:mc:auth:priv:mcvideo:automatic_remote_video_push"
	2	MCVideo Ambient viewing call request (TS 23.281).	"3gpp:mc:auth:priv:mcvideo:ambient_viewing"

Table J.3.3.3-3: MCData privileged signalling authorisations (mc-priv-mcdata)

Byte	Bit	Privileged signalling authorisation	Idm scope definition
0	0	MCData standalone data request for application consumption (TS 23.282).	"3gpp:mc:auth:priv:mcdata:sds:unnotified_req"
	1	MCData standalone session data request for application consumption (TS 23.282).	"3gpp:mc:auth:priv:mcdata:sds:unnotified_standalone_session_req"
	2	MCData session data request for application consumption (TS 23.282).	"3gpp:mc:auth:priv:mcdata:sds:unnotified_session_req"
	3	MCData group standalone data request for application consumption (TS 23.282).	"3gpp:mc:auth:priv:mcdata:sds:unnotified_group_standalone_req"
	4	MCData group data request for application consumption (TS 23.282).	"3gpp:mc:auth:priv:mcdata:sds:unnotified_group_req"
	5	MCData FD request with mandatory indication (TS 23.282).	"3gpp:mc:auth:priv:mcdata:fd:mandatory_req"
	6	MCData group standalone FD request with mandatory indication (TS 23.282).	"3gpp:mc:auth:priv:mcdata:fd:mandatory_group_req"

J.3.3.4 Authorisations for off-network signalling

Table J.3.3.4-1: MCPTT Off-network signalling authorisations (mc-offnet-mcptt)

Byte	Bit	Off-network signalling authorisation	Idm scope definition
0	0	Permission to transmit MCPTT off-network	"3gpp:mc:auth:offnet:mcptt:use"
	1	MCPTT Group call announcement (TS 23.379).	"3gpp:mc:auth:offnet:mcptt:group_call_announcement"
	2	MCPTT emergency alert announcement (TS 23.379).	"3gpp:mc:auth:offnet:mcptt:emergency_alert_announcement"
	3	MCPTT Call setup request (TS 23.379).	"3gpp:mc:auth:offnet:mcptt:call_setup_req"

Table J.3.3.4-2: MCVideo Off-network signalling authorisations (mc-offnet-mcvideo)

Byte	Bit	Off-network signalling authorisation	Idm scope definition
0	0	Permission to transmit MCPTT off-network	"3gpp:mc:auth:offnet:mcvideo:use"
	1	MCVideo Group communication announcement (TS 23.281).	"3gpp:mc:auth:offnet:mcvideo:group_communication_announcement"
	2	MCVideo emergency alert announcement (TS 23.281).	"3gpp:mc:auth:offnet:mcvideo:emergency_alert_announcement"
	3	MCVideo Private communication request (TS 23.281).	"3gpp:mc:auth:offnet:mcvideo:private_communication_req"
	4	MCVideo Capability request (TS 23.281).	"3gpp:mc:auth:offnet:mcvideo:capability_req"
	5	MCVideo Activity request (TS 23.281).	"3gpp:mc:auth:offnet:mcvideo:activity_req"

Table J.3.3.4-3: MCData Off-network signalling authorisations (mc-offnet-mcdata)

Byte	Bit	Off-network signalling authorisation	Idm scope definition
0	0	Permission to transmit MCPTT off-network	"3gpp:mc:auth:offnet:mcdata:use"
	1	MCData standalone data request (Clause 7.4.3.3.2, TS 23.282).	"3gpp:mc:auth:offnet:mcdata:standalone_data_req"
	2	MCData group standalone data request (Clause 7.4.3.4.2, TS 23.282).	"3gpp:mc:auth:offnet:mcdata:group_standalone_data_req"

J.3.4 Example Authorised Identities

J.3.4.1 General

This clause contains examples of Authorised Identities using the names from Clause J.3.2 and the values from Clause J.3.3.

J.3.4.2 PTT User (on and off-network)

If a user has the following MC Service ID (without authorisation):

sip:mc.user@example.org

If the user is authorised to use a mcptt client, on and off-network (but no privileged signalling), then the IdM-provided access token sent to the KMS will contain the following values in the scope:

"3gpp:mc:auth:role:client:ptt"

"3gpp:mc:auth:offnet:mcptt:use"

"3gpp:mc:auth:offnet:mcptt:group_call_announcement"

"3gpp:mc:auth:offnet:mcptt:emergency_alert_announcement"

"3gpp:mc:auth:offnet:mcptt:call_setup_req"

The following is the user's authorised MC Service ID:

sip:mc.user@example.org?mc-role-client=01&mc-offnet-mcptt=0f

If supported, the KMS shall provision keys to the user's KM client for both the original MC Service ID and the authorised MC Service ID.

J.3.4.3 Dispatcher

If we assume a dispatcher has full permission to take any action (on-network) and the following MC Service ID:

sip:mc.dispatcher@example.org

Then the authorised MC Service ID is:

sip: mc.dispatcher@example.org?mc-role-client=07&mc-priv-mcptt=07&mc-priv-mcvideo=07&mc-priv-mcdata=7f

Annex K (informative): Non-3GPP security mechanisms

K.1 General

This clause provides some details of non-3GPP security mechanisms which may be in use in the 3GPP network. The purpose of including it in this specification is to inform 3GPP vendors and system owners about the existance of such mechanisms. The definition of these mechanisms is out of scope of this document.

K.2 LMR E2EE

K.2.1 General

LMR end-to-end security allows the IWF to pass protected media unmodified from the 3GPP system to the LMR system. The LMR end-to-end security mechanisms are out of scope of this document.

This clause assumes a non-3GPP (LMR) layer operating below the 3GPP layer defined in this specification at the UE, and potentially at the IWF. This layer may pass media packets to the 3GPP layer for further processing. The 3GPP layer and the non-3GPP layer act independently of each other.

K.2.2 Interworking E2EE keys and key management

Void.

K.2.3 Interworking E2EE media for MCPTT

Non-3GPP RTP or SRTP packets are generated within the non-3GPP layer of the 3GPP MC UE. The generation method of these media packets within the non-3GPP layer of the 3GPP MC UE is out of scope for this document. The non-3GPP layer may or may not apply non-3GPP security to the media. Any non-3GPP security applied to the media packets within the non-3GPP layer is out of scope for this document. Management of the non-3GPP E2EE interworking keys is defined in clause 11.2.

Once processed by the non-3GPP layer, the packet is passed to the 3GPP application layer for further 3GPP processing. The 3GPP application layer views the packet as an unencrypted RTP stream regardless of whether security has been applied at the non-3GPP layer. If the interworking communication is a private MCPTT call, the 3GPP application layer applies MCPTT private call security to the media packet as defined in clause 7.2. If the interworking communication is a group MCPTT call, the 3GPP application layer applies MCPTT group call security to the media packet as defined in clause 7.3. Once processed by the MC application layer, the media is sent by the MC client to the IWF.

As defined in clause 11.2, the IWF is the 3GPP security endpoint for any private or group call security applied to the interworking RTP packets that is sent to, or received from, the 3GPP system. The IWF applies SeGy security functionality to remove security from the messages sent by the 3GPP system before processing the unencrypted message. Consequently, the IWF processes inbound interworking RTP packets prior to applying SeGy security functionality and sending them into the 3GPP system.

K.2.4 Interworking E2EE media for MCData

Non-3GPP MCData Data payloads sent from a 3GPP MC UE to the IWF are generated within the non-3GPP layer of the 3GPP MC UE. The generation method of the payload within the non-3GPP layer of the 3GPP MC UE is out of scope for this document. The non-3GPP layer may or may not apply MCData security to the payload. Any E2EE non-3GPP security applied to the payload within the non-3GPP layer is out of scope for this document. Management of the non-3GPP E2EE interworking keys is defined in clause 11.2.

For MCData messages sent by the 3GPP system, the non-3GPP layer creates the MCData Data payload and passes to the 3GPP application layer for further 3GPP processing. The 3GPP application layer views the packet as an unencrypted payload regardless of whether security has been applied at the non-3GPP layer. If the interworking communication is a private MCData call, the 3GPP application layer applies MCData private communication security to the payload as defined in clause 8. If the interworking communication is a group MCData communication, the 3GPP application layer applies MCData group communication security to the payload as defined in clause 8. Once processed by the MC application layer, the media is sent by the MC client to the IWF.

As defined in clause 11.2, the IWF is the 3GPP security endpoint for any MCData security applied to the interworking MCData message that is sent to, or received from, the 3GPP system. The IWF applies SeGy security functionality to remove security from the MCData messages sent by the 3GPP system before processing the unencrypted message. Consequently, the IWF processes inbound MCData messages prior to applying SeGy security functionality and sending them into the 3GPP system.

Annex L (normative): MC Security Gateway (SeGy)

L.1 General

The MC Security Gateway is a network function that terminates all 3GPP MC security functionality from a protected 3GPP MC system to allow MC signalling and media to be provided to an unprotected MC system or external system. The use of a MC Security Gateway is required when the system that the user wishes to communicate with does not support MC security mechanisms defined in this specification.

The SeGy is a network element in its own right. It may also be used as part of a Interworking Function (IWF) when the IWF requires MC security functionality to be terminated.

The use of a MC Security Gateway terminates end-to-end 3GPP protected media and signalling security. For this reason, a notification shall be provided to the MC user by the MC client when the user's communication involves a MC Security Gateway.

Protection of media and signalling within an external system is out of scope for this standard, and therefore the external system is responsible for ensuring that both signalling and media within the external system are appropriately protected.

L.2 Functional model for the MC Security Gateway (SeGy)

A MC Security Gateway (SeGy) communicates with 3GPP MC systems as a 3GPP partner interconnected MC domain. The SeGy has two interfaces. On the encrypted interface, the SeGy acts as a 3GPP MC domain that uses MC security mechanisms defined in this specification. On the encrypted interface, the SeGy communicates with protected MC systems (MC systems that use the security mechanisms defined in this specification). On the unencrypted interface the SeGy acts as a 3GPP MC domain that does not use the security mechanisms defined in this specification. On the unencrypted interface, the SeGy communicates with unprotected MC systems (MC systems that do not use the security mechanisms defined in this specification) or external systems. Consequently, on the encrypted interface media shall be encrypted and signalling may be encrypted. On the unencrypted interface, media and signalling are unencrypted. Figure L.2-1 shows the role of the SeGy in context.

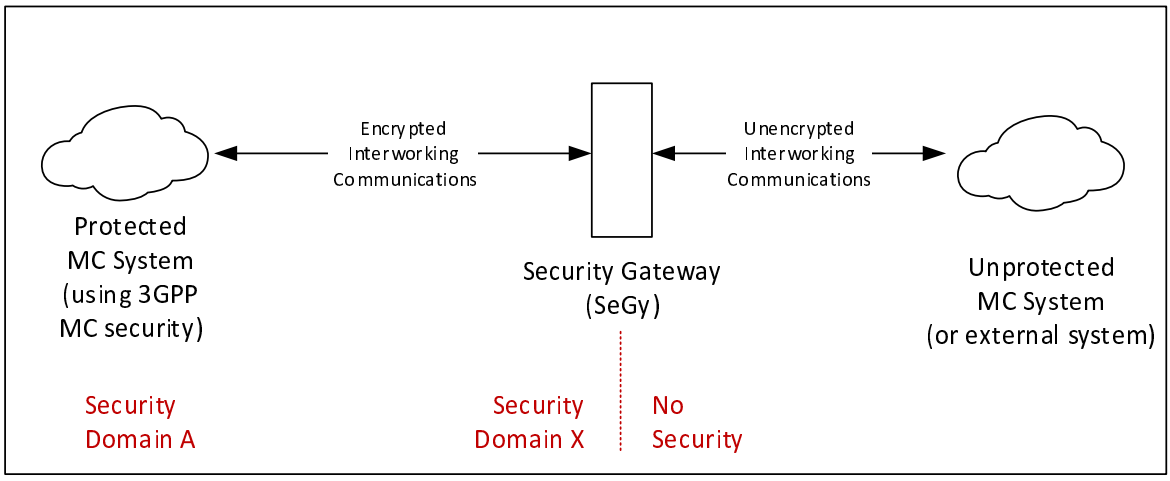

Figure L.2-1: MC Security Gateway (SeGy)

The SeGy shall be configured as an independent security domain to existing MC security domains. In Figure L.2-1, the MC Domain is in Security Domain A, whereas the SeGy is in Security Domain X. This allows the risk of terminating security to be isolated to the SeGy, and allows the use of the SeGy to be communicated to clients.

L.3 Functions of a MC Security Gateway (SeGy)

L.3.1 Components of a MC Security Gateway (SeGy)

At a high-level, the MC Security gateway is composed of four components:

- Pseudo KMS.
- Pseudo GMS.
- Pseudo MCX Server(s).
- Pseudo MC clients.

The term "pseudo" in this case is used to indicate that the security functionality of these components shall be implemented as part of a SeGy however physical entities and servers (i.e. KMS, GMS, MC service servers and MC clients) are not required. The method used to implement a pseudo KMS, GMS, MC service server or MC clients and associated key material within a SeGy is left to the manufacturer and is outside the scope of 3GPP. These components are shown in Figure L.3.1-1.

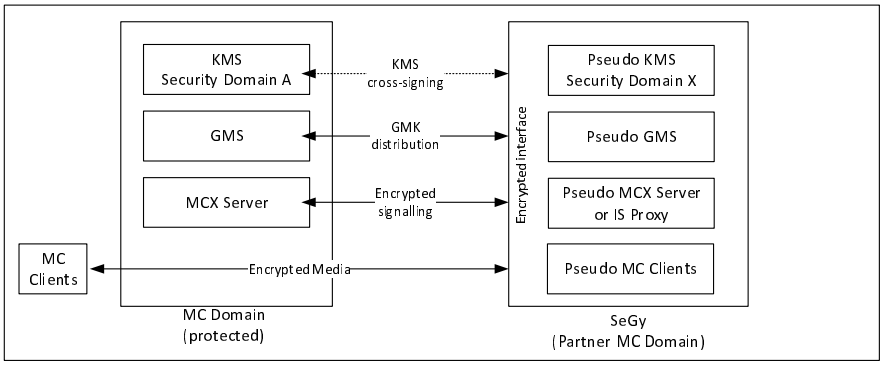

Figure L.3.1-1: Components of a MC Security Gateway (SeGy)

L.3.2 Pseudo KMS

The SeGy contains a KMS function. This establishes the SeGy as containing its own security domain (Security Domain X). The Pseudo KMS does not perform key management functions with any clients, but allows the SeGy to represent external system functions and users as members of the SeGy's security domain within the 3GPP MC System.

The Pseudo KMS shall cross-sign with KMSs in partner protected MC systems that use the SeGy. This means that the SeGy's KMS Certificate shall be provided to the KMS in a partner protected MC system (and vice-versa). As a consequence of cross-signing, users in partner security domains will be able to securely communicate with (external users and groups represented by) the SeGy. As cross-signing is a manual process, no communication is required between KMSs in partner protected MC systems and the SeGy's pseudo KMS.

The SeGy shall create a KMS Certificate as defined in Annex D. The KMS Certificate generated by the SeGy shall include the information that the Certificate originates from an MC Security Gateway. The SeGy's KMS Certificate represents the security domain for the external users that use the SeGy's unencrypted interface and the pseudo network entities within the SeGy itself. This type of KMS Certificate is known as a SeGy KMS Certificate. The use of a SeGy KMS Certificate ensures that 3GPP MC systems and 3GPP MC clients that use the SeGy are aware that a gateway is in use. A visual reference shall be provided to MC users when communicating with a user whose KMS URI corresponds to a SeGy.

In partner systems, the Pseudo KMS shall never be a Migration KMS, but shall be an External KMS.

L.3.3 Pseudo GMS

Should the MC Security Gateway support group communications, the SeGy shall contain a Pseudo GMS. The SeGy's GMS will perform the security functionality of a GMS towards partner GMSs on the encrypted interface. In terms of security, the SeGy GMS will create and add GMKs to Notification messages sent to GMSs in partner protected systems. The SeGy GMS will also receive GMKs from within Notification messages sent by GMSs in partner protected systems. Specifically, on the encrypted interface the SeGy:

- Shall support inter-GMS GMK distribution functionality defined in Clauses 5.7 and 11.1.2.2.
- May support verification of EAR elements attached to incoming signalling messages from external security domains as defined in Clause 9.6.
- May support attaching EAR elements to outgoing signalling messages as defined in Clause 9.6.

The SeGy is able to sign and encrypt messages on the encrypted interface as the pseudo GMS using key material provided by the SeGy's Pseudo KMS.

L.3.4 Pseudo MCX Server or IS Proxy

The SeGy performs the security functions of an IS Proxy (or equivalently, the MCX Server) towards protected MC systems. Specifically, the on the encrypted interface SeGy may:

- Establish and use a SPK with protected MC systems as defined in Clauses 5.5 and 9.
- Verify EAR elements attached to incoming signalling messages from external security domains as defined in Clause 9.6.
- Attach EAR elements to outgoing signalling messages as defined in Clause 9.6.

The SeGy is able to sign and encrypt messages as the IS Proxy using key material provided by the Pseudo KMS.

L.3.5 Pseudo MC clients

For each client in an external or unprotected MC system that uses the SeGy's unenrypted interface, the SeGy performs the security functions of an MC client on behalf of the external user. As an external user is signalled from the protected MC system, or sends signalling from within the unprotected MC system, the SeGy creates security credentials on behalf of the user using the Pseudo KMS. Consequently, any group or private communications directed towards a user in the unprotected MC system can be decrypted by the SeGy. Unencrypted communications can then be sent towards the client in the unprotected MC system over the unencrypted interface.

Specifically, on the encrypted interface the SeGy:

- Shall support end-to-end security functionality for MCPTT, MCVideo and MCData defined in Clauses 7 and 8.
- May support verification of EAR elements attached to incoming signalling messages from external security domains as defined in Clause 9.6.
- May support attaching EAR elements to outgoing signalling messages as defined in Clause 9.6.

L.4 Security procedures for the MC Security Gateway (SeGy)

L.4.1 General

The following procedures describe how a MC Security Gateway performs security functions on behalf of an external system (e.g. LMR) or unprotected MC system. For the purposes of these procedures it is assumed that a protected MC system is interconnecting with the encrypted interface, and an unprotected MC system is interconnected with the unencrypted interface.

For all these procedures, signalling sent on the unencrypted interface is not protected. Signalling sent on the encrypted interface may be protected with a SPK shared with a partner protected MC system.

L.4.2 Security procedures for private communication (initiated in the protected MC system)

The following private communication security procedures provide a mechanism for establishing a security context as part of the following private communication requests:

- Private Call Request (MCPTT)
- Private Call Request (MCVideo)
- MCData standalone data request
- MCData session data request
- MCData FD request

These requests are sent from an initiating MC client (on the unencrypted interface) to the terminating MC client (on the encrypted interface) via the MC Security Gateway.

The security procedure for a private communication via an MC Security Gateway is summarized in Figure L.4.2-1, In these procedures, the initiating client follows the security procedures defined in Clause 7.2.2 (MCPTT or MCVideo) or Clause 8.3 (MCData). The terminating client on the unencrypted interface does not use the MC security procedures defined in Clause 7 and 8. Prior to beginning this procedure, it is assumed that the initiating MC client has been provisioned with key material associated with a user's MC service ID by the initiating user's KMS as described in clause 5.3. It is also assumed that the SeGy has established its own ‘pseudo KMS'. Finally, it is assumed that the SeGy's KMS Certificate has been provisioned as an External KMS Certificate to the initiating client by the initiating user's KMS (as defined in Clause 5.3). The SeGy's KMS Certificate shall have the ‘IsSecurityGateway' attribute set to ‘true'.

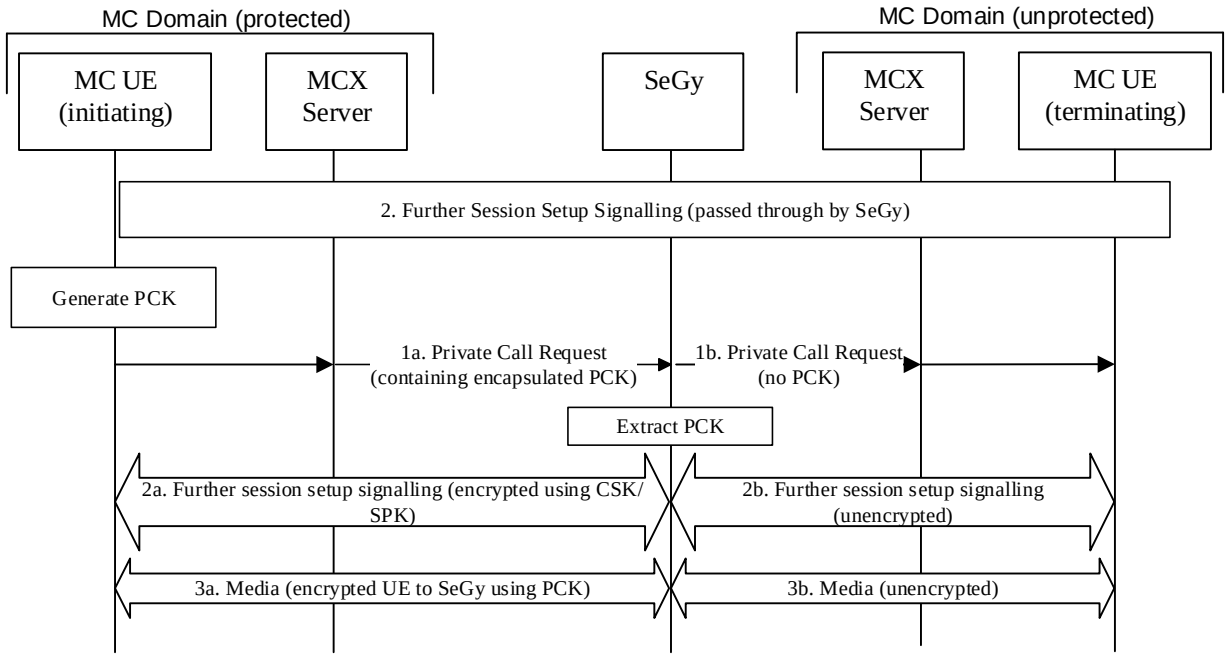

Figure L.4.2-1: Private call security procedure for SeGy (call initiated on the encrypted interface)

The procedure in Figure L.4.2-1 is now described step-by-step.

1a. The initiating MC client generates the PCK and sends a private call request to the terminating entity as defined in Clause 7.2.2 (MCPTT and MCVideo) or Clause 8.3 (MCData). The message is routed via a SeGy. The SeGy

receives the I_MESSAGE and generates the terminating user's decryption key material using its ‘pseudo KMS'. The SeGy uses this key material to decrypt the PCK and stores the PCK and the PCK-ID for future use.

1b. The SeGy removes the I_MESSAGE from the communication request and extracts the PCK. The SeGy forwards the modified communication request towards the terminating MC client.

2a. Further session signalling that occurs between the client and MCX server is protected using the CSK and protected from the MCX server to the SeGy using the SPK.

2b. Further session signalling that occurs between the SeGy and the unprotected MC domain is unencrypted.

3. Communication media sent and received on the encrypted interface is encrypted using the PCK (3a) as defined in Clause 7.5 or 8.5. Communication media sent and received on the unencrypted interface is unencrypted (3b). On receipt of media on the encrypted interface, the SeGy decrypts the media using the PCK and forwards the media on the unencrypted interface. On receipt of media on the unencrypted interface, the SeGy encrypts the media using the PCK and forwards the media on the encrypted interface.

The initiating MC client is aware a MC Security Gateway is in use based upon the ‘IsSecurityGateway' flag in the KMS Certificate used by the SeGy. During the communication, the initiating MC client shall warn the MC user that the communication is via an MC Security Gateway.

L.4.3 Security procedures for private communication (initiated in the unprotected MC system)

The following private communication security procedures provide a mechanism for establishing a security context between the SeGy and a MC client as part of the following private communication requests:

- Private Call Request (MCPTT).
- Private Call Request (MCVideo).
- MCData standalone data request.
- MCData session data request.
- MCData FD request.

These requests are sent from an initiating MC client (on the unencrypted interface) to the terminating MC client (on the encrypted interface) via the MC Security Gateway.

The security procedure for an on-network MCPTT or MCVideo private call via an MC Security Gateway is summarized in Figure L.4.3-1, In these procedures, the terminating client follows the security procedures defined in Clause 7.2.2 (MCPTT and MCVideo) or Clause 8.3 (MCData). The initiating client on the unencrypted interface does not use the MC security procedures defined in Clause 7 and 8.

Prior to beginning this procedure, it is assumed that the terminating MC client has been provisioned with key material associated with a user's MC service ID by the terminating user's KMS as described in Clause 5.3. It is also assumed that the SeGy has established its own ‘pseudo KMS'. Finally, it is assumed that the SeGy's KMS Certificate has been provisioned as an External KMS Certificate to the terminating client by the terminating user's KMS (as defined in Clause 5.3). The SeGy's KMS Certificate shall have the ‘IsSecurityGateway' attribute set to ‘true'.

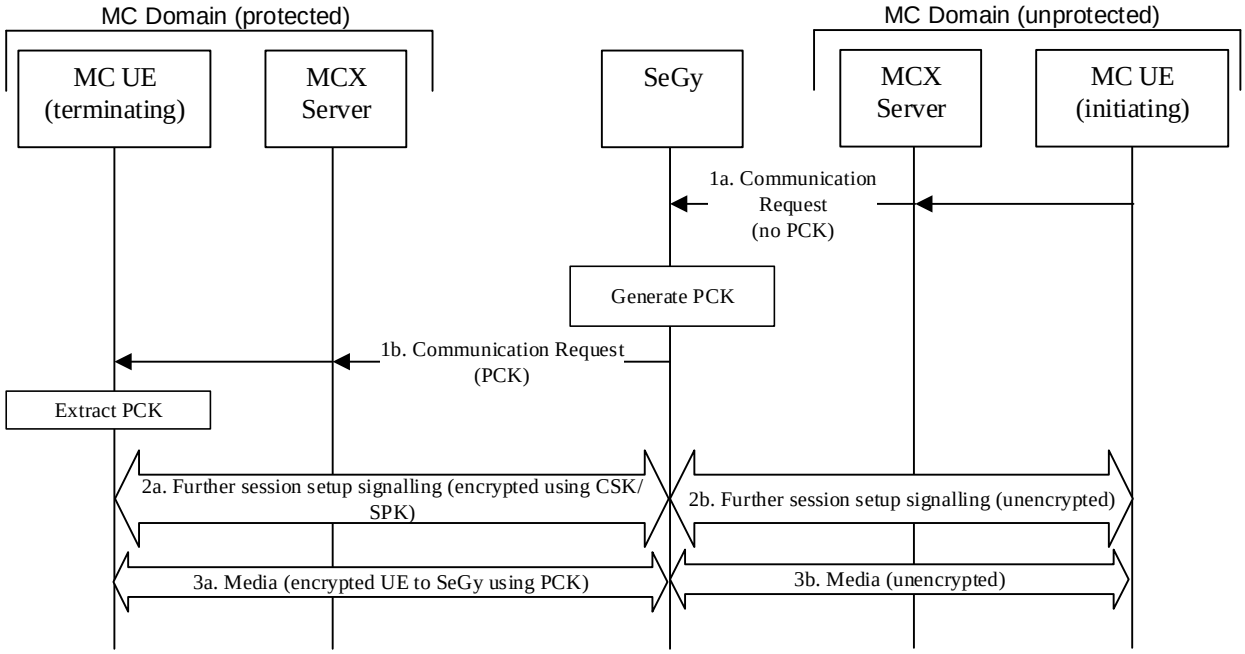

Figure L.4.3-1: Private call security procedure for SeGy (call initiated on the unencrypted interface)

The procedure in figure L.4.3-1 is now described step-by-step.

1a. The initiating client sends a private communication request to the terminating MC client. The message is routed via a SeGy. The SeGy receives the message and generates a PCK and PCK-ID. The SeGy creates an I_MESSAGE and encapsulates the PCK and PCK-ID as defined in Clause 5.6. The SeGy attaches the I_MESSAGE to the received communication request within an SDP offer as defined in IETF RFC 6509 [11]. The modified communication request is forwarded towards the terminiating MC client.

1b. The terminating MC client receives the communication request and processes it as described in Clause 7.2.2 (MCPTT and MCVideo) or Clause 8.3 (MCData).

2a. Further session signalling that occurs between the SeGy and MCX server is protected using the SPK and protected between the MCX server and terminating MC UE client using the CSK.

2b. Further session signalling that occurs between the SeGy and the unprotected MC domain is unencrypted.

3. Communication media sent and received on the encrypted interface is encrypted using the PCK (3a) as defined in Clause 7.5 or 8.5. Communication media sent and received on the unencrypted interface is unencrypted (3b). On receipt of media on the encrypted interface, the SeGy decrypts the media using the PCK and forwards the media on the unencrypted interface. On receipt of media on the unencrypted interface, the SeGy encrypts the media using the PCK and forwards the media on the encrypted interface.

The terminating MC client is aware a MC Security Gateway is in use based upon the ‘IsSecurityGateway' flag in the KMS Certificate used by the SeGy. During the communication, the terminating MC client shall warn the MC user that the communication is via an MC Security Gateway.

L.4.4 Security procedures for group communications (group homed in the protected MC system)

This procedure uses a MIKEY payload to distribute a GMK from the GMS to another GMS to support group interconnection. The GMS follows the procedures in Clause 5.7 and 11.1.2.2. In this clause, it is assumed that at least one group member is in the unprotected system and hence the Notify group request containing the GMK is routed to the GMS in the unprotected system.

Prior to beginning this procedure, it is assumed that the GMS has been provisioned by its KMS with key material associated with its identity. It is also assumed that the SeGy has established its own ‘pseudo KMS'. Finally, it is assumed that the SeGy's KMS Certificate has been provisioned as an External KMS Certificate to the GMS by the GMS's KMS (as defined in Clause 5.3). The SeGy's KMS Certificate shall have the ‘IsSecurityGateway' attribute set to ‘true'.

Figure L.4.4-1 shows the security procedures for creating a security association for a group with a SeGy.

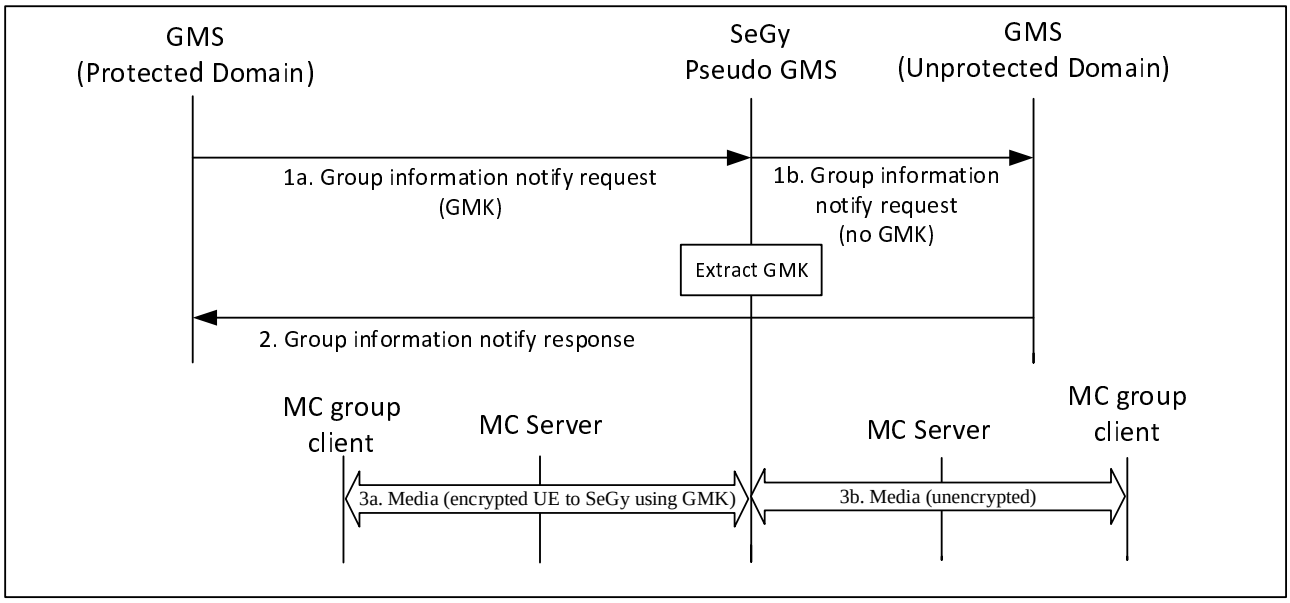

Figure L.4.4-1: Security configuration for MC groups (where a group member is behind a SeGy)

A description of the procedures depicted in figure L.4.4-1 follows:

1a. The GMS shall send a MIKEY payload containing a GMK to the GMS in an interconnected system within a 'Group information notify request' message as defined in Clause 11.1.2.2. Where the interconnected system is unprotected and hence is behind a SeGy, the 'Group information notify request' is sent via the SeGy. The SeGy shall generate the Pseudo GMS's identity-based key material using its pseudo-KMS and use this key material to extract the GMK and GMK-ID from the I_MESSAGE within the ‘Notify group request'.

1b. The SeGy shall remove the I_MESSAGE from the 'Group information notify request' and forward the modified request towards the unprotected MC system's GMS.

2. The SeGy shall forward on further signalling invisibly (including the ‘Notify response').

3a. Group media sent and received on the encrypted interface is encrypted using the GMK (3a) as defined in Clause 7.5 or 8.5. Group signalling sent and received on the encrypted interface is protected as defined in clause 9. On receipt of media on the encrypted interface, the SeGy decrypts the media using the GMK and GMK-ID and forwards the media on the unencrypted interface.

3b. Group media sent and received on the unencrypted interface is unencrypted (3b). Group signalling sent and received on the unencrypted interface is unprotected. On receipt of media on the unencrypted interface, the SeGy encrypts the media using the GMK and forwards the media on the encrypted interface.

The GMS is aware a MC Security Gateway is in use based upon the ‘IsSecurityGateway' flag in the KMS Certificate used by the SeGy. When any group member is behind a Security Gateway, the GMS shall set the ‘Security Gateway' flag within the ‘Status' field of the group GMK's key parameters (as defined in Clause E.6.9).

The MC group clients within the protected MC system are aware the MC Security Gateway is in use based upon the ‘Security Gateway' flag within the ‘Status' field of the GMK's key parameters (as defined in Clause E.6.9). During a communication encrypted with the GMK, the MC group client shall warn the MC user that the communication may be via an MC Security Gateway.

L.4.5 Security procedures for group communications (group homed in the unprotected MC system)

In this clause, it is assumed that the group is owned by a GMS inside the unprotected system and group members and their GMS are inside the protected domain. The GMS in the protected domain follows the procedures in Clause 5.7 and 11.1.2.2. A Notify group request is routed from the GMS in the unprotected domain to the GMS in the protected system.

Prior to beginning this procedure, it is assumed that the GMS in the protected domain has been provisioned by its KMS with key material associated with its identity. It is also assumed that the SeGy has established its own ‘pseudo KMS'. Finally, it is assumed that the SeGy's KMS Certificate has been provisioned as an External KMS Certificate to the GMS in the protected system by the GMS's KMS (as defined in Clause 5.3). The SeGy's KMS Certificate shall have the ‘IsSecurityGateway' attribute set to ‘true'.

Figure L.4.5-1 shows the security procedures for creating a security association for a group with a SeGy.

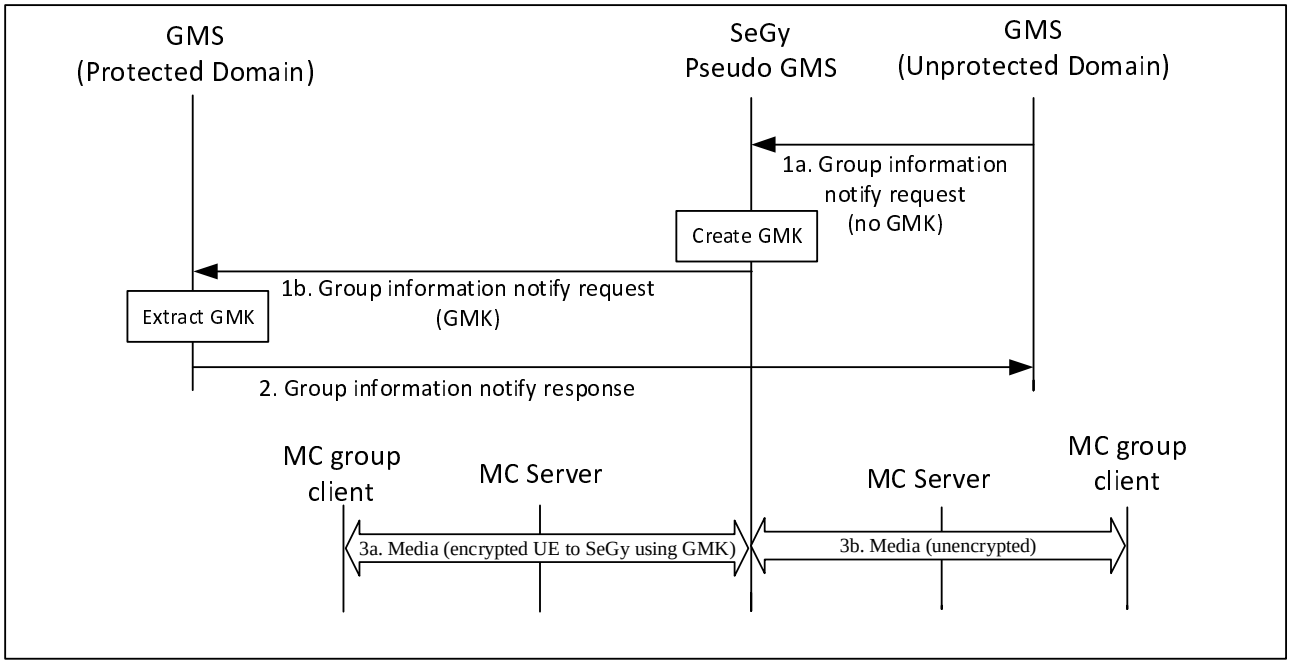

Figure L.4.5-1: Security configuration for MC groups (where group is homed in an unprotected domain)

A description of the procedures depicted in figure L.4.5-1 follows:

1a. The GMS in the unprotected system send a Group Notify to the GMS in an interconnected system within a 'Group information notify request' message as defined in Clause 11.1.2.2. Where the interconnected system is protected and hence is behind a SeGy, the 'Group information notify request' is sent via the SeGy.

1b. On receipt of a Notify group request on the unencrypted interface, the SeGy shall generate a GMK and GMK-ID and encrypt the GMK to the GMS in the protected system using the SeGy's Pseudo GMS's identity-based key material. SeGy shall set the ‘Security Gateway' flag within the ‘Status' field of the group GMK's key parameters (as defined in Clause E.6.9). The encapsulated GMK and GMK-ID is attached to the ‘Notify group request' within an I_MESSAGE as defined in Clause 5.7. The modified ‘Notify group request' is sent on by the SeGy to the GMS in the protected system.

2. The SeGy shall forward on further signalling invisibly (including the ‘Notify response').

3a. Group media sent and received on the encrypted interface is encrypted using the GMK (3a) as defined in Clause 7.5 or 8.5. Group signalling sent and received on the encrypted interface is protected as defined in clause 9. On receipt of media on the encrypted interface, the SeGy decrypts the media using the GMK and GMK-ID and forwards the media on the unencrypted interface.

3b. Group media sent and received on the unencrypted interface is unencrypted (3b). Group signalling sent and received on the unencrypted interface is unprotected. On receipt of media on the unencrypted interface, the SeGy encrypts the media using the GMK and forwards the media on the encrypted interface.

The GMS in the protected system is aware a MC Security Gateway is in use based upon the 'IsSecurityGateway' flag in the KMS Certificate used by the SeGy and as the 'Security Gateway' flag will be set within the 'Status' field of the group GMK's key parameters (as defined in Clause E.6.9).

On receipt of the GMK, the GMS in the protected domain shall distribute the key to group clients as defined in Clause 5.7. MC group clients in the protected system are aware the MC Security Gateway is in use based upon the 'Security Gateway' flag within the 'Status' field of the GMK's key parameters (as defined in Clause E.6.9). During a communication encrypted with the GMK, the MC group client shall warn the MC user that the communication may be via an MC Security Gateway.

L.5 Interworking using a MC Security Gateway

L.5.1 General

Interworking with Land Mobile Radio Systems is defined in TS 23.283[48]. An interworking function (IWF) is required to allow the MC System to interwork with Land Mobile Radio Systems.

L.5.2 MC Security Gateway and the IWF

The functional model for the SeGy as used within the IWF is shown in Figure L.5.2-1. Where the IWF terminates the security of the 3GPP MC Domain, the IWF performs the functions of a SeGy for that purpose.

For interworking communications sent towards the non-3GPP system, an MC gateway with an IS Proxy and the HTTP proxy are used to provide topology hiding and terminate external routing as defined in clause 11.1.3 and the IWF processes the signalling and media for use in the Land Mobile Radio System after terminating the 3GPP MC system security. Where the media and signalling between an MC Domain and IWF is not encrypted using 3GPP MC security mechanisms, the SeGy functionality is not applied by the IWF, allowing the media and signaling to pass directly through for processing by the IWF.

For interworking communications sent from a Land Mobile Radio system towards the 3GPP system, the IWF processes the signalling and media from the Land Mobile Radio system prior to applying 3GPP security and sending it into the 3GPP system. Where the media and signalling between an MC Domain and IWF is not encrypted using 3GPP MC security mechanisms, the SeGy functionality is not applied by the IWF, allowing the processed media and signaling to pass directly from the IWF into the 3GPP system.

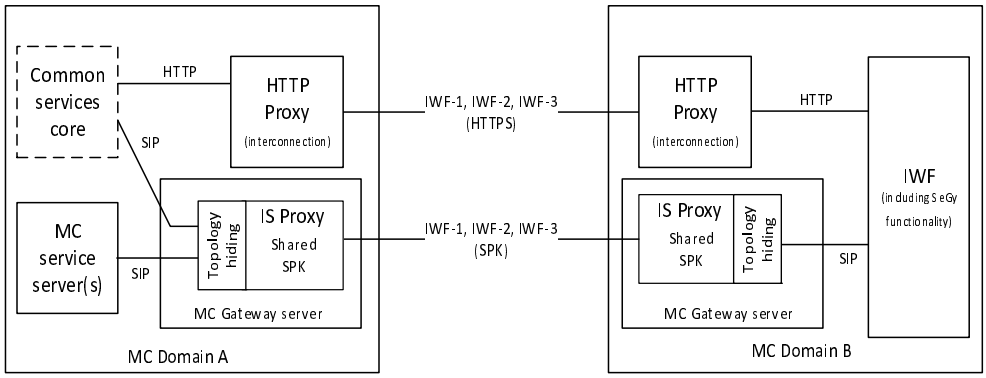

Figure L.5.2-1: Functional model for MC Security Gateway use during interworking

The IWF-1 reference point is defined in 23.283 [48] and provides for the transfer of MCPTT media and signalling between a 3GPP MC domain MCPTT server and the IWF. Authentication and security of this interface shall be as described in clause 6.

The IWF-2 reference point is defined in 23.283 [48] and provides for the transfer of MCData media and signalling between a 3GPP MC domain MCData server and the IWF. Authentication and security of this interface shall be as described in clause 6.

The IWF-3 reference point is defined in 23.283 [48] and provides for the transfer of group management information between a 3GPP MC domain GMS and the IWF. Authentication and security of this interface shall be as described in clause 6.

Any security applied by the non-3GPP system to MCPTT or MCData media and signalling, or any interfaces within the non-3GPP system is defined by the non-3GPP system and is out of scope for this document.

Annex M (informative): Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2017-06	SA#76					Upggrade to change control version	14.0.0
2017-09	SA#77	SP-170639	0001	-	F	Ambient Listening and ambient viewing	14.1.0
2017-09	SA#77	SP-170639	0002	1	F	Group communications and emergencies	14.1.0
2017-09	SA#77	SP-170639	0005	-	F	Fix IdM token response message	14.1.0
2017-09	SA#77	SP-170639	0006	-	F	Token revocation	14.1.0
2017-09	SA#77	SP-170639	0008	-	F	Video push and video pull	14.1.0
2017-09	SA#77	SP-170639	0009	-	F	Clarifications of key period calculation	14.1.0
2017-09	SA#77	SP-170639	0010	-	F	Clarifications of security domain parameters and UK-ID	14.1.0
2017-09	SA#77	SP-170639	0011	-	F	Clarifications and editorial corrections related to SRTCP protection	14.1.0
2017-09	SA#77	SP-170639	0012	1	F	Correction of parameters for use of MIKEY-SAKKE	14.1.0
2017-09	SA#77	SP-170639	0014	1	F	Corrections to MCData security procedures	14.1.0
2017-09	SA#77	SP-170639	0015	1	F	General Corrections to TS 33.180	14.1.0
2017-09	SA#77	SP-170639	0016	-	F	MCData payload authentication correction	14.1.0
2018-01	SA#78	SP-170874	0017	-	F	Corrections to MCData security procedures	14.2.0
2018-01	SA#78	SP-170874	0019	-	F	Add transmission control for MCVideo	14.2.0
2018-01	SA#78	SP-170874	0020	-	F	MCPTT to MCX fixes	14.2.0
2018-01	SA#78	SP-170874	0021	-	F	SIP MESSAGE clarification for MCData	14.2.0
2018-01	SA#78	SP-170874	0030	1	F	A Clarification on SSRC use in group communications	14.2.0
2018-01	SA#78	SP-170874	0032	1	F	Fix inter-domain IdM token exchange procedure	14.2.0
2018-01	SA#78	SP-170874	0035	-	F	Fix reference to 33.179	14.2.0
2018-01	SA#78	SP-170874	0036	-	F	Fix media security for private call	14.2.0
2018-01	SA#78	SP-170874	0037	1	F	Fix client check during GMK provisioning	14.2.0
2018-01	SA#78	SP-170874	0038	1	F	Alignment with MuSiK Stage 3 in CT1 specs 24.379 and 24.481	14.2.0
2018-01	SA#78	SP-170874	0039	1	F	Key parameters payload correction	14.2.0
2018-01	SA#78	SP-170877	0026	1	B	Adding KMS Redirect Responses	15.0.0
2018-01	SA#78	SP-170877	0027	1	B	KMS enhancement, including Migration KMS	15.0.0
2018-01	SA#78	SP-170877	0028	1	B	Addition of Clause on Logging, Audit and Discreet Monitoring	15.0.0
2018-01	SA#78	SP-170877	0029	1	B	Addition of Signalling Proxies	15.0.0
2018-01	SA#78	SP-170877	0040	1	B	Addition of Element for Authenticating Requests (EAR)	15.0.0
2018-01	SA#78	SP-170877	0041	-	B	Addition of KMS Requests to support KMS Discovery	15.0.0
2018-01	SA#78	SP-170877	0043	1	B	Addition of Security Gateway	15.0.0
2018-03	SA#79	SP-180043	0045	3	B	Interconnection, Interworking media and signaling	15.1.0
2018-03	SA#79	SP-180043	0046	1	F	Interworking key management (InterSD)	15.1.0
2018-03	SA#79	SP-180043	0048	1	B	Interworking SeGy clarification	15.1.0
2018-03	SA#79	SP-180043	0049	-	B	[eMCSEC] Addition of indicators on the use of Security Gateways	15.1.0
2018-03	SA#79	SP-180043	0051	-	B	Adding Integrity Key for KMS communications	15.1.0
2018-03	SA#79	SP-180043	0054	2	A	GMK management clarification	15.1.0
2018-03	SA#79	SP-180043	0055	2	A	MC key storage and persistence	15.1.0
2018-03	SA#79	SP-180051	0056	2	B	Security of functional alias(es)	15.1.0
2018-03	SA#79	SP-180051	0057	1	B	Security of Multi-talker	15.1.0
2018-03	SA#79	SP-180043	0059	-	B	Providing details of EARs into Annex J	15.1.0
2018-03	SA#79	SP-180043	0060	1	F	Clarification of purpose of Inter-domain user service authorisation	15.1.0
2018-03	SA#79	SP-180043	0061	-	F	[eMCSEC] Correction of reference to SA1 specification	15.1.0
2018-06	SA#80	SP-180447	0064	-	F	Interconnection references clarification	15.2.0
2018-06	SA#80	SP-180447	0065	-	F	Mixing of encrypted media	15.2.0
2018-06	SA#80	SP-180447	0066	-	B	Migration user authentication and authorisation	15.2.0
2018-06	SA#80	SP-180447	0067	-	F	Various technical clarifications	15.2.0
2018-06	SA#80	SP-180447	0068	1	F	Removal of Editor's note in Clause I.3.4	15.2.0
2018-06	SA#80	SP-180447	0069	1	C	Resolution of editor's notes within Clause 10 on logging, audit and discreet monitoring.	15.2.0
2018-06	SA#80	SP-180447	0071	-	A	Addition of test vector for MIKEY-SAKKE UID	15.2.0
2018-06	SA#80	SP-180447	0073	-	A	Removal of Editor's note in Clause 5.1.3.1.	15.2.0
2018-06	SA#80	SP-180446	0075	1	A	[eMCSec] 33180 R15 technical clarification for a proxy usage	15.2.0
2018-06	SA#80	SP-180446	0076	1	F	[eMCSec] 33180 R15 Migration KMS clarification	15.2.0
2018-06	SA#80	SP-180445	0078	-	A	Definition of KMS XML namespace	15.2.0
2018-06	SA#80	SP-180446	0080	1	A	Addition of note to say that temporary group regroup mechanism is not secured.	15.2.0
2018-06	SA#80	SP-180446	0082	-	A	Inclusion of MCData message types as defined by CT1	15.2.0
2018-06	SA#80	SP-180447	0083	-	F	Making Annex J normative	15.2.0
2018-06	SA#80	SP-180447	0084	-	B	Definition of KMS Redirect Request message format	15.2.0
2018-09	SA#81	SP-180702	0086	1	A	[MCSec] 33180 R15. Examples of MC service ID shall be URI	15.3.0
2018-09	SA#81	SP-180702	0088	1	A	[MCSec] 33180 R15. Clarification for MIKEY-SAKKE values	15.3.0
2018-09	SA#81	SP-180702	0091	-	A	[MCSec] 33180 R15 Fix XML schema (mirror)	15.3.0
2018-09	SA#81	SP-180702	0093	-	A	[MCSec] 33180 R15 FC values for MCData (mirror)	15.3.0
2018-09	SA#81	SP-180703	0094	-	F	[MCSec] 33180 R15 registered media type	15.3.0
2019-03	SA#83	SP-190101	0097	1	A	Annex D.3.5.2 XSD correction (mirror)	15.4.0
2019-03	SA#83	SP-190101	0099	1	A	[33.180] R15 IdMS interface security (mirror)	15.4.0
2019-03	SA#83	SP-190101	0103	1	A	[33.180] R15 InK clarifications (mirror)	15.4.0

2019-03	SA#83	SP-190101	0105	1	A	[33.180] R15 MCX identity clairfication (mirror)	15.4.0
2019-06	SA#84	SP-190356	0107	1	A	[MCSec] 33180 R15. Clarification of the references to RFC 3711	15.5.0
2019-06	SA#84	SP-190356	0109	-	A	[33.180] R15 XSD Corrections (mirror)	15.5.0
2019-06	SA#84	SP-190356	0111	1	A	[33.180] R15 Remove IANA editor's notes (mirror)	15.5.0
2019-06	SA#84	SP-190357	0112	1	B	[33.180] R16 Establishment of PCK for MCData	16.0.0
2019-09	SA#85	SP-190680	0114	-	A	[33.180] R16 - Fix hash result (mirror)	16.1.0
2019-12	SA#86	SP-191209	0117	-	A	[MCXSec] 33180 R16 Missing Abbreviations (Mirror)	16.2.0
2019-12	SA#86	SP-191209	0118	-	A	[MCXSec] 33180 R16 Reference Addition (Mirror)	16.2.0
2019-12	SA#86	SP-191209	0119	-	A	[MCXSec] 33180 R16 Correction concerning IdM client (Mirror)	16.2.0
2019-12	SA#86	SP-191209	0128	-	A	[33.180] R16 Fix bad reference	16.2.0
2019-12	SA#86	SP-191136	0129	1	F	[33.180] R16 Consistent use of off-network	16.2.0
2019-12	SA#86	SP-191136	0130	-	F	[33.180] R16 KM client to KMS security	16.2.0
2019-12	SA#86	SP-191136	0131	1	F	[33.180] R16 TrK-ID and InK-ID	16.2.0
2019-12	SA#86	SP-191136	0132	-	C	[33.180] R16 InterSD KM record	16.2.0
2019-12	SA#86	SP-191136	0133	-	F	[33.180] R16 ETSI Plugtest clarifications	16.2.0
2019-12	SA#86	SP-191136	0134	1	B	Algorithm selection for MCData signalling protection	16.2.0
2020-03	SA#87E	SP-200135	0135	-	D	[33.180] Formatting corrections	16.3.0
2020-03	SA87E	SP-200135	00136	1	B	[33.180] R16 Gateway security	16.3.0
2020-03	SA87E	SP-200135	0137	-	B	[33.180] R16 - MC location authorization	16.3.0
2020-03	SA87E	SP-200135	0138	1	F	[33.180] R16 SeGy IWF corrections	16.3.0
2020-03	SA87E	SP-200135	0139	-	F	Correction to definition about temporary group call related procedures	16.3.0
2020-07	SA88E	SP-200362	0146	-	F	[33.180] R16 Fix IdM client terminology	16.4.0
2020-07	SA88E	SP-200362	0147	-	D	[33.180] R16 Fix XML references	16.4.0
2020-07	SA88E	SP-200362	0148	-	F	[33.180] R16 TrK-ID and InK-ID indication	16.4.0

History

Document history		
V16.4.0	August 2020	Publication