Service and Content Protection for Mobile Broadcast Services

Approved Version 1.0.1 – 09 Jan 2013

Open Mobile Alliance

OMA-TS-BCAST_SvcCntProtection-V1_0_1-20130109-A

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Use of this document is subject to all of the terms and conditions of the Use Agreement located at .

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the "OMA IPR Declarations" list at http://www.openmobilealliance.org/ipr.html. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE "OMA IPR DECLARATIONS" LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.

THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

Contents

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Figures

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Tables

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

1. Scope

This document specifies the service protection and content protection systems, and affiliated mechanisms, which support various business models of OMA BCAST enabled mobile broadcast services delivery. On behalf of broadcast service providers and content providers, means are provided to protect the access to, and control the consumption of, broadcast content in either streaming or file delivery format. Two main systems can be used to provide service protection or content protection: the DRM Profile and the Smartcard Profile.

Fundamental components of the service and content protection systems consist of various content encryption mechanisms, protection signalling, and key management related messages which may carry rights objects, other post-reception consumption attributes (such as recording permission), key material, and parental rating criteria. In addition to server-client (i.e., network-to-terminal) interactions, this document also normatively specifies the server-side interfaces pertaining to service and content protection.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

2. References

2.1 Normative References

The version and release numbers specified for the 3GPP and 3GPP2 references in this section are the minimum version and release numbers that can be used. The references are not meant to be restricted to these versions and releases except if explicitly mentioned with the wording "Restricted to…"; subsequent versions and releases can also be used because they are required to be backward compatible. For example, the minimum version of 3GPP TS 33.222 is the release 6 but the use of the relesae 7 is acceptable as well.

[3GPP TS 23.003 v6] "Numbering, Addressing and Identification (Release 6)", 3rd Generation Partnership Project, 3 GPP TS 23.003,
URL: http://www.3gpp.org/

[3GPP TS 23.032 v6] "Universal Geographical Area Description (GAD) (Release 6)", 3rd Generation Partnership Project, 3 GPP TS 23.032,
URL: http://www.3gpp.org/

[3GPP TS 26.346 v7] "Multimedia Broadcast/Multicast Service (MBMS), Protocols and codecs (Release 7)", Technical Specification Group Services and System Aspects, 3rd Generation Partnership Project, 3GPP TS 26.346,
URL: http://www.3gpp.org/

[3GPP TS 31.101 v7] "UICC-terminal interface; Physical and logical characteristics (Release 7)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 31.101,
URL: http://www.3gpp.org/

[3GPP TS 31.102 v7] "Characteristics of the Universal Subscriber Identity Module (USIM) application (Release 7)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 31.102,
URL: http://www.3gpp.org/

[3GPP TS 31.111 v9.1.0] "Universal Subscriber Identity Module (USIM) Application Toolkit (USAT)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 31.111, Restricted to Version 9.1.0
URL: http://www.3gpp.org/

[3GPP TS 33.110 v7] "Key Establishment between a Universal Integrated Circuit Card (UICC) and a Terminal (Release 7)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 33.110,
URL: http://www.3gpp.org/

[3GPP TS 33.220 v6] "Generic Authentication Architecture, Generic Bootstrapping Architecture (Release 6)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 33.220,
URL: http://www.3gpp.org/

[3GPP TS 33.222 v6] "Generic Authentication Architecture, Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Release 6)", 3rd Generation Partnership Project, Technical Specification 3GPP TS 33.222,
URL: http://www.3gpp.org/

[3GPP TS 33.246 v7] "3G Security; Security of Multimedia Broadcast/Multicast Service (Release 7)", Technical Specification Group Services and System Aspects, 3rd Generation Partnership Project, 3GPP 33.246,
URL: http://www.3gpp.org/

[3GPP TS 51.011 v4] "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (Release 4)", rd Generation Partnership Project, Technical Specification 3GPP TS 51.011,
URL: http://www.3gpp.org/

[3GPP2 C.S0002-0] "Physical Layer Standard for cdma2000 Spread Spectrum Systems", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0002-0,
URL: http://www.3gpp2.org/

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

[3GPP2 C.S0005-D] "Upper Layer (Layer 3) Signalling Standard for cdma2000 Spread Spectrum Systems", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0005-D,
URL: http://www.3gpp2.org/

[3GPP2 C.S0023-C] "Removable User Identity Module for Spread Spectrum Systems", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0023-C,
URL: http://www.3gpp2.org/

[3GPP2 C.S0024-A] "cdma2000 High Rate Packet Data Air Interface Specification", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0024-A,
URL: http://www.3gpp2.org/

[3GPP2 C.S0035-A] "CDMA Card Application Toolkit (CCAT)", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0035-A,
URL: http://www.3gpp2.org/

[3GPP2 C.S0054-0] "cdma000 High Rate Broadcast-Multicast Packet Data Air Interface Specification", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0054-0,
URL: http://www.3gpp2.org/

[3GPP2 C.S0065-0] "cdma2000 Application on UICC for Spread Spectrum Systems", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0065-0,
URL: http://www.3gpp2.org/

[3GPP2 C.S0072-0] "Mobile Station Equipment Identifier (MEID) Support for cdma2000 Spread Spectrum Systems", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 C.S0072-0,
URL: http://www.3gpp2.org/

[3GPP2 S.S0083-A] "Broadcast-Multicast Service Security Framework", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 S.S0083-A,
URL: http://www.3gpp2.org/

[3GPP2 X.S0022-A] "Broadcast and Multicast Service in cdma2000 Wireless IP Network", 3rd Generation Partnership Project 2, Technical Specification 3GPP2 X.S0022-A,
URL: http://www.3gpp2.org/

[BCAST10-Architecture] "Mobile Broadcast Services Architecture", Open Mobile Alliance™, OMA-AD- BCAST-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-BCMCS-Adaptation] "Broadcast Distribution System Adaptation – 3GPP2/BCMCS", Open Mobile Alliance™, OMA-TS-BCAST_BCMCS_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-Distribution] "File and Stream Distribution for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_Distribution-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-DVBH-IPDC-Adaptation] "Broadcast Distribution System Adaptation – IPDC over DVB-H", Open Mobile Alliance™, OMA-TS-BCAST_DVB_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-MBMS-Adaptation] "Broadcast Distribution System Adaptation – 3GPP/MBMS", Open Mobile Alliance™, OMA-TS-BCAST_MBMS_Adaptation-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-ServContProt] "Service and Content Protection for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_SvcCntProtection-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-Services] "Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-BCAST_Services-V1_0,
URL: http://www.openmobilealliance.org/

[BCAST10-SG] "Service Guide for Mobile Broadcast Services", Open Mobile Alliance™, OMA-TS-

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	BCAST_ServiceGuide-V1_0, URL: http://www.openmobilealliance.org/
[BCAST10-XMLSchema-SPCP-Backend]	"Mobile Broadcast Services – XML Schema SP/CP Backend Messages", Open Mobile Alliance™,OMA-SUP-XSD_bcast_spcp_backend-V1_0, URL: http://www.openmobilealliance.org/
[DRM Enabler-v2.0]	*OMA-DRM-V2_0 enabler*, Open Mobile Alliance™, URL: http://www.openmobilealliance.org/
[DRMCF-v2.0]	"DRM Content Format V2.0", Open Mobile Alliance™, OMA-DRM-DCF-V2_0, URL: http://www.openmobilealliance.org/
[DRMDRM-v2.0]	"DRM Specification V2.0", Open Mobile Alliance™, OMA-DRM-DRM-V2_0, URL: http://www.openmobilealliance.org/
[ETSI EN 300 468 V1.6.1]	"Digital Video Broadcasting (DVB); Specification for Service Infor-mation (SI) in DVB systems", November 2004, URL: http://www.etsi.org/
[ETSI EN 302 304 V1.1.1]	"Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)", URL: http://www.etsi.org/
[ETSI TS 102.221]	"Smart Cards; UICC-Terminal interface; Physical and Logical Characteristics", URL: http://www.etsi.org/
[ETSI TS 102.484]	"Secure Channel between a UICC and an End Point Terminal", ETSI SmartCard Platform, URL: http://www.etsi.org/
[FIPS197]	*ADVANCED ENCRYPTION STANDARD (AES)*, Federal Information Processing Standards Publication 197, URL: http://csrc.nist.gov/publications/fips/
[FIPS198]	*The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standards Publication 198, URL: http://csrc.nist.gov/publications/fips/
[IOPPROC]	"OMA Interoperability Policy and Process", Version 1.1, Open Mobile Alliance™, OMA-IOP-Process-V1_1, URL: http://www.openmobilealliance.org/
[ISMACRYP11]	"ISMA 1.0 Encryption and Authentication, Version 1.1", release version, URL: http://www.isma.tv
[ISMACRYP20]	"ISMA Encryption and Authentication, Version 2.0", URL: http://www.isma.tv
[ISO-3166]	"Codes for the representation of names of countries and their subdivisions", URL: http://www.iso.org/iso/en/prods-services/iso3166ma/index.html
[ISO/IEC 7816-4]	"Identification cards – Integrated circuit cards; Part4: Organization, security and commands for interchange", URL: http://www.iso.org/
[ITU-MCC]	"List of Mobile Country or Geographical Area Codes", ITU-T Telecommunication Standardization Sector of ITU Complement To ITU-T Recommendation E.212 (05/2004), URL: http://www.itu.int/dms_pub/itu-t/opb/sp/T-SP-E.212A-2007-PDF-E.pdf Note: This List will be updated regularly by numbered series of amendments published in ITU Operational Bulletin. For the latest version see: URL: http://www.itu.int/itu-t/bulletin/annex.html
[ITU-T Recommendation E.118]	"Series E: Overall Network Operation, Telephone Service, Service Operation and Human Factors; The international Telecommunication Charge Card", URL: http://www.itu.int/ITU-T/publications

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

[ITU-T Recommendation E.164] "Series E: Overall Network Operation, Telephone Service, Service Operation and Human Factors; The international public telecommunication numbering plan",
URL: http://www.itu.int/ITU-T/publications

[OMA MLP] "Mobile Location Protocol 3.2", Open Mobile Alliance™, OMA-TS-MLP-V3_2-20051124-C,
URL: http://www.openmobilealliance.org/

[OMA Push] "OMA Push V2.1", Open Mobile Alliance™, OMA-ERP-Push-V2_1-20051122-C,
URL: http://www.openmobilealliance.org/

[OMNA] *Open Mobile Naming Authority*, Open Mobile Alliance™,
URL: http://www.openmobilealliance.org/tech/omna

[RFC1982] "Serial Number Arithmetic", R. Elz, R. Bush, August 1996,
URL: http://www.ietf.org/rfc/rfc1982.txt

[RFC2045] "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed, N. Borenstein, November 1996,
URL: http://www.ietf.org/rfc/rfc2045.txt

[RFC2104] "HMAC: Keyed-Hashing for Message Authentication", H. Krawczyk, M. Bellare, R. Canetti, February 1997,
URL: http://www.ietf.org/rfc/rfc2104.txt

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997,
URL: http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] "The TLS Protocol, Version 1.0", T. Dierks, C.Allen, January 1999,
URL: http://www.ietf.org/rfc/rfc2246.txt

[RFC2392] "Content-ID and Message-ID Uniform Resource Locators", E. Levinson, August 1998,
URL: http://www.ietf.org/rfc/rfc2392.txt

[RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, August 1998,
URL: http://www.ietf.org/rfc/rfc2396.txt

[RFC2401] "Security Architecture for the Internet Protocol", S. Kent, R. Atkinson, November 1998,
URL: http://www.ietf.org/rfc/rfc2401.txt

[RFC2404] "The Use of HMAC-SHA-1-96 within ESP and AH", C. Madson, R. Glenn, November 1998,
URL: http://www.ietf.org/rfc/rfc2404.txt

[RFC2406] "IP Encapsulating Security Payload (ESP)", S. Kent, R. Atkinson, November 1998,
URL: http://www.ietf.org/rfc/rfc2406.txt

[RFC2451] "The ESP CBC-Mode Cipher Algorithms", R. Pereira, R. Adams, November 1998,
URL: http://www.ietf.org/rfc/rfc2451.txt

[RFC2617] "HTTP Authentication: Basic and Digest Access Authentication", J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, June 1999,
URL: http://www.ietf.org/rfc/rfc2617.txt

[RFC3237] "Requirements for Reliable Server Pooling", M. Tuexen, Q. Xie, R. Stewart, M. Shore, L. Ong, J. Loughney, M. Stillman, January 2002,
URL: http://www.ietf.org/rfc/rfc3237.txt

[RFC3548] "The Base16, Base32, and Base64 Data Encodings", S. Josefsson, Ed., July 2003,
URL: http://www.ietf.org/rfc/rfc3548.txt

[RFC3566] "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", S. Frankel, H. Herbert, September 2003,
URL: http://www.ietf.org/rfc/rfc3566.txt

[RFC3602] "The AES-CBC Cipher Algorithm and Its Use with IPsec", S. Frankel, R. Glenn, S. Kelly, September 2003,

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

URL: http://www.ietf.org/rfc/rfc3602.txt

[RFC3629] "UTF-8, a transformation format of ISO 10646", F. Yergeau, November 2003,
URL: http://www.rfc-editor.org/rfc/rfc3629.txt

[RFC3640] "RTP Payload Format for Transport of MPEG-4 Elementary Streams", J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, P. Gentric, November 2003,
URL: http://www.ietf.org/rfc/rfc3640.txt

[RFC3664] "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", P. Hoffman, January 2004,
URL: http://www.ietf.org/rfc/rfc3664.txt

[RFC3711] "The Secure Real-time Transport Protocol (SRTP)", M. Baugher, D. McGrew, M. Naslund, E. Carrara, K. Norrman, March 2004,
URL: http://www.ietf.org/rfc/rfc3711.txt

[RFC3830] "MIKEY: Multimedia Internet KEYing", J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman, August 2004,
URL: http://www.ietf.org/rfc/rfc3830.txt

[RFC3986] "Uniform Resource Identifier (URI): Generic Syntax ", T. Berners-Lee, R. Fielding, L. Masinter, January 2005,
URL: http://www.ietf.org/rfc/rfc3986.txt

[RFC4281] "The Codecs Parameter for Bucket Media Types", R.Gellens, D. Singer, P. Frojdh, November 2005,
URL: http://www.ietf.org/rfc/rfc4281.txt

[RFC4301] "Security Architecture for the Internet Protocol", S. Kent and K. Seo, December 2005,
URL: http://www.ietf.org/rfc/rfc4301.txt

[RFC4563] "The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY)", E. Carrara, V. Lehtovirta, K. Norrman, June 2006,
URL: http://www.ietf.org/rfc/rfc4563.txt

[RFC4566] "SDP: Session Description Protocol", M. Handley, V. Jacobson, C. Perkins, July 2006,
URL: http://www.ietf.org/rfc/rfc4566.txt

[RFC4568] "Session Description Protocol (SDP) Security Descriptions for Media Streams", F. Andreasen, M. Baugher, D. Wing, July 2006,
URL: http://www.ietf.org/rfc/rfc4568.txt

[RFC4771] "Integrity Transform Carrying Roll-Over Counter for the Secure Real-time Transport Protocol (SRTP)", V. Lehtovirta, M. Naslund, K. Norrman, January 2007,
URL: http://www.ietf.org/rfc/rfc4771.txt

[RFC5159] "Session Description Protocol (SDP) Attributes for Open Mobile Alliance (OMA) Broadcast (BCAST) Service and Content Protection", L. Dondeti, Ed., A. Jerichow, March 2008,
URL: http://www.ietf.org/rfc/rfc5159.txt

[RFC5410] "Multimedia Internet KEYing (MIKEY) General Extension Payload for Open Mobile Alliance BCAST 1.0", A. Jerichow, Ed., L. Piron, January 2009,
URL: http://www.ietf.org/rfc/rfc5410.txt

[SIMULCRYPT] "Digital Video Broadcasting (DVB); Head-end implementation of DVB SimulCrypt", ETSI Publication ETSI TS 103 197 V1.5.1, March 2007,
URL: http://www.etsi.org/

[SSL30] "SSL 3.0 Specification", Netscape Communications, November 1996,
URL: http://wp.netscape.com/eng/ssl3/draft302.txt

[XBS DRM extensions-v1.0] "OMA DRM v2.0 Extensions for Broadcast Support", Open Mobile Alliance™, OMA-TS-DRM-XBS-V1_0,
URL: http://www.openmobilealliance.org/

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

[XMLNames] "Namespaces in XML 1.0 (Second Edition)", T. Bray, D. Hollander, A. Layman and Richard Tobin, W3C Recommendation, 16 August 2006,
URL: http://www.w3.org/TR/REC-xml-names/

2.2 Informative References

[ISO-14496-2] "Information Technology - Coding of audio-visual objects, Part 2: Visual", ISO/IEC 14496-2, 3rd edition, 2004,URL: http://www.etsi.org/

[ISO-14496-3] "Information Technology - Coding of audio-visual objects, Part 3: Audio", ISO/IEC 14496-3, 3rd edition, 2005,
URL: http://www.iso.org

[ISO-14496-10] "Information Technology - Coding of audio-visual objects, Part 10: Advanced Video Coding", ISO/IEC 14496-10, 3rd edition, 2005,
URL: http://www.iso.org

[ISO-14496-12] "Information technology – Coding of audio-visual objects, Part 12: ISO base media file format", ISO/IEC 14496-12,
URL: http://www.iso.org

[OMA SUPL] "OMA Secure User Plane Location V 1.0", Open Mobile Alliance™, OMA-ERP-SUPL-V1_0-20070122-C,
URL: http://www.openmobilealliance.org/

[RFC5159] "Session Description Protocol (SDP) Attributes for Open Mobile Alliance (OMA) Broadcast (BCAST) Service and Content Protection", L. Dondeti, Ed., Anja Jerichow, March 2008,
URL: http://www.ietf.org/rfc/rfc5159.txt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

(U)SIM A SIM or a USIM application residing in the memory of the UICC.

Application IDentifier (AID) Data element that identifies an application in a Smartcard.

BCAST Permissions Issuer The BCAST Permissions Issuer (BCAST PI, or simply PI) is a logical entity that issues to BCAST terminals key material or consumption rules, the latter in the form of permissions and constraints. These rules in turn allow and control a user’s consumption of live or stored content pertaining to broadcast services. For the DRM Profile, consumption rules are defined by Generalized Rights Objects (GRO) as specified in [XBS DRM extensions-v1.0], and the BCAST Permissions Issuer is synonymous with the “Rights Issuer” in OMA DRM. For the Smartcard Profile, such rules are defined by the contents of the EXT BCAST payload included in the LTKM, and may indicate the number of times the SEK/PEK can be used to replay content.

BCAST Smartcard Smartcard that supports one of the following sets of applications:

- 3GPP USIM with support for BCAST processing, as indicated by the presence of Service n°75 (BCAST) in the USIM Service Table (EF_UST defined in [3GPP TS 31.102 v7]);
- 3GPP USIM with support for BCAST processing and BCAST BSIM, where support for BSIM is indicated by the presence of the BSIM AID in EF_DIR, where EF_DIR is defined in [ETSI TS 102 221];
- 3GPP2 (R-) UIM with support for BCAST processing;
- 3GPP2 CSIM with support for BCAST processing;
- 3GPP2 CSIM with support for BCAST processing and BCAST BSIM, where support for BSIM is indicated by the presence of the BSIM AID in EF_DIR, and where EF_DIR is defined in [ETSI TS 102 221].

Broadcast Device A device that does not support an interactive communication channel and cannot communicate with other entities except using the broadcast channel.

Note that a Broadcast Device can still have an implicit return channel: it may present information, triggers and dialogs to the user who may “implement” the interactive channel in various ways (e.g. telephone, web portal, service desk).

Broadcast Rights Object This is a Rights Object used by DRM Profile of the Service and Content Protection for rights delivered over the broadcast channel. Encoding of the BCRO is specified in [XBS DRM extensions-v1.0].

BSIM BCAST application residing on the UICC.

Content Encryption The cipher algorithm is applied on the data before packetization for transport or encapsulations occur.

Content Protection This involves the protection of content (files or streams) during the complete lifetime of the content i.e. it is NOT an access control mechanism as it involves post-acquisition rules. Content protection is enabled for encrypted content through the use of appropriate rules or rights, e.g. using DRM Profile or Smartcard Profile based solution for file and stream distributed content. Content remains protected in the Terminal.

Usage rules are enforced at "consumption time" (based on DRM or Smartcard Profile). In addition to subscription and pay-per-view, typically associated with Service Protection, Content Protection enables more fine-grained usage rules, such as for displaying, saving in unencrypted form, printing, processing, re-distributing, etc.

CSIM Acronym for ‘cdma2000 Subscriber Identify Module’ corresponding to an application defined in [3GPP2 C.S0065-0] residing on the UICC to register services provided by 3GPP2 mobile networks with the appropriate security.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

DRM Profile	The DRM Profile uses the Service and Content Protection solution for BCAST receivers in which the long term key management and registration of devices is based on OMA DRM and the broadcast extensions [XBS DRM extensions-v1.0]. The Service & Content Protection solution for the DRM Profile is described in Section 5.
Generalized Rights Object	This term is used in this document as a more generic term whenever an RO or a BCRO is meant.
Interactive Device	A device that supports an interactive communication channel and can communicate with other entities without using the broadcast channel for the communication. For example, an Interactive Device can execute interactive protocols, like the DRM 2.0 ROAP protocol or HTTP towards a Broadcast Permissions Issuer.
Long-Term Key Message	Collection of keys and possibly, depending on the profile, other information like permissions or other attributes that are linked to items of content or services.
MBMS only Smartcard	Smartcard that does not support any of the combination of applications required to be classified as a BCAST Smartcard but does support the processing defined for MBMS [3GPP TS 33.246 v7], as indicated by the presence of Service n°69 (MBMS Security) in the USIM Service Table (EFUST defined in [3GPP TS 31.102 v7]).
MIKEY (Multimedia Internet KEYing)	IETF defined key management protocol to support multimedia security protocols, as defined in [RFC3830]
Program	A logical portion of a service or content with a distinct start and end time. In the case the program is not free-to-air, it can be offered individually for purchase, such as “Pay-Per-View”, or as part of a parent service (e.g. subscription service).
Rights Object	This is the Rights Object used by the DRM Profile of the Service and Content Protection for rights delivered over the interactive channel. Encoding of the RO is specified in [DRMDRM-v2.0], and some extensions are specified in [XBS DRM extensions-v1.0].
R-UIM	Acronym for ‘Removable User Identity Module’ corresponding to a non-UICC platform based standalone module as defined in [3GPP2 C.S0023-C] to register services provided by 3GPP2 mobile networks with the appropriate security.
Secure Storage Entity	The secure storage entity protects sensitive data such as cryptographic keys introduced by either the DRM Profile or the Smartcard Profile. Only an authorized agent is allowed to access the sensitive data. To ensure that the sensitive data is not manipulated fraudulently, it is integrity protected. The sensitive data are also cryptographically protected to guarantee its confidentiality. The secure storage entity can be implemented on either the Smartcard or the terminal.
Service Protection	This involves protection of content (files or streams) during its delivery i.e. it is an access control mechanism only. In the absence of any subsequent Content Protection, content is freely available (thus unencrypted) once it is securely delivered. For the benefit of allowing Content Protection to be provided for the same service, Service Protection is limited to immediate consumption / rendering only.
Short-Term Key Message	Message delivered alongside a protected service, carrying key material to decrypt and optionally authenticate the service, and access rights to delivered content.
SIM	A Subscriber Identity Module is a standalone module defined in [3GPP TS 51.011 v4] to register services provided by 2G mobile networks with the appropriate security.
Smartcard	A non-UICC secure function platform which may contain the SIM or R-UIM module, or a UICC-based secure function platform which may contain one or more of the following applications: a 3GPP USIM, or 3GPP2 CSIM. Note that the set of applications/modules residing on the Smartcard are typically governed by the affiliation of the Smartcard to 3GPP or 3GPP2 specifications, as indicated by the definition below for “Smartcard Profile”.
Smartcard Profile	Alias for a set of Smartcard-based technologies and mechanisms which provide key establishment and key management, as well as permission and token handling for the Service and Content Protection solution for BCAST Terminals. In particular, subscriber key establishment and both short and long term key management are based on GBA mechanisms and a Smartcard with (U)SIM as defined by 3GPP, or based on a pre-provisioned shared secret key and a Smartcard with R-UIM/CSIM or a UIM as defined by

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	3GPP2. The Smartcard Profile is described in Section 6.
Transport Encryption	The cipher algorithm is applied on the data that have been packetized for transport on a network.
UICC	A Universal Integrated Circuit Card is a physically removable secured device as defined in [3GPP TS 31.101 v7] for communication purposes not restricted to mobile convenience only. It is a platform to all the resident applications (e.g. USIM, BSIM, or CSIM).
UIM	Acronym for 'User Identity Module', representing a standard device or functionality which provides secure procedures in support of registration, authentication, and privacy functions in mobile telecommunications. In the context of BCAST, the UIM refers specifically to the non-removable version of this standard device or functionality which is employed by (some) mobile terminals which operate according to 3GPP2 specifications. In addition, Smartcard Profile based service and content protection functionality can be provided on UIM-equipped BCAST Terminals.
USIM	A Universal Subscriber Identity Module is an application defined in [3GPP TS 31.102 v7] residing in the memory of the UICC to register services provided by 3GPP mobile networks with the appropriate security.

3.3 Abbreviations

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
ADF	Application Dedicated File
AES	Advanced Encryption Standard
AID	Application Identifier
AU	Access Unit
AVC	Advanced Video Codec
BCD	Binary Coded Decimal
BCI	Binary Content ID
BCMCS	Broadcast and Multicast Services
BCRO	Broadcast Rights Object
BDS	Broadcast Distribution System
BDS-SD	BDS Service Distribution
BM-SC	Broadcast-Multicast Service Centre
BSDA	BCAST Service Distribution and Adaptation
BSF	Bootstrapping Server Functionality
bslbf	Bit String, Left Bit First
BSM	BCAST Subscription Management
CSIM	cdma2000 subscriber Identify Module
DCF	DRM Content Format
DF	Dedicated File
DK	Device Key
EF	Elementary File
FCP	File Control Parameters
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

GBA_U	GBA with UICC-based enhancements
GMK	Group Management Key
GRO	Generalized Rights Object
H-AAA	Home Authentication, Authorization and Accounting
HMAC	Hashed Message Authentication Code
ICC	Integrated Circuit(s) Card
IIN	Issuer Identifier Number
IPsec	IP Security
ISMA	Internet Streaming Media Alliance
KV	Key Validity
LI	Language Indication
LSB	Least Significant Bit
LTKM	Long Term Key Message
MAC	Message Authentication Code
MBMS	Multimedia Broadcast Multicast Service
ME	Mobile Equipment
MF	Master File
MII	Major Industry Identifier
MIKEY	Multimedia Internet KEYing
MJD	Modified Julian Date
mjdutc	Modified Julian Date Coordinated Universal Time
MK	Master Key
MKI	Master Key Index
MRK	MBMS Request Key
MS	Master Salt
MSK	MBMS Service Key
MTK	MBMS Transport Key
MTU	Maximum Transmission Unit
MUK	MBMS User Key
NAF	Network Application Function
NALu	Network Abstraction Layer Unit
OMA	Open Mobile Alliance
OMNA	Open Mobile Naming Authority
PAK	Program Authentication Key
PAS	Program Authentication Seed
PDCF	Packetized DCF
PEAK	Program Encryption / Authentication Key
PEK	Program Encryption Key
PIX	Proprietary application Identifier eXtension
PKI	Public Key Infrastructure

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

PL	Preferred Languages
PPT	Pay Per Time
PPV	Pay Per View
PRF	Pseudo Random Function
REK	Rights Encryption Key
RFC	Request For Comments
RIAK	Right Issuer Authentication Key
RID	Registered application provider IDentifier
RK	Registration Key
RO	Rights Object
ROAP	Rights Object Acquisition Protocol
RTP	Real-time Transport Protocol
R-UIM	Removable User Identity Module
SA	Security Association
SAC	Secure Authenticated Channel
SAK	Service Authentication Key
SAS	Service Authentication Seed
SCK	SmartCard Key
SDP	Session Description Protocol
SEAK	Service Encryption / Authentication Key
SEK	Service Encryption Key
SG	Service Guide
SHA-1	Secure Hash Algorithm
SIM	Subscriber Identity Module
SK	Short-term Key (appears in 3GPP2 BCMCS specifications)
SKI	Symmetric Key Infrastructure
SM	Subscription Manager
SMK	Subscriber Management Key
SPE	Security Policy Extension
SPI	Security Parameters Index
SRK	Subscriber Request Key
SRTP	Secure Real-time Transport Protocol
STKM	Short Term Key Message
TAK	Traffic Authentication Key
TAS	Traffic Authentication Seed
TBK	Terminal Binding Key
TEK	Traffic Encryption Key
TK	Temporary Key
TKM	Traffic Key Message
TOI	Transport Object Identifier

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

TS	TimeStamp
UDN	Unique Device Number
UE	User Equipment
UICC	Universal Integrated Circuit(s) Card
UIM	User Interface Module
uimsbf	Unsigned Integer Most Significant Bit First
URI	Uniform Resource Indicator
USIM	Universal Subscriber Identity Module
UTC	Universal Time, Co-ordinated
XBS	Extensions for Broadcast Support

3.4 Symbols

E{K}(M)	Encryption of message ‘M’ using key ‘K’
D{K}(M)	Decryption of message ‘M’ using key ‘K’
A ‖ B	Concatenation of A and B
LSBm(X)	The bit string consisting of the m least significant bits of the bit string X.
MSBm(X)	The bit string consisting of the m most significant bits of the bit string X.
HEX(X)	The hexadecimal presentation of the parameter containing hexadecimal characters 0-9 and a-f (in lowercase) with possible preceding zeros. As an example, for a 16 bit value 2748, HEX() returns "0abc". Note that two characters are always generated for each byte.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

4. Introduction

An architectural overview of Service Protection and Content Protection appears in [BCAST10-Architecture].

This specification describes the Service Protection and Content Protection system for OMA BCAST services. Not only does such system enable the restriction of access to services to authorised users during broadcast delivery, it also controls the consumption of the associated content throughout its lifetime.

OMA BCAST has requirements to provide both protection for broadcast content and services. However, the protection of broadcast content and services are required for different purposes:

- Content Protection: This involves the protection of content (files or streams) during the complete lifetime of the content. Content providers require securing the content not only at the present time of broadcasting, but also in the future. Some content providers might want to determine post-acquisition usage rules or so called digital rights. These can be obtained on an individual basis by the end user. Other content providers have content to offer, for which they do not require technical restrictions but limit it to fair use cases and rely on copyright acts.

- Service Protection: This involves protection of content (files or streams) during its delivery. Service providers require a secure access mechanism. They are only concerned with managing access to the content at the time of broadcasting. This is independent of the offered content and independent of the presence of digital rights for certain types of content. Only an access/no-access mechanism is required to distinguish between subscribed and not-subscribed users.

4.1 Version 1.0

In BCAST ERP 1.0, Service Protection and Content Protection may be handled by two different security mechanisms. The complete protection system consists of Service or Content Protection. The possible key management systems and encryption are as defined in this document. There are two possibilities:

◊ DRM Profile: OMA DRM based solution for managing the keys. This is described in Section 5. The DRM Profile is derived from, and almost identical to, DVB-H 18Crypt.

  - For file download delivered over the broadcast channel, the Service or Content Protection is as per OMA DRM 2.0 specifications or using DCF or IPsec as specified in this specification. In this case normal usage rules are as defined in the OMA DRM 2.0 Rights Object.

  - For real-time broadcast streaming using RTP, Service or Content Protection is applied using the relevant broadcast extensions and appropriate encryption (IPsec, SRTP, ISMACryp). Post delivery usage rules associated with the service and / or specific program content are delivered in Rights Objects and STKMs. These rules can apply to content recorded in an appropriate file format, as defined in this specification for broadcast streams, which may be recorded either encrypted or unencrypted.

◊ Smartcard Profile: Smartcard based solutions for managing the keys. These are described in Section 6.

  - For file download delivered over the broadcast channel, the Service or Content Protection uses DCF or IPsec as specified in this specification. In this case normal usage rules are as defined in the LTKMs and STKMs.

  - For real-time broadcast streaming using RTP, Service or Content Protection is applied using the appropriate encryption (IPsec, SRTP or ISMACryp). Post delivery usage rules associated with the service or specific program content MAY be delivered in LTKMs and STKMs. These rules can apply to content recorded in an appropriate file format, as defined in this specification for broadcast streams, which may be recorded either encrypted or unencrypted.

In addition to the key management, the encryption solution can operate on one of the following ways:

◊ The Internet Protocol (IP) layer based on the IPsec security standard, in which case it is transparent to IP based receiver applications like video players.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

◊ The transport layer, based on the SRTP security standard.

◊ The content level, i.e. by encrypting Access Units before packetization occurs (ISMACryp).

For Service or Content Protection, both IPsec and SRTP allow the solution to be completely independent of the content format by protecting content at the transport level. On the other hand, content encryption is provided at the content level by using ISMACryp, allowing the solution to be completely independent of formats used on the transport level. Service or Content Protection may include message authentication/integrity protection and detecting replay attacks.

A service provider may use content level encryption instead of transport level encryption for streaming to provide Service Protection and support Content Protection for the same encrypted stream. In this case, the service offered depends on the nature of implicit or explicit rights delivered (access-only right or post-acquisition rights). To allow this scenario, recording of content-encrypted content shall be allowed in encrypted format only if content encryption is used for the purpose of providing optional Content Protection.

An OMA BCAST Terminal MAY implement Service Protection and MAY implement Content Protection, as shown in Table 1.

Table 1: Service Protection and Content Protection in OMA BCAST Terminals

	BCAST Terminal
Service Protection	OPTIONAL
Content Protection	OPTIONAL

For BCAST Terminals with Service Protection:

Table 2 summarises the possible scenarios. At least one Profile SHALL be implemented. Both Profiles MAY be implemented.

- A BCAST Terminal with a cellular radio interface and a Smartcard SHALL implement the Smartcard Profile. The DRM Profile is OPTIONAL. Hence terminals MAY implement both profiles.
- A BCAST Terminal with a cellular radio interface and no Smartcard SHALL implement the DRM Profile (the Smartcard Profile is not applicable).
- A BCAST Terminal that does not have a cellular radio interface SHALL implement the DRM Profile (the Smartcard Profile is not applicable based on current technology).

Table 2: OMA BCAST Terminal Profile Support for Service Protection

	DRM Profile	Smartcard Profile
Terminal without cellular radio interface or without Smartcard	MANDATORY	N/A
Terminal with cellular radio interface and Smartcard	OPTIONAL	MANDATORY

For BCAST Terminals with Content Protection:

Table 3 summarises the possible scenarios. At least one profile SHALL be implemented. Both profiles MAY be implemented.

- A BCAST terminal with a cellular radio interface and a Smartcard MAY implement the Smartcard Profile or MAY implement the DRM Profile. The Terminal SHALL implement at least one profile. Hence terminals MAY implement both profiles.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- A BCAST terminal with a cellular radio interface and no Smartcard SHALL implement the DRM Profile (the Smartcard Profile is not applicable).

- A BCAST terminal that does not have a cellular radio interface SHALL implement the DRM Profile (the Smartcard Profile is not applicable).

Note that 'Terminal Implementation' of a content protection profile means that the Terminal is capable of it, but does not necessarily mandate its use. Decision to use (or not to use) an implemented content protection profile is made at the time of service deployment.

Table 3: OMA BCAST Terminal Profile Support for Content Protection

	DRM Profile	Smartcard Profile
Terminal without cellular radio interface or without Smartcard	MANDATORY	N/A
Terminal with cellular radio interface and Smartcard	OPTIONAL	OPTIONAL

Adaptations of the described service and content protection mechanisms to underlying Broadcast Distribution Systems (BDSs) are possible and are described in Section 15 and in the respective adaptation specifications, e.g. [BCAST10-MBMS-Adaptation], [BCAST10-BCMCS-Adaptation], and [BCAST10-DVBH-IPDC-Adaptation].

4.1.1 Selected Technologies

These are the main standards on which the solution is based:

- Advanced Encryption Standard (AES, see [FIPS197]) in the Cipher Block Chaining mode with 128 bit keys, for actual content encryption. Furthermore, OMA DRM uses AES-WRAP in its Rights Objects and optionally AES CBC-MAC. AES-ECB is also used by the terminal binding scheme to protect the TEKs.

- Secure Internet Protocol (IPsec, see [RFC2406]) using the Encapsulating Security Payload (ESP) protocol, for implementing transport encryption and decryption as a function of the IP stack. Only transport mode is used.

- Secure Real Time Protocol (SRTP, see [RFC3711]) for implementing service protection at the transport layer. SRTP uses AES-CTR (counter mode).

- Content encryption as specified in [ISMACRYP11]. Appropriate extensions are provided in this specification for codec agnostic RTP transport of ISMACryp protected streams.

- Traffic Encryption Key (TEK) delivery protocol and management is specified in this document.

- Terminal Binding Key (TBK) delivery protocol and use is specified in this document.

- Open Mobile Alliance (OMA) Digital Rights Management version 2.0 [DRM Enabler-v2.0] for service and content protection, managing rights and the associated service and program encryption keys, and the cryptographic protection of those keys themselves. This specification makes some adaptations to OMA DRMv2 for OMA BCAST, mainly for devices without interactive channel.

- DRM Rights Object delivery and device registration over the OMA BCAST channel, without using an interaction channel, are also newly specified. They are described in this document and [XBS DRM extensions-v1.0]. (Devices with access to the interactive channel do not need to implement those extensions for broadcast-only devices, as they typically do registration and Rights Object acquisition over the interactive channel only.)

- GBA [3GPP TS 33.220 v6] for (U)SIM and pre-provisioning [3GPP2 S.S0083-A] for (R-)UIM/CSIM for establishing a shared secret. Their applications for service protection are as specified in this document.

The reasons for choosing these particular technologies as the basis of the solution include the following:

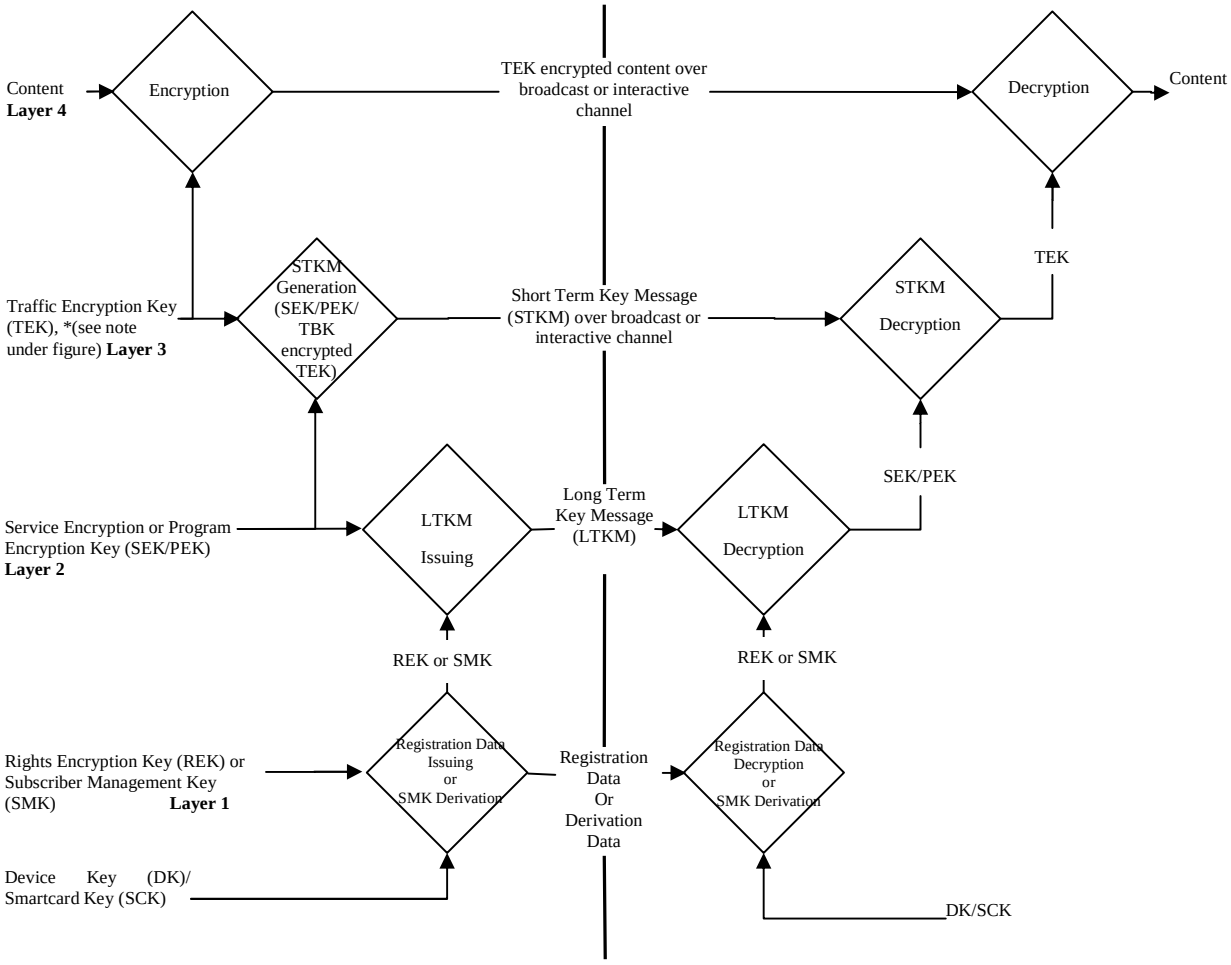
© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- AES is an open standard symmetric encryption algorithm which is widely used in various standards including OMA DRM v2.0.
- IPsec ESP is the standard way of keeping service decryption at receiving devices within the IP stack, invisible to the receiving applications, which thus remain independent of service protection and the carriers of the IP packet streams (of which IPDC may be only one). IPsec ESP has been used in various existing applications.
- SRTP is a standard way of performing service decryption at receiving devices within the transport layer. SRTP has been used to protect all common forms of streaming content.
- ISMACryp allows encrypted content to be streamed. This means encrypted content stored in a file can be streamed at the server side and directly recorded in a file at the terminal side, without the need for decryption and re-encryption. Content encryption may be used to protect content during its complete lifetime, not only during transport.
- TEK management framework and protocol are specified in this document. Guidelines are provided on TEK management based on two different assumptions:
  - First, where the terminal is untrusted, the solution is made robust by using a key delivery protocol and management scheme for frequently changing TEKs to make it expensive for a misbehaving terminal to share TEKs with unauthorized devices.
  - The alternative is to trust the terminals to behave according to certain rules. In the context of the Smartcard Profile, the terminal is expected to delete TEKs after use, cache TEKs only for authorized use, for example to rewind and play content, and never transmit TEKs to external entities.
- GBA is a general architecture that allows to share securely a secret between a server and a client; it has already been used in 3GPP MBMS. Currently, 3GPP2 uses pre-provisioning to establish shared secret between (R-)UIM/CSIM and home network.

4.1.2 Overview of Operations for Streaming of Content

Streaming can be done with content coming either from a live source or from a file. Protection of streamed content can be done using service protection or content protection. Both protection mechanisms use the 4-layer model of Figure 1.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.



*Note: For the Smartcard profile, Traffic Encryption Key (TEK) may be encrypted by Terminal Binding Key (TBK) before encryption by the SEK/PEK for the STKM Generation.

Figure 1 – Protection via the 4-Layer Model

4.1.2.1 The 4-Layer Model

As illustrated in Figure 1, the solution is based on a 4-layer model key management architecture, with an optional optimisation to provide both secure subscription and pay-per-view purchase options for a single service. Traffic Encryption Keys (TEKs) are applied to the actual content (Layer 4) following different mechanisms depending on the actual encryption method used.

The TEKs are themselves sent encrypted by a Service or Program Encryption Key (SEK/PEK). These messages carrying TEKs are called Short Term Key messages (STKMs). STKMs are distributed over the same channels used by the corresponding content. When using the Smartcard Profile, the TEKs MAY optionally be encrypted with a Terminal Binding Key (TBK) before being encrypted by the SEK/PEK, to provide for terminal binding.

Separate SEK and PEK keys can have different lifetimes and can be used to provide, for a single service, different granularities of purchase periods to different customers. This allows for the efficient implementation of both subscription and pay-per-view business models for the same service. Pay-per-view customers are provided with a PEK that is only valid for a single program while subscribers would be provided with a SEK, valid for reception of the service for some longer period. For the DRM Profile, within the STKM, the TEK is encrypted with a PEK, and the PEK is also carried in the STKM, encrypted with the SEK. Thus, pay-per-view customers can directly decrypt the TEK, while subscription-based customers can decrypt the PEK by using the SEK, which can then be used to decrypt the TEK.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

For DRM profile, STKMs contain extension of content IDs for the program or service. Devices use this ID to identify which Long Term Key Message (LTKM) contains the necessary keys to use for decryption of Short Term Key messages. For the Smartcard Profile, STKMs contain the SEK or PEK ID directly to identify the SEK/PEK used to protect the STKM; LTKMs also contain a flag indicating whether or not a TBK is used. The LTKMs are delivered over the broadcast or interactive channel in case of the DRM Profile and over the Interactive channel in case of the Smartcard Profile.

Where the service and program functionality differentiation is not required or supported, the TEK can be directly encrypted with the SEK, and the SEK-encrypted PEK can be omitted from the STKM.

For the Smartcard Profile key management, please refer to Section 6.

Depending on the key management profile (DRM Profile or Smartcard Profile), either the Rights Encryption Key (REK) or the Subscriber Management Key (SMK) is used to protect the LTKM delivery. The key material (REK or SMK) and meta-data are delivered as a result of the registration phase (DRM Profile) / Subscriber Key Establishment phase (Smartcard Profile).

Cryptographic keys introduced by the 4-layer model SHALL be stored securely within a secure storage entity to guarantee the access control, the confidentiality and the integrity of the sensitive data and SHALL never be exposed outside of the secure storage.

Only the TEK among cryptographic keys MAY be allowed to be exposed outside the secure storage upon request from authorized applications.

4.1.2.2 Streaming Using Service Protection

For service protection, encryption is carried out using the AES algorithm with 128 bit symmetric traffic keys. TEKs are retrieved from the secure storage entity and are applied:

- as part of standard IPsec security associations (SAs), or
- as an SRTP master key, from which the session key is derived as per SRTP specification, or
- directly to encrypt the content, presented as Access Units (AUs), before packetization for transport occurs (ISMACryp).

Depending on the chosen encryption, the keys are used to perform decryption automatically before passing the packets to the receiving application.

The SEKs or PEKs are transmitted to each receiving device within Long Term Key messages (LTKMs) and SHALL be stored within the secure storage entity, and SHALL never be exposed outside of the secure storage. If OMA DRM 2.0 extensions [XBS DRM extensions-v1.0] are used, the LTKMs are referred to as Generalized Rights Objects. Such transmission of LTKMs can be done in two different ways, depending on whether the receiving device can make use of an interactivity channel:

- Via broadcast over OMA BCAST broadcast channel, or
- Via an interactivity channel.

As already mentioned, there are two key management systems:

- Using OMA DRM 2.0 Extensions [XBS DRM extensions-v1.0]. When delivering LTKMs over the OMA BCAST broadcast channel in the form of Rights Objects (ROs), bandwidth is a major constraint. This specification addresses this problem in two complimentary ways. Firstly, a new binary form of an RO, called a Broadcast Rights Object (BCRO), is defined. Secondly, a method is described for securely delivering BCROs to groups of devices at the same time. Valuable portions of ROs are protected by group or unit keys, and when necessary, broadcast encryption can be used to allow messages to be decrypted only by arbitrary sets of devices within a larger group. When delivering ROs to devices that have access to an interactive channel, implementation complexity is a major constraint. Thus, such devices, which are expected to support OMA DRM 2.0 for interactive content services, use standard OMA DRM 2.0 mechanisms as much as possible, e.g. they acquire ROs for broadcast content via the interactive channel using the DRM 2.0 ROAP protocol, as they would do for non-broadcast content as well. This specification defines also an efficient and user-friendly process for the registration of devices which do not have an interactivity channel. Rights Encryption Keys (REKs) are also

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

delivered to receive-only devices during a device registration process protected using the public key of the individual devices. When an interactivity channel is available, the registration process is according to standard OMA DRM v2.0.

- Using Smartcard Profile. An overview of operation is given in Section 6.

4.1.2.3 Streaming Using Content Protection

For content protection, encryption is carried out according to AES using 128 bit symmetric traffic keys. While service protection provides protection of the stream only at the time of service reception, content protection provides protection of the content even after the service reception, i.e. content remains stored protected in the Terminal. On one hand, content protection may be achieved by using TEKs to encrypt the content before packetization for transport or when encapsulation in a file occurs (ISMACryp). On the other hand, content protection may also be provided using transport encryption (SRTP or IPsec) and appropriate measures in the receiving device to protect content inside the device.

4.1.3 Overview of Operation for Download of Content

Protection of files is as defined by OMA DRM 2.0 specifications [DRM Enabler-v2.0] for the DRM Profile. For the Smartcard Profile, a modified version of the DCF file format is defined in this specification.

The mechanisms supported for the protected download of content using file delivery are dependent on the profile used.

For the DRM Profile, the protection of files is achieved by at least one of the following:

- as defined by the OMA DRM 2.0 specifications [DRM Enabler-v2.0], or
- by using an additional box in the extended headers field of the DCF file format and encryption by TEKs as defined in [XBS DRM extensions-v1.0], or
- by using IPsec.

For the Smartcard Profile, the protection of files is achieved by at least one of the following:

- by using an additional box in the extended headers field of the DCF file format and encryption by TEKs as defined in [XBS DRM extensions-v1.0], or
- by using IPsec.

Note that combining the above methods allows compatibility with OMA DRMv2 DCF file format and operation with both DRM Profile and Smartcard Profile.

4.1.3.1 Content Download Using Service Protection

Content download by using Service Protection is specified in Section 5.6.2.1 for the DRM Profile and Section 6.8.2.1 for the Smartcard Profile.

4.1.3.2 Content Download Using Content Protection

Content download by using Content Protection is specified in Section 5.6.2.2 for the DRM Profile and Section 6.8.2.2 for the Smartcard Profile.

4.1.4 Key Management

The 4-layer model described in [BCAST10-Architecture] allows different key management systems to be used. (See also Section 4.1.2.1.) This section outlines the key management profiles defined for BCAST 1.0, namely the Smartcard Profile and the DRM Profile.

The Smartcard Profile defines a key management system based on the symmetric key model used by either the 3GPP MBMS [3GPP TS 33.246 v7] security model based on the (U)SIM or 3GPP2 BCMCS [3GPP2 S.S0083] security model based on (R-)UIM/CSIM. An overview of the Smartcard Profile is provided in Section 4.1.4.2, while a full description is provided in Section 6.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The DRM Profile defines a key management system based on the Public Key Infrastructure (PKI) offered by OMA DRM v2.0 [DRMDRM-v2.0]. An overview of the DRM Profile is provided in Section 4.1.4.1, while a full description is provided in Section 5.

In order to ensure maximum interoperability, OMA BCAST defines a common layer for traffic encryption (Layer 4) and allows the other layers of key management to be implemented using either the DRM Profile or the Smartcard Profile.

Adaptation of the 4-layer model used in OMA BCAST to underlying BDSes is specified for 3GPP MBMS, 3GPP2 BCMCS and IPDC over DVB-H. This adaptation allows the existing functionalities provided by the underlying BDS to be re-used. Information on the appropriate adaptation is provided in Section 15.

4.1.4.1 DRM Profile Overview

The DRM Profile is based on public key based mechanisms. It uses the OMA DRMv2.0 Enabler [DRMDRM-v2.0] to support key management. For non-interactive devices, broadcast extensions for OMA DRMv2.0 as specified in [XBS DRM extensions-v1.0] are used.

The key management for the DRM Profile is based on the BCAST 4-layer model key hierarchy introduced in Section 4.1.2.1.

Layer 1 is for registration purpose and uses the public/private key pair stored in the BCAST terminal. The public key is used to secure the delivery of the Rights Encryption Key (REK), and, with the corresponding private key, the Generalized Rights Objects (GROs) can be processed. The REK may be delivered over an interactive or broadcast channel. In case of delivery via the broadcast channel the REK may refer to several keys, which are delivered to the BCAST terminal using the registration process specified in [XBS DRM extension-v1.0]. In case of an interactive channel, the ROAP registration procedure [DRMDRM-v2.0] is applied. Note that the actual provisioning of the public/private key pair is out of scope for this specification.

In Layer 2, the Long Term Key Messages (LTKMs) are delivered. The LTKM is a Generalized Rights Object, which may take two alternative formats. In case LTKM is delivered over a broadcast channel, the format used is of a Broadcast Rights Object (BCRO) as specified in [XBS DRM extensions-v1.0]. If an interactive channel is used, the GRO is a Rights Object (RO) as specified for OMA DRMv2.0 [DRMDRM-v2.0]. The LTKM transports the Service or Program Encryption Key (SEK/PEK), as well as permissions and attributes. SEK/PEK is encrypted using the keys delivered or broadcasted during the Layer 1 registration procedure.

Layer 3 securely transports short term keys, i.e. the Traffic Encryption Keys (TEK), in the Short Term Key Message (STKM) that are broadcasted over the same network as the media streams. Furthermore, data can be protected in case of streaming and file delivery respectively for both service and content protection. In the case where the TEK is encrypted with a PEK, the STKM may also carry the SEK-encrypted PEK.

Finally, Layer 4 is responsible for traffic encryption using the TEK for stream or file delivery respectively for both service and content protection.

The DRM Profile key management is described in detail in Section 5.

4.1.4.2 Smartcard Profile Overview

The Smartcard Profile is based on existing security technologies and standards defined for 3GPP or 3GPP2 broadcast/multicast services.

In the context of the BCAST 4-layer model key hierarchy, the Smartcard Profile provides a key management solution that uses a Smartcard and an interactive cellular radio interface. Assuming key provisioning has taken place, this solution enables authentication and Subscriber Key Establishment (Layer 1), LTKM delivery (Layer 2) and STKM delivery (Layer 3), as specified in Section 6. Access to the protected content (Layer 4) is supported irrespective of the type of encryption used (SRTP or ISMACryp or IPsec), as specified in Section 9.

This specification defines two variants of the Smartcard Profile. The two variants are referred to as the (U)SIM Smartcard Profile and the (R-)UIM/CSIM Smartcard Profile respectively. The two variants differ in the way that the Smartcard establishes the Layer 1 key(s) but are otherwise the same (Layers 2, 3 and 4).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The Subscriber Key Establishment layer (Layer 1) makes use of a secret key stored on a Smartcard based identity module. This key is referred to as "SmartCard Key" (SCK) in the Smartcard Profile. The SCK is a pre-provisioned secret key that is shared between the Smartcard and the Smartcard issuer. If the Smartcard issuer is not also the broadcast service provider, then the SCK is unknown to the broadcast service provider.

The SCK is used to create the Layer 1 key, the Subscriber Management Key (SMK), using the Generic Bootstrapping Architecture (GBA) as defined in [3GPP TS 33.220 v6] for the (U)SIM Smartcard Profile, or using the pre-shared key mechanism as defined in [3GPP2 S.S0083] for the (R-)UIM/CSIM Smartcard Profile. The SMK is established between the broadcast service provider and the Smartcard or the terminal depending on the key management implementation. If the smart-card contains the key management system, the SMK SHALL be established between the broadcast service provider and the smart-card (using GBA-U, or respectively pre-shared key mechanism of 3GPP2); otherwise it is established with the terminal (using GBA-ME or 2G GBA).

The SMK SHALL be stored on the Smartcard or the terminal depending on the variant of the Smartcard Profile key management implemented. For the (U)SIM Smartcard Profile, the SMK SHALL be stored on a USIM when using GBA_U, and on a terminal when using GBA_ME or 2G GBA. For the (R-)UIM/CSIM Smartcard Profile, the SMK SHALL be stored on a (R-)UIM/CSIM.

The SMK is a user-specific key used to protect the Long Term Key Messages (LTKM) that are delivered in Layer 2. Depending on the service configuration, within the LTKM a Program Encryption Key (PEK) or a Service Encryption Key (SEK), used respectively for pay per view or subscription customers, is delivered protected by SMK. For the (U)SIM Smartcard Profile, the SEK or PEK SHALL be stored on a USIM when using GBA_U or on a terminal when using GBA_ME or 2G GBA. For the (R-)UIM/CSIM Smartcard Profile the SEK or PEK SHALL be stored on a (R-)UIM/CSIM.

Layer 3 delivers the Short Term Key Message (STKM) within which Traffic Encryption Keys (TEKs) are protected using SEK or PEK, as well as optionally by a Terminal Binding Key (TBK).

Layer 4 is for traffic encryption using the TEK for stream or file delivery respectively for both service and content protection.

Table 4 gives a brief outline of the 4-layer model key hierarchy:

Table 4: Smartcard Profile key hierarchy model

Key layer	Key name	Key hierarchy		Storage location
Key Provisioning	SmartCard Key (SCK)	SCK	Pre-provisioned secret key shared with the Smartcard issuer. Provisioning of this key is out of the scope of this specification.	Smartcard
Layer 1: Subscriber Key Establishment	Subscriber Management Key (SMK)	SMK	For the (U)SIM Smartcard Profile SMK is generated as a result of a successful run of the GBA bootstrapping procedure. For (R-)UIM/CSIM Smartcard Profile, SMK is derived from the SCK. SMK is equivalent to the MBMS User Key (MUK).	Smartcard (for GBA_U or if security is based on registration key RK) or Terminal (for GBA_ME or 2G GBA)
Layer 2: LTKM	Service / Program Encryption Key (SEK/PEK)	SMK[SEK] or SMK[PEK]	Protected by SMK and sent to the Smartcard via the terminal using a point to point channel (in the case of GBA_U). SEK/PEK is equivalent to the MBMS Service Key (MSK).	Smartcard (for GBA_U or if security is based on registration key RK) or Terminal (for GBA_ME or 2G GBA)
Layer 3: STKM	Traffic Encryption Key (TEK)	SEK[TEK] or PEK[TEK]	Protected by SEK or PEK ((U)SIM variant) or derived from SEK or PEK ((R-)UIM/CSIM variant), and sent over the broadcast channel. Optionally also encrypted with TBK. TEK is equivalent to the MBMS Traffic Key (MTK).	Terminal

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Layer 4: traffic encryption		TEK[content]	TEK encrypted content; traffic encryption with SRTP, ISMACryp, or IPsec	

The Smartcard Profile key management is described in detail in Section 6.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5. DRM Profile

5.1 Introduction

OMA BCAST DRM Profile uses OMA DRMv2.0 specified solutions [DRMDRM-v2.0] for the registrations and rights management over the interactive channel and specifies a set of protocols for use in broadcast [XBS DRM extensions-v1.0] and out-of-band channels.

The following sections describe the four layers of the 4-layer model key hierarchy, as well as key provisioning required to access the first layer for DRM Profile. Section 5.2 briefly describes key provisioning. Section 5.3 describes registration. Section 5.4 describes the LTKM structure, while Section 5.5 describes that of the STKM. Section 5.6 and Section 5.6.2 describe how to protect data in case of streaming and file delivery respectively for both service and content protection. Recording aspects are described in Section 5.7, while SG signalling is explained in Section 5.8.

5.2 Key Provisioning

The OMA DRM Profile uses PKI-based mechanism. Access to the registration layer (Layer 1) is implemented using a device key (or public/private key pair) that is stored in the mobile device. How the device key is provisioned is out of scope for this specification.

5.3 Layer 1: Registration

The device must first register with the Rights Issuer to receive protected broadcast service. Registration can be performed either via an interaction or broadcast channel.

In the case that an interaction channel is used, the registration protocol is as defined in OMA DRMv2.0 [DRMDRM-v2.0] and right encryption keys (used to protect Layer 2 RO) are delivered protected with the public key of the device. In this case, the registration procedure is initiated by the device, e.g. on reception of a ROAP Registration Trigger, typically returned whenever an unregistered device executes any of the procedures for interactive service provisioning defined in [BCAST10-Services], or in response to an out-of-band mechanism.

For the devices that do not support an interaction channel, an alternative process for the registration is defined in [XBS DRM extension-v1.0] and a set of keys (used to protect Layer 2 BCRO) are delivered over the broadcast channel protected with the public key of the device.

OMA DRM Profile supports a notion of Broadcast Domains and Interactive Domains to facilitate sharing of content and services among the registered terminals, see [XBS DRM extensions-v1.0].

5.4 Layer 2: Long Term Key Message – LTKM

For the DRM Profile service encryption key (SEK)/program encryption key (PEK) is packaged in a special LTKM format. This special format is called Rights Object (RO) and in addition to the provided keys, it may contain permissions and attributes linked to the protected content. The profile supports the delivery of ROs over interactive and broadcast channel.

Before a device can start receiving LTKMs, it must be subscribed to the service or pay-per-view program that the LTKs protect. For devices that support an interaction channel, this is e.g. done with a “Service Request” or “LTKM Renewal Request” message as defined in [BCAST10-Services]. For devices that support only the broadcast channel, an out-of-band procedure is used (see Section 5.4.4.1.2 in [BCAST10-Architecture]). The information needed to perform the subscription is announced in the Service and various purchase-related fragments of the Service Guide (see Section 5.8). Services and pay-per-view programs that are available for purchase are generically referred to as “purchase items”.

Section 5.4.1 introduces and describes use of ROs. Section 5.4.2 gives OMA DRMv2.0 extensions for BCRO. Section 5.4.3 describes how ROs are used for service protection at Long Term Key Delivery layer

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

5.4.1 Use of ROs and BCROs

Service Encryption Keys (SEK) and Program Encryption Keys (PEK) described in Layer 2 of the Key Hierarchy for Service Protection MAY be transmitted to each terminal within Generalized Rights Objects (GROs). Two formats are available for the purpose. One is the format of an OMA DRM 2.0 Rights Object (RO), as specified in [DLRDRM-v2.0]. The other format is Broadcast Rights Object (BCRO) specified in XBS document [XBS DRM extensions-v1.0], and is used when GRO is delivered over a broadcast channel. In addition to SEKs/PEKs, GROs also contain permissions and other attributes linked to protected service. SEKs would typically be utilized for subscription services. Each SEK protects a single subscription service that can be purchased as a unit. A unit is the minimum granularity of services that a service provider offers to an end user, and a unit, therefore, MAY correspond to a single program channel, to a portion of a channel, or to a collection of program channels that are all purchased as a unit. The SEK is an intermediate key, i.e. it does not directly encrypt the content but instead encrypts a Traffic Encryption Key (TEK) or PEK. The SEKs themselves are encrypted by keys transmitted in Layer 1 of the Key Hierarchy. PEKs carried in GROs are encrypted by keys transmitted in Layer 1, and used to decrypt the TEK. In the context of BCAST Enabler, these GROs are called Long Term Key Messages (LTKMs).

A terminal periodically receives a set of SEKs/PEKs that MUST be encrypted and authenticated. Depending on the capabilities of underlying transport networks, multiple SEKs/PEKs MAY be combined into one LTKM directed to a terminal. There MAY also be multiple such messages that relay different sets of SEKs/PEKs to the same terminal.

SEKs SHOULD be periodically updated so that when someone drops a subscription, their access to a service will be terminated cryptographically once a SEK changes. For example, SEKs MAY change once per billing period (e.g., on a monthly basis). PEKs, when provided, SHOULD change once per program.

The transmission of LTKM to a terminal can be done over an interaction channel or over a broadcast channel, depending on whether the terminal has access to an interaction channel or not.

If the LTKM is transmitted over the broadcast channel, then the GRO MUST be encoded using a suitable binary encoding or compression. A GRO thus encoded is called a BCRO. The syntax for BCRO is introduced in XBS document [XBS DRM extensions-v1.0].

In addition, if the LTKM is transmitted over the broadcast channel, then digital signatures or MACs over the GRO MAY be verifiable over the BCRO itself without having to decode or de-compress the BCRO.

In addition, if the LTKM is transmitted over the broadcast channel, then all content of the LTKM other than the BCRO MUST be compressed or encoded.

If the LTKM is transmitted over the interaction channel, then the LTKM, including the GRO, digital signatures or MACs, MAY be encoded, compressed, or text-based.

5.4.2 OMA DRM v2.0 Extensions for Broadcast Rights Objects

Extensions to OMA DRM v2.0 for broadcast rights objects including design and format, appear in the OMA DRM v2.0 Extensions for Broadcast Support document [XBS DRM extension-v1.0][*].

An alternative Content Protection solution to that depicted in OMA DRM v2.0 Extensions for Broadcast Support document, or appropriate modifications thereto, is specified in Section 6.

5.4.3 GROs in Long Term Key Delivery Layer for service protection

In case of **subscription**, the Service Encryption and Authentication Key material (SEAK) associated with the service is securely delivered to the authorized terminal in a GRO. Such a GRO is called a **Service RO**. SEAK consists of 128 bits SEK (Service Encryption Key) and 128 bits SAS (Service Authentication Seed). SAS is used as a seed in a generic authentication

[*] (Informative Footnote) Where Generalized Rights Objects (particularly for post-acquisition rights associated with BCAST Stream Delivery of protected content) are stored in secure removable Smartcards, i.e. (U)SIM/(R)UIM/CSIM in 2G/3G mobile terminals, an alternative Content Protection scheme to handle such Broadcast Rights Objects may be applicable as an option. Such an alternative is explored in the OMA DRM working group.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

function to derive SAK (Service Authentication Key). In general, a Service RO will contain key material associated with more than one service (when associated with a service bundle).

In case of **pay-per-view**, the Program Encryption and Authentication Key material (PEAK) associated with a pay-per-view event is securely delivered to the authorized terminal directly within a GRO. Such a GRO is called a **Program RO**. PEAK consists of 128 bits PEK (Program Encryption Key) and 128 bits PAS (Program Authentication Seed). PAS is used as a seed in a generic authentication function to derive PAK (Program Authentication Key).

The ID of GROs that contain SEAKs or a PEAK needs to be structured, to allow for the management of purchase transactions in the device, or more specifically, to create an association between the purchase item in the service guide and the successful completion of the purchase transaction (when the GRO related to the purchase has finally been received in the device). This is valid for both connected and especially for unconnected operation (see [DRMDRM-v2.0] for the definition of "connected" and "unconnected"), where the GRO may be received by the device much later than the purchase transaction is initiated. A connected device has a direct 2-way connection to the Rights Issuer (RI) through interaction channel. On the other hand, the unconnected devices do not have access to the RI through an interaction channel but they are capable of making connection via an intermediary interactive device.

Defining a structured ID for GRO will also allow the device to check later on whether GROs for all subscribed services are available (and have been renewed). The rekeying_period_number is an increasing number by which the ID of the GRO related to the same purchase item can be made unique.

The ID of a GRO linked with subscription (Service RO) or pay-per-view (Program RO), and bound to a device or to a domain, SHALL be constructed respectively as follows:

deviceRoID = "E" || deviceID || "_S" || stringtomakeitunique || "_I" || purchaseItemID || "_" || HEX(rekeying_period_number)

domainRoID = "O" || domainID || "_S" || stringtomakeitunique || "_I" || purchaseItemID || "_" || HEX(rekeying_period_number)

- **deviceID** is the Unqiue Device Number (UDN) as discussed in [XBS DRM extensions-v1.0].
- **stringtomakeitunique** Note that 'deviceRoID' and 'domainRoID' SHALL be globally unique. Note further that because of the specification of 'purchaseItemID' in the OMA BCAST SG, the global uniqueness is already guaranteed and therefore, 'stringtomakeitunique' SHALL be the empty string.
- **purchaseItemID** is the GlobalPurchaseItemID associated with the purchase item and signalled in the Purchase Item Fragment of the SG(see Section 5.8). According to [DRM Enabler-v2.0], Rights Object IDs are of type xml:id, in which only a limited character set is allowed as specified (see grammar of 'NCName' in [XMLNames]. As the purchaseItemID is of type anyURI according to [BCAST10-SG], some characters can be present in purchaseItemID which are not allowed in NCName. Those characters, in particular the colon character (":"), SHALL be replaced by the underscore character ("_") before including the purchaseItemID into deviceRoID or domainRoID
- **rekeying_period_number** is a 7-bit counter that is used to differentiate between different ROs with the same purchase_item_id (defined in Section 7.2 of [XBS DRM extensions-v1.0])

In the case of BCROs, the link with the corresponding subscription (Service RO) or pay-per-view (Program RO) is obtained by using the BCRO fields purchase_item_id and rekeying_period_number ([XBS DRM extensions-v1.0]).

A **Service RO** SHALL contain at least one (<CID>, <SEAK>) pair. The <CID> (Content Identifier) SHALL be constructed as specified in the paragraph defining the Short Term Key Message (see Section 5.5).

The <SEAK> contains SEK and SAS. SEK and SAS are obtained from a GRO as specified in sections C.14.2.1 and C.14.2.2 of [XBS DRM extensions-v1.0]).

A **Program RO** SHALL contain at least one (<CID>, <PEAK>) pair. The <CID> SHALL be constructed as specified in the paragraph defining the traffic key message (see Section 5.5).

The <PEAK> contains PEK and PAS. PEK and PAS are obtained from a GRO as specified in sections C.14.2.1 and C.14.2.2 of [XBS DRM extensions-v1.0]).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

5.5 Layer 3: Short Term Key Message - STKM

This Section describes the format and role of STKM (Short Term Key Message) in the transport of short term traffic keys (TEKs) for DRM Profile at the Short Term Key Delivery layer.

Each STKM SHALL be encapsulated in exactly 1 UDP packet.

In order to keep access times low for devices that start accessing a service, a STKM SHALL be transmitted periodically.

The STKM SHALL be transported over the same network as the media streams that are protected with the traffic keys contained in the STKM. The STKM stream MAY be transported in an own session, e.g. transported in an own IP stream.

If the traffic_protection_protocol equals to TKM_ALGO_DCF, then the STKM MAY be delivered as a separate object inside a FLUTE session, together with the protected traffic, having its own FDT entry.

Table 5: Format of STKM for DRM Profile

Short_Term_Key_Message_Description	Length (in bits)	Type
short_term_key_message() {		
selectors_and_flags {		
protocol_version	**4**	**uimsbf**
protection_after_reception	**2**	**uimsbf**
reserved_for_future_use	**1**	**bslbf**
access_criteria_flag	**1**	**uimsbf**
traffic_protection_protocol	**3**	**uimsbf**
traffic_authentication_flag	**1**	**uimsbf**
next_traffic_key_flag	**1**	**uimsbf**
timestamp_flag	**1**	**uimsbf**
program_flag	**1**	**uimsbf**
service_flag	**1**	**uimsbf**
}		
if (traffic_protection_protocol == TKM_ALGO_IPSEC) {		
security_parameter_index	**32**	**uimsbf**
if (next_traffic_key_flag == TKM_FLAG_TRUE) {		
next_security_parameter_index	**32**	**uimsb**
}		
}		
if (traffic_protection_protocol == TKM_ALGO_SRTP) {		
master_key_index_length	**8**	**uimsbf**
master_key_index	**8*master_key_index_length**	**uimsbf**
reserved_for_future_use	**5**	**bslbf**
next_master_key_index_flag	**1**	**uimsbf**
next_master_salt_flag	**1**	**uimsbf**
master_salt_flag	**1**	**uimsbf**
if (master_salt_flag == TKM_FLAG_TRUE) {		
master_salt	**112**	**bslbf**
}		
if (next_traffic_key_flag == TKM_FLAG_TRUE) {		
if (next_master_key_index_flag == TKM_FLAG_TRUE) {		
next_master_key_index	**8*master_key_index_length**	**uimsbf**
}		
if (next_master_salt_flag == TKM_FLAG_TRUE) {		
next_master_salt	**112**	**bslbf**
}		
}		
}		
if (traffic_protection_protocol == TKM_ALGO_ISMACRYP) {		
key_indicator_length	**8**	**uimbf**
key_indicator	**8*key_indicator_length**	**uimsbf**
if (next_traffic_key_flag == TKM_FLAG_TRUE) {		
key_indicator	**8*key_indicator_length**	**uimsbf**
}		
}		
if (traffic_protection_protocol == TKM_ALGO_DCF) {		
key_identifier_length	**8**	**uimsbf**

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

key_identifier	**8*key_identifier_length**	**bit string**
}		
encrypted_traffic_key_material_length	**8**	**uimsbf**
encrypted_traffic_key_material	**8*encrypted_traffic_key_material_length**	**bslbf**
if (next_traffic_key_flag == TKM_FLAG_TRUE) {		
next_encrypted_traffic_key_material	**8*encrypted_traffic_key_material_length**	**bslbf**
}		
reserved_for_future_use	**4**	**bslbf**
traffic_key_lifetime	**4**	**uimsbf**
if (timestamp_flag == TKM_FLAG_TRUE) {		
Timestamp	**40**	**mjdutc**
}		
if (access_criteria_flag == TKM_FLAG_TRUE) {		
reserved_for_future_use	**8**	**bslbf**
number_of_access_criteria_descriptors	**8**	**uimsbf**
access_criteria_descriptor_loop() {		
access_criteria_descriptor()		
}		
}		
if (program_flag == TKM_FLAG_TRUE) {		
program_selectors_and_flags {		
reserved_for_future_use	**7**	**bslbf**
permissions_flag	**1**	**uimsbf**
}		
if (permissions_flag == TKM_FLAG_TRUE) {		
permissions_category	**8**	**uimsbf**
}		
if (service_flag == TKM_FLAG_TRUE) {		
encrypted_PEK	**128**	**bslbf**
}		
program_CID_extension	**32**	**uimsbf**
program_MAC	**96**	**bslbf**
}		
if (service_flag == TKM_FLAG_TRUE) {		
service_CID_extension	**32**	**uimsbf**
service_MAC	**96**	**bslbf**
}		
}		

Reserved_for_future_use – These bits are reserved for future use, and SHALL be set to zero when not used.

5.5.1 Coding and Semantics of Attributes

Section 7 introduces the coding and semantics of all Attributes common between the DRM Profile and the Smartcard Profile. Any DRM Profile specific attributes are introduced below.

next_traffic_key_flag – indicates whether or not the Short Term Key Message contains the next traffic key material:

TKM_FLAG_FALSE	The Short Term Key Message contains only the current traffic key material.
TKM_FLAG_TRUE	The Short Term Key Message contains both the current and the next traffic key material.

The next traffic key material SHALL be included at least 1 second before it becomes current. This is to enable the devices to process the traffic key material and put the necessary security associations in place before the media packets that are encrypted with the next traffic encryption key start arriving.

The above time SHALL be relative to the moment of transmission of the key stream messages.

If PEK is used to protect the traffic key material, then next traffic key material that protects a program different from the current program SHALL NOT be included.

timestamp_flag – indicates whether or not the STKM contains a timestamp:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

TKM_FLAG_FALSE	The STKM does not contain a timestamp.
TKM_FLAG_TRUE	The STKM contains a timestamp.

program_flag – indicates whether or not the program key layer is present in the Short Term Key Message:

TKM_FLAG_FALSE	The PEK is not present, i.e. the optional program key layer is not used for the service.
TKM_FLAG_TRUE	The PEK is present, i.e. the optional program key layer is used for the service.

<program_flag> and <service_flag> SHALL NOT both be 0. All other combinations are allowed, indicating that either one or both of the key layers are present.

service_flag – indicates whether or not the service block is present in the Short Term Key Message:

TKM_FLAG_FALSE	The SEK is not present, i.e. the optional service key layer is not used for the service.
TKM_FLAG_TRUE	The SEK is present, i.e. the optional service key layer is used for the service.

<program_flag> and <service_flag> SHALL NOT both be 0. All other combinations are allowed, indicating that either one or both of the key layers are present.

security_parameter_index – provides the link to the IPsec ESP header:

Upon reception of a protected IP packet, the terminal SHALL use the security parameter index (SPI) to identify (look up) the correct security association and thereby find the decryption and authentication keys to be used for the received IPsec ESP packet. The SPI value SHALL be in the range 0x00000100 – 0xFFFFFFFF. An incoming ESP packet containing the SPI value specified in this field SHALL use the keymaterial provided in the encrypted traffic key material field as keymaterial for the decryption operation.

next_security_parameter_index – provides the link to the IPsec ESP header:

This field is present in the packet only if next traffic key flag is set to true. This field then contains the IPsec SPI value corresponding to the next_encrypted_traffic_key_material field. The value of the SPI SHALL be in the range 0x00000100 – 0xFFFFFFFF. An incoming ESP packet containing the SPI value specified in this field SHALL use the keymaterial provided in the next encrypted traffic key material field as keymaterial for the decryption operation.

master_key_index_length – provides the length of the master_key_index field

This field gives the length of the master_key_index field in bytes.

master_key_index – provides the link to the SRTP header:

Upon reception of a protected RTP packet, the terminal SHALL use the master key index (MKI) to identify (look up) the correct security association and thereby find the decryption and authentication keys to be used for a received SRTP packet.

This field is a sequence of Octets. The sequence consists of master_key_index_length bytes. The bytes are in the same order that they will be in an SRTP packet and SHALL be in SRTP [RFC3711] network byte-order when extracting the MKI value.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

next_master_key_index_flag – specifies if the master key index (MKI) for the next TEK is explicitly included in the SRTP parameters (as the next_master_key_index field). In the case that the next_master_key_index is not present in the message, the value of current MKI+1 SHALL be assumed. In the case when the next_traffic_key_flag is false there is no information related to the next traffic key included in the message and this parameter does not apply.

next_master_salt_flag – specifies if the next SRTP master salt value corresponding to the next TEK is explicitly included in the SRTP parameters (as the next_master_salt field). In the case that the next_master_salt is not present in the message, the same value as for the current master salt SHALL be assumed. In the case when the next_traffic_key_flag is false there is no information related to the next traffic key included in the message and this parameter does not apply.

master_salt_flag – specifies if the master salt is included in the SRTP parameters. In the case that the master salt is not present in the message, a NULL value consisting of 112 0-bits SHALL be assumed.

master_salt – SRTP master salt that is used along with the master key to derive SRTP session keys as defined by SRTP [RFC3711].

next_master_key_index – provides the link to the SRTP header:

This field is present in the packet only if the next_traffic_key_flag and the next_master_key_index_flag are both set to true. This field then contains the SRTP MKI value corresponding to the next_encrypted_traffic_key_material field. An incoming protected RTP packet containing the MKI value specified in this field SHALL use the key material provided in the next encrypted traffic key material field as key material for the decryption operation.

next_master_salt – next value of the SRTP master salt that is used along with the next master key to derive SRTP session keys as defined by SRTP [RFC3711].

This field is present in the packet only if the next_traffic_key_flag and the next_master_salt_flag are both set to true. This field then contains the SRTP master salt value corresponding to the next_encrypted traffic key material field. An incoming protected RTP packet containing the next MKI value SHALL use the next master salt value provided in this field during the SRTP session key derivation.

key_indicator – value of the KeyIndicator used to identify the TEK transported in the STKM. This is used to identify the particular TEK key needed to decrypt AUs (as indicated in the ISMACrypContextAU field defined in [ISMACRYP11] and [ISMACRYP20]).

key_identifier_length – indicates the length in bytes of the key_identifier. For ISMACryp, key_indicator_length is signaled in SDP. For DRM Profile, the key_indicator_length is also signaled in STKM. Note that the Smartcard Profile STKM does not contain such field for ISMACryp. The key_indicator_length parameter is part of the Session Description Protocol (SDP) and is described in Section 0.

key_identifier – value of the identifier used to identify the TEK transported in the STKM. This is used to identify the particular TEK needed to decrypt DCF encoded files.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

encrypted_traffic_key_material_length – is the length in bytes of the encrypted traffic key material.

The length of the traffic key material depends on the encryption and authentication algorithm, and is obtained by adding the respective key sizes. Encryption MAY require the clear-text key material to be padded.

encrypted_traffic_key_material – is the key material currently used for encryption and optional authentication of the traffic, encrypted using AES-128-CBC, with fixed IV 0, and with 0 padding in the last block, if needed.

If <program_flag> == TKM_FLAG_TRUE, the traffic key material is encrypted with the Program Encryption Key (PEK).

If <program_flag> == TKM_FLAG_FALSE and <service_flag> == TKM_FLAG_TRUE, the traffic key material is encrypted with the Service Encryption Key (SEK).

After decryption (and discarding any padding), the Traffic Encryption Key (TEK) and the Traffic Authentication Key (TAK) are obtained in a way that depends on the protocol used for traffic protection:

IPsec: If no traffic authentication is used, the IPsec encryption key is identical to the decrypted traffic key material (16 bytes).

If traffic authentication is used, IPsec encryption key and Traffic Authentication Seed (TAS) are obtained by splitting the decrypted traffic key material into two parts, where the IPsec encryption key is identical to the first 16 bytes, and the TAS is identical to the second 16 bytes. The TAK (20 bytes) is derived from the TAS, as described in Section 9.1.

SRTP: The master key is identical to the decrypted traffic key material and SHALL always be a 16-byte key. How the keys for traffic decryption and authentication are derived from the master key is defined by SRTP.

ISMACRYP: If no traffic authentication is used, the decrypted traffic key material is identical to the key used for the AES-CTR decryption and its length is 16 bytes. If authentication is used, the first 16 bytes of the decrypted traffic key material are used as the 128 bit master key (MK) together with the 112 bit master_salt (MS) to derive encryption and authentication keys as described by STRP.

For the DRM Profile, when traffic authentication is used, the MS, from which the actual salt keys are derived, SHALL be signalled via SDP. When traffic authentication is not used, the salt keys as such are signaled in SDP.

Note that, for the Smartcard Profile, the MK is sent in the MIKEY STKM, and the MS is also sent in the MIKEY STKM.

DCF: If no traffic authentication is used, the encryption key is identical to the decrypted traffic key material (16 bytes).

If traffic authentication is used, the encryption key and the Traffic Authentication Seed (TAS) are obtained by splitting the decrypted traffic key material into two parts, where the encryption key is identical to the first 16 bytes, and the TAS is identical to the second 16 bytes. The authentication key (20 bytes) is derived from the TAS in the same way as specified for IPsec (see Section 9.1, Authentication for IPsec).

next_encrypted_traffic_key_material – is the encrypted key material used for encryption and optional authentication of the traffic after the current crypto period is over and the next crypto period starts. The structure of this attribute is the same as for the encrypted_traffic_key_material attribute.

timestamp – Field containing a timestamp at the point of sending the STKM. The timestamp SHALL be used as a reliable time of reception of the associated media stream for post-acquisition permissions. The device SHALL not use the timestamp as a reliable source for DRM time.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The format of the 40-bit mjdutc field is specified in Section 14. This 40-bit field contains the timestamp of the STKM in Universal Time, Co-ordinated (UTC) and Modified Julian Date (MJD). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 digits in 4-bit Binary Coded Decimal (BCD).

As an example, 93/10/13 12:45:00 is coded as "0xC079124500".

permissions_flag – indicates whether or not permissions category is defined for the program:

TKM_FLAG_FALSE	No permissions category is defined.
TKM_FLAG_TRUE	Permissions category is defined.

permissions_category – indicates the permissions category for the program:

0x00	No permissions category, RO applies as such,
0x01...0x3F	Permissions_category is included in the post- acquisition permissions lookup.
0x40...0xEF	Reserved for future standardization.
0xFF	No post-acquisition content protection (export in plaintext is allowed)

If permissions_category is in the range 0x01...0x3F,

- In case of a RO that is not a BCRO, the device SHALL use as service_CID for post-acquisition permissions lookup the text string

  service_CID = ***"cid:"*** || ***"b"*** || ***"#S"*** || ***baseCID*** || ***"@"*** || ***HEX(service_CID_extension)*** || **"_"** || ***HEX(permissions_category)***

and then apply the permissions specified in the service RO for this asset. Note that 'service_CID' shall be globally unique. Note further that, because of the specification of 'baseCID' in the Service Guide, the global uniqueness is guaranteed. (See Appendix H of [BCAST10-SG].) The baseCID component MUST NOT contain characters which are disallowed either by [RFC2396] URI syntax or by [RFC2392] cid-url syntax, such as ":".

- In case of BCRO, the device SHALL look up the permissions specified in the service BCRO for the asset that has a matching permissions_category field.

If permissions_category is in the (reserved for future standardization) range 0x40...0xEF, and the device does not support it, the device SHALL drop (i.e. ignore) all post-acquisition permissions (like play, redistribute etc.) indicated in the service RO, or if the device cannot do such permissions dropping, allow real-time rendering of the streaming content only (i.e. refuse to record the content, or to redistribute it in real time). Permissions_category has no impact on a Program RO. The permissions delivered in a Program RO apply as such.

If permissions_category = 0xFF, there is no need to protect the content after service protection has been removed; in other words, export in plaintext is allowed. This is comparable to setting protection_after_reception to 0x03. If protection_after_reception = 0x03 and permissions_category value is included in the STKM, the permission_category SHALL be set to 0xFF.

encrypted_PEK – is the Program Encryption Key (PEK) used within the current STKM to decrypt the traffic key material, encrypted using AES-128-CBC with a fixed IV equal to 0. The PEK is encrypted with the SEK.

program_CID_extension – is the extension of the program_CID, which allows to identify the program key material that has been delivered to the device within a LTKM for a program.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Note that for BCRO, a binary, fixed-size version of the content ID (CID) is needed. This ID is called BCI in this specification.

The CID/BCI of the service key is constructed as:

program_CID = "cid:" || "b" || "#P" || baseCID || "@" || HEX(program_CID_extension)

program_BCI = hash("cid:" ||"b" || "#P" || baseCID || "@") || program_CID_extension

The baseCID is a string value announced in the Service Guide; (see section 5.1.2 of [BCAST10-SG]). Upon reception of a STKM, the terminal can assemble the program_CID/BCI and look up the PEK (wrapped inside a LTKM). Note that 'program_CID' shall be globally unique. Note further that, because of the specification of 'baseCID', the global uniqueness is guaranteed. (See Appendix H of [BCAST10-SG].) The baseCID component MUST NOT contain characters which are disallowed either by [RFC2396] URI syntax or by [RFC2392] cid-url syntax, such as ":".

The HEX() function is a hexadecimal presentation of the parameter containing hexadecimal characters 0-9 and a-f (in lowercase) with possible preceding zeros. As an example, for a 16 bit value 2748, HEX() returns "0abc". Note that two characters are always generated for each byte.

The hash function for the construction of program_BCI is SHA1-64. It does not depend on the contents of the STKM, and can thus be pre-computed.

program_MAC – is the HMAC-SHA-1-96 according to [RFC2104] and [RFC2404] calculated over all preceding fields of the Short Term Key Message. It is used to authenticate the relevant part of the STKM in case of pay-per-view, where a PEK from a LTKM for a program is used to directly decrypt the traffic key material.

In case the terminal is accessing the STKM with a LTKM for a program, the terminal SHALL compute the program MAC, and drop the message if authentication fails. In this case, <program_MAC> MAY also be used to detect and drop duplicates (it can be expected that a particular STKM is repeated multiple times, in order to keep access times short for terminals that newly start receiving a broadcast transmission).

In case the terminal is accessing the STKM with a LTKM for a service, it will not be able to compute the program MAC, and there is no need for it to do so.

service_CID_extension – is the extension of the service_CID, which allows identifying the service key material that has been delivered to the device within a LTKM for a service.

Note that for BCRO, a binary, fixed-size version of the content ID (CID) is needed. This ID is called BCI in this specification.

The CID/BCI of the service key is constructed as:

service_CID ::= "cid:" || "b" || "#S" || baseCID || "@" || HEX(service_CID_extension)

service_BCI ::= hash("cid:" ||"b" || "#S" || baseCID || "@") || service_CID_extension

The baseCID is a string value announced in the Service Guide; (see section 5.1.2 of [BCAST10-SG]). Upon reception of a STKM, the terminal can assemble the service_CID/BCI and look up the SEK (wrapped inside a LTKM). Note that 'service_CID' shall be globally unique. Note further that, because of the specification of 'baseCID', the global uniqueness is guaranteed. (See Appendix H of [BCAST10-SG].) The baseCID component MUST NOT contain characters which are disallowed either by [RFC2396] URI syntax or by [RFC2392] cid-url syntax, such as ":"

The hash function for the construction of service_BCI is SHA1-64. It does not depend on the contents of the STKM, and can thus be pre-computed.

If the permissions_category field is present and has a nonzero value, the Service_CID of the service is constructed as specified at description of the permissions_category field.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

service_MAC – is the HMAC-SHA-1-96 according to [RFC2104] and [RFC2404] calculated over all preceding fields of the Short Term Key Message. It is used to authenticate the STKM with SAK in case of subscription, where a SEK from a LTKM for a service is used to decrypt the PEK and further decrypt the traffic key material.

In case the terminal is accessing the STKM with a LTKM for a service, the terminal SHALL compute the service MAC, and drop the message if authentication fails, i.e. if the computed MAC doesn't correspond to <service_MAC>. In this case, <service_MAC> MAY also be used to detect and drop duplicates (it can be expected that a particular traffic key message is repeated multiple times, in order to keep access times short for terminals that newly start receiving a broadcast transmission).

In case the terminal is accessing the STKM with a LTKM for a program, it need not compute the service MAC.

5.5.2 Authentication for STKMs for OMA DRM 2.0 Extensions

A STKM can contain two MAC fields: The program MAC and the service MAC. If only one MAC field would be used, the authentication key could only be renewed when both SEK and PEK change at the same time. Having two MAC fields and two authentication keys makes it possible to authenticate the message and check for its integrity while only having one key set. The Service Authentication Key (SAK) and the Program Authentication Key (PAK) will be derived from the Service Authentication Seed and the Program Authentication Seed respectively which are transmitted together with the encryption keys in the LTKMs. (How this is carried in the BCRO and RO is explained in [XBS DRM extensions-v1.0], Section C.14.2.1 and C.14.2.2, respectively.) A RO for a service will contain Service Encryption and Authentication Keys (SEAK) and a RO for a program will contain Program Encryption and Authentication Keys (PEAK).

To obtain the SAS or PAS from the BCRO the encrypted SEAK/PEAK is decrypted with the Inferred Encryption Key (IEK, see Section 4.1 in [XBS DRM-extensions-v1.0]):

$$SAS = LSB_{128}(D\{IEK\}(E\{IEK\}(SEAK))$$

$$PAS = LSB_{128}(D\{IEK\}(E\{IEK\}(PEAK))$$

The authentication key is generated from the authentication seed:

$$SAK = f_{auth}\{SAS\}(CONSTANT\_SAK)$$

$$PAK = f_{auth}\{PAS\}(CONSTANT\_PAK)$$

where :

CONSTANT_SAK = 0x02020202020202020202020202020202 (120 bit)

CONSTANT_PAK = 0x01010101010101010101010101010101 (120 bit)

The SAK or PAK is used in the MAC generation / verification of the STKM. The algorithm used to calculate the MAC field is HMAC-SHA1-96 according to [FIPS198] and [RFC2104], using authentication keys of 160 bit in both cases.

The function f_{auth} consists of several steps:

1. Denote by PRF{key}(text) as the AES-XCBC-MAC-PRF with output blocksize 128 bits as defined by IPsec WG in IETF. Please note:
   - ◊ Refer to [RFC3566] for the AES-XCBC-MAC-PRF based key generation function.
   - ◊ Refer to [RFC3664] for the requirement NOT to truncate the generated key material.
2. Apply the generated input key according to ideas of IKEv2 to generate authentication key. Define a key generator function f-kg{key}(constant). Keying material will always be derived as the output of the negotiated PRF algorithm.. PRF^+ describes the function that outputs a pseudo-random stream of n blocks based on the inputs to a PRF as follows:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

$$T1 = AES\_XCBC\_MAC\_PRF\{AS\}(CONSTANT \| 0x01)$$

$$T2 = AES\_XCBC\_MAC\_PRF\{AS\}(T1 \| CONSTANT \| 0x02)$$

....

$$Tn = AES\_XCBC\_MAC\_PRF\{AS\}(T1 \| CONSTANT \| n)$$

where *AS* is the appropriate authentication seed (be it TAS, PAS, SAS or RIAK) and *CONSTANT* is the appropriate constant as described in this section, Section 9.1 and [XBS DRM extensions-v1.0]. The amount of blocks to derive is defined by the amount of key material needed, i.e. n is the amount of needed key bits divided by 128 and rounded up.

This means that if 160 bits were needed then PRF*() would be computed as:

$$T1 \| T2 = PRF^{+}\{K\}(S)$$

3. The 160 bit authentication key is taken from the generated key material as follows:

$$AK = MSB_{160}(T1 \| T2)$$

The generated authentication key is applied as described in this section and Section 9.1.

5.5.3 Parental control processing (Informative)

DRM Profile signals parental control information about content via access criteria in STKMs. How this information is provisioned on the Terminal is outside the scope of this specification (e.g. the rating_type and level_granted information can be provided by the parent/user himself). The usage of a parental control PINCODE is optional in DRM Profile and when a PINCODE is used, its storage location and management is outside the scope of this specification.

The rating_value transmitted in the STKM is checked against the level_granted stored in the Terminal for the rating_type.The Terminal should compare the rating_type received in the STKM against all of the rating_type values stored in the Terminal. If there is a level granted, depending on the rating_value and the rating_type, the outcome is success or failure. If the processing of the parental_control access criteria ends with success, the Terminal proceeds with other STKM processing.

Success If there is a level_granted for the rating_type in the Terminal and if it is an equal or more restrictive value than the rating_value received in the STKM, the checking of rating_value ends with success and the processing of STKM resumes.

If there is no level_granted for the rating_type in the Terminal , the user is authorized to view the content. The checking of rating_value ends with success and the processing of the STKM resumes.

Failure If there is a level_granted for the rating_type in the Terminal and if it is less restrictive than the rating_value received in the STKM, the checking of rating_value ends with failure. The Terminal MAY trigger a request for a parental control PINCODE (if one is used). Otherwise the Terminal should indicate to the user that he is not allowed access the content.

Note that the term ‘more restrictive’ means that there are more constraints on having access to the content. This typically means the user age is higher. Note that actual numerical values of rating_value for certain rating_types do not always follow a linear scale, either from less restrictive to more restrictive or vice-versa. The corresponding logical order (from least restrictive to most restrictive) is based on the semantics of the individual rating values. An informative example can be found in Table 130 in Appendix H.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Note that the value for “not rated” or “undefined” should be treated by default as “least restrictive”, unless its semantics is explicitly stated by the rating scheme.

5.6 Layer 4: Traffic Encryption

Layer 4 corresponds to the BCAST 4-layer key hierarchy model and describes how to protect data. The services considered for the BDS delivery are streaming sessions and file downloads, for which service and content protection is described in the following sections.

5.6.1 Streaming Delivery

5.6.1.1 Service Protection of Streams

Broadcast streams that are signalled as having service protection are securely delivered to authorized users. The service protection mechanism protects streams only at the delivery time. The streamed content after the removal of service protection can be stored in clear if post delivery content protection is not signalled.

For DRM Profile, Layer 4 protection is provided through encryption. The encryption mechanisms are described in Section 9 of this document.

5.6.1.2 Content Protection of Streams

Broadcast streams that are signalled (through protectionType value in Service Guide and protection after reception value in STKM) as having content protection may be recorded as defined in this specification. However, for recorded material having content protection, appropriate rights need to be obtained via Rights Issuer.

For terminals using the DRM Profile, the appropriate key material can be requested based on the Program or Service ID.

As the content encryption key provides access to recorded content stored in the terminal, preventing unauthorized access to content encryption key is extremely important. However, the exact storage and handling of content encryption key in the device is specific to an implementation.

5.6.2 File Delivery

5.6.2.1 Service Protection of Files

BCAST terminal and server MAY support download protection using DCF.

The same mechanism can be used to protect PDCF files. This is optional for both terminal and server.

Service protection of download data uses IPsec or DCF encryption protocol. In case of DCF encryption protocol, DCF file is used as a container for ciphered file data. The DCF container also identifies the keys used in protecting the data.

Each file is encrypted using a single TEK, as explained in Section 9.4.

If a file is transmitted in a FLUTE carousel, the same key value of the TEK SHALL be used during the whole time the file is transmitted using the same TOI in an ongoing FLUTE session.

The correct TEK for decrypting and verifying the integrity of the download data is indicated by the KeyID field in the Key Info box.

For the DRM Profile, KeyID takes its value as follows:

- If SEK is used for protecting STKMs, KeyID is defined as the base64 encoded concatenation (service_CID_extension || ";" || TEK ID).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- If PEK is used in protectig STKMs and the PEK is not protected by an SEK, KeyID is defined as the base64 encoded concatenation (program_CID_extension || ";" || TEK ID).
- If PEK is used in protecting STKMs and the PEK is protected by an SEK, KeyID is defined as the base64 encoded concatenation (service_CID_extension || ";" || program_CID_extension || ";" || TEK ID).

The RightsIssuerURL MAY be indicated within the Key Info box in the KeyIssuerURL, or MAY be indicated in the RightsIssuerURL in the OMADRMCommonHeaders box.

5.6.2.2 Content Protection of Files

When using the DRM Profile, Content Protection for files SHALL follow OMA DRM 2.0 specification [DRMCF-v2.0].

For audio or video content either the PDCF or the DCF formats SHALL be used.

5.7 Recording

Please refer to Section 8 for details on recording.

5.8 SG Signalling

SG signalling is described in [BCAST10-SG]. The relevant fragments linking SG signalling to service and content protection are the Service, Content, Access, Purchase Item, Purchase Data and Purchase Channel Fragments.

The Access Fragment describes how the service may be accessed during the validity time of the access fragment. The fragment links to Session Description and indicates the delivery method. KeyManagementSystem element identifies the type of KMS that can be used to contact the RI. The value of this element for DRM Profile is oma-bcast-drm-pki. The associated attributes are ProtectionType and RightsIssuerURI. The ProtectionType attribute specifies the protection type (service protection only, content protection only or both service & content protection) offered by the DRM Profile. The RightsIssuerURI specifies the URI of RightsIssuer that should be contacted to obtain ROs.

The Purchase Channel Fragment represents a system from which access and content rights can be purchased by the terminal. The associated attribute RightsIssuerURI specifies the identity of the rights issuer associated with the BSM. For DRM Profile, RightsIssuerURI SHALL be specified.

For devices that support an interaction channel, the PurchaseURL in the Purchase Channel Fragment specifies the URL to which the interactive service provisioning messages defined in [BCAST10-Services] are to be addressed. An interactive service ordering procedure will result in the delivery of a ROAP trigger to the device, which in turn uses the trigger to initiate a Rights Object Acquisition as specified in [DRMDRM-v2.0].

For broadcast-only devices, the Purchase Channel contains information on how to initiate an out-of-band purchase. For an overview of the purchase message flow, see [BCAST10-Architecture].

The Purchase Item fragment contains the GlobalPurchaseItemID, used to refer to the services, service bundles or pay-per-view programs when subscribing via the BSM.

The Purchase Data fragment contains additional information on how the purchase item can be subscribed to. Depending on the chosen purchase data, the resulting LTKM will contain different access rights.

To identify the asset in the RO needed for a service or a program, the following parameter is used in SG: baseCID. The parameter baseCID is announced in the Service fragment and Content fragment of the SG.

5.9 Usage Metering for DRM Profile

Extensions to OMA DRM v2.0 for usage metering appear in the OMA DRM v2.0 Extensions for Broadcast Support document [XBS DRM extensions-v1.0].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6. Smartcard Profile

Caution: The term "Smartcard" is used in this document in the restricted sense specified in the definition provided in Section 3.2.

6.1 Introduction

The Smartcard Profile is based on an existing security framework for service protection defined for broadcast/multicast services based on smartcards defined by 3GPP MBMS [3GPP TS 33.246 v7] and may include the key provisioning mechanism defined for 3GPP2 BCMCS [3GPP2 S.S0083-A]. The solution requires an interactive channel to obtain key material.

Two variants of the Smartcard Profile are defined in this specification: the (U)SIM Smartcard Profile and the (R-)UIM/CSIM Smartcard Profile. The two variants differ in the way that the Layer 1 key(s) are established (see Section 6.5 but are otherwise the same (Layers 2, 3 and 4).

The following sections describe the four layers of the 4-layer model key hierarchy, as well as the key provisioning required to access the first layer.

Section 6.4 briefly describes the provisioning of the SmartCard Key (SCK). Section 6.5 describes Subscriber Key Establishment. Section 6.6 details the structure and delivery of the LTKM while Section 6.7 describes those of the STKM. Section 6.8.1 and Section 6.8.2 describe how to protect content in case of streaming and file delivery respectively for both service and content protection. Recording aspects are detailed in Section 6.9 while SG signalling is explained in Section 6.10.

6.2 Relationship between MBMS Security and the Smartcard Profile

Appendix G provides a description of BCAST compatibility with MBMS Smartcards and clarification of how BCAST 1.0 enables the use of MBMS only Smartcards. As stated above, the Smartcard Profile uses the key management defined by 3GPP MBMS [3GPP TS 33.246 v7]. To clarify the relationship between the two specifications the following tables provide a mapping between the keys and key IDs used in [3GPP TS 33.246 v7] and this specification. The remainder of this specification uses the terminology defined for the Smartcard Profile.

Table 6: Mapping between MBMS keys and Smartcard Profile Keys

MBMS key	Smartcard Profile key
MBMS User Key (MUK)	Subscriber Management Key (SMK)
MBMS Registration Key (MRK)	Subscriber Request Key (SRK)
MBMS Service Key (MSK)	Service Encryption Key (SEK) [1]
MBMS Traffic Key (MTK)	Traffic Encryption Key (TEK)

[1] The Smartcard also supports the concept of a Program Encryption Key (PEK); see Section 11.1 for further details.

Table 7: Mapping between MBMS key IDs and Smartcard Profile Key IDs

MBMS		Smartcard Profile	
Key ID	**Construction**	**Key ID**	**Construction**
MUK ID	MUK ID is received by combining IDi and IDr, where IDi is the identity of the initiator and the IDr is the identity of the responder. IDr is Bootstrapping – Transaction ID (B-TID) and IDi is the Network	SMK ID	As for MUK ID

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	Application Function ID (without the Ua security protocol identifier), as defined in [3GPP TS 33.246 v7].		
MRK ID	The B-TID is used as the username when MRK is used as the password within HTTP digest and so can be thought of as the MRK ID (although it is never defined as such within the MBMS specification). See [3GPP TS 33.246 v7] for further details.	SRK ID	As for MRK ID
MSK ID	MSK ID is 4 bytes long and with byte 0 and 1 containing the Key Group part, and byte 2 and 3 containing the Key Number part. Every MSK is uniquely identifiable by its Key Domain ID and MSK ID where Key Domain ID = Mobile Country Code || Mobile Network Code, and is 3 bytes long (see [3GPP TS 33.246 v7] for further details).	SEK/PEK ID	As for MSK ID. Note that for the Smartcard Profile, the Key Group part value 0x01 is reserved (see Section 6.6.3). MBMS reserves the Key Group part 0x00 for future use. This value SHALL also be reserved for the Smartcard Profile.
MTK ID	MTK ID is 2 bytes long sequence number and is used to distinguish MTKs that have the same Key Domain ID and MSK ID. Every MTK is uniquely identifiable by its Key Domain ID, MSK ID and MTK ID (see [3GPP TS 33.246 v7] for further details).	TEK ID	As for MTK ID

The Smartcard Profile BSM provides the functionality that in MBMS is provided by the MBMS Broadcast-Multicast Service Centre (BM-SC) security functions. As such the Smartcard Profile BSM SHALL support the following MBMS BM-SC security functions:

- Key Management function
    - Key Request function
    - Key Distribution function
- Membership function

as defined in [3GPP TS 33.246 v7], with the modifications described in this specification. Note that the Session and Transmission functionality is not required to be supported by the BSM as this functionality is provided by the BSDA.

MBMS uses the Generic Bootstrapping Architecture (GBA) [3GPP TS 32.220] to establish a MUK and MRK between the BM-SC, an instance of a GBA Network Application Function (NAF), and the USIM/terminal. GBA requires the implementation of a Bootstrapping Server Function (BSF) to enable the bootstrapping procedure required to establish MUK and MRK. Within this specification the (U)SIM Smartcard Profile BSM is assumed to support BSF functionality required to establish SMK and SRK, which are equivalent to the MBMS MUK and MRK respectively. However, the BSF may be shared between the BSM NAF and NAFs for other services.

The (R-)UIM/CSIM Smartcard Profile derives SMK and SRK from the SmartCard Key (i.e. RK) pre-provisioned on the (R-)UIM/CSIM and in the BCMCS Subscription Manager (SM) function. Within this specification the (R-)UIM/CSIM Smartcard Profile BSM is assumed to support the SM functionality requied to establish SMK and SRK.

Smartcard Profile terminals SHALL support the key management functionality specified for MBMS terminals, as defined in [3GPP TS 33.246 v7], with the modifications described in this specification.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

(U)SIM Smartcard Profile Smartcards (i.e. (U)SIMs) SHALL support all key management functionality specified for MBMS capable (U)SIMs, as defined in [3GPP TS 31.102 v7] and MAY support the additions and modifications described in this specification.

(R-)UIM/CSIM Smartcard Profile Smartcards SHALL support the key management functionality specified for MBMS capable (U)SIMs related to the processing of MBMS MSK and MTK messages , as defined in [3GPP TS 33.102] and MAY support the additions and modifications described in this specification. (R-)UIM/CSIM Smartcard Profile Smartcards SHALL support the functionality defined in [3GPP2 S.S0083-A] to derive the Temporary Key (TK) and Authentication Key (Auth-Key), which correspond to the SMK and SRK respectively, from the pre-provisioned Registration Key (RK).

The SEK/PEK ID are mapped to the MSK ID as described in the above table. The SEK/PEK ID SHALL comply with the following rule:

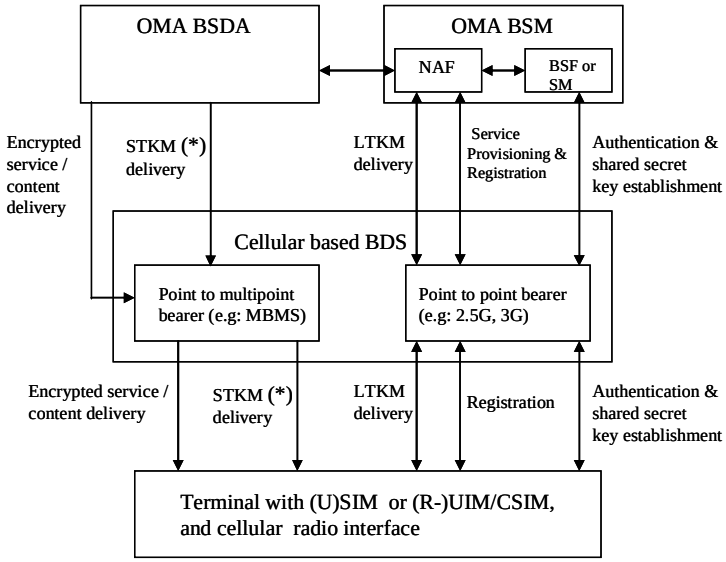
Within a Key Group (i.e. SEKs/PEKs with the same Key Group part of the SEK/PEK ID), the Key Number part of the SEK/PEK ID SHALL increase for every new SEK/PEK used by the BSM. This guarantees that, within a same Key Group, a SEK/PEK ID will have a Key Number part greater than the Key Number part of a SEK/PEK ID belonging to a previously delivered SEK/PEK.

6.3 Use of the Smartcard Profile for Various BDS Architectures

Different BDS architectures can be used with the Smartcard Profile using MBMS key management. The Smartcard Profile is applicable to cellular based BDS architectures, which natively can use a point-to-multipoint or point-to-point bearer, and also to broadcast-only BDS architectures with the additional support of a cellular interaction channel.

6.3.1 Smartcard Profile using a pure Cellular Based BDS

In the pure cellular based BDS case, both multicast/broadcast and unicast bearers are available.



(*) Short-term key message may be delivered over the point-to-point bearer instead.

Figure 2 – Pure Cellular based BDS Scenario

As a clarifying note for the scenario shown in, there may exist one or more broadcast service providers, each represented by a separate instance of the BSM. One of these broadcast service providers also serves as the cellular BDS network operator, such that the BDS-SD is functionally integrated with the BSDA and BSM.

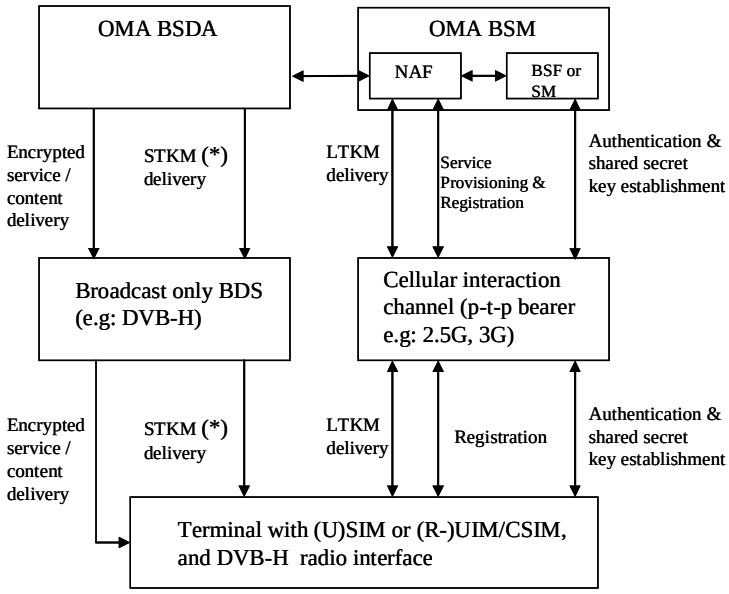
A basic overview of the operations of the Smartcard Profile in this scenario is as follows:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Broadcast Service Discovery: A user selects a protected service on the BCAST service guide available over the cellular based BDS.
- Authentication and Subscriber Key Establishment: This corresponds to Layer 1 of the BCAST 4-layer model key hierarchy and takes place via the point-to-point bearer. See Section 6.5 for details of how Authentication and Subscriber Key Establishment is handled for the two variants of the Smartcard Profile.
- LTKM Delivery: This corresponds to Layer 2 of the BCAST 4-layer model key hierarchy. After Layer 1 keys have been established between the BSM and the Smartcard/terminal and the terminal has subscribed to one or more services, the terminal may request the relevant LTKMs from the BSM, or alternatively, the BSM may send them automatically. LTKMs are delivered by the BSM to the terminal, via the point-to-point bearer. The construction, delivery and processing of Smartcard Profile LTKMs is explained in Section 6.6.
- STKM Delivery: This corresponds to Layer 3 of the BCAST 4-layer model key hierarchy. STKMs are used to deliver the TEKs and may be delivered over the point-to-multipoint bearer or the point-to-point bearer to the terminal. The construction, delivery and processing of Smartcard Profile STKMs is explained in Section 6.7.
- Access to protected content: This corresponds to Layer 4 of the BCAST 4-layer model key hierarchy. The cellular based BDS delivers a service, e.g. a file download or streaming session, which may be transmitted over the cellular network in unicast, multicast or broadcast mode.

6.3.2 Smartcard Profile using a broadcast BDS and cellular interactive channel

In a mixed or hybrid scenario (e.g. IPDC over DVB-H + cellular interaction channel) a pure broadcast BDS is complemented with an additional interaction channel given by a cellular network.



(*) Short-term key message may be delivered over the cellular interaction channel instead.

Figure 3 – Broadcast-only BDS with Cellular Interaction Channel Scenario, using either GBA or derivation of Layer 1 Key from RK

The same clarification note as indicated for applies here as well.

A basic overview of the operation of the BCAST Smartcard Profile in this scenario can be the following:

- Broadcast Service Discovery: As for cellular BDS above but available over the broadcast BDS (e.g. IPDC over DVB-H).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Authentication and Registration: As for cellular BDS above i.e. via the cellular interaction channel.
- LTKM Delivery: As for cellular BDS above i.e. via the cellular interaction channel
- STKM Delivery: As for cellular BDS above but STKMs may be delivered over the broadcast only BDS (e.g. IPDC over DVB-H) or via the cellular interaction channel.
- Access to protected content: As for cellular BDS above but available over the broadcast BDS (e.g. IPDC over DVB-H).

6.4 Use of Pre-provisioned Keys

The Smartcard Profile uses a pre-provisioned secret key - the "SmartCard Key" (SCK) - stored on the Smartcard to establish the shared Layer 1 key(s) between the BSM and the Smartcard/terminal, as described in Section 6.5. The SCK corresponds to the authentication key "K" stored on 3GPP compliant USIMs [3GPP TS 31.102 v7], to the authentication key "Ki" on 3GPP compliant SIMs [3GPP TS 31.111 v9.1.0], and to the key "RK" on 3GPP2 compliant (R-)UIM/CSIMs [3GPP2 C.S0023-C] or CSIM [3GPP2 C.S0065-0].

How the SCK is provisioned is out of scope of this specification.

6.5 Layer 1: Subscriber Key Establishment

6.5.1 Subscriber Key Establishment using a (U)SIM

This layer enables the establishment of two shared keys to secure communiction between the BSM and the terminal: The Subscriber Management Key (SMK), which is used to protect the delivery of SEK/PEKs within LTKM from the BSM to the terminal, and the Subscriber Request Key (SRK), which is used to secure communication between the terminal and the BSM. The SMK corresponds to the MBMS User Key (MUK) while the SRK corresponds to the MBMS Request Key (MRK), where the MBMS keys are as defined in [3GPP TS 33.246 v7].

The (U)SIM Smartcard Profile is based on MBMS security and therefore SMK and SRK are derived by running the GBA bootstrap procedure, as defined in Section 6.1 "Using GBA for MBMS" of [3GPP TS 33.246 v7]. In particular, the Ua protocol used for this GBA procedure is the Ua protocol defined for MBMS. The Ua protocol value is defined in annex H.3 of [3GPP TS 33.220 v6]: (0x01 0x00 0x00 0x00 0x01): Ua security protocols according to [3GPP TS 33.246 v7]. Appendix J describes a method to derive the Zn URL (the URL to which Zn requests are directed when Zn is over web services) used in the GBA procedure when a shared NAF and many BSFs are used. The relationship of the BSM to the GBA NAF and BSF elements is described in Section 4.1.4.2.

6.5.2 Subscriber Key Establishment using a (R-)UIM/CSIM

BCMCS uses pre-provisioning to establish a unique 128-bit Registration Key (RK) in the (R)-UIM/CSIM and the Subscription Manager (a functional entity, SM) prior to providing service. This is referred to as the SmartCard Key (SCK) within this specification. The SM performs accounting, authentication and authorization for BCMCS. The SM also calculates the "Auth-Key", derived from the RK, which is used to secure communication between the terminal and the BSM. (The RK is functionally equivalent to the SMK and, therefore, the MUK in MBMS. The "Auth-Key" is functionally equivalent to the SRK and, therefore, the MRK in MBMS.) The SM may be the subscriber's home AAA (H-AAA) or an independent entity.

6.6 Layer 2: Service Provisioning and LTKM Delivery

To access a protected service a terminal must obtain the necessary LTKM(s). To receive the LTKM(s) the terminal must subscribe to or purchase a BCAST purchase item. Subscription MAY be achieved using one of the following Service Provisioning messages, as defined in [BCAST10-Services]:

- "Service Request"
- "Token Purchase Request"

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Alternatively, subscription MAY be achieved via other channels, e.g. the user may subscribe to the service via a web portal/shop (see Section 6.10.3 for more details).

The BSM SHALL authenticate the sender of the Service Provisioning or Registration message(s) sent by the terminal, by following the HTTP DIGEST authentication procedure defined in section 6.3.2.1A of [3GPP TS 33.246 v7], e.g. the BSM shall ensure that a valid SRK is used for in the HTTP DIGEST authentication. If authenticated is successful the request SHALL be acknowledged using an HTTP 200 OK message. Note that the requirement for a valid SRK also ensures that a valid SMK has been established.

The Smartcard profile procedures for which this HTTP DIGEST authentication applies SHALL be: Pricing Information, Service Request, Subscription Renewal, Unsubscription, Token Purchase, Account Inquiry, LTKM Request, Registration and De-registration.

The terminal SHALL authenticate itself to the BSM in the first request of the concerned procedure, whenever it assumes to hold the valid authentication credentials for the realm in scope. In this case, the terminal SHOULD use in digest-response of Authorization header the nonce provided by “nextnonce” directive in last Authentication-Info response received for this realm, or if “nextnonce" directive not present or not supported, SHALL use the nonce provided by “nonce” directive in last digest-challenge received for this realm.

HTTP DIGEST authentication directives SHALL be specified as follows:

- the “realm” directive in digest-challenge SHALL contain two parts delimited by the "@" sign. The first part is the constant string "3GPP-bootstrapping" (when SMK and SRK where established using GBA_ME) or "3GPP-bootstrapping-uicc" (when SMK and SRK where established using GBA_U), and the latter part shall be the FQDN of the BSM (NAF).
- the “stale” directive SHALL be included in digest-challenge and set to “TRUE” to indicate to terminal that the request digest in digest-response (and consequently also the username B-TID/NAI and password SRK) is valid but the nonce used for this digest is stale. The terminal SHOULD then retry to send the request using in the digest-response the nonce value provided in digest-challenge.
- the “qop-options” directive SHALL always be specified in digest-challenge, with possible values “auth” or “auth-int”. Consequently, “cnonce” and “nonce-count” directives SHALL always be specified in digest-response.
- the “nextnonce” directive MAY be specified in Authentication-Info header.

Following a successful service registration, the LTKMs corresponding to the services to which the terminal is subscribed SHALL be delivered by the BSM to the terminal as a result of a push or pull procedure as defined in sections 6.3.2.2 and 6.3.2.3 of [3GPP TS 33.246 v7]. This provides support for the scenarios described below:

- The BSM MAY push an LTKM to the terminal in order to provide a new SEK/PEK to a terminal. Pushing LTKMs to registered terminals allows the BSM to spread the delivery of SEKs/PEKs required by a large number of users to manage network congestion, e.g. the BSM determines when the LTKM is pushed to the terminal.

  The “Registration” message, as defined in [BCAST10-Services], SHALL be sent by the terminal after the application is started and the terminal re-establishes connectivity to the interactive network associated with its service provider. In addition, the “Registration” message SHALL be sent by the terminal in response to a BSM Solicited Pull Procedure where the BM-SC Solicited Pull message is formatted according to Section 6.6.3 below. This message indicates to the BSM that the terminal is available to receive any LTKMs that it may have missed while it was unreachable. Note that when a terminal establishes connectivity with an interactive network that is not associated with its service provider, e.g. in the case of roaming between cellular networks, the terminal MAY send the Registration message.

  The sending of the “Registration” message also ensures that the terminal establishes the necessary IP connectivity required to enable the BSM to push the LTKM(s) over UDP. Note that the “Registration Request” message corresponds to the MBMS “User Service Registration” message, as defined in [3GPP TS 33.246 v7].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

When the BSM wishes to push an LTKM, if the network is able to retrieve a valid IP address for the terminal, the LTKM can be pushed over UDP. Otherwise, the BSM can use the BSM Solicited Pull Procedure Initiation over SMS Bearer feature described below to deliver the LTKM.

- The terminal SHALL request the LTKM associated to a particular service when the terminal realises that it has missed an LTKM update, e.g. due to being out of coverage. The terminal SHALL use the "LTKM Request" message, as defined in [BCAST10-Services], to request the missed LTKM. Note that the "LTKM Request" message corresponds to the MBMS "MSK Request message", as defined in [3GPP TS 33.246 v7].

- The BSM MAY trigger the terminal to request the current LTKM for a particular service. This process SHALL be as defined for the "BM-SC solicited pull procedure" in section 6.3.2.2.4 of [3GPP TS 33.246 v7]. The solicited pull procedure can be used to provide a means to update the terminal with a new SEK when:

    - the SMK is no longer valid, e.g. the BSM can respond to the "Registration Request" message from the terminal with an HTTP 401 WWW Authenticate message, thereby initiating a new run of GBA;

    - the terminal is not trusted to provide acknowledgment of LTKM delivery, e.g. with the solicited pull procedure the BSM can assume successful delivery if the terminal does not repeat the "Registartion Request" message;

    - the UE has just registered to a User Service, and needs to initiate the delivery of the SEK/PEK

  The BSM SHALL NOT send a BSM solicited pull procedure initiation message to a terminal deregistered with this BSM, regardless of the bearer in use for this type of message (UDP or SMS).

  A terminal SHALL ignore a received BSM solicited pull procedure initiation message if BCAST client is not started or if the terminal is currently not registered with the BSM identified by MIKEY IDi payload (i.e. terminal deregistered with this BSM, or never registered with this BSM), regardless of the bearer used for this type of message (UDP or SMS).

  The terminal SHALL send the LKTM request to a Request-URI compliant with [3GPP TS 33.246 v7] section G.2.3, that SHALL be structured as follows:

  http://***IDi***/keymanagement?requesttype=msk-request

  where ***IDi*** is the BSM FQDN carried in the MIKEY IDi payload of the LTKM initiating the procedure (for the (U)SIM Smartcard Profile, IDi payload = NAF_Id without the Ua security protocol identifier). As specified in [3GPP TS 33.246 v7], the terminal MAY add additional URI parameters to the Request-URI.

When a terminal has successfully unsubscribed from a BCAST service using an "Unsubscribe Request", as defined in [BCAST10-Services], or via other means like a web shop, the BSM MAY invalidate the SEK/PEKs on the terminal that are associated with the relevant purchaseItemID and that are not used by any other purchase items to which the device is subscribed. The BSM invalidates SEKs/PEKs by sending an LTKM with invalid Key Validity data, i.e. the lower bound is greater than the upper bound, where the bounds define the allowed range of either TEK or TimeStamp values.

Note that once a purchaseItemID has expired, i.e. the content associated with this purchaseItemID has been broadcast, the terminal SHALL remove the purchaseItemID from all subsequent "RegistrationRequest" and "Deregistration request" messages, as defined in [BCAST10-Services], sent to the BSM.

Table 8 below shows the MIKEY message format used for LTKMs in the Smartcard Profile. The message structure SHALL be identical to the MIKEY message used by MBMS to deliver the MBMS Service Key (MSK) (defined by [3GPP TS 33.246 v7]) apart from the addition of the (optional) EXT BCAST payload as defined in [RFC5410]. The EXT BCAST payload is described in Section 6.6.4.

Table 8: The Logical Structure of the MIKEY Message used for LTKMs. The use of brackets is according to section 1.3 of RFC 3830 (MIKEY)

Common HDR
EXT MBMS

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

{EXT BCAST}
TS*
MIKEY RAND
IDi
IDr
{SP}
KEMAC

* The TS (timestamp) field in the LTKM pertains strictly to the LTKM, for the purpose of LTKM replay detection. It is not the same timestamp that exists in the STKM (the latter serves for replay detection of STKMs).

The structure of the EXT MBMS payload, depicted in Table 8 above, is defined in Section 6.4.4 "General extension payload" of [3GPP TS 33.246 v7] and reproduced below for convenience. For the Smartcard Profile LTKM the EXT MBMS payload is the extension payload defined in [3GPP TS 33.246 v7] for use with MBMS MIKEY MSK message.

Table 9: The Logical Structure of the EXT MBMS Payload

Key Domain ID sub-payload
Key Type ID sub-payload (MSK ID)

All fields in the Smartcard Profile LTKM, with the exception of the EXT BCAST payload and the modifications defined in this section, SHALL be populated as defined in [3GPP TS 33.246 v7] for the MBMS MSK message. The mappings described in Section 6.2 for the Smartcard Profile parameters SHALL be used, i.e. SEK/PEK ID is mapped to MSK ID and SEK/PEK is mapped to MSK.

All fields in the BCAST MIKEY LTKM SHALL be populated as defined in [3GPP TS 33.246 v7] with the above mapping for BCAST parameters. BCAST MIKEY LTKM messages SHALL be transported to UDP port number 4359, registered with IANA under the name "omabcastltkm".

The EXT BCAST for LTKMs SHALL be populated as defined in Section 6.6.4. If the LTKM message includes the EXT BCAST payload and the security_policy_ext_flag is set to LTK_FLAG_TRUE or the consumption_reporting flag is set to LTK_FLAG_TRUE, then the Key Validity data subfield of the KEMAC payload in the LTKM defines the Key Validity interval for the SEK/PEK in terms of a specified interval of STKM Timestamp values:

From (32-bits): Lower limit of STKM Timestamp ("TS low")

To (32bits): Upper limit of STKM Timestamp ("TS high")

It should be noted that the use of STKM Timestamps to define Key Validity, as described above, is a deviation from the MBMS Security specification.

It should be noted that [3GPP TS 33.246 v7] omits to specify the ID type of IDr and IDi payloads. For BCAST 1.0, the following applies:

- The ID type of IDi payload SHALL NOT be interpreted by the terminal and the Smartcard, and
- the ID type of IDr payload SHALL be set to NAI, as IDr payload contains a B-TID and B-TID is generated in format of NAI according to [3GPP TS 33.220 v6].

SP is present only when the LTKM addresses a streaming service which uses SRTP (unless empty map is used as defined in Section 6.7.4). If the LTKM message does not include the EXT BCAST payload or when the security_policy_ext_flag is set to LTK_FLAG_FALSE and the consumption_reporting flag set to LTK_FLAG_FALSE, the standard Key Validity data is

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

constructed as per [3GPP TS 33.246 v7], i.e. the Key Validity data subfield in the KEMAC payload is defined in terms of sequence number interval (i.e. lower limit of MTK ID and upper limit of MTK ID, where MTK ID is a 16 bits identifier).

6.6.1 LTKM Related Terminology

In the sections below, the following terms are used for LTKM:

Resending check: Resending LTKMs with the same key material is allowed provided the 32-bit counter timestamp in the TS field is increased, as mandated by MBMS [3GPP TS 33.246 v7].

LTKM replay detection processing or verification: This procedure is used to detect replay attacks for LTKMs. This procedure operates by comparing the TS field in the LTKM with the corresponding LTKM replay detection counter in the secure function. Detection of an LTKM replay implies a replay attack and the secure function SHALL discard the LTKM and send an error message as defined in Section 6.6.7.4.

LTKM replay detection (or freshness failure): This occurs when the LTKM contains a TS field value which is less than or equal to the current LTKM replay detection counter.

Message validation: The LTKM Message Validation check consists of the verification of integrity of the LTKM message, using the SMK. This procedure is equivalent to that described in the section 7.1.1.6 for MSK messages of [3GPP TS 31.102 v7].

Play-back counter: An internal counter in the secure function that contains the number of play-backs authorized.

SEK/PEK key group: A group of SEK/PEKs that are identified by the same Key group part of the SEK/PEK ID. The SEK/PEK key group is uniquely identifiable by its Key Domain ID and Key group part of the SEK/PEK ID.

6.6.2 BSM Solicited Pull Procedure Initiation over SMS Bearer

The first message in the BSM solicited pull procedure referenced in Section 6.6 is sent over UDP, requiring the terminal to have a valid IP address. In networks in which it cannot be guaranteed that terminals maintain a valid IP address the BSM MAY trigger a solicited pull procedure by sending a LTKM over SMS to the terminal. That LTKM SHALL NOT include an MSK, but it SHALL be encoded according to this specification, with MSK ID Key Number part = 0, KEMAC Encr Data Len = 0, and V-bit in HDR not set. The SMS SHALL satisfy the following conditions:

- The SMS carries a WAP connectionless push (WDP/WSP encoding) as defined in [OMA Push] ;
- The WSP content type header contains the Content Type code registered by OMNA for ‘application/mikey’, i.e. the binary value 0x52;
- The WSP X-Wap-Application-Id header contains the binary code registered by OMNA for the PUSH Application ID identifying the BCAST Push client, as specified in [BCAST10-Distribution].

The terminal SHALL process this LTKM carried over SMS exactly as it processes the LTKMs carried over UDP that initiate a BSM solicited pull procedure. That means the message SHALL trigger the terminal to request the current MSK for the specified Key Group over UDP, and the terminal SHALL send the LKTM request to the Request-URI specified in Section 6.6 for the case of BSM solicited pull procedure initiated over UDP.

Support for this BSM solicited pull procedure initiation over SMS bearer is:

- for the BSM: optional;
- for BCAST terminal: mandatory if Smartcard Profile using (U)SIM is supported, optional otherwise.

6.6.3 BSM Solicited Pull Procedure to Initiate the Registration Procedure

If the terminal receives a BSM Solicited Pull message, formatted as defined in [3GPP TS 33.246 v7], that contains an MSK ID Key Group part = 1, the terminal SHALL initiate a Registration procedure [BCAST10-Services] with the MBMS User Service ID = “oma-bcast-allservices” and SHALL not send an LTKM Request. Note that according to [3GPP TS 33.246 v7]

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

the MSK ID Key Number part in a BSM Solicited Pull message SHALL always be set equal to 0. The BSM Solicited Pull message used to initiate the Registration procedure SHALL therefore contain an MSK ID where the Key Group part set equal to 1 and Key Number part are set = 0. There is no change in the processing required by the Smartcard, i.e. the message is processed as a 'normal' BSM Solicited Pull message. The MSK ID Key Group value '1' SHALL be reserved in BCAST and SHALL not be used for live services.

The BSM SHALL NOT send a BSM Solicited Pull message to a terminal deregistered with this BSM, regardless of the bearer in use for this type of message (UDP or SMS).

A terminal SHALL ignore a received BSM Solicited Pull message if the BCAST client is not started or if the terminal is currently not registered with the BSM identified by MIKEY IDi payload (i.e. terminal deregistered with this BSM, or never registered with this BSM), regardless of the bearer used for this type of message (UDP or SMS).

The terminal SHALL send the Registration request to a Request-URI compliant with [3GPP TS 33.246 v7] section G.2.1, that SHALL be structured as follows:

http://***IDi***/keymanagement?requesttype=register

where ***IDi*** is the BSM FQDN carried in the MIKEY IDi payload of the LTKM initiating the procedure (for the (U)SIM Smartcard Profile, IDi payload = NAF_Id without the Ua security protocol identifier). As specified in [3GPP TS 33.246 v7], the terminal MAY add additional URI parameters to the Request-URI.

Note: the BSM Solicited Pull procedure can only be used after the keys SMK and SRK have been established between the terminal and the BSM, i.e. after GBA has been run.

BSM Solicited Pull message delivery over UDP SHALL be supported by both BSM and BCAST terminal.

Support of BSM Solicited Pull message delivery over SMS bearer is as follows:

- for BSM: OPTIONAL;
- for BCAST terminal: MANDATORY if Smartcard Profile using (U)SIM is supported, OPTIONAL otherwise.

When the SMS bearer is used, the BSM Solicited Pull message SHALL be encoded and transported like the message initiating a BSM Solicited Pull procedure over SMS bearer, specified in Section 6.6.2. Actually, the terminal can only distinguish between the two types of messages by looking at the value of MSK ID Key Group part (set to 1 or to other value).

6.6.4 EXT BCAST for LTKM

To include Smartcard Profile specific information in LTKMs that can not be supported by the MBMS MSK message, a new MIKEY Extension payload MAY be included in the LTKM. This payload is referred to as the EXT BCAST for LTKMs. The EXT BCAST for LTKMs is used to transport additional information governing the use of the SEK/PEK carried within the LTKM. The EXT BCAST for LTKMs enables the following functionalities:

- Subscription to live services
- Pay-Per-View access to a live event (PPV)
- Pay-Per-Time (PPT) access to a live service or recorded content, whereby the amount of time is governed by the number of TEKs that can be decrypted
- Unlimited playback of recorded content
- Pay-Per-Play (PPP) access to recorded content, whereby the maximum number of times the content can be played back is possible can be set by the service provider.
- Send tokens to be added to a purse
- Service/program termination for a user and SEK/PEK ID key deletion

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 10: Smartcard Profile LTKM Extensions and Supported Modes of Operation

Security policy extension	LIVE support	PLAYBACK support	Subscription / PPV support	PPP support	PPT support	Tokens support
0x00	X				X	X (live_ppt_purse)
0x01		X			X	X (playback_ppt_purse)
0x02	X				X	X (user_purse)
0x03		X			X	X (user_purse)
0x04	X		X			
0x05		X	X			
0x06	N/A	N/A	N/A	N/A	N/A	N/A
0x07		X		X		
0x08	X		X			X(user_purse)
0x09		X		X		X(user_purse)
0x0A	N/A	N/A	N/A	N/A	N/A	N/A
0x0B	N/A	N/A	N/A	N/A	N/A	N/A
0x0C	X				X	
0x0D		X			X	

The BSM and BSD/A SHALL support the use of the EXT BCAST for LTKMs. The terminal SHALL support the use of the EXT BCAST for LTKMs. A BCAST Smartcard SHALL support the use of the EXT BCAST for LTKMs and SHALL support the use of security policy extension 0x04.

As MBMS MIKEY implementations will ignore the EXT BCAST payload, if the BSM / BSD/A sends a LTKM to a MBMS only Smartcard, the LTKM MAY include the EXT BCAST payload, in which case the security_policy_ext_flag, consumption_reporting_flag, and access_criteria_flag SHALL be set to LTK_FLAG_FALSE. In all cases in which the BSM / BSD/A sends a LTKM to a MBMS only Smartcard, the KV data payload SHALL define the Key Validity interval for the SEK/PEK in terms of a specified interval of 16 bit TEK ID values.

In all cases in which the BSM / BSD/A sends a LTKM to a BCAST Smartcard, the LTKM SHALL include the EXT BCAST payload, the security_policy_ext_flag SHALL be set to LTK_FLAG_TRUE and the KV data payload SHALL define the Key Validity interval for the SEK/PEK in terms of a specified interval of 32 bit STKM Timestamp values.

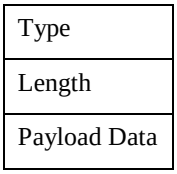
How the BSM / BSD/A determines the capabilities of the Smartcard that it addressing is implementation specific.

The EXT BCAST payload is an instance of the General Extension Payload for MIKEY defined in section 6.15 of [RFC3830] and reproduced below for convenience:

Table 11: Logical Structure of the MIKEY General Extension Payload

Next Payload

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.



For the EXT BCAST for LTKMs the MIKEY General Extension Payload fields SHALL be populated as defined below:

Next Payload (8 bits): This field SHALL be populated as defined in [RFC3830].

Type (8 bits): This field defines a new type for MIKEY in addition to the existing types for MIKEY. The new type is named "OMA BCAST STKM/LTKM MIKEY General Extension" and is assigned the value of 5.

Length (16-bits): This field SHALL be populated as defined in [RFC3830]. No change is required.

Payload Data (Variable length): The Subtype is equal to 1 and the Subtype specific data SHALL be populated with the Smartcard Profile LTKM Management Data as defined in Table 12.

Table 12: Format of Smartcard Profile LTKM Management Data

Smartcard Profile LTKM Management Data	Length (bits)	Type
LTKM_management_data() {		
protocol_version	**4**	**uimsbf**
security_policy_ext_flag	**1**	**bslbf**
consumption_reporting_flag	**1**	**bslbf**
access_criteria_flag	**1**	**uimsbf**
terminal_binding_flag	**1**	**bslbf**
if (security_policy_ext_flag == LTK_FLAG_TRUE) {		
security_policy_extension	**8**	**uimsbf**
purse_flag	**1**	**bslbf**
reserved_for_future_use	**7**	**uimsbf**
if security_policy_extension == 0x00 || 0x01 || 0x02 || 0x03|| 0x08 || 0x09) {		
cost_value	**16**	**uimsbf**
}		
if security_policy_extension == 0x0C) {		
add_flag	**1**	**bslbf**
keep_credit_flag	**1**	**bslbf**
number_TEKs	**22**	**uimsbf**
}		
if security_policy_extension == 0x0D) {		
add_flag	**1**	**bslbf**
number_TEKs	**23**	**uimsbf**
}		
if (security_policy_extension == 0x07) {		
add_flag	**1**	**bslbf**
number_playback	**7**	**Uimsbf**
}		
if (purse_flag == LTK_FLAG_TRUE) {		
purse_mode	**1**	**bslbf**
token_value	**31**	**uimsbf**
}		
}		
if (access_criteria_flag == LTK_FLAG_TRUE) {		
reserved_for_future_use	**8**	**bslbf**
number_of_access_criteria_descriptors	**8**	**uimsbf**
access_criteria_descriptor_loop() {		
access_criteria_descriptor()		
}		
}		
if (terminal_binding_flag == LTK_FLAG_TRUE) {		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

terminalBindingKeyID	**32**	**uimsbf**
permissionsIssuerURILength	**8**	**uimsbf**
permissionsIssuerURI	**8***permissionsIssuerURILength	**bslbf**
}		
if (consumption_reporting_flag == LTK_FLAG_TRUE) {		
security_policy_extension	**8**	**uimsbf**
}		
}		

6.6.4.1 Constant Values

LTK_FLAG_FALSE 0
LTK_FLAG_TRUE 1

6.6.4.2 Coding and Semantics of Attributes

protocol_version (4 bits): This field indicates the protocol version of this LTKM.

The terminal SHALL ignore messages that have a protocol_version number it doesn't support.

If the protocol version is set to 0x0 the format specified in this version of the specification SHALL be used. If set to anything else than 0x0, then the format is beyond the scope of this version of the specification.

security_policy_ext_flag (1 bit): This field indicates whether or not a security extension payload is carried in this LTKM.

LTK_FLAG_FALSE	Indicates no security extension payload is present and the MBMS security policy in section 6.5.3 "MSK processing" in [3GPP TS 33.246 v7] applies.
LTK_FLAG_TRUE	Indicates a security extension payload is present and that a Smartcard Profile specific security policy associated with the security_policy_extension applies.

Specifically, if the security_policy_ext_flag is equal to LTK_FLAG_TRUE, the counter in TS field in STKMs is used to detect replay attacks and facilitate key validity data check (both procedures associated with the TEK) while the TEK ID field of the EXT MBMS payload is used to detect the resending of the same TEK.

consumption_reporting_flag (1 bit): This field indicates whether or not a consumption reporting payload is carried in this LTKM. If the consumption_reporting_flag is equal to LTK_FLAG_TRUE, the security_policy_ext_flag and the terminal_binding_flag shall be set to LTK_FLAG_FALSE, the V bit in the common header of LTKM SHALL be set equal to 0, and a consumption reporting message (as defined in 6.6.7.8) SHALL be sent by the terminal to the BSM.

terminal_binding_flag (1 bit): This field indicates whether or not terminal binding applies for the STKM streams protected by the SEK or PEK transported in this LTKM. LTK_FLAG_FALSE indicates it is not used, LTK_FLAG_TRUE indicates it is used.

security_policy_extension (8 bits): This field indicates the security_policy_extension (SPE) to associate to the SEK/PEK contained in the LTKM. The following table describes the semantics of the different security_policy_extension values. The processing of the LTKM and STKM related to the security_policy extension is described in Sections 6.6.7 and 6.7.3, respectively.

In the following descriptions the permission associated to a SEK/PEK is only granted if the TS value in the received STKM is within the range of the key validity (KV) data associated to an instance of the SPE stored in the Smartcard.

All LTKM security policy extensions giving permissions are valid within a time window defined by the KV data.

A cryptoperiod corresponds to the lifetime of a TEK.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 13: security_policy_extension (SPE) Values

Value	Description
0x00	**Service Token Pay Per Time (PPT) LIVE** Permission to access a live service provided that there are enough tokens in the live_ppt_purse. These tokens are valid for a specific service. Tokens may be sent in the LTKM. For every cryptoperiod, the purse is decreased by "cost_value" tokens. If insufficient tokens are available, the user is unable to access the live service until additional tokens are obtained.
0x01	**Service Token Pay Per Time (PPT) PLAYBACK** Permission to access recorded content related to a service provided that there are enough tokens in the playback_ppt_purse. These tokens are valid for a specific service. Tokens may be sent in the LTKM. For every cryptoperiod, the purse is decreased by "cost_value" tokens. If insufficient tokens are available, the user is unable to access the recorded content until additional tokens are obtained.
0x02	**User Token Pay Per Time (PPT) LIVE** Permission to access live services, provided that there are enough tokens in the user_purse. These tokens are valid for all services. Tokens may be sent in the LTKM. For every cryptoperiod, the purse is decreased by "cost_value" tokens. If insufficient tokens are available, the user is unable to access the live service until additional tokens are obtained.
0x03	**User Token Pay Per Time (PPT) PLAYBACK** Permission to access recorded content related to services, provided that there are enough tokens in the user_purse. These tokens are valid for all services. Tokens may be sent in the LTKM. For every cryptoperiod, the purse is decreased by "cost_value" tokens. If insufficient tokens are available, the user is unable to access the recorded content until additional tokens are obtained.
0x04	**Subscription and Pay Per View (PPV) LIVE** Permission to allow access to a live service.
0x05	**Unlimited PLAYBACK** Permission to allow the unlimited playback of recorded content related to a channel**.**
0x06	**Reserved for future use**
0x07	**Pay Per Play (PPP) PLAYBACK** Permission to playback a piece of recorded content a set number of times. When a user accesses the recorded content they consume one playback "permission". Playback is no longer possible when there are no playback permissions left.
0x08	**User Token Pay Per View (PPV) LIVE** Permission to access a live PPV event, provided that there are enough tokens in the user_purse. These tokens are valid for all services. Tokens may be sent in the LTKM. When a user starts to access the service during the allowed period, he/she consumes a pre-defined number of tokens (the cost_value) from the user_purse. If insufficient tokens are available the user is unable to access the PPV event.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

0x09	**User Token Pay Per Play (PPP) PLAYBACK** Permission to playback a piece of recorded content provided that there are enough tokens in the user_purse. These tokens are valid for all services. Tokens may be sent in the LTKM. When a user accesses the piece of recorded content he/she consumes a pre-defined number of tokens (the cost_value) from the user_purse. If insufficient tokens are available the user is unable to access the recorded content.
0x0A	**Service/Program termination** Whatever the KV interval is, the Smartcard deletes the stored SEK/PEK and related key material corresponding to that SEK/PEK ID, for all SPEs and all KVs.
0x0B	**Reserved for future use**
0x0C	**Pay Per Time (PPT) LIVE** Permission to access a live service provided that the TEK counter is not zero. For every cryptoperiod, the TEK counter is decreased by one. If the TEK counter is equal to zero the user is unable to access the live channel. The initial value of the TEK counter is set by the LTKM. The keep_credit_flag allows unused TEK counters to be kept when the SEK/PEK changes.
0x0D	**Pay Per Time (PPT) PLAYBACK** Permission to access recorded content related to a service, provided that the TEK counter is not zero. For every cryptoperiod, the TEK counter is decreased by one. If the TEK counter is equal to zero the user is unable to access the live service. The initial value of the TEK counter is set by the LTKM.
0x0E …0x8F	**Reserved for future standardization**
0x90 …0xFF	**Reserved for proprietary implementation**

purse_flag (1 bit): This field indicates whether or not a purse extension is carried in this LTKM.

LTK_FLAG_TRUE	Indicates purse data is present and the Smartcard SHALL perform appropriate update of the purse as indicated by the purse_mode value.
LTK_FLAG_FALSE	Indicates no purse data is present.

access_control_flag (1bit): This field indicates whether or not an access_control_descriptor is carried in this LTKM. Server MAY support access_control_descriptor. In case server does not support access_control_descriptor, the access_control_flag SHALL be set to LTK_FLAG_FALSE.

LTK_FLAG_TRUE	Indicates that at least one access_control_descriptor is present in the LTKM.
LTK_FLAG_FALSE	Indicates that no access_control_descriptor is present.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

cost_value (16 bits): If the security_policy_extension is set to:

0x00	Indicates the number of tokens per TEK decrypted by the Smartcard to decrement from the live_ppt_purse.
0x01	Indicates the number of tokens per TEK decrypted by the Smartcard to decrement from the playback_ppt_purse.
0x02 or 0x03	Indicates the number of tokens per TEK decrypted by the Smartcard to decrement from the user_purse.
0x08 or 0x09	Indicates the number of tokens per playback to decrement from the user_purse.

add_flag (1 bit): This field indicates whether or not number_TEKs / number_playback is to be added to an existing TEK / playback counter.

LTK_FLAG_TRUE	Indicates that the number_TEKs / number_playback SHOULD be added to an existing TEK / playback counter.
LTK_FLAG_FALSE	Indicates that the number_TEKs / number_playback replaces an existing TEK / playback counter,

keep_credit_flag (1 bit): This field indicates whether or not the value of the TEK counter is to be kept by adding it to the next appropriate TEK counter when the key validity of the current SPE expires.

LTK_FLAG_TRUE	Indicates that the value of the TEK counter SHOULD be added to the next appropriate TEK counter, i.e. the credit is kept.
LTK_FLAG_FALSE	Indicates that the value of the TEKcounter SHOULD NOT be added to the next appropriate TEK counter, i.e. the credit is lost.

number_TEKs (22 or 23 bits): Indicates the number of TEKs that can be decrypted for the SEK/PEK. The associated counter is decreased by one for each TEK decrypted, until the counter reaches zero, when an error message is produced.

number_play_back (7 bits): Indicates the maximum number of times content recorded under a SEK/PEK can be played back.

purse__mode (1 bit): This field indicates the purse update mode.

Table 14: Purse Update Mode Indication

0x00	**Set mode:** The relevant purse SHALL be set to token_value: • The user_purse if the security_policy_extension is equal to 0x02, 0x03, 0x08 or 0x09. • The live_ppt_purse if the security_policy_extension is equal to 0x00. • The playback_ppt_purse if the security_policy_extension is equal to 0x01.
0x01	**Add mode:** The token_value SHALL be added to the relevant purse:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	• The user_purse if the security_policy_extension is equal to 0x02, 0x03, 0x08 or 0x09. • The live_ppt_purse if the security_policy_extension is equal to 0x00. • The playback_ppt_purse if the security_policy_extension is equal to 0x01.

In order to detect overflow in a purse when an update occurs using the Add mode, the following SHALL apply:

If the purse_flag is set equal to LTK_FLAG_TRUE and the purse_mode is set equal to 0x01, the V bit in the common header of LTKM SHALL be set and a verification message containing the status of the update SHALL be sent by the secure function according to Section 6.6.6.

token_value (31 bits): This field indicates the number of tokens to use in the update procedure toward the purse.

access_criteria_flag (1bit): This field indicates whether or not an access_criteria_descriptor is carried in this LTKM. LTK_FLAG_FALSE indicates that no access_criteria_descriptor is present, LTK_FLAG_TRUE indicates that at least one access_criteria_descriptor is present in the LTKM.

Server MAY support access_criteria_descriptor. In case server doesn't support access_criteria_descriptor, the access_criteria_flag SHALL be set to LTK_FLAG_FALSE.

number_of_access_criteria_descriptors (8bits): This field indicates the number of access_criteria_descriptors present in the LTKM.

access_criteria_descriptor_loop

Tag	8	uimsbf
Length	8	uimsbf
Value	8xlength	bit string

The access_criteria_descriptor_loop element provides an extension mechanism to allow the addition of new access_criteria_descriptor elements carried in LTKM in future versions of this specification. The secure function SHALL ignore access_criteria_descriptor elements, which it does not support. It is OPTIONAL for the BCAST Terminal to support access_criteria_descriptor. A single access_criteria_descriptor can carry one or more access criteria. Currently no access_criteria for LTKM is defined in this specification.

TerminalBindingKeyID (32 bits): This field contains the identifier of the Terminal Binding Key. See Section 12 for further details. This field is ignored by the USIM as it is used only by the terminal.

PermissionsIssuerURILength (8 bits): This field specifies the length in bytes of the Permission Issuer URI specified below. This field is ignored by the USIM as it is used only by the terminal.

PermissionsIssuerURI (Variable Length): This field is the URI of the Permission Isssuer that can be contacted to obtain the Terminal Binding Key. See Section 12 for further details. This field is ignored by the USIM as it is used only by the terminal.

6.6.5 Parental Control Message Structure and Processing

A MIKEY message for parental control is introduced. The parental control message SHALL be formatted as defined below:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 15: Logical Structure of the Parental Control Message

Common HDR
EXT BCAST
TS
MIKEY RAND
IDi
IDr
KEMAC

The terminal SHALL forward the parental control message to the secure function. Values in the message are as follows:

- In the Common HDR, if the V-bit is set to 1 to request, the secure function SHALL send a verification message, as described in Section 6.6.6.1.
- In the EXT BCAST, the parental_control_management_data are carried as defined in Section 6.6.5.1.
- TS, MIKEY RAND, IDi, IDr payloads SHALL be populated as defined in [3GPP TS 33.246 v7] for the MBMS MSK message.
- In the key sub-payload of the KEMAC, the value of the "Key data len" field (16 bits) SHALL be reset to 0 if no PINCODE is carried in this message, otherwise the value of the "Key data len" field SHALL be set to 16. The material to be encrypted is a PINCODE (64 bits) padded to 128 bits with 1 for the 64 least significant bits. As a convention, the most significant bit is on the left hand side of the string, i.e. the PINCODE part of the message (64 most significant bits) is constructed as specified for PIN and PIN2 in section 7 of [3GPP TS 31.101 v7]. The mechanism for encrypting the PINCODE is the mechanism used for encrypting the SEK/PEK in a LTKM. The key used for the AES engine is the key used for encrypting the SEK/PEK (derived from the SMK key and the RAND carried in the same message). In the Key data sub-payload, the type is 2 and there is no KV data. The MAC (included in the KEMAC) is computed over the full message as defined in RFC 3830 [9]. The key used in the MAC computation is the authentication key derived from SMK as described in RFC 3830 [9]. As an example, pincode "020579" is decoded as follows:

30	32	30	35	37	39	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF
Bit 0 (MSB)								(LSB) Bit 127							

6.6.5.1 EXT BCAST Parental Control

The EXT BCAST parental control data payload is an instance of the General Extension Payload for MIKEY defined in Section 6.15 of [RFC3830]. The Subtype is equal to 4 as defined in [RFC5410].

The Subtype specific data SHALL be populated as follows:

Table 16: Format of the Smartcard Profile parental_control Management Data

Smartcard Profile Parental Control Management Data	Length (bits)	Type
parental_control_management_data() {		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

reserved_for_future_use	2	
operation	1	
number_of_rating_types	5	uimsbf
for (i=0; i < number_of_rating_types; i++) {		
rating_type	8	uimsbf
level_granted	8	uimsbf
}		
}		

operation (1bit): This field specifies the type of operation performed on the list of rating_type/level_granted pairs values stored in the secure function. The following operations are defined:

- 0x0: Overwrite the list of rating_type/level_granted pairs already stored in the secure function with the list specified in the parental control message. If this operation is used with number_of_rating_types = 0 all stored rating_type/level_granted pairs SHALL be deleted.
- 0x1: Merge the list of pairs in pairs already stored in the secure function with the list specified in the parental control message. If a specific rating_type is already stored in the secure function, its level_granted SHALL be overwritten with the one specified in the parental control message. If this operation is used with number_of_rating_types = 0 all stored rating_type/level_granted pairs remain unmodified (which enables update of the PINCODE only).

number_of_rating_types (5bits): This field indicates the number of rating types transmitted within the descriptor.

rating_type (8bits): This field indicates the rating_type. Possible values are as specified in the OMA BCAST Parental Rating System Registry available at [OMNA].

level_granted (8 bits): This field is an integer defining the maximum authorized value with a coding that is dependent on the rating_type.

6.6.5.2 Parental Control Message Processing

When receiving the parental control message, if the secure function supports the enforcement of the parental control, it does the following:

- Compares the received Time Stamp field (TS) with the stored Parental Control Message replay detection counter. This replay detection counter is associated to the NAF_ID part of the SMK and then there is one replay detection counter per NAF.
- Extracts the list of rating_type/level_granted pairs. This list of pairs is a user specific setting and is associated to the SMK used to protect the parental control message. The secure function SHALL process the list depending on the operation specified in the message. Note: A default setting for rating_type/level_granted pairs specific to a service provider may be possible during the Smartcard manufacture.
- If the encrypted PINCODE is present in the KEMAC of the message, the secure function decrypts the PINCODE, unblocks the PIN if blocked, and replaces the current PINCODE value with the received value. The PINCODE SHALL be associated to the BCAST functionality (there is only one parental control PINCODE regardless of the BSM).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

If the secure function is located in the Smartcard, the command used to transmit the parental control message from the terminal to the Smartcard is the AUTHENTICATE Command in MBMS security context and MSK update mode. The response to the AUTHENTICATE command is as described in Appendix E.2.2.

- If the enforcement of the parental control is supported, the Smartcard SHALL return the new list of the rating_type/level_granted pairs. Additionally, the Smartcard SHALL include in this response message the status 0x0F, 0x10, or 0x11, if either PINCODE or rating_type/level_granted pair or both have been successfully changed. The terminal MAY then inform the user that the PINCODE has been changed and that the old PINCODE is no more usable.

- If the enforcement of the parental control access criteria is not supported, the response message includes the status 0x0E “Parental control not supported”.

6.6.6 LTKM Verification Message and Reporting Message Structure

Two types of messages can be sent by the secure function to the BSM in response to a received LTKM:

1. An LTKM Verification Message sent to confirm successful reception of an LTKM
2. An LTKM Reporting Message sent to do the following:
    a. report consumption of permissions in the secure function (see Section 6.6.7.8),
    b. or indicate an overflow of a counter or purse during LTKM processing (see Section 6.6.7.9) ,
    c. or indicate that a security_policy_extension value is unsupported (see Section 6.6.7.10).

The format of the LTKM Verification message and LTKM Reporting message is detailed in Section 6.6.6.1 and Section 6.6.6.2, respectively.

6.6.6.1 LTKM Verification Message

The LTKM verification message SHALL be constructed as defined in section 6.4.5.2 of [3GPP TS 33.246 v7].

6.6.6.2 LTKM Reporting Message Format

Table 17: Logical Structure of the LTKM Reporting Message

Common HDR
EXT BCAST
TS
Idr
V

The MAC included in the verification payload (V), shall be computed over both the initiator's and the responder's ID as well as the timestamp in addition to be computed over the response message as defined in RFC 3830 [RFC3830]. The key used in the MAC computation is the authentication key derived from SMK as described in RFC 3830 [RFC3830]. The EXT BCAST payload SHALL be formatted as defined in Section 6.6.6.3.

The terminal SHALL forward the LTKM Reporting messages received from the secure function to the BSM/Permissions issuer.

6.6.6.3 EXT BCAST Reporting Management Data

The EXT BCAST Reporting Management Data payload is an instance of the General Extension Payload for MIKEY defined in section 6.15 of [RFC3830]. The subtype is equal to 3 and the SubType specific data SHALL be populated as in Table 18. (Note that SubType payload 3 does not currently exist in RFC 3830 [RFC3830].) The EXT BCAST Reporting Management Data payload is included in an LTKM Reporting message. The parameters included in the payload are dependent on why the

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

LTKM Reporting message is being sent by the secure function (see Section 6.6.7.8, Section 6.6.7.9 and Section 6.6.7.10 for further details).

Table 18: Format of the Smartcard Profile Reporting Management Data

Smartcard Profile Reporting Management Data	Length (bits)	Type
Reporting_management_data() {		
consumption_reporting_flag	**1**	**bslbf**
overflow_flag	**1**	**bslbf**
unsupported_extension_flag	**1**	**bslbf**
not_found_flag	**1**	**bslbf**
reserved_for_future_use	**4**	**uimsbf**
if (consumption_reporting flag == LTK_FLAG_TRUE) {		
security_policy_extension	**8**	**uimsbf**
reserved_for_future_use	**8**	**uimsbf**
if (security_policy_extension == 0x00 || 0x01 || 0x02 || 0x03|| 0x08 || 0x09) {		
cost_value	**16**	**uimsbf**
reserved_for_future_use	**1**	**bslbf**
purse_value	**31**	**uimsbf**
}		
if (security_policy_extension == 0x07) {		
reserved_for_future_use	**1**	**bslbf**
playback_counter	**7**	**uimsbf**
}		
if (security_policy_extension == 0x0C) {		
reserved_for_future_use	**1**	**bslbf**
keep_credit_flag	**1**	**bslbf**
TEK_counter	**22**	**uimsbf**
}		
if security_policy_extension == 0x0D) {		
reserved_for_future_use	**1**	**bslbf**
TEK_counter	**23**	**uimsbf**
}		
}		
}		

consumption_reporting_flag: Indicates whether or not that the message includes consumption reporting data. If set to LTK_FLAG_TRUE, the message contains consumption reporting data. If set to LTK_FLAG_FALSE, the message does not contain consumption reporting data.

overflow_flag: Indicates whether or not that the message includes an indication that an overflow occurred during the processing of an LTKM..If set to LTK_FLAG_TRUE, the message contains overflow information. If set to LTK_FLAG_FALSE, the message does not contain overflow information.

unsupported_extension_flag: Indicates that the secure function does not support the SPE sent in the LTKM if set to LTK_FLAG_TRUE. If set to LTK_FLAG_FALSE the message indicates that the SPE is supported by the secure function.

not_found_flag: Indicates whether or not the SEK/PEK ID, SPE and KV tuple contained in the received LTKM exist in the secure function. If set to LTKM_FLAG_TRUE, the SEK/PEK ID, SPE and KV tuple does not exist in the secure function. If set to LTKM_FLAG_FALSE, the SEK/PEK ID, SPE and KV tuple does exist in the secure function.

security_policy_extension The SPE value received in the LTKM message that triggered the secure function to send the LTKM Reporting Message.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

keep_credit_flag: It is the keep_credit_flag value specific to the LTKM identified by the SEK/PEK ID, KV and the SPE.

cost_value: The cost_value associated to the instance of the SPE stored in the secure function identified by the SEK/PEK ID, KV and the SPE in the received LTKM.

TEK_counter: The value of the TEK counter associated to the instance of the SPE stored in the secure function identified by the SEK/PEK ID, KV and the SPE in the received LTKM.

playback_counter: The value of the playback counter associated to the instance of the SPE stored in the secure function identified by the SEK/PEK ID, KV and the SPE in the received LTKM.

purse_value: The number of tokens remaining in the relevant purse, i.e. the value of the user purse if the SPE is 0x02,0x03,0x08 or 0x09, or the value of the live_ppt_purse if the SPE value is 0x00 or the value of the playback_ppt_purse if the SPE value is 0x01.

6.6.7 OMA BCAST LTKM Processing

LTKMs are processed by a secure function located on either the Smartcard or terminal. Following MBMS [3GPP TS 33.246 v7], where GBA_U is used the secure function is located on the Smartcard, and where GBA_ME is used the secure function is located on the terminal.

The following sections define the processing of LTKMs in BCAST.

6.6.7.1 LTKM Terminal Processing

When a MIKEY message indicating MSK/SEK delivery, i.e. an LTKM, arrives at the Terminal, the message SHALL be processed as described below.

If the secure function is located on the Terminal, the Terminal SHALL perform the the LTKM replay detection check as described in Section 6.6.7.4. If the check ends in success, the Terminal forwards the LTKM to the secure function for further processing. The secure function processes the LTKM as described in the following sections.

If the secure function is located on the Smartcard, in order to support the use of both MBMS only and BCAST Smartcards, the Terminal SHALL process a LTKM received over the UDP port 4359 (i.e. the UDP port defined for BCAST) as defined below:

When the Terminal is paired with an MBMS only Smartcard, the following occurs:

- If the LTKM does not contain an EXT BCAST payload, the Terminal SHALL:
  - perform the LTKM replay detection check as described in Section 6.6.7.4;
  - if the LTKM replay detection check ends in success, then the Terminal forwards the message to the Smartcard;
- If the LTKM contains an EXT BCAST payload and the security_policy_ext_flag and the consumption_reporting_flag and the access_criteria flag are set to LTK_FLAG_FALSE, the Terminal SHALL:
  - perform the LTKM replay detection check as described in Section 6.6.7.4;
  - if the LTKM replay detection check ends in success, then the Terminal forwards the message to the Smartcard – the EXT BCAST payload is being used to deliver data only intended for use by the Terminal (e.g. information relating to the Terminal Binding Key; the terminalBindingKeyID, permissionsIssuerURILength and permissionsIssuerURI) that will be ignored by the MBMS only Smartcard;
- If the LTKM contains an EXT BCAST payload and the security_policy_ext_flag or the consumption_reporting_flag or the access_criteria flag are set to LTK_FLAG_TRUE, then the Terminal SHALL NOT forward the message to the Smartcard – this message cannot be processed by the MBMS only card and has been incorrectly sent by the BSM.

If the Terminal is paired with a BCAST Smartcard, the following occurs:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- If the LTKM does not contain an EXT BCAST payload, then the Terminal SHALL NOT forward the message to the Smartcard – this message has been incorrectly sent by the BSM;
- If the LTKM contains an EXT BCAST payload and the security_policy_ext_flag, consumption_reporting_flag and access_criteria flag are set to LTK_FLAG_FALSE , then the Terminal SHALL NOT forward the message to the Smartcard – this message has been incorrectly sent by the BSM;
- If the LTKM contains an EXT BCAST payload and the security_policy_ext_flag or the consumption_reporting_flag or access_criteria flag are set to LTK_FLAG_TRUE, then the Terminal SHALL forward the message to the Smartcard.

Note that, unlike MBMS only Smartcards, BCAST Smartcards perform the LTKM replay detection check (see Section 6.6.7.4). Therefore when a Terminal is paired with a BCAST Smartcard the Terminal does not perform the LTKM replay detection check.

In all the above cases in which the LTKM is forwarded to the secure function, the Security Policy payload is stored temporarily in the Terminal if it was present.

If the secure fucntion indicates to the Terminal that it has successfully processed the LTKM, the temporarily stored Security Policy payload is taken into use. Otherwise the Security Policy payload is deleted.

If the LTKM indicated a BM-SC solicited pull procedure or a BSM solicited pull procedure to initiate the registration procedure, the Terminal SHALL behave as described in Sections 6.6 and 6.6.3 respectively.

The formating rules defined for LTKMs in Section 6.6.4 require that the BSM / BSD/A format the LTKM appropriately for the type of Smartcard (MBMS only or BCAST) to which they are sending the LTKM.

The Terminal filtering rules defined above SHALL apply only to MIKEY messages received on the UDP port reserved for BCAST, i.e. UDP port 4359. MIKEY messages addressed to the MIKEY UDP port 2269 are not subject to Terminal filtering. This allows a BCAST Smartcard to work as part of an MBMS security solution as defined in [3GPP TS 33.246 v7].

A secure function SHALL be able to manage different LTKMs with the same SEK/PEK ID but with different Key Validity (KV) intervals or different security_policy_extensions values (SPEs).

Details on how parameters relating to the LTKM are to be updated when receiving an STKM are provided in the STKM processing Section 6.7.3.

6.6.7.2 Initial LTKM Processing in the Smartcard

When a BCAST Smartcard receives a MIKEY message, the Smartcard SHALL first determine the type of message by examining the MIKEY General Extension payload(s). If the MIKEY message contains a General Extension payload of Type 3 (Key ID Information), i.e. the EXT MBMS payload defined for MBMS in [RFC4563], the Smartcard SHALL inspect the contents of this payload. If the payload indicates delivery of an MSK/SEK, the Smartcard SHALL checks the MIKEY message for the presence of a General Extension payload of Type 5, i.e. the EXT BCAST payload defined for BCAST in [RFC5410]:

- If the EXT BCAST payload is present, then the following occurs:
  - If the security_policy_extenstion flag or consumption_reporting flag or access_criteria_flag are set to LTK_FLAG_TRUE, the selected Smartcard application SHALL process the LTKM as defined in the following sections of this document;
  - If the security_policy_extenstion flag, consumption_reporting_flag and access_criteria_flag are set to LTK_FLAG_FALSE, the selected Smartcard application SHALL abort the processing of the MIKEY message and send an error message to the Terminal. The error message returned by the Smartcard SHALL take the form of the status word ‘6A80’ (Incorrect parameters in the data field). This case can not occur if the BSM and Terminal are correctly implemented.
- If the EXT BCAST payload is not present, then the MIKEY message is an MBMS MSK message. If the selected Smartcard application supports MBMS processing, then the Smartcard application (e.g. selected application is USIM) SHALL process the message as defined in [3GPP TS 33.246 v7]. If the Smartcard application does not support

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

MBMS processing (e.g. selected application is BSIM), then it SHALL abort the processing of the MIKEY message and send an error message to the terminal. The error message returned by the Smartcard SHALL take the form of the status word '9864' (Authentication error, security context not supported). Note: It is mandatory for the USIM application on BCAST Smartcard to support MBMS processing as defined in Section 6.2.

Note: The above processing is required to ensure that a 3GPP BCAST Smartcard can work in a pure MBMS system, i.e. where Service Protection is implemented according to [3GPP TS 33.246 v7].

6.6.7.3 LTKM Message Validation

The secure function SHALL perform the message validation using the SMK (MUK) as described in [3GPP TS 31.102 v7] (including the update of the EF_{MUK} and the check of the usage of the last successfully used MUK). If the message validation is successful, then the LTKM replay detection procedure shall be executed.

6.6.7.4 LTKM Replay Detection

The LTKM replay detection check SHALL compare the received Time Stamp field (TS) with the stored LTKM replay detection counter associated with the given SMK (i.e. TS stored in the record associated to the SMK in the EF_{MUK}).

The LTKM replay detection check SHALL be performed by the Terminal unless the Terminal is paired with a BCAST Smartcard, in which case the Smartcard SHALL perform the check.

Success	If the received TS is greater than the stored LTKM replay detection counter value, the LTKM replay detection check ends in success and the stored LTKM replay detection counter SHALL be set to the received TS value. If the LTKM replay detection check ends in success, the secure function SHALL check whether it has already stored the SEK/PEK ID, i.e. whether it has previously successfully processed an LTKM containing the same SEK ID. If the SEK/PEK ID is not present the secure function SHALL set the STKM replay detection counter associated to the SEK/PEK ID to 0. If the SEK/PEK ID is present the secure function SHALL NOT update the STKM replay detection counter.
Failure	If the received TS is equal or lower than the stored LTKM replay counter value, then LTKM replay detection check ends in failure. If the LTKM replay detection check is being performed by the secure function, in the case of failure, then the secure function SHALL return a failure message to the terminal. If the secure function is located on the Smartcard, then the returned failure message SHALL take the form of the status word '9862' (Authentication error – incorrect MAC).

Note: Less than or equal is to be taken in the meaning of RFC 1982 [RFC1982]. If the less than or equal relation is undefined in the sense of RFC 1982 [RFC1982], the message SHOULD be considered as being replayed and shall be discarded.

As described for MBMS in [3GPP TS 31.102 v7], the USIM stores the last successfully used MUK along with its MUK ID for further use.

6.6.7.5 LTKM Controlled SEK/PEK and SPE Deletion

If the Key Validity data in the LTKM is invalid, i.e. the lower bound (“TS low”) is greater than the upper bound (‘TS High”), then the following applies:

- If the TS low = 0xFFFFFFFF and the TS High = 0, then the secure function SHALL delete all instances of the SPE and their associated data (e.g. cost_value, TEK_counter, etc.) that match the SEK/PEK ID and SPE value in the received LTKM.

- Otherwise, i.e. if the TS low is not 0xFFFFFFFF or the TS High is not 0, then the secure function SHALL delete the instance (if any) of the SPE having the opposite Key Validity data (High/Low inversed) and its associated data (e.g. cost_value, TEK_counter, etc.) that match the SEK/PEK ID and SPE value in the received LTKM.

The Smartcard SHALL support both aforementioned methods. The BSM SHALL support the first method and MAY support the second. Following the deletion of one or more SPEs (and associated data), the secure function SHALL delete the SEK/PEK (and related data) if no other SPE are associated to the SEK/PEK ID.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.6.7.6 LTKM Processing based on security_policy_extension (SPE)

Further details on the parameters used below can be found in Section 6.6.7.13 and Section 6.6.7.14 below.

SPE = 0x00 (Token PPT Live)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), SEK/PEK ID, the KV data, and the SPE and the cost_value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

live_ppt_purse	If purse_flag is set to LTK_FLAG_TRUE, the secure function SHALL update the live_ppt_purse with the token_value according to the received purse_mode value: • If the purse_mode is set to LTK_FLAG_FALSE, the live_ppt_purse SHALL be set to token_value. • If the purse_mode is set to LTK_FLAG_TRUE, the token_value SHALL be added to the live_ppt_purse.
live_ppt_purse overflow	If an overflow occurs on the live_ppt_purse during this update (live_ppt_purse > 0x7FFFFFFF), then: • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x01 (Token PPT Playback)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data, and the SPE and the cost_value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

playback_ppt_purse	The token_value is stored in the playback_ppt_purse associated to the SEK/PEK key group and SPE. If purse_flag is set to LTK_FLAG_TRUE, the secure function SHALL update the playback_ppt_purse with the token_value according to the received purse_mode value: • If the purse_mode is set to LTK_FLAG_FALSE, the playback_ppt_purse SHALL be set to token_value. • If the purse_mode is set to LTK_FLAG_TRUE, the token_value SHALL be added to the playback_ppt_purse.
playback_ppt_purse overflow	If an overflow occurs on the playback_ppt_purse during this update (playback_ppt_purse > 0x7FFFFFFF), then: • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x02 or 0x03 (User Token PPT)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data, the SPE and the cost_value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

user_purse	If purse_flag is set to LTK_FLAG_TRUE, the secure function SHALL update the user_purse with the token_value according to the received purse_mode value: • If the purse_mode is set to LTK_FLAG_FALSE, the user_purse SHALL be set to token_value.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

	• If the purse_mode is set to LTK_FLAG_TRUE, the token_value SHALL be added to the user_purse.
user_purse overflow	If an overflow occurs on the purse during this update (user purse > 0x7FFFFFFF), then: • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x04 or 0x05 (Subscription/PPV)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data and the SPE. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 00x07 (PPP Playback)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data and the SPE. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

current_TS_counter	The current_TS_counter is initialised to the LTKM validity "TS high".
playback counter	If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_FALSE, the playback counter SHALL be set to number_playback. If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_TRUE, the number_playback SHALL be added to the playback counter. If the KV is new for the SEK/PEK ID and SPE, the playback counter SHALL be set to number_playback.
playback counter overflow	If an overflow occurs on the playback counter during this update (playback counter > 0x7F), then: • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x08 or 0x09 (User Token PPV / PPP)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data, the SPE, and the cost_value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

current_TS_counter	The current_TS_counter is initialised to the LTKM validity "TS high".
user_purse	If purse_flag is set to LTK_FLAG_TRUE, the secure function then updates the user_purse with the token_value according to the received purse_mode value: • If the purse_mode is set to LTK_FLAG_FALSE, the user_purse SHALL be set to token_value. • If the purse_mode is set to LTK_FLAG_TRUE, the token_value SHALL be added to the user_purse.
user_purse	If an overflow occurs on the purse during this update (user_purse > 0x7FFFFFFF), then:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

overflow	• execute Section 6.6.7.9.

Details on the LTKM parameters updates when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x0A (Key deletion)

Whatever the KV interval is, the Smartcard SHALL delete all the previously stored SEK/PEK with the same SEK/PEK ID and their related key material (i.e, KV data, SPEs, playback counter, TEK number and cost_value).

No STKM is associated to this LTKM.

SPE = 0x0C (PPT Live)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the SEK/PEK ID, the KV data, the keep_credit_flag value and the SPE value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

TEK counter	If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_FALSE, the TEK counter SHALL be set to number_TEKs. If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_TRUE, the number_TEKs SHALL be added to the TEK counter. If the KV is new for the SEK/PEK ID and SPE, the TEK counter SHALL be set to number_TEKs.
TEK counter overflow	If an overflow occurs on the TEK counter during this update (TEK counter > 0x3FFFFF), • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processing Section 6.7.3.

SPE = 0x0D (PPT Playback)

The secure function SHALL store the received SEK/PEK (if the KEMAC Key Data sub-payload is present), the KV data, and the SPE value. (See the Storage of SEK/PEK and associated data in the secure function paragraph below.)

TEK counter	If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_FALSE, the TEK counter SHALL be set to number_TEKs. If the same SPE with the same SEK/PEK ID and KV exists and the add_flag is set to LTK_FLAG_TRUE, the number_TEKs SHALL be added to the TEK counter. If the KV is new for the SEK/PEK ID and SPE, the TEK counter SHALL be set to number_TEKs.
TEK counter overflow	If an overflow occurs on the TEK counter during this update (TEK counter > 0x7FFFFF), • execute Section 6.6.7.9.

Details on the LTKM parameters updated when receiving an STKM are given in the STKM processingSection 6.7.3.

Storage of SEK/PEK and associated data in the secure function

The SEK/PEK and any relevant associated data SHALL be stored within the secure function. Where the secure function is located on the Smartcard, the number of SEKs/PEKs that the Smartcard is able to store SHALL be defined at the pre-issuance of the card by the service provider. The maximum number of SEKs/PEKs that the Smartcard is able to store with the UsedForRecording flag set to LTK_FLAG_TRUE, i.e. that are required for access to protected recordings, SHOULD also be

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

defined at the pre-issuance of the card by the service provider. This ensures that a certain number of SEKs/PEKs can always be stored for access to live sevices.

In the case that there is not enough memory available in the secure function to store the information resulting from the processing of the LTKM, an error message SHALL be returned to the terminal. If the secure function is located on the Smartcard, the secure function SHALL abandon the function and return the status word '9866' (Authentication error, no available memory space) to the terminal. In this case, the terminal MAY run a SEK/PEK Deletion procedure (see Section 6.6.7.12) to free memory before re-sending the LTKM to the Smartcard.

6.6.7.7 LTKM Verification Message Request

If the V-bit in the HDR field of the received LTKM is set equal to 1, then the secure function SHALL produce a LTKM Verification message with the format as described in Section 6.6.6.1, unless a LTKM reporting message SHALL be sent to report a Counter or Purse Overflow as described in Section 6.6.7.9 or to report Unsupported SPE values as described in Section 6.6.7.10. In these cases of overflow or error in the SPE value, the LTKM reporting message is sent instead of the verification message.

6.6.7.8 Reporting Consumption using the LTKM Reporting Message

If an LTKM contains a consumption_reporting_flag set equal to LTK_FLAG_TRUE, the Verification bit of the MIKEY message SHALL be set equal to 0. If the LTKM consumption_reporting flag is equal to LTK_FLAG_TRUE, the secure function SHOULD try and find a matching SEK/PEK ID, KV and SPE.

The SEK/PEK ID, KV and the SPE of the received LTKM are used to identify a previously received LTKM. The stored data associated to the SEK/PEK ID, KV and the SPE values are sent back in the Reporting Message.

If successful, the secure function SHALL send a LTKM Reporting message as described in Section 6.6.6.2, with the following parameters:

- consumption_reporting_flag set to LTK_FLAG_TRUE
- overflow_flag set to LTK_FLAG_FALSE
- unsupported_extension_flag set to LTK_FLAG_FALSE
- not_found_flag set to LTK_FLAG_FALSE
- security_policy_extension set to the SPE of the LTKM requesting the message
- relevant SPE-specific parameters

If no matching SEK/PEK ID, KV and SPE is found the secure function SHALL send a LTKM Reporting message as described in Section 6.6.6.2, with the following parameters:

- consumption_reporting_flag set to LTK_FLAG_FALSE
- overflow_flag set to LTK_FLAG_FALSE
- unsupported_extension_flag set to LTK_FLAG_FALSE
- not_found_flag set to LTK_FLAG_TRUE

This message exchange allows the server to retrieve the relevant stored parameters for the given SPE value (as defined in Table 18) from the secure function, without updating the SEK/PEK.

6.6.7.9 Reporting a Counter or Purse Overflow during LTKM Processing using the LTKM Reporting Message

During LTKM processing counters or purses can overflow due to the addition of a value sent in the LTKM (e.g. the counter/purse value may become larger than the maximum value that can be stored when the incoming value is added). When this arises; the following SHALL apply:

- the counter or purse that has overflowed SHALL remain unchanged, and

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- an LTKM Reporting message SHALL be sent with the following parameters:
    - consumption_reporting_flag set to LTK_FLAG_TRUE
    - overflow_flag set to LTK_FLAG_TRUE
    - security_policy_extension set to the SPE value of the LTKM that caused the overflow
    - relevant SPE-specific parameters (as defined in Table 4: Format of the Smartcard Profile Reporting Management Data), which includes the unchanged parameter for which there was an overflow (e.g. for SPE 0x0C or 0x0D if the TEK counter overflows due to LTKM processing, the value of the TEK counter prior to LTKM processing and overflow is sent)

6.6.7.10 Reporting Unsupported SPE Values using the LTKM Reporting Message

If an LTKM is received for which the SPE is NOT supported by the secure function, the secure function SHALL:

- return the status code 'security policy extension not supported' in the MBMS operation response Data Object
- send a LTKM-Reporting message as described in Section 6.6.6.2. with the following parameters:
    - consumption_reporting_flag set to LTK_FLAG_FALSE
    - overflow_flag set to LTK_FLAG_FALSE
    - unsupported_extension_flag set to LTK_FLAG_TRUE
    - not_found_flag set to LTK_FLAG_FALSE

There SHALL be no further processing of the LTKM containing the unsupported SPE.

6.6.7.11 Terminal_binding_flag

After the successful processing of an LTKM by the secure function (no integrity error due to integrity or validation or bootstrapping failure), if the terminal_binding_flag is set to LTK_FLAG_TRUE, the terminal SHALL store the TerminalBindingKeyID and the PermissionsIssuerURI.

6.6.7.12 SPE Deletion by the Terminal

If the secure function is located on the Smartcard, and the Smartcard is an MBMS only Smartcard the policy of deleting SEK/PEK records to free up space in the file in EF_{MSK} SHALL be controlled by the Terminal using the Authenticate Command in MSK Deletion Mode defined in [3GPP TS 31.102 v7], e.g. the Terminal SHOULD delete any SEKs/PEKs that are no longer needed. How the terminal decides which SEKs/PEKs are no longer needed is implementation specific.

If the secure function is located on the Smartcard, and the Smartcard is a BCAST Smartcard, the management (deletion) of the SEKs/PEKs related to recorded content is described here after.

The terminal SHOULD delete any SPE that are no longer needed in order to free up storage space in the Smartcard for new SEKs/PEKs or SPE. The terminal SHALL control the deletion of SPE stored on the Smartcard using the Authenticate command for the MBMS security context in OMA BCAST operation and in SPE Deletion Mode (AppendixE: E.2.3.1). This command is used to delete SPEs that are not used for recording.

The terminal SHALL send the Authenticate command for the MBMS security context in OMA BCAST operation and in Recording Deletion Mode (AppendixE: E.2.3.2) when it erases a piece of a recorded content (received through streaming or download) protected by one or more SEKs/PEKs on the Smartcard. On reception of this command the secure function, SHALL delete the content identifier stored in the smartcard and its association to the flagged SPEs and remove the UsedForRecording flag associated to the SPE if no more content identifier is linked to this SPE, thereby indicating that the SPE is no longer required for the playback of recorded content (see Section 8.5).

When the secure function removes the UsedForRecording flag it SHALL NOT erase the corresponding SPE. PLAYBACK SPEs with their UsedForRecording flag set to LTK_FLAG_FALSE will be deleted by the secure function when it detects that this SPE is no more valid (see deletion of expired PLAYBACK security policy extensions and SEK/PEK in Section 6.7.3.10) or by a new Authenticate command in SPE Deletion Mode sent by the terminal to the Smartcard on this SPE.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

To discover which SPEs are stored in the Smartcard, the terminal SHALL use the OMA BCAST command in SPE audit mode (AppendixE: E.3.2). The terminal MAY use the returned information, along with other local information, to determine whether or not any SPEs should be deleted.

6.6.7.13 Association between Parameters and IDs Stored in Secure Function

The following table gives the association between Smartcard Profile parameters used in the processing of LTKM or STKM and identifiers stored on the Smartcard, i.e. it indicates which parameters can be linked to which key identifiers (or identifier sub-parts).

Table 19: Association between Smartcard Profile Parameters and Key Identifiers

	SEK/PEK ID, KV and SPE	**SEK/PEK ID key group part and SPE**	**SEK/PEK ID**	**SMK ID**	**NAF ID part of SMK ID**
SMK				X	
SEK/PEK			X		
Key validity data (STKM TS low & TS high)	X				
SPE			X		
cost_value	0x00, 0x01, 0x02, 0x03, 0x08, 0x09				
playback counter	0x07				
kept TEK counter		0x0C			
TEK counter	0x0C, 0x0D				
LTKM replay detection counter				X	
STKM replay detection counter			X		
current_TS_counter	0x07, 0x08, 0x09				
user_purse					0x02, 0x03, 0x08, 0x09
live_ppt_purse		0x00			
playback_ppt_purse		0x01			
Parental Control Message replay detection counter					X

If the secure function is located on the terminal, or the secure function is located on the Smartcard and the Smartcard is BCAST Smartcard, it SHOULD be possible to store a variable number of SEK/PEK per Key Domain ID/Key Group ID pair in the Smartcard.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6.6.7.14 Implementation Notes

- The first STKM sent by the network (BSD/A) for the associated SEK/PEK SHALL contain a timestamp (TS) value equal to or greater than "TS Low" + 1.
- When the LTKM contains an EXT BCAST payload in which the SPE flag is set to LTK_FLAG_TRUE, the Key Validity Data subfield in the KEMAC payload in the LTKM SHALL define the Key Validity interval for SEK/PEK in terms of STKM TIMESTAMP interval:

  From (32-bits): Lower limit of Timestamp ("TS low")

  To (32-bits): Upper limit of Timestamp ("TS high")

- When the LTKM does not contain an EXT BCAST payload or the LTKM contains an EXT BCAST payload in which the SPE flag is set to LTK_FLAG_FALSE, the Key Validity Data subfield in the KEMAC payload in the LTKM SHALL define the Key Validity interval for SEK/PEK in terms of TEK ID interval (i.e. as defined in [3GPP TS 33.246 v7]):

  From (16-bits): Lower limit of TEK ID

  To (16-bits): Upper limit of TEK ID

- Note that the case in which the LTKM contains an EXT BCAST payload in which the SPE flag is set to LTK_FLAG_FALSE is allowed to enable the delivery of information relating to the Terminal Binding Key (TBK) to a Terminal paired with an MBMS only Smartcard.
- To enable the use of 32 bit Timestamps within the KV data of the LTKM and 16 bit TEK IDs within STKMs, it SHALL be possible to re-use a TEK ID value within a set of STKMs protected by a specific SEK. TEK ID management SHOULD ensure that a Terminal never has two STKMs containing the same TEK ID for the same content stream at the same time.
- There is one internal STKM replay detection counter per SEK/PEK ID to support replay detection for STKM delivery.
- To avoid security failures during the STKM key validity data and replay detection checks, the Timestamp field (TS) in STKM associated with a SEK/PEK key group SHALL only be reset when the SEK/PEK is updated.
- The secure function SHALL be able to associate more than one SPE (and associated data) to a SEK/PEK ID. To identify uniquely the SPE to use during STKM or LTKM processing, the secure function SHALL the tuple: SEK/PEK ID, SPE value and KV data. Each instance of a SPE is associated to it its own associated data. Table 3 shows which stored data can be associated to which SPE.
- A separate playback counter is used for each SEK/PEK ID, KV and SPE (0x07) tuple. The playback counter is used for PPV without tokens and corresponds to the available number of plays left.
- A separate TEK counter is used for each SEK/PEK ID, KV and SPE (0x0C or 0x0D) tuple. The TEK counter is used for PPT without tokens and corresponds to the available number of TEKs left.
- A kept TEK counter is used for each SEK/PEK ID key group associated to a SPE 0x0C. It is used to allow unused TEKs to be brought forward.
- A user purse is associated to the NAF ID of the BSM, i.e. the NAF-ID part of an SMK ID stored in the file EF_{MUK}. The user_purse corresponds to the available number of tokens left that can be used by SPE 0x02, 0x03, 0x08 and 0x09.
- A single LTKM replay detection counter is used to keep track of the Timestamp (TS) of the latest valid LTKM received for a given SMK. The LTKM replay detection for each SMK is stored in the file EF_{MUK}.
- A current_TS_counter is used for SPEs 0x07, 0x08 and 0x09 to detect whether or not a "playback" has occurred. This counter is local to a SEK/PEK ID, KV and a relevant SPE.

6.6.8 Purses

The use of certain values of SPE requires sufficient credit to be available in a purse stored on the Smartcard.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

A purse stores tokens that are consumed when the permission defined by a SPE is used. The number of tokens that are consumed when the SPE is used is defined by the cost_value associated to the instance of the permission.

The Smartcard profile defines three different purses:

live_ppt_purse

The live_ppt_purse used by the SPE 0x00. It is associated to the SEK/PEK key group. There is only once instance of the live_ppt_purse per SEK key group. Tokens in the live_ppt_purse can be consumed by any instances of SPE value equal to 0x00 and belonging to the same SEK key group.

playback_ppt_purse

The playback_ppt_purse is used by the SPE 0x01. There is only once instance of the playback_ppt_purse per SEK/PEK key group. Tokens in the playback_ppt_purse can be consumed by any instances of the SPE with value equal to 0x01 and belonging to the same SEK key group.

user_purse

The user_purse is used by the SPEs 0x02, 0x03, 0x08 and 0x09. It is associated to the NAF Id of the BSM (i.e. the IDi value transmitted in the LTKM and stored on the Smartcard in the file EF_{MUK} as MUK IDi [3GPP TS 31.102 v7]). There is only one instance of the user_purse per NAF ID. Tokens in the user_purse can be consumed by any instance of the security_policy_extension with value equal to 0x02, 0x03, 0x08 or 0x09 protected by an SMK associated to the same NAF ID.

Note: For the live_ppt_purse and the playback_ppt_purse descriptions above it is assumed that the SEK/PEK key group belongs to the same SMK, i.e. if the SEK/PEK key group is the same but the SMK is different the live/playback_ppt_purse is also different.

6.6.8.1 Updating a Purse Balance

Identifying the correct purse to update

If the the purse_flag in the LTKM is set to LTK_FLAG_TRUE, the LTKM contains an update to a purse. To identify the relevant purse to update the secure function SHALL:

- Identify the SMK used to protect the LTKM and;
- If the security_policy_extension is 0x02, 0x03, 0x08 or 0x09, update the user_purse associated to the SMK;
- If the security_policy extension is 0x00, update the live_ppt_purse associated to the identified SMK and the SEK key group transmitted in the LTKM;
- If the security_policy extension is 0x01, update the playback_ppt_purse associated to the identified SMK and the SEK key group transmitted in the LTKM.

Updating the purse balance

The balance of a purse SHALL be updated when an LTKM is received in which the:

- security_policy_ext_flag is set to LTK_FLAG_TRUE;
- AND the a security-policy-extension is equal to 0x00, 0x01, 0x02, 0x03, 0x08 or 0x09
- AND the purse_flag is set to LTK_FLAG_TRUE;

In this case the relevant purse SHALL be updated according to the received purse_mode, i.e. the number of tokens indicated in the token_value will be:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- added to the balance of the relevant purse, if the purse_mode is 0x01;
- OR used to set the purse if the purse_mode is 0x00

6.7 Layer 3: Short Term Key Message - STKM

The table below shows the MIKEY message format used for STKMs in the Smartcard Profile. The message structure SHALL be identical to the MIKEY message used by MBMS to deliver the MBMS Traffic Key (MTK), as defined by [3GPP TS 33.246 v7], with the addition of the EXT BCAST payload. The EXT BCAST payload is described in Section 6.6.4.

Table 20: Logical Structure of the MIKEY Message Used

Common HDR
EXT MBMS
EXT BCAST
TS
KEMAC

The EXT MBMS payload, depicted in Table 21, is defined in section 6.4.4 of [3GPP TS 33.246 v7] and reproduced below for convenience.

Table 21: EXT MBMS Used within the MBMS MTK Message

Key Domain ID sub-payload
Key Type ID sub-payload (MSKID)
Key Type ID sub-payload (MTK ID)

All fields within the STKM SHALL be populated as defined in [3GPP TS 33.246 v7] for the MBMS MTK message, with the exception of the EXT BCAST payload. Mappings are as described in Section 6.2, i.e. SEK/PEK ID is mapped to MSK ID and SEK/PEK is mapped to MSK, TEK ID is mapped MTK ID and TEK is mapped to MTK.

Each STKM stream MUST only be secured using a single SEK/PEK. In some cases multiple STKM streams can deliver the same TEKs secured by different SEKs/PEKs. The Terminal MUST use the Service Guide (SG) to locate the relevant STKM stream for the encrypted traffic stream it needs to decrypt.

Each STKM SHALL be encapsulated in exactly 1 UDP packet. One UDP packet only contains at most one STKM.

The EXT BCAST payload SHALL be populated as defined in Section 6.7.2.

6.7.1 STKM Related Terminology

In the sections below, the following terms apply for STKMs:

Resending check: Resending of the same STKM/TEK SHALL be detected by the terminal using the TEK_ID (MTK ID) field of the MBMS EXT payload. See Section 6.7.3.2 for details.

STKM replay detection processing or verification: This procedure is used to detect replay attempts for STKMs. This procedure operates by comparing the TS field in the STKM with the corresponding STKM replay detection counter in the secure function.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

STKM replay detection (or freshness failure): This occurs when the STKM contains a TS field value which is less than or equal to the current STKM replay detection counter.

Key Validity Data check: The Key Validity Data check verifies that the SEK/PEK key is still valid. This procedure compares the STKM Timestamp value against the stored relevant Key Validity data in the secure function. That Key Validity data stored in the secure function is defined as an interval of STKM timestamps (i.e. Lower limit of Timestamp ("TS low") and Upper limit of Timestamp ("TS high"). Key Validity data check failure corresponds to the condition that the timestamp in the STKM received is higher than the 'TS high' or lower than the 'TS low'.

STKM Message validation: The STKM Message Validation check consists of the verification of integrity of the STKM, using the SEK/PEK. This procedure is equivalent to that described in Section 7.1.1.8 for MTK messages of [3GPP TS 31.102 v7].

Play-back counter: An internal counter in the secure function that contains the number of play-backs authorized.

SEK/PEK key group: A group of SEK/PEKs that are identified by the same Key group part of the SEK/PEK ID. The SEK/PEK key group is uniquely identifiable by its Key Domain ID and Key group part of the SEK/PEK ID.

6.7.2 EXT BCAST for STKMs

To include Smartcard Profile specific information in STKMs that can not be supported by the MBMS MTK message, a new MIKEY Generic Extension Header payload SHALL be included in the STKM. This payload is reffered to as the EXT BCAST for STKMs. The EXT BCAST for STKMs is used to transport information related to the use of the TEKs contained within the STKM.

The terminal SHALL support the processing of all fields included in the EXT BCAST for STKM with the exception of the Access Criteria Descriptor. The terminal MAY support the processing of Access Criteria Descriptors.

If the Smartcard supports the use of the EXT BCAST payloads, the Smartcard MAY process and enforce the access criteria (if it is transmitted in the EXT BCAST for STKM). Note that MBMS MIKEY implementations [3GPP TS 33.246 v7] ignore the EXT BCAST for STKMs and therefore do not support the enforcement of the access criteria.

Table 22: Logical Structure of the MIKEY General Extension Payload

Next Payload
Type
Length
Payload Data

For the EXT BCAST for STKMs the fields of the MIKEY Generic Extension Header MUST contain the following data:

Next Payload (8 bits): This field SHALL be populated as defined in [RFC3830]. No change is required.

Type (8 bits): This field defines a new type for MIKEY in addition to the existing types for MIKEY. The new type is named "OMA BCAST STKM/LTKM MIKEY General Extension" and is assigned the value of 5.

Length (16-bits): This field SHALL be populated as defined in [RFC3830]. No change is required.

Payload Data (Variable Length): The subtype is equal to 2 and the SubType specific data SHALL contain Smartcard Profile STKM Management Data defined below.

Table 23: Format of Smartcard Profile STKM Management Data

Smartcard Profile STKM Management Data	Length (in bits)	Type
short_term_key_message() {		
selectors_and_flags {		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

protocol_version	**4**	**Uimsbf**
protection_after_reception	**2**	**Uimsbf**
terminal_binding_flag	**1**	**Uimsbf**
access_criteria_flag	**1**	**uimsbf**
traffic_protection_protocol	**3**	**uimsbf**
traffic_authentication_flag	**1**	**uimsbf**
}		
traffic_key_lifetime	**4**	**uimsbf**
if (access_criteria_flag == TKM_FLAG_TRUE) {		
reserved_for_future_use	**7**	**bslbf**
secure_channel_flag	**1**	**bslbf**
number_of_access_criteria_descriptors	**8**	**uimsbf**
access_criteria_descriptor_loop() {		
access_criteria_descriptor()		
}		
}		
}		

reserved_for_future_use – these bits are reserved for future use, and SHALL be set to zero when not used.

6.7.2.1 Coding and Semantics of Attributes

Section 7 introduces the coding and semantics of all attributes common between the DRM Profile and the Smartcard Profile. Any Smartcard Profile specific attributes are introduced below.

terminal_binding_flag – indicates whether or not terminal binding is required for the Smartcard Profile. TKM_FLAG_FALSE indicates it is not required, whereas TKM_FLAG_TRUE indicates it is required.

secure_channel_flag – indicates whether or not a TEK must be transmitted from the Smartcard to the Terminal inside a secure channel for the Smartcard Profile. TKM_FLAG_FALSE indicates that a secure channel is not required, TKM_FLAG_TRUE indicates that a secure channel is required. If the access_criteria_flag is set to TKM_FLAG_FALSE, i.e. the secure_channel_flag is absent, then the secure_channel_flag SHALL be considered to be set to TKM_FLAG_FALSE, thus indicating that the secure channel is not required.

6.7.3 OMA BCAST STKM Processing

STKMs are processed by a secure function located on either the Smartcard or terminal. Where GBA_U is used the secure function is located on the Smartcard-and where GBA_ME is used the secure function is located on the terminal.

The following sections describe the processing of the STKM peformed in the terminal and the secure function. The terminal and the secure function SHALL identify a MIKEY message as a BCAST STKM if the MIKEY message includes the EXT MBMS payload (indicating MTK delievery) and the EXT BCAST payload.

It should be noted that MBMS only Smartcards can be used within a BCAST system, in which case the secure function processing of STKMs is defined in Section 6.5.4 of [3GPP TS 33.246]. As MBMS MIKEY implementations will ignore the EXT BCAST payload in the STKM, the access_criteria_flag in the EXT BCAST payload in any STKM that could be be received by a MBMS only Smartcard SHOULD be set to TKM_FLAG_FALSE.

6.7.3.1 Event Information Sent by the Terminal (moved)

Note: This section has been moved to the section 6.7.3.16.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.7.3.2 STKM Resending Check in the Terminal

Resending the same STKM allows faster changing between channels because the terminal does not have to wait for the arrival of a new STKM before being able to access the protected content, e.g. a new STKM/TEK may only be sent every minute, but the STKM/TEK is resent every 500ms meaning that the terminal has to wait a much shorter period for the required STKM/TEK after a channel change.

The terminal SHALL detect that a STKM has been resent by the BSM if the TEK_ID (MTK ID) field of the MBMS EXT payload is equal to the TEK ID contained in the last STKM sent by the terminal to the secure function. The terminal SHALL NOT forward resent STKMs to the Smartcard.

This shall not be confused with the STKM replay detection check (described in Section 6.7.3.3), which uses the TS field in the STKM message.

In MBMS for each STKM sent the TS field is increased, even if this STKM carries the same TEK as the previous STKM message.

However, in BCAST the server MAY resend the same STKM, containing the same TEK, without increasing the TS field. This avoids the need for generating new STKMs within the same crypto period.

Note: this is an improvement to the MBMS specification version 6 since BM-SC handling needs less processing for building subsequent authenticated STKM with the same key material included.

Filtering at the terminal side keeps the solution consistent with the MBMS replay protection, since in the terminal resending of the STKM/TEK is detected by checking the TEK_ID (MTK ID) field of the MBMS EXT payload.

6.7.3.3 STKM Replay Detection Protection in the Terminal

If the secure function is located on the Smartcard and the Smartcard is an MBMS only Smartcard, then the terminal SHALL perform the MBMS replay protection check as defined in section 6.4.3 of [3GPP TS 33.246 v7]. If the secure function is located on the terminal or the terminal is paired with a BCAST Smartcard, then the terminal SHALL NOT perform the MBMS replay protection check. In this case the STKM replay detection check is completed by the secure function as explained in Section 6.7.3.6.

6.7.3.4 STKM Processing in a Smartcard Supporting BCAST and MBMS

When a BCAST Smartcard receives a MIKEY message, the Smartcard SHALL first determine the type of message by examining the MIKEY General Extension payload(s) present in the message. If the message contains a General Extension payload of Type 3 (Key ID Information), i.e. the EXT MBMS payload defined for MBMS in [RFC4563], then the Smartcard SHALL inspect the contents of this payload. If the payload indicates that the delivery of an TEK (MTK), then the Smartcard SHALL check the message for the presence of a General Extension payload of Type 5, i.e. the EXT BCAST payload defined for BCAST in [RFC5410], and the following occurs:

- If the EXT BCAST payload is present, then the Smartcard SHALL process the STKM as defined in the following sections of this document;
- If the EXT BCAST payload is not present, then the MIKEY message is an MBMS MTK message. If the selected Smartcard application supports MBMS processing (e.g. selected application is USIM), then it SHALL process the message as defined in [3GPP TS 33.246 v7]. If the selected Smartcard application does not support MBMS processing (e.g. selected application is BSIM), then it SHALL abort the processing of the MIKEY message and send an error message to the terminal. The error message returned by the Smartcard SHALL take the form of the status word ‘9864’ (Authentication error, security context not supported). Note: It is mandatory for the USIM application on BCAST Smartcard to support MBMS processing as defined in Section 6.2.

Note: The above processing is required to ensure that a 3GPP BCAST Smartcard can work in a pure MBMS system, i.e. where Service Protection is implemented according to [3GPP TS 33.246 v7].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6.7.3.5 STKM Message Validation in the Secure Function

On reception of the STKM, the secure function first retrieves, from the EXT MBMS payload, the Key Domain ID and the SEK/PEK ID, which it uses to retrieve the SEK/PEK that is required to process the STKM.

If the secure function can not retrieve a SEK/PEK matching the Key Domain ID and SEK/PEK ID pair contained in the STKM, the secure function SHALL return an error message to the terminal. If the secure function is located on the Smartcard the message SHALL be the status word "6A88" (referenced data not found).

If the secure function can retrieve a SEK/PEK matching the Key Domain ID and SEK/PEK ID pair contained in the STKM, , the secure function SHALL perform the message validation according to [RFC3830].

6.7.3.6 STKM Replay Detection in the Secure Function

Following a successful STKM message validation check, the secure function SHALL perform the STKM replay detection check by comparing the received Time Stamp field (TS), i.e. the STKM TS, with the stored STKM replay detection counter value of the associated SEK/PEK. The conditions for success and failure are defined in the table below:

Success	If the received TS is greater than the stored STKM replay detection counter value, the replay detection check ends in success.
Failure	If the received TS is equal or lower than the stored STKM replay detection counter value, the replay detection check ends in failure.

Note: Less than or equal is to be taken in the meaning of RFC 1982 [RFC1982]. If the less than or equal relation is undefined in the sense of RFC 1982 [RFC1982], the message should be considered as being replayed and shall be discarded.

Note: A single STKM replay detection counter is used per SEK/PEK ID, irrespective of the number of different instances of security policy extensions that are stored in the secure function for that SEK/PEK ID.

6.7.3.7 Choice of the Security Policy Extension (SPE) for Processing the STKM

To select the security policy extension to use for the processing of the STKM the secure function SHALL first perform the STKM replay detection check, as defined in Section 6.7.3.6.

STKM replay detection check succeeds:

If the STKM replay detection check results in success, i.e. if the STKM TS is greater than the STKM replay detection counter, the secure function SHALL limit its choice of SPEs to those that allow access to LIVE content, i.e. 0x00, 0x02, 0x04, 0x08 and 0x0C) and SHALL continue to the Key Validity data check.

STKM replay detection check fails:

If the STKM replay detection check results in failure, i.e. if the STKM TS is less than or equal to the STKM replay detection counter, the secure function SHALL limit its choice of SPEs to those that allow the PLAYBACK of recorded content, i.e. 0x01, 0x03, 0x05, 0x07, 0x09 and 0x0D) and SHALL continue to the Key Validity data check.

Key Validity data check:

Once the STKM replay detection check has been completed the secure function SHALL complete the key validity data check by checking the received TS, i.e. the TS field in the STKM, against the Key Validity data associated to each applicable security policy extension, i.e. each LIVE or PLAYBACK security policiy extension (dependent on the result of the STKM replay detection check) associated to the SEK/PEK. For each applicable security policy extension, if the received TS is equal to or lower than “TS low” or is greater than “TS high” the SPE fails the key validity data check and the SPE is not applicable to this STKM.

When the secure function has completed the key validity data checks for all applicable SPEs, if no SPEs have passed the key validity data check, the secure function SHALL return a an error message to the terminal. If the secure function is located on the Smartcard, the message SHALL be the status word "9865" (Key freshness failure).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

If only one applicable security policy extension passed the key validity data check, the secure fucntion SHALL select that SPE for the processing of the STKM and SHALL process the message as defined in Section 6.7.3.8.

If several applicable SPEs passed the key validity data check with different SPE values, the following priorities SHALL be used by the secure function to select the SPE to use to handle the incoming STKM:

For LIVE security policy extensions

1. SPE for a subscription mode (0x04)
2. SPE for a pay-per-view (PPV) mode (0x04 or 0x08)
3. SPE for a pay-per-time (PPT) mode (0x00 or 0x02 or 0x0C)

For PLAYBACK security policy extensions

1. SPE for a subscription mode (0x05)
2. SPE for a pay-per-play (PPP) mode (0x07 or 0x09)
3. SPE for a pay-per-time (PPT) mode (0x01 or 0x03 or 0x0D)

If a PPV or PPT permission for a given SEK/PEK exists without tokens and also with tokens, the priority SHALL be to use the LTKM without tokens.

Hence the following priority SHALL apply on the pay-per-view / pay-per-play and pay-per-time SPE:

- No tokens vs tokens PPT: SPEs 0x0C and 0x0D will take precedence over 0x00, 0x01, 0x02 and 0x03
- No tokens vs tokens PPV: SPEs 0x04 and 0x07 will take precedence over 0x08 and 0x09

The table below summarises the order of priority when choosing an LTKM SPE to use.

Table 24: LTKM security_policy_extension Priorities

	LIVE	PLAYBACK
Highest Priority	0x04 (subscription)	0x05 (unlimited playback)
	0x04 (PPV)	0x07 (PPP)
	0x08 (user token PPV)	0x09 (user token PPP)
	0x0C (PPT)	0x0D (PPT)
	0x00 (service token PPT)	0x01 (service token PPT)
Lowest Priority	0x02 (user token PPT)	0x03 (user token PPT)

If several applicable SPEs passed the key validity data check with the same SPE value, the following rules SHALL be used by the secure function to select the SPE to use to handle the incoming STKM:

- the secure function SHALL select the SPE with the lowest “TS Low” value;
- if there is more than one SPE with the same SPE value and “TS Low” value, the secure function SHALL select the SPE with the lowest “TS High” value;

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6.7.3.8 STKM Processing based on the LTKM security_policy_extension (SPE)

Note that the processing described below is done AFTER having successfully selecting the security policy extension to use based on the key validity check, as explained above in Section 6.7.3.7.

The secure function SHALL NOT send a verification message as a response to an STKM even in the case where the V-bit in the STKM message is equal to 1.

In the following descriptions, the term "decrypted material" is used to denote the TEK and Salt key (if Salt key is available), which may be returned in the clear or wrapped by the TBK.

LTKM SPE = 0x00 (Service Token PPT Live)

Success	• If the live_ppt_purse is greater or equal to the cost_value, the secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • decrease the live_ppt_purse by the stored cost value, • return the decrypted material.
Failure	If the live_ppt_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12.

LTKM SPE = 0x01 (Service Token PPT Playback).

Success	If the playback_ppt_purse is greater or equal to the cost_value, the secure function SHALL: • decrease the playback_ppt_purse by the stored cost value, • return the decrypted material.
Failure	If the playback_ppt_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12.

LTKM SPE = 0x02 (User Token PPT Live)

Success	• If the user_purse is greater or equal to the cost_value, the secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • decrease the user_purse by the stored cost value, • return the decrypted material.
Failure	If the user_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

LTKM SPE = 0x03 (User Token PPT Playback)

Success	If the user_purse is greater or equal to the cost_value , the secure function SHALL: • decrease the user_purse by the stored cost value, and • return the decrypted material.
Failure	If the user_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12.

LTKM SPE = 0x04 (Subscription and PPV Live)

Success	The secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • return the decrypted material.
Failure	• N/A

LTKM SPE = 0x05 (Subscription unlimited playback)

Success	The secure function SHALL: • return the decrypted material.
Failure	• N/A

LTKM SPE = 0x07 (PPP Playback)

Success	If the playback counter is not equal to zero AND the STKM TS is less than or equal to the current_TS_counter, then the secure function SHALL: • set current_TS_counter to the STKM TS value and, • decrease the playback counter by one, and • returns the decrypted material. If the STKM TS is greater than to the current_TS_counter, then the secure function SHALL: • set current_TS_counter to the STKM TS value, and • NOT decrease the playback counter, and • return the decrypted TEK material.
Failure	If the playback counter is equal to zero AND the STKM TS is less than the current_TS_counter, then the secure function SHALL: • execute Section 6.7.3.12.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

LTKM SPE = 0x08 (User Token PPV Live).

Success	If the user_purse is greater or equal to the cost_value AND the STKM TS is equal to or less than the current_TS_counter, the secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • set current_TS_counter to the STKM TS value and, • decrease the user_purse by the stored cost value and, • return the decrypted material. If the STKM TS is greater than the current_TS_counter, then the secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • NOT decrease the user_purse, and • return the decrypted material.
Failure	If the STKM TS is equal to or less than the current_TS_counter and the user_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12.

LTKM SPE = 0x09 (User Token PPP Playback).

Success	If the user_purse is greater or equal to the cost_value, AND the STKM TS is less than or equal to the current_TS_counter, the secure function SHALL: • set current_TS_counter to the STKM TS value and, • decrease the user_purse by the stored cost value, • return the decrypted material. If the STKM TS is greater than the current_TS_counter, then the secure function SHALL: • set current_TS_counter to the STKM TS value, and • NOT decrease the user_purse, and • return the decrypted TEK material.
Failure	If the STKM TS is less than the current_TS_counter and either the user_purse is less than the cost_value, then the secure function SHALL: • execute Section 6.7.3.12

LTKM SPE = 0x0C (PPT Live)

Success	The kept TEK counter value SHALL be added to the TEK counter. If the kept TEK counter is NOT equal to zero, the kept TEK counter SHALL then be set to zero. If the TEK counter overflows, it SHALL be set to maximum. If the TEK counter is greater than zero the secure function SHALL: • set the STKM anti-replay counter to the STKM TS value, and • decrease the TEK counter by one and, • return the decrypted material.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Failure	If the TEK counter is equal to zero, then the secure function SHALL: • execute Section 6.7.3.12.

LTKM SPE = 0x0D (PPT Playback)

Success	If the TEK counter is greater than zero the secure function SHALL: • decrease the TEK counter by one and, • return the decrypted material.
Failure	If the TEK counter is equal to zero, then the secure function SHALL: • execute Section 6.7.3.12.

6.7.3.9 Deletion of Expired LIVE Security Policy Extensions and SEK/PEK

The following text assumes that the STKM Message Validation check (see Section 6.7.3.4) has been passed. The processing to support the following functions is implementation specific.

The secure function SHALL delete all stored data related to an instance of a LIVE SPE (including the SPE value itself) if the instance of the LIVE SPE is associated to the SEK/PEK ID in the received STKM and the TS value contained in the received STKM is greater than the "TS high" value of the KV data associated to that instance of the SPE. Such action corresponds to the identification and deletion of expired LIVE SPEs

Before the secure function deletes an instance of SPE 0x0C, if the keep_credit_flag associated to that instance of SPE 0x0C is set to LTK_FLAG_TRUE, the secure function SHALL add the value of the associated TEK counter to the to the value of the kept TEK counter associated to the SEK/PEK key group of the SEK associated to the instance of the SPE 0x0C being deleted.

When the secure function processes an STKM, if the STKM is protected by a SEK/PEK belonging to the same SEK/PEK Key Group as one or more SEKs/PEKs stored in the secure function, the secure function SHALL delete all stored data related to instances of LIVE SPEs associated to the older SEKs/PEKs. The secure function SHALL use the Key Number part of the SEK/PEK IDs to determine which SEK/PEKs are older than the SEK/PEK used to protect the current STKM.

Information relating to PLAYBACK security policy extensions SHALL NOT be deleted.

If any of the processes described above results in there being no valid SPEs associated to a SEK/PEK, the secure function SHALL delete the SEK/PEK and all associated data.

If there are no more SEK/PEK and SPE associated to a Key Group the Secure Function SHALL NOT delete the SEK/PEK ID Key Group part and associated data as purse could be associated to this Key Group. Only the terminal MAY delete the Key Group and associated data.

6.7.3.10 Deletion of Expired PLAYBACK Security Policy Extensions and SEK/PEK

When a SPE for playback stored in the smartcard is obsolete (how the secure function determines the obsolescence of the SPE is implementation specific), and this SPE has not been flagged as SPE used for recorded content (see Section 8.5), then the secure function SHALL delete the SPE. If the deletion of this SPE results in there being no valid security policy extension associated to the SEK/PEK ID, the secure function SHALL delete the SEK/PEK and associated data.

If there are no more SEK/PEK and SPE associated to a Key Group the Secure Function SHALL NOT delete the SEK/PEK ID Key Group part and associated data as purse could be associated to this Key Group. Only the terminal MAY delete the Key Group and associated data.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6.7.3.11 Access Criteria

The secure function checks the presence of access criteria in the message and controls that the access criteria conditions are met using internal information. This internal information depends on the type of the access criteria. For the current version of the specification, access criteria defined are for parental control and location based restriction.

The terminal SHALL implement all necessary processing and SHALL support associated messaging to handle Smartcard based access criteria enforcement.

6.7.3.11.1 Parental Control

Enforcement of the parental control is done by checking the level_granted against the rating_value received in the STKM for the same rating_type.

In the STKM the country_code_flag SHALL be set to LTK_FLAG_FALSE.

If the parental_control access criteria are transmitted in the STKM and if the secure function is in the Smartcard, parental control enforcement SHALL be done by the Smartcard as explained below. Note that MBMS MIKEY implementations [3GPP TS 33.246 v7] will ignore the EXT BCAST for STKMs and therefore will not support the enforcement of parental control as described in this document. In this case, the Terminal MAY choose to enforce the parental_control. Alternatively, Terminal enforcement MAY be used in parallel with the Smartcard enforcement mechanism for providing an additional, locally controlled restriction on access. Note that in this case the most restricted level from the smartcard or the terminal will apply.

It is out of scope of the BCAST 1.0 specification how parental control applies to multiple instances of possibly different services when those have to be simultaneously treated by the terminal. The result of the whole parental control checking process is as follows:

Failure	If the processing of the parental_control access criteria ends with failure, the secure function SHALL abort the processing of the STKM. If the secure function is located on the Smartcard, it SHALL send an Operation Status code corresponding to 'User not authorized' with the current rating_value (received in the STKM) and the level_granted for this rating_type stored in the Smartcard. These data are sent as a response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for parental control operation (see Appendix E). If the secure function is located on the Smartcard, it MAY send the proactive command 'DISPLAY TEXT' (as described in [3GPP TS 31.111 v9.1.0] or [3GPP2 C.S0035-A]) in order to inform the user that the level_granted stored in the card for the rating_type received in the STKM does not allow to view this service as they are not authorized to view services with the associated rating transmitted in the STKM.
Success	If the processing of the parental_control access criteria ends with success, the secure function performs the checks as defined in previous sections if needed. This will then allow the secure function to send the decrypted material to the terminal.

Parental control management in the Smartcard:

If the secure function is in the Smartcard, the terminal SHALL implement PINCODE requested processing (described below), operation on PINCODE (described below) and associated messaging to handle parental control management with the related processing (i.e.: response of AUTHENTICATE command corresponding to OMA BCAST operation for parental control operation (as described in Appendix E), VERIFY PIN as defined in [ETSI TS 102.221]). The terminal MAY implement UNBLOCK PIN as defined in [ETSI TS 102.221] and proactive command DISPLAY TEXT as defined in [3GPP TS 31.111 V9.1.0] or [3GPP2 C.S0035-A].

The enforcement of the parental control is divided in several processing phases:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- Check the rating_value transmitted in the STKM against the level_granted stored in the Smartcard for the rating_type.
- Check if the PINCODE has been verified.
- Request a PINCODE if necessary. A PINCODE provided by the user is checked against the PINCODE stored in the Smartcard.
- Unblock a locked Parental Control PINCODE, if applicable.

The following gives details on these different steps:

- **Check the rating_value transmitted in the STKM against the level_granted stored in the Smartcard for the rating_type:**

The secure function SHALL first compare the rating_type received in the STKM against all of the rating_type values stored in the Smartcard. If there is a level granted, depending on the rating_value and the rating_type, the outcome is success or failure:

Success If there is a level_granted for the rating_type in the Smartcard and if it is an equal or more restrictive value than the rating_value received in the STKM, the checking of rating_value ends with success and the processing of STKM resumes. Requesting the PINCODE is not needed.

If there is no level_granted for the rating_type in the Smartcard, the user is authorized to view the content. The checking of rating_value ends with success and the processing of the STKM resumes. Requesting the PINCODE is not needed.

Failure If there is a level_granted for the rating_type in the Smartcard and if it is less restrictive than the rating_value received in the STKM, the checking of rating_value ends with failure and the secure function triggers a request for the PINCODE. If the PINCODE is not defined in the Smartcard, the Smartcard aborts the processing of STKM and indicates to the user that they are not allowed to view this content.

Table 25 gives an example of comparison of the rating_value in the STKM against the level_granted stored in the Smartcard. In this example, the rating_type 9 (as defined in the OMA BCAST Parental Rating System Registry available at [OMNA]) is taken as an example. Table 25 uses the following symbols:

X means that the secure function stops processing the STKM unless a valid PINCODE is provided.

O means that the secure function accepts processing the STKM without requesting a PINCODE.

Table 25: Example of Comparing STKM rating_value against Smartcard level_granted

		Smartcard level_granted					
		none defined	1 (least restrictive)	2	3	4	5 (most restrictive)
STKM rating_value	none defined	O	O	O	O	O	O
	1 (least restrictive)	O	O	O	O	O	O
	2	O	X	O	O	O	O
	3	O	X	X	O	O	O
	4	O	X	X	X	O	O
	5 (most restrictive)	O	X	X	X	X	O

Note that the term ‘more restrictive’ means that there are more constraints on having access to the content. This typically means the user age is higher. Note that actual numerical values of rating_value for certain rating_types do not always follow a linear scale, either from less restrictive to more restrictive or vice-versa. The corresponding logical order (from least restrictive to most restrictive) is based on the semantics of the individual rating values. An informative example can be found in Table 130 in Appendix H.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Note that the value for “not rated” or “undefined” SHALL be treated by default as “least restrictive”, unless its semantics is explicitly stated by the rating scheme.

- **Check if PINCODE has been verified:**

A PINCODE is defined in the Smartcard for the parental control function. For using this PINCODE in the VERIFY PIN and UNBLOCK PIN commands, a key reference is assigned at the manufacture of the Smartcard. The PINCODE function is optional in the Smartcard for the parental control.

Depending on the result of checking of rating_value against the granted_level value and if a PINCODE is defined in the Smartcard, the Smartcard SHALL check if the PINCODE has been verified previously for the same content. This verification results in the following.

Success	If the PINCODE has been previously verified with success the parental control ends with success and the processing of STKM resumes. The Smartcard SHALL NOT request that a PINCODE is entered if the PINCODE has been previously verified with success for the same content (i.e. when the SEK/PEK_ID and rating_type/rating_value pair is the same in the STKM). Information that the PINCODE has been verified SHALL be stored in the Smartcard and SHALL be reset if the content changes (SEK/PEK_ID or rating_type/rating_value change in the incoming STKM, or reception of an Event Signalling Mode command (Section E.3.5) when such command indicates a content change) if the terminal is switched off or if the transmission of STKM has been interrupted. This interruption in the transmission MAY be detected by a gap in the timestamp value in the incoming STKM (width of the gap MAY be adjusted by the service provider at the manufacture stage of the Smartcard) against the value stored in the replay counter of the SEK/PEK_ID. As a complement to this usual detection, this interruption MAY be signaled by the terminal with the Event Signalling Mode command (Section E.3.5).
Failure	If the PINCODE has not been verified or the verification process ended with failure the Smartcard proceeds to request the PINCODE.

- **Request a PINCODE if necessary. A PINCODE provided by the user is checked against the PINCODE stored in the Smartcard:**

If the Smartcard needs to request a PINCODE, the following applies:

The Smartcard aborts the STKM processing by sending a response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for parental control operation (see Appendix E) with:

- A status code corresponding to 'PINCODE blocked' and the key reference corresponding to the PIN used for parental control if the Parental control PIN has been previously blocked or
- A status code corresponding to ‘PINCODE required’ and the key reference corresponding to the PIN used for parental control in order to request from the terminal a PINCODE verification processing.

At the reception of the response with a status code corresponding to ‘PINCODE required’, the terminal asks the user to enter the PINCODE and sends this PINCODE to the Smartcard using the APDU command VERIFY PIN defined in [ETSI TS 102.221] on the PIN corresponding to the key reference value transmitted in the response of AUTHENTICATE command.

The result of the VERIFY PIN command is success or failure:

Success If the VERIFY PIN ends with success, the terminal SHALL resend the STKM to the secure function in the Smartcard for the processing.

Failure If the VERIFY PIN ends with failure, the terminal MAY request another entry of the PINCODE. Three false entries SHALL block the PINCODE.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- **Unblock a locked Parental Control PINCODE:**

When the user has entered the wrong PINCODE three times in the verification process, the PINCODE is blocked in the Smartcard. After receiving a response with a status code corresponding to 'PINCODE blocked', the terminal MAY ask the user to unblock the PINCODE.

When unblocking the PINCODE, the terminal MAY request the user to input an UNBLOCK PIN value and a new personal PINCODE. The new PINCODE value SHALL be sent to the Smartcard using the APDU command UNBLOCK PIN, together with the UNBLOCK PIN value, as specified in [ETSI TS 102.221].

The terminal MAY use the command UNBLOCK PIN defined in [ETSI TS 102.221] with the key reference received in the response of AUTHENTICATE command.

NOTE: The acquisition of the UNBLOCK PIN value uses out-of-band mechanism, e.g. by post or by calling to operator's customer service center.

6.7.3.11.2 Location Restriction

- For Smartcard based location_based_restriction enforcement, the terminal SHALL support the proactive command PROVIDE LOCAL INFORMATION as defined in [3GPP TS 31.111 v9.1.0] or [3GPP2 C.S0035-A], the proactive command DISPLAY TEXT as defined in [3GPP TS 31.111 v9.1.0] or [3GPP2 C.S0035-A] and the response of AUTHENTICATE command corresponding to OMA BCAST operation for location based restriction operation (see Appendix E).

If the location_based_restriction access criteria are transmitted in the STKM the following applies:

- If the Smartcard supports the use of EXT BCAST payloads and supports the enforcement of location_based_restriction, this enforcement SHALL be done by the Smartcard as explained in Section 6.7.3.15. Note that MBMS MIKEY implementations [3GPP TS 33.246 v7] will ignore the EXT BCAST for STKMs and therefore will not support the enforcement of the access criteria.

Location control failure	If the location based restriction ends with the status 'blackout', then the secure function aborts the processing of the STKM. If the secure function is located on the Smartcard, then it SHALL either send a response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for location based restriction operation (see Appendix E) with a status code corresponding to 'blackout', or SHALL send a proactive command 'DISPLAY TEXT' (as described in [3GPP 31.111 v9.1.0] or [3GPP2 C.S.0035A]) in order to inform the user that the program can not be displayed in this area. The terminal SHALL support the proactive command 'DISPLAY TEXT'.
Location control success	If the location based restriction ends with the status 'need specific permissions', then the secure function checks if a security_policy_extension for a PPV is available for this content. Two cases result from this check: - If a security_policy_extension for a PPV is available for this content, then the processing of the STKM continues as as we discuss below. - If a security_policy_extension for a PPV is not available for this content, then the secure function aborts the processing of the STKM. If the secure function is located on the Smartcard, then it SHALL either send a response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for location based restriction operation (see Appendix E) with a status code corresponding to 'need specific permissions' or SHALL send a proactive command 'DISPLAY TEXT' (as described in [3GPP 31.111 v9.1.0] or [3GPP2 C.S.0035-A]) in order to inform the user that the program can not be displayed in this area without a specific permission. The terminal SHALL support the proactive command 'DISPLAY TEXT'.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.7.3.12 Lack of Credit in Purse or Playback Counter or TEK Counter

For the SPE modes using the live_ppt_purse, playback_ppt_purse, user_purse, the play-back counter, or the TEK counter (i.e. 0x00, 0x01, 0x02, 0x03, 0x07, 0x08, 0x09, 0x0C or 0x0D), the secure function SHALL ignore the nominal operation in cases whereby the live_ppt_purse, playback_ppt_purse or user_purse cannot be decreased by the cost_value, or play-back counter or TEK counter equals zero.

If the secure function is located on the Smartcard the following processing is applicable:

- When the live_ppt_purse cannot be decreased the Smartcard SHALL return the response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for SPE operation (see Appendix E) with a status code corresponding to 'lack of credit in the live_ppt_purse'.

- When the playback_ppt_purse cannot be decreased the Smartcard SHALL return the response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for SPE operation (see Appendix E) with a status code corresponding to 'lack of credit in the playback_ppt_purse'.

- When the user_purse cannot be decreased the Smartcard SHALL return the response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for SPE operation (see Appendix E) with a status code corresponding to 'lack of credit in the user_purse'.

- When the play-back counter is invalid or equals zero, the Smartcard SHALL return the response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for SPE operation (see Appendix E) with a status code corresponding to 'play_back counter invalid or equal to zero'.

- When the TEK counter is invalid or equals zero, the Smartcard SHALL return the response to the terminal for the current AUTHENTICATE command corresponding to OMA BCAST operation for SPE operation (see Appendix E) with a status code corresponding to 'TEK counter invalid or equal to zero'.

6.7.3.13 Protection of the TEK after STKM Processing

When the TEK is returned by the secure function to the terminal, the TEK is in the clear unless the secure function is located on the Smartcard and the Terminal Binding Key (TBK) is used, in which case the TEK is wrapped by the TBK, or the Secure Channel is used, in which case the return message containing the TEK is protected using the Connection SA as defined in [ETSI TS 102 484]. In the case that both the TBK and Secure Channel are used, the Secure Channel contains the TEK wrapped with the TBK.

6.7.3.14 Illustration of Parameters Used (Informative)

This section provides diagrams to illustrate the different parameters used when processing STKMs to determine whether or not the STKM being presented to the secure function corresponds to a LIVE or PLAYBACK situation.

Figure 4 illustrates a scenario in which two LTKMs have been sent to the secure function with the same KV data (SPE TS low and TS high). For the purposes of this example the SPEs in the two LTKMs were 0x04 and 0x05 but they could be any combination of LIVE and PLAYBACK SPEs with overlapping KVs. The STKM anti-replay counter, associated to the SEK/PEK used to protect the STKM, is within the KV data range of both SPEs, i.e. the secure function has previously processed one or more STKMs corresponding to this SEK//PEK. The figure shows that any STKM processed by the secure function, within the KVs of the two SPEs, will be treated as part of a PLAYBACK if its TS is less than or equal to the STKM anti-replay counter and as LIVE if its TS is great than the STKM anti-replay counter.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

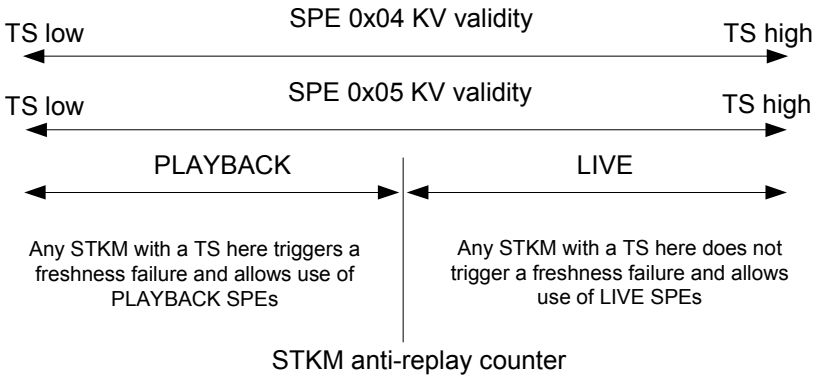


Figure 4 – Illustration of LIVE vs PLAYBACK Relative to the STKM Anti-Replay Counter

Figure 5 illustrates the use of a PLAYBACK SPE that uses the current_TS_counter to detect whether or not the STKM is part of an existing playback or whether the STKM is part of a new playback. The example uses SPE 0x07 in which the detection of a new playback results in the playback counter being decremented, however, the current_TS_counter is used in a similar way for SPE 0x09 but the detection of a new playback results in the user_purse being decremented by the associated cost_value.

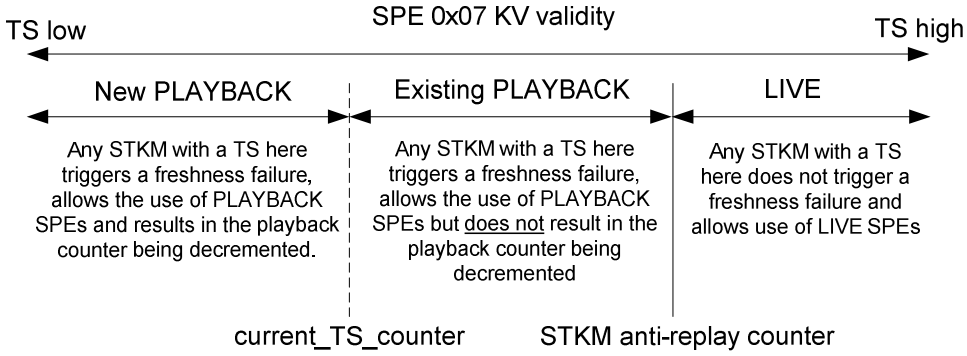


Figure 5 – Illustration of PLAYBACK and use of Current_TS_Counter to Detect Local Playback

6.7.3.15 Enforcement of location_based_restriction

The enforcement of the location_based_restriction is processed as follow:

Beforehand, the secure function has requested to the terminal current location information sending a proactive command 'PROVIDE LOCAL INFORMATION' as described in [3GPP 31.111 v9.1.0] or [3GPP2 C.S.0035-A]. To request this information the secure function, on a polling of the terminal returns a status code '91XX' to the terminal to indicate that a proactive command is pending. The terminal fetches the pending proactive command 'PROVIDE LOCAL INFORMATION', performs it and sends to the secure function the response of the proactive command execution. The terminal SHALL support proactive command 'PROVIDE LOCAL INFORMATION' as described in [3GPP 31.111 v9.1.0] or [3GPP2 C.S.0035-A].

The secure function will be able to handle some target_area_type that relies on information provided by the proactive command 'PROVIDE LOCAL INFORMATION'.

If target_area_type is 0x2, then the mobile_country_code of the current location of the terminal is used and is compared to the mobile_country_code specified in the STKM. The comparison results in the following cases:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

o If there is a match and if polarity flag in the STKM is set to 0 (“normal”) and if the override flag is set to 0, then the location_based_restriction ends with the status ‘blackout’, i.e. the terminal SHALL not render the media streams and then the STKM processing SHALL ends without the processing of the TEK and then is aborted (see Section 6.7.3).

o If there is no match and if polarity flag in the STKM is set to 1 (“spotbeam”) and if the override flag is set to 0, the location_based_restriction ends with the status ‘blackout’ i.e. the terminal SHALL not render the media streams and then the STKM processing SHALL ends without the processing of the TEK and then is aborted (see Section 6.7.3).

o If there is a match and if polarity flag in the STKM is set to 0 (“normal”) and if the override flag is set to 1, the location_based_restriction ends with the status ‘need specific permissions’, i.e. the restriction may be ignored if the terminal is able to obtain the necessary permissions. Then the STKM processing resume to check if a PPV for this program is active (see Section 6.7.3).

o If there is no match and if polarity flag in the STKM is set to 1 (“spotbeam”) and if the override flag is set to 1, the location_based_restriction ends with the status ‘need specific permissions’, i.e. the restriction may be ignored if the terminal is able to obtain the necessary permissions. Then the STKM processing resume to check if a PPV for this program is active (see Section 6.7.3).

o Otherwise the location_based_restriction enforcement ends without blackout, i.e. the terminal MAY render the media streams and then the STKM processing SHALL resume.

If target_area_type is 0x5, then depending of the cell_target_area_type, the Cell Global Identifier or the location Area Identifier or the SID, or the SID+NID…of the current location of the terminal is used and compared to the cell_area_values received in the STKM. The comparison results in the following cases:

o If there is a match and if polarity flag in the STKM is set to 0 (“normal”) and if the override flag is set to 0, then the location_based_restriction ends with the status ‘blackout’, i.e. the terminal SHALL not render the media streams and then the STKM processing SHALL ends without the processing of the TEK and then is aborted (see Section 6.7.3).

o If there is no match and if polarity flag in the STKM is set to 1 (“spotbeam”) and if the override flag is set to 0, the location_based_restriction ends with the status ‘blackout’ i.e. the terminal SHALL not render the media streams and then the STKM processing SHALL ends without the processing of the TEK and then is aborted (see Section 6.7.3).

o If there is a match and if polarity flag in the STKM is set to 0 (“normal”) and if the override flag is set to 1, the location_based_restriction ends with the status ‘need specific permissions’, i.e. the restriction may be ignored if the terminal is able to obtain the necessary permissions. Then the STKM processing resume to check if a PPV for this program is active (see Section 6.7.3).

o If there is no match and if polarity flag in the STKM is set to 1 (“spotbeam”) and if the override flag is set to 1, the location_based_restriction ends with the status ‘need specific permissions’, i.e. the restriction may be ignored if the terminal is able to obtain the necessary permissions. Then the STKM processing resume to check if a PPV for this program is active (see Section 6.7.3).

o Otherwise the location_based_restriction enforcement ends with the status ‘without blackout’, i.e. the terminal MAY render the media streams and then the STKM processing SHALL resume.

BCAST 1.0 provides above a basic signalling framework for how to override location based restrictions but does not specify how the terminal and Smartcard determine if the restriction can be ignored when the override flag of the location_based_restriction Access Criteria Descriptor is set to 1. Therefore, BCAST 1.0 terminals and Smartcards SHALL always interpret the override flag as if it were set to 0.

6.7.3.16 Event Information Sent by the Terminal

When the parental control function is located on the Smartcard, the Smartcard has some means (e.g. SEK/PEK ID change, TS gap detection) to determine e.g. whether to request the parental control PINCODE again.

The Terminal MAY send an "Event Signalling Mode" command to inform the Smartcard of an event that can impact the parental control decision process to specifically cater for deployment scenarios where the Smartcard does not have all the information necessary to perform such decision on its own.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The "Event Signalling Mode" command as specified in section E.3.5.2 is a generic command that allows the signalling of an event that MAY be used by the Smartcard for the parental control function. The command MAY be sent to the Smartcard as soon as possible, latest prior submitting an STKM, which "number_of_access_criteria_descriptors" value is not equal to zero.

Note: in BCAST 1.0, the event Zapping '0x00' is currently the only specified event (see Table 127). However, Table 127 is designed so that other extensions can be specified in a future release of this specification in order to address local regulatory requirements for other event signalling (e.g. loss of signal, terminal OMA BCAST application switch-off/switch-on).

6.7.4 STKMs and traffic encryption protocols

STKM can be sent over UDP. It is possible to multiplex STKM/UDP with FLUTE packets (on the same IP transport address but on a separate IP port – refer to Section 10.1 on how this is signaled)

If the traffic_protection_protocol equals to TKM_ALGO_DCF, then the STKM MAY be delivered as a separate object inside a FLUTE session, together with the protected traffic, having its own FDT entry.

SRTP

3GPP MBMS security [3GPP TS 33.246 v7], on which the Smartcard Profile is based, is designed for use with SRTP. It follows that the STKM defined in Section 6.7 is compatible with SRTP. SRTP encryption SHALL be indicated by the traffic_protection_protocol value in the STKM. The SRTP Master Key (MK, 128 bits) and Master Salt (MS, 112 bits) SHALL be sent within the STKM. For compatibility with the DRM Profile a NULL MS MAY be sent.

The correct TEK to use to decrypt the data is indicated using the MKI (Master Key identifier) field, which SHALL be included in the SRTP packets as defined in [RFC 3711]. The MKI SHALL be the TEK ID, unless compatibility with MBMS terminals is required in which case the MKI SHALL be a concatenation of SEK/PEK ID and TEK ID, i.e. MKI = (SEK/PEK ID || TEK ID). See Section 11.3.2 for further details on the requirements related to sharing protected traffic streams

The key derivation rate MAY be zero.

The CS ID map type subfield in the LTKM message MAY be set to value '1' (empty map) as defined in [RFC4563], which indicates that SP is absent. In this case, the SRTP Policy corresponding to the Smartcard Profile and DRM Profile interoperability case shall be assumed as described in Section 9.2:

- AES_128_CTR Encryption algorithm
- HMAC-SHA-1-80 Authentication algorithm
- A NULL 112 byte Master Salt (MS)
- A key derivation rate of 0

Note: [BCAST10-MBMS-Adaptation] sets restrictions with regards to usage of the CS ID map type.

ISMACryp

The Smartcard Profile STKM, defined in Section 6.7, is compatible with ISMACryp. For content encryption, the usage of ISMACryp SHALL be signalled by traffic_protection_protocol value in the STKM. The Smartcard Profile TEK ID corresponds to the key_indicator in the DRM Profile STKM. The key_indicator sent in the ISMACrypContextAU field (defined in [ISMACRYP11] and [ISMACRYP20]) as part of the encrypted stream SHALL correspond to the TEK ID (2 bytes) sent in the EXT MBMS payload of the STKM. Note that, unlike for SRTP, there is no requirement for compatibility with MBMS only terminals and therefore the key_indicator is never required to be a concatenation of SEK/PEK ID and TEK ID, i.e. SEK/PEK ID || TEK ID. The 128 bit TEK SHALL be transported, as for SRTP, in the KEMAC field of the STKM.

The CS ID map type subfield in both LTKM and STKM message SHALL be set to value '1' (empty map) as defined in [RFC4563], regardless of SRTP authentication is used or not. The MIKEY inner key derivation follows [RFC3830] section 4.1.4.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

If no SRTP authentication is used, the 128 bit encryption key SHALL be sent instead of the MK. The MS is not used. Salt keys SHALL be signalled in SDP. No SRTP key derivation is done in ISMACryp.

If SRTP authentication is used, MK (128 bits) and MS (112 bits) SHALL be sent within the STKM and used to derive encryption and authentication keys as per SRTP [RFC3711].

IPsec

IPsec encryption SHALL be signaled by traffic_protection_protocol value in the BCAST STKM. The 4-byte SPI sent in IPsec packets SHALL consist of constant prefix 0x0001 followed by the 2-byte MTK ID. In other words, SPI = (0x0001 || MTK ID).

The security policy information is as specified in this document, and the CS ID map type subfield in both LTKM and STKM MIKEY message SHALL be set to value '1' (empty map) as defined in [RFC4563]. Consequently cs_id SHALL be set to 0x00000000 within the IPsec key derivation of section 4.1.3 in [RFC3830].

The MTK SHALL be transported in the KEMAC field. The 16-byte IPsec encryption key (the key for ESP encryption) SHALL be derived from the MTK as specified by MIKEY ([RFC3830], Section 4.1.3) using the "encryption key" constant. If traffic authentication is used, the 16-byte Traffic Authentication Seed (TAS) SHALL be derived from the MTK as specified by MIKEY using the "authentication key" constant. The IPsec authentication key (TAK, 20 bytes) is derived from the TAS, as described in Section 9.1. No salt is used.

6.8 Layer 4: Traffic Encryption

Layer 4 corresponds to the BCAST 4-layer model key hierarchy. The protection of data in case of streaming and file delivery respectively for both service and content protection is described for the Smartcard Profile.

6.8.1 Streaming Delivery

6.8.1.1 Service Protection of Streams

Broadcast streams that are signalled as having service protection by the SG via the protectionType field with the value = 1 are associated with STKM stream(s). The broadcast streams are encrypted by TEKs using IPsec, SRTP or ISMACryp, or not encrypted at all in case no encryption type is specified in service guide.

How to obtain the relevant information from the SG to request the appropriate SEK or PEK (used for TEK protection) to access with the TEK the protected stream is explained in Section 6.10. If the LTKM extension payload is absent, and the “protection_after_reception” value in the STKM = 0x03 (i.e. “Service Protection”), then upon obtaining the TEK from the Smartcard, the terminal can either store the TEK or record the content in the clear or do both.

BCAST related communication between the Smartcard and the Terminal MAY be protected using a Secure Channel as described in Section 6.13. Using a Secure Channel ensures that the TEKs are not exposed in the clear over the Smartcard – Terminal interface.

6.8.1.2 Content Protection of Streams

Broadcast streams that are signaled as having content protection may be recorded as defined in this specification. However, for recorded material having content protection, appropriate rights need to be obtained via a Broadcast Permissions Issuer.

For terminals using the Smartcard Profile, the appropriate key material can be requested based on the Program or Service ID.

The Permissions Issuer can provide content protection for the Smartcard Profile allowing an implicit play once right. Once the server issues the appropriate SEK or PEK to the terminal / Smartcard, the terminal SHALL interpret the obtained keys relating to the recorded stream as being "play once" unless otherwise indicated by a a security policy extension contained in the EXT BCAST payload in the LTKM (see Section 6.6.4). If the EXT BCAST payload is not present in the LTKM or does not contain a security policy extension, it SHALL not be possible to use the SEK/PEK to access the same content more than once. This is achieved through the processing described in Section 6.6.7.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

As the key material provides access to recorded content stored in the terminal, preventing unauthorized access to these keys is extremely important. It is therefore recommended that they are stored in a secure storage area and protected appropriately during their limited lifetime. For an implementation using GBA_U, the Smartcard can deliver TEKs to the terminal if the adapted PDCF is used to record a TEK key stream. For content protection, the Smartcard-terminal interface SHOULD be secured.

The protection of the Smartcard-terminal interface, when supported, SHALL be implemented as described in Section 6.13.

6.8.1.3 Permissions Management using the Smartcard Profile for Content Protection of Streams

If the EXT BCAST payload is not present in the LTKM, the SEK/PEK in Smartcard Profile is based on an implicit "play once" permission. This “play once” functionality can be used by the BSM to enable more complex constraints relating to the use of a SEK/PEK, e.g. unlimited access to the appropriate SEK/PEK for a given time period or controlled access to the SEK/PEK for a given number of times. In all cases, where the EXT BCAST payload is not present in the LTKM, the Smartcard Profile terminals are forced to request a new SEK/PEK for every access to content.

If the EXT BCAST payload is also present in the LTKM, the security_policy_extension value defines the rights applicable to the LTKM. The security_policy_ext enables the provision of extended rights such as Pay Per View (PPV) and Pay Per Time. The types of extended rights that can be offered are described in Section 6.6.4. In order to request the tokens and / or consumption rules required to provide PPT and PPV functionality, the "Token Purchase Request" message defined in BCAST in [BCAST10-Services] SHALL be used.

If broadcast streams are protected and need content protection consumption rules, this is signaled via ProtectionType in the SG and via the protection_after_reception values in the STKM. Before the delivery of the related LTKM, this means there SHALL be mutual terminal-server authentication and there SHALL be the standard Smartcard BSM authentication.

The following steps SHOULD be followed when requesting key material for content protected streams:

1. Identify the Permissions Issuer URI and SEK/PEK ID
2. Initiate mutual terminal BSM authentication (see Section 6.5)
3. Initiate mutual Smartcard BSM authentication (see Section 6.11.2)
4. Establish / enable the secure channel between the Smartcard and terminal (see Section6.13)
5. Request the appropriate SEK or PEK using the "Token Request" message in [BCAST10-Services] (see Section 6.6). The requested key identifier is the SEK / PEK ID.

6.8.2 File Delivery

6.8.2.1 Service Protection of Download Data using DCF

This section contains material from MBMS text in [3GPP TS 33.246 v7]. The mechanism described in this section was adopted from [3GPP TS 33.246 v7] and adapted to BCAST needs.

BCAST terminals SHALL support download protection using DCF. BCAST servers MAY support download protection using DCF.

The same mechanism can be used to protect PDCF files. This is optional for both terminal and server.

Service protection of download data uses IPsec or DCF encryption protocol. In case of DCF encryption protocol, DCF file is used as a container for ciphered file data. The DCF container also identifies the keys used in protecting the data. Use of IPsec for Service Protection of download data is Optional.

Each file is encrypted using a single TEK, as explained in Section 9.4.

If a file is transmitted in a FLUTE carousel, the same key value of the TEK SHALL be used during the whole time the file is transmitted using the same TOI in an ongoing FLUTE session.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

For the Smartcard Profile, KeyID takes its values as follows:

- KeyID is defined as the base64 encoded concatenation of (SEK or PEK ID || TEK ID) (i.e. equivalent to Key Domain ID || SEK/PEK ID || TEK ID).

Keys can be acquired by using the PermissionsIssuerURI indicated via the KeyIssuerURL in the Key Info box.

6.8.2.2 Content Protection of Download Data using DCF for Smartcard Profile

BCAST terminals SHALL support download protection using DCF. BCAST servers MAY support download protection using DCF.

The same mechanism can be used to protect PDCF files. This is optional for both terminal and server.

The DCF format defined in Section 6.8.2.1 above can also be used for content protection for the Smartcard Profile. Content protection rules are identified to the terminal by the protection_after_reception value in the STKM, and to the Smartcard by the security_policy_extension value in the LTKM.

Keys can be acquired by using the PermissionsIssuerURI indicated via the KeyIssuerURL in the Key Info box.

OMA DRM v2.0 MAY be used for download content protection together with the Smartcard Profile.

6.9 Recording

Please refer to Section 8 for details on recording.

6.9.1 Playback of Content Protected Recorded Streams

The mechanisms described in this sub-section SHOULD be implemented for terminals using content protection with the Smartcard Profile.

This section describes how streamed content encrypted at the content level using ISMACryp and recorded in the adapted PDCF together with STKM track can be played back locally. Content protection is indicated to the terminal by the protection_after_reception value in the STKM.

1. Read the first STKM from the STKM track and send it to the Secure Function.
2. If the TEK is returned then decrypt the encrypted content. Otherwise, go to step 4. Note: This condition indicates that the SEK/PEK corresponding to the desired TEK is not available at the Secure Function, such that the terminal must request its delivery from the BSM.
3. Repeat 1 to 2 until the end of the file or until the TEK is not returned; (this is indicated by a failure message sent by the Secure Function). If the end of file is reached, then this procedure terminates
4. Read the PermissionsIssuerURI in the RightsIssuerURL field of the OMADRMCommon HeadersBox.
5. Identify the SEK/PEK from the recorded STKM track.
6. Identify the Timestamp field (TS) from the current STKM in the STKM track.
7. Identify the Timestamp (TS) from the last STKM in the STKM track.
8. If terminal is not in a registered state with the BSM identified by PermissionsIssuerURI, then perform a Registration procedure with this BSM. The Request-URI is this recorded PermissionsIssuerURI, appended by “requesttype” parameter set to “register”.
9. In case of BCAST Smartcard, request a Playback SPE to the BSM via a Service Provisioning procedure (Service Request, Token Purchase Request, Web portal). Note that in the current version of this specification, the purchase information needed to perform this Service Provisioning procedure (GlobalPurchaseItemID, PurchaseDataID, PurchaseURL, PortalURL,…) is not recorded in the Adapted PDCF. As such, it is the responsibility of the terminal, at

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

the time of content recording, to make sure it can later on be able to retrieve this purchase information at the time of playback. How the storage of this purchase information is achieved by the terminal is out of the scope of this version of the specification.

10. Receive the LTKM with the requested SEK/PEK (and eventually the playback SPE) from the BSM. go to step 1.

In case the mechanism described in this subsection is supported, the Secure Function is located on a Smartcard, i.e.as in the case of the GBA_U variant of the Smartcard Profile, and the STKM message is for playback of recorded content (i.e. SPE used for the processing of the STKM has one of the following value: SPE = 0x01 or SPE = 0x03 or SPE = 0x05 or SPE= 0x07 or SPE= 0x09 or SPE = 0x0D), then the corresponding TEK SHALL only be returned by the Smartcard if it has been secured by at least one of the following methods:

- By using the mechanism for Secure Channel described in Section 6.13, or
- By using the Terminal Binding Key (TBK) mechanism (as described in Section 12 and indicated through the terminal_binding_flag, described in Section 6.7.2.1).

If the Secure Channel is supported, then it SHALL be used.

The following must be noted when using the above mechanism:

- The MBMS replay protection mechanisms mean any “rewind” forces a new SEK/PEK request unless TEKs are buffered in the terminal. Hence buffering is recommended until end of play.
- The security policies extensions described in Section 6.6.4.2 permit the playback of content according a security policy associated to the SEK/PEK.
- The Timestamp (TS) lower and upper limits allow a finer management of rights on the server side rather than basing charging on the full duration of the program defined by the TEK ID.
- The Permissions Issuer (BSM) must keep a history of SEK/PEKs.
- The delivery of the STKM must only be done through a Secure Channel to ensure TEKs are returned via a secure channel and not in the clear, unless the Terminal Binding Key is used in which case the SAC is optional.

6.10 SG Signalling (Description of Service Access)

6.10.1 SG Signalling for SEK/PEK Acquisition

The Service Guide (SG) provided by OMA BCAST provides information regarding available services and allows a user to subscribe to or acquire purchase items. For example, information regarding available services is delivered via the Service fragments, and information regarding available purchase items is delivered via the PurchaseItem fragments and PurchaseData fragments. Each fragment contains its own unique identifier. GlobalPurchaseItemID is defined in the PurchaseItem fragment and PurchaseDataID is defined in the PurchaseData fragment. The concatenation of GlobalPurchaseItemID and PurchaseDataID results in a parameter equivalent to the MBMS User Service ID defined in MBMS. The MBMS User Service ID can be created as follows:

MBMS User Service ID = Base64Enc(GlobalPurchaseItemID) || “#” || Base64Enc(PurchaseDataID)

Where Base64Enc(id) represents Base64 encoding.

Having completed subscription/purchase of a purchase item/broadcast service, to subsequently enable rendering of service/content, the appropriate SEK/PEK must be acquired by the BCAST Terminal. The Access fragment clearly identifies the type of protection offered (service protection or content protection or both) and the supported key management systems, e.g. the DRM Profile or the Smartcard Profile. In the case of the Smartcard Profile two possible means exist for acquiring the appropriate SEK/PEK, via the LTKM:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- via the MBMS USD contained in the Session Description fragment (Section 6.10.1.1) or
- via session description information extended to include BCAST protection-specific information; these may be provided directly in the Access fragment or in the Session Description fragment (Section 6.10.1.2).

The PurchaseChannel fragment can be linked to a PurchaseItem fragment to provide further information via the PortalURL or indicate to the terminal that it must contact the PortalURL for any subscription (see Section 6.10.3), rather than send a Service Request directly to the PurchaseURL.

The association between the protected services or contents and the corresponding keys (SEK/PEK) necessary to access them is done via the "ProtectionKeyID" element, which can be present in the Service, Content or Purchase Item fragments. This allows the mapping of keys with a specific service, content or group of services/contents, respectively, so that the terminal can determine whether it already has valid access keys or not.

6.10.1.1 MBMS USD Method for Acquiring SEK/PEK

As specified in [BCAST10-SG], the SessionDescription fragment, referenced by the Access fragment, may contain MBMS User Service Description (USD), the latter specified by [3GPP TS 26.346 v7]. If the MBMS USD is used as the entry point, it SHALL contain the relevant service information required by the terminal to register to for the services that it is advertising. For convenience these steps are summarised below:

1. During the MBMS announcement procedure, the terminal receives the full domain name of the BSM (BM-SC) from which it can deduce the IP address to send the "Registration Request" and "LTKM Request" messages, as defined in [BCAST10-Services]. Note that the Smartcard Profile "Registration Request" and "LTKM Request" messages correspond to the MBMS "User Service Registration" and "MSK Request" messages respectively.
2. The terminal sends a "Registration Request" message to the BSM (BM-SC) for the services to which it is subscribed. As defined in [BCAST10-Services], the following information SHALL be included in the "Registration Request" message:

- Indication that the UE requests to register to the MBMS User Service;
- One or more MBMS User Service ID(s), where each MBMS User Service ID corresponds to the generating method (See Section 6.10.1), or one MBMS User Service ID corresponding to the value "oma-bcast-allservices".

In this situation the PermissionsIssuerURI contained in the Access fragment and the BaseCID contained in the Service fragment are to be ignored as the relevant parameters are provided in the MBMS USD fragments. This is summarised in the table below.

Table 26: Parameters used when using MBMS USD

Parameter	Value / Description
Session Description Fragment contents	MBMS USD.
PermissionsIssuerURI (in Access fragment)	Not used / ignored. MBMS USD contains a Service Protection Description, which identifies the key management server which the terminal should register with, and request SEK/PEK from.
BaseCID (in Service or Content fragment)	Not used / ignored as this applies to DRM Profile only. Note: Equivalent identifier in Smartcard Profile is provided in two possible ways: 1) in the MBMS USD (represented by the serviceID attribute of userServiceDescription element in the User Service Description; this serviceID is equivalent to the MBMS User Service ID); 2) in the PurchaseItem and PurchaseData fragments of the BCAST Service Guide [BCAST10-SG].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

6.10.1.2 Session Description Method for Acquiring SEK/PEK

In this scenario, session description information, either embedded in the Access fragment or provided in a standalone Session Description fragment, and containing Smartcard Profile specific protection information (in addition to nominal session information) is used. The session description is formatted according to the syntax of Session Description Protocol (SDP). The BCAST Service Guide provides the global purchase item identifier (*globalPurchaseItemID* of Purchase Item fragment) and purchase data identifier (*id* attribute of Purchase Data fragment). These two identifiers are used to create the MBMS User Service ID by using the method (See Section 6.10.1).

In this method, the SDP file provides information on the data and STKM streams, as well as other service protection parameters equivalent to those found in MBMS USD's Service Protection Description. This would typically be the case for a non-MBMS bearer used to deliver the data, with the interactive communication channel being used to provide LTKMs. The TEK delivery could be done in-band with the data. Depending on the bearer, this could be an MBMS or non-MBMS network.

Registration to the service is achieved by sending the "Service Registration" message as explained above in Section 6.10.1.1.

The relevant parameters are summarised in the table below.

Table 27: Parameters used when using Session Description

Parameter	Value / Description
Session Description information type	SDP in Access fragment, or SDP-formatted data in Session Description fragment referenced by Access fragment.
PermissionsIssuerURI (in Access fragment)	Used to reference the key management server, i.e. the BSM, which the terminal should register with and to request SEK/PEK from.
BaseCID	Not used / ignored as applies to DRM Profile only. Note: Equivalent identifier is provided in the PurchaseItem and PurchaseData fragments of the BCAST Service Guide [BCAST10-SG].

6.10.2 Description of Service Access for Smartcard Profile using BCMCS Information Acquisition

Section 5 of BCMCS specification [3GPP2 X.S0022-A] describes BCMCS service discovery, subscription and registration procedures using BCMCS information acquisition process. For terminals with (R)-UIM smartcards, those procedures are used for OMA BCAST application information and security parameters, including keys. The PermissionsIssureURI may be pre-provisioned or acquired via BCMCS Information Acquisition exchange(s).

When using BCMCS Information Acquisition, the PermissionsIssuerURI contained in the Access fragment and the BaseCID contained in the Service fragment are to be ignored as the relevant parameters are provided in the BCMCS Information Acquisition elements. This is summarised in the table below.

Table 28: Parameters used when using BCMCS Information Acquisition

Parameter	Value / Description
Session Description Reference Type	BCMCS Information Acquisition
PermissionsIssuerURI	Not used / ignored. Such information is provided by BCMCS Security Parameters.
BaseCID	Not used / ignored.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	BCMCS Application Information and BCMCS Security Parameters are linked.

6.10.3 Web Portal used as Entry Point

While the Service Guide can provide all the information to obtain information on available services as well as information relating to acquisition of LTKMs, as explained above, another possibility for terminals having access to an interaction channel is to use a Web Portal.

If the PortalURL in the PurchaseChannel fragment linked to a PurchaseItem indicates that the PortalURL should be contacted to obtain further information and subscribe to services, the terminal SHOULD contact the PortalURL. The supportedService value under PortalURL element of the PurchaseChannel fragment indicates the expected behaviour of the terminal regarding service provisioning. Furthermore, while initiating the access to the Portal, provided that the following information is available to the terminal (e.g from the Service Guide), the terminal MAY send the GlobalPurchaseItemID, and MAY include the idRef of the PurchaseData fragment, to the web portal associated with the PortalURL to indicate to the portal the PurchaseItem (or PurchaseData) of interest. The (optional) sending of the PurchaseData idRef enables the user to identify a specific pricing option for the purchase item of interest, obtained from the Service Guide. For example, that could represent the lowest among different price offers from multiple broadcast service providers with which the user maintains business relationships.

When the user attempts to subscribe to a service via the portal, the portal is unable to determine whether or not the Smartcard/Terminal has established a valid SMK and SRK with the BSM (e.g. whether or not the bootstrapping procedure has been run in the case of (U)SIM, or whether TK and Auth-Key have been derived from the pre-provisioned RK in the case of (R-)UIM/CSIM). Once the terminal has completed the web-based purchase, the portal informs the BSM of the completed transaction (via means that are outside the scope of this specification), and the BSM then sends the Terminal a BSM solicited pull messag as defined in section 5.1.8 of [BCAST10-Services] to force the terminal to run the Registration procedure, which in turn requires that the bootstrapping procedure has been run in the case of (U)SIM. The message flow for this scenario is described in section 5.4.7.3.2 of [BCAST10-Architecture].

6.11 BCAST Client ID for Smartcard Profile

This section describes how a BCAST Client identifier MAY be sent by the Terminal or MAY be requested by the BSM (Permissions Issuer) during MBMS User Service Registration.

This MAY allow the BSM (Permissions Issuer) to check software / firmware versions and make a decision as to whether or not access can be granted to the terminal requesting the service.

The mechanisms described in this section are OPTIONAL for the network to use and MANDATORY for the terminal to support if they have a BCAST client ID, a terminal certificate and if they support the Smartcard Profile for service protection. The mechanisms are MANDATORY for the terminal to support for the Smartcard Profile for content protection, i.e. the BCAST client ID and terminal certificate are MANDATORY.

Security: Message integrity and authentication is guaranteed by using certificate-based mutual authentication between Terminal and Application Server for access to NAF using HTTPS as specified in [3GPP TS 33.222 v6], where HTTPS SHALL be based on .SSL3.0 [SSL30] and TLS 1.0 [RFC2246].

6.11.1 BCAST Client Identifier

The format defined below SHALL be used as a unique BCAST client identifier for the Smartcard Profile.

The BCAST_Client_ID SHOULD be stored in the BCAST Management Object (BCAST MO) as specified in [BCAST10-Services]. Note that the BCAST_Client_ID may be stored elsewhere in the device.

Note that it is NOT mandatory for every terminal to have a BCAST Client Identifier for service protection, it is only MANDATORY for content protection.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 29: BCAST Client ID

BCAST_Client_ID	Length	Type
TerminalIdentifierType	1	byte
TerminalIdentifier	16	byte
TerminalFirmwareVersionNo	2	byte
ClientManufacturerCode	2	byte
ClientModelNo	2	byte
ClientSerialNo	3	byte
ClientSoftwareVersionNo	2	byte

Coding and Semantics of Attributes:

The Terminal identifiers are specific to the actual device used to receive mobile broadcast services and are defined in the Table below:

Table 30: Terminal Identifiers

Parameter	Definition	
TerminalIdentifierType	**Value**	**Type**
	0	IMEI (International Mobile Equipment Identity) as defined in [3GPP TS 23.003 v6].
	1	MEID (Mobile Station Equipment Identifier) as defined in [3GPP2 C.S0072-0]
	2	Globally Unique Identifier (GUID)
	3	Media Access Control (MAC) address in EUI-48 or EUI-64 format
	4-127	*for future use*
	128-255	*for private use*
TerminalIdentifier	The identifier of the terminal, in the format specified through TerminalIdentifierType. Identifiers that occupy less space than 16 bytes are padded with leading zeros to fill 16 bytes after padding.	
TerminalFirmwareVersion	Version number indicating the firmware version of the terminal. This version number is assigned by the Terminal manufacturer. This version number SHALL be increased following a secure firmware upgrade.	

The Client identifiers are specific to the BCAST client installed in the Terminal allowing access to the BCAST services and are indicated in the Table below:

Table 31: BCAST Client Identifiers

Parameter	Definition
ClientManufacturerCode	Indicates the BCAST client manufacturer. Values for ClientManufacturerCode are available in an OMNA [OMNA] registry.
ClientModelNo	Model number for a specific manufacturer code. Numbering assignment is left to the manufacturer.
ClientSerialNo	Unique serial number specific to the BCAST client manufacturer code and model number. Serial number assignment is left to the manufacturer.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

	Note that this is unique for a given ClientManufacturerCode and ClientModelNo pair
ClientSoftwareVersion	Version number indicating the software (or firmware) version of the terminal. This version number is assigned by the BCAST client manufacturer. This version number SHALL be increased following a secure software (or firmware) upgrade.

6.11.2 Signalling Protocols used for Smartcard Profile

This section explains how the BCAST_Client_ID presented above MAY be sent or requested for the Smartcard Profile during the BCAST service provisioning message sequence or Registration procedure as defined in [BCAST10-Services]. The Figures below summarise the possible messages exchanged. Italics are used to indicate the parameters / messages related to the BCAST_Client_ID. The first Figure illustrates the case where the BSM/Permissions Issuer requests the BCAST_Client_ID. The second Figure illustrates the case where the Terminal sends its BCAST_Client_ID to the BSM/Permissions Issuer.

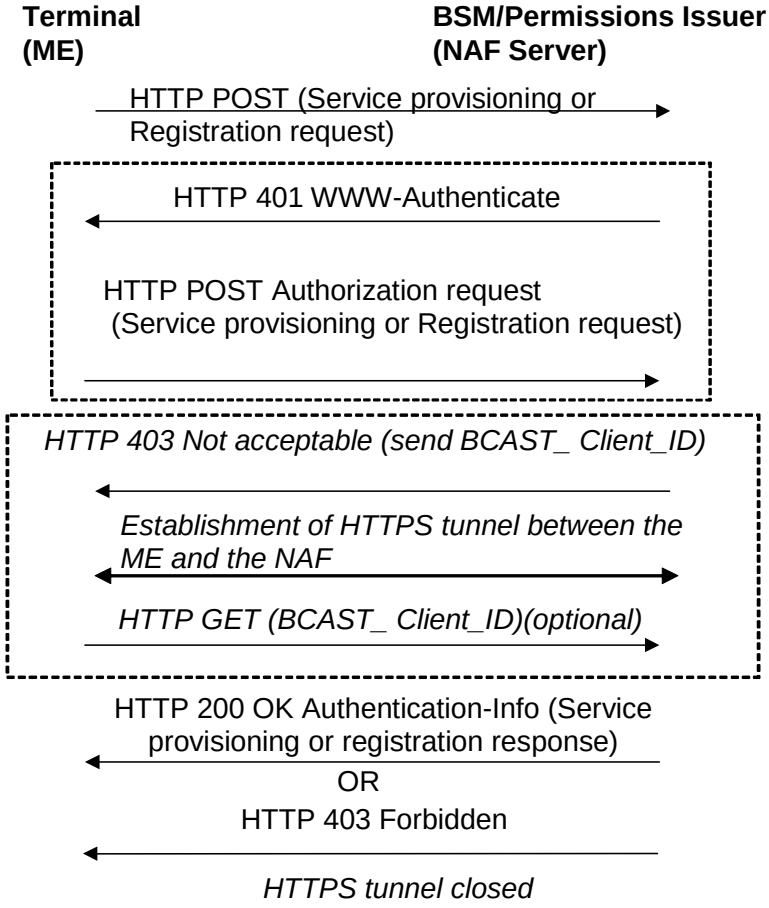


Figure 6 – Mutual Authentication and BCAST Service Provisioning or Registration Messages when BSM/Permissions Issuer requests BCAST_Client_ID

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

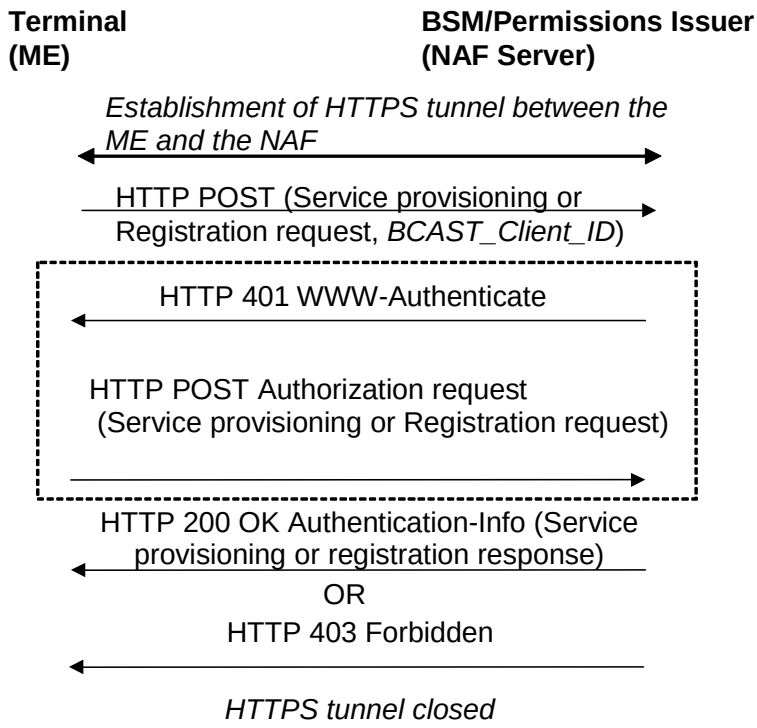


Figure 7 – Mutual Authentication and BCAST Service Provisioning or Registration Messages when Terminal sends BCAST_Client_ID

The following sections describe the messages in more detail.

6.11.2.1 Certificate-based Mutual Authentication between Terminal and BSM/Permissions Issuer

Before initiating a BCAST Service Provisioning message or Registration request, the Terminal and the BSM/Permissions Issuer MAY establish a HTTPS tunnel with certificate-based mutual authentication between the Terminal and the application server as described in TS [3GPP TS 33.222 v6] Section 5.5 “Certificate based mutual authentication between Terminal and application server” for Release 6. This SHALL be done if the Terminal intends to send a BCAST Client ID at the start of BCAST Service Provisioning message or Registration request as described below.

6.11.2.2 Terminal Sending BCAST_Client_ID at start of BCAST Service Provisioning Message or Registration Request

The BCAST_Client_ID identifier MAY be sent automatically by the Terminal in the initial HTTP Post Request during the start of the BCAST Service Provisioning message or Registration request as defined in [BCAST10-Services]. If it sends a BCAST_Client_ID it SHALL do so through an HTTPS tunnel, as described above in Section 6.11.2.

The BCAST_Client_ID SHALL be sent using the following notation:

User-Agent:BCAST_Client_ID=*BCAST_Client_ID*

Where:

"BCAST_Client_ID=" is text allowing the server to identify the BCAST client ID and *BCAST_Client_ID* is the actual value.

BCAST_Client_ID is Base64 encoded.

If the Terminal does not send the BCAST_Client_ID in the HTTP Post Request, then the BSM/Permissions Issuer MAY ask for it as described below in Section 6.11.2.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

6.11.2.3 BSM/Permissions Issuer Requesting BCAST_Client_ID

If the Terminal has NOT sent the BCAST_Client_ID in the HTTP Post Request, the BSM/Permissions Issuer MAY request the BCAST client identifier using the following request:

Table 32: BSM/Permissions Issuer Requesting BCAST_Client_ID

```
HTTP/1.1 403 Not acceptable
Server: BCAST BSM
Date: Thu, 08 Jan 2004 10:13:18 GMT
send_BCAST_Client_ID
```

where send_BCAST_Client_ID is text.

6.11.2.4 Terminal Sending BCAST_Client_ID to BSM/Permissions Issuer following Request

Upon reception of the request for the BCAST_Client_ID, the terminal MAY be incapable of sending this identifier (as it is optional).

If the terminal recognizes the request for a BCAST_Client_ID (as defined in Section 6.11.1), it SHOULD establish an HTTPS tunnel between the Terminal and BSM/Permissions Issuer as described in Section 6.11.2 and then send the BCAST_Client_ID using the following response:

Table 33: Terminal Sending BCAST_Client_ID to BSM/Permissions Issuer

```
GET / HTTP/1.1
User-Agent: BCAST_Client_ID=BCAST_Client_ID
Date: Thu, 08 Jan 2004 10:13:18 GMT
```

Where "BCAST_Client_ID=" is text allowing the server to identify the BCAST_Client_ID and *BCAST_Client_ID* is the actual value.

BCAST_Client_ID is Base64 encoded.

If the BCAST Client does not have a BCAST_Client_ID it is recommended that the above message is sent using an empty User-Agent entry without establishing an HTTPS tunnel. Note that this may result in the BSM/Permissions Issuer refusing registration.

6.11.2.5 BSM/Permissions Issuer Accepting BCAST_Client_ID

If the BCAST_Client_ID provided by the terminal to the BSM/Permissions Issuer is deemed acceptable, then the normal BCAST Service Provisioning or Registration Response message is sent, i.e. HTTP 200 OK as defined in [BCAST10-Services].

6.11.2.6 BSM/Permissions Issuer Refusing Access to Terminal

If the Terminal does not send a BCAST_Client_ID following the request from the BSM/Permissions Issuer, it MAY refuse access to the Terminal by sending an HTTP 403 Forbidden message.

If the BSM/Permissions Issuer refuses access to the Terminal after receiving its BCAST_Client_ID, it SHALL send an HTTP 403 Forbidden message.

6.11.3 Security Requirements on BCAST_Client_ID and Terminal Private Key

The BCAST_Client_ID SHALL be stored securely and updated accordingly following secure upgrades.

The Terminal private key SHALL also be stored securely.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

If the BCAST_Client_ID is used by the BSM/Permissions Issuer to check the BCAST client version, then clearly the information provided by the terminal must be trusted. The BCAST_Client_ID SHALL be transported over a secure, authenticated channel between the terminal and the BSM/Permissions Issuer, as described in [3GPP TS 33.222 v6].

6.12 Terminal-Smartcard Interface

The interface between the Terminal and the 3GPP (U)SIM smartcard SHALL comply with the specifications in [3GPP TS 31.101 v7], [3GPP TS 31.102 v7], [ETSI TS 102.221] and [3GPP TS 31.111 v9.1.0]. Specific commands defined for OMA BCAST and OMA BCAST extensions to MBMS commands are defined in Appendix E of this specification.

6.12.1 Use of the Secure Channel between the Smartcard and the Terminal

The text in this section only applies when the secure function is located in the Smartcard.

Support for the secure channel, as defined in Section 6.12, is optional for both the terminal and Smartcard. Support of the BSIM, as defined in Appendix E.5, is a pre-condition of support of the Secure Channel as specified in this document.

If the terminal and the Smartcard support the secure channel, the secure channel SHALL be established before any BCAST messages, as defined in Appendix E.3 and Appendix E.4, are sent or received by the Terminal. Once the Secure Channel is established, all BCAST commands SHALL be transported through mechanism defined in [ETSI TS 102.484].

If either the terminal or Smartcard do not support the Secure Channel, then it SHALL be possible for the terminal to send and receive BCAST messages without establishing a Secure Channel provided that the signalling in STKM messages allows communication of the TEK in the clear. To provide this capability the terminal and Smartcard SHALL process STKMs as follows:

- If a secure channel has not been set up and either the secure_channel_ flag in the STKM is set to TKM_FLAG_TRUE, or the STKM message is for PLAYBACK of recorded content (i.e. SPE used for the processing of the STKM has one of the following value: SPE = 0x01 or SPE = 0x03 or SPE = 0x05 or SPE= 0x07 or SPE= 0x09 or SPE = 0x0D) with a the terminal_binding_flag set to TKM_FLAG_FALSE, the Smartcard SHALL return the operation status code 0x13 'Security policy not satisfied; Secure Channel is required' in the response of the AUTHENTICATE command containing the STKM.
  To avoid unnecessary communication and processing of STKMs, if the terminal receives the status code 0x13 in response to an STKM and the secure_channel_flag in subsequent STKMs protected by the same SEK is set to TKM_FLAG_TRUE, the terminal SHOULD not send the STKM to the Smartcard.

- If a secure channel has not been set up and the secure_channel_flag in the STKM is set to TKM_FLAG_FALSE, and the STKM message is not for PLAYBACK of recorded content (i.e. SPE used for the processing of the STKM has one of the following value: SPE=0x00 or SPE=0x02 or SPE=0x04 or SPE=0x08 or SPE=0x0C), the Smartcard SHALL process the STKM based on the relevant LTKM security_policy_extension.

6.13 Secure Channel between the Smartcard and the Terminal

The text in this section only applies when the secure function is located in the Smartcard.

The following text describes how the ETSI “Smart Cards; Secure channel between a UICC and an end-point terminal”, as defined in [ETSI TS 102 484], can be used by the Smartcard Profile to secure the communication of TEKs between the Smartcard and the terminal. The protection of TEKs over of the Smartcard – terminal interface is especially relevant in the context of content protection (see Section 6.8.1.2).

[ETSI TS 102 484] specifies the architecture, functional capabilities and characteristics of a Secure Channel protocol for securing communication between an end-point on the terminal and an end-point on a Smartcard.

The Secured APDU – Application to Application secured data transport protocol, as defined in [ETSI TS 102 484], is OPTIONAL for the Smartcard and Terminal to implement.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Terminal and Smartcard implementations that support the Secure Channel SHALL use the Secured APDU – Application to Application secured data transport protocol, as defined in [ETSI TS 102 484].

The Terminal SHALL determine Smartcard support for the Secure Channel as defined in [ETSI TS 102 221].

If a logical channel has not been set up between the Terminal and the BSIM, then a new logical channel SHALL be set up for the Secure Channel. Once a Secure Channel has been established, all communication between the Terminal and the BSIM SHALL take place over the Secure Channel.

The Terminal SHALL use the "Manage Secure Channel APDU – Retrieve UICC Endpoints" command to discover the endpoint on the UICC. The terminal SHALL only attempt to use the Secure Channel to secure the transportation of Smartcard Profile messages if an endpoint with the UICC_appli_ID set to the OMA BCAST AID is indicated in the response to the "Manage Secure Channel APDU – Retrieve UICC Endpoints" command. For the (U)SIM Smartcard Profile, the Strong Pre-shared Keys – GBA mechanism, as defined in [ETSI TS 102 484], SHALL be used to setup the security context of the secured data transport protocol. The Strong Pre-shared Keys – GBA mechanism uses the key agreement procedures defined in [3GPP TS 33.110 v7] to establish a shared key between the UICC and the Terminal. For the CSIM Smartcard Profile, the mechanism to setup the security context of the secured data transport protocol is to be defined in a future, relevant 3GPP2 specification.

The following values SHALL be used for the parameters used in the setup of the security context:

UICC_ID: Content of EF_{ICCID} under the MF, as defined in [ETSI TS 102 221];

UICC_appli_ID: BSIM Application Identifier (AID) as defined in Appendix E.5.1;

Terminal_ID: The IMEI of the Terminal in case of the (U)SIM Smartcard Profile, and MEID of the Terminal in case of the CSIM Smartcard Profile;

Terminal_appli_ID:

Byte	Content / Description	Value
Specification_Reference		
1-2	Tag and length of SpecificationReference	06 04

3-6	SpecificationReference:= OID {
	joint-isu-itu-t (2)
	identified-organizations (23)
	wap (43)
	oma-bcast (8)
	oma-bcast-spcp(1)
	}
		67 2B 08 01

Terminal_appli_ID		
7-8	Tag and length of BCAST Terminal_appli_ID	04 08
9	BCAST Terminal_appli_ID coding scheme	01
10	BCAST version 1.0 (Incremented for each major and minor specification version change.)	01
11-12	BCAST client manufacturer code as allocated at [OMNA]	defined by OMNA

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13-14	Client manufacturer extension	defined by client manufacturer
15-16	Client version	defined by client manufacturer

For the (U)SIM Smartcard Profile, the Key Lifetime and Counter Limit parameters are set during the establishment of the key Ks_local, as described in [3GPP TS 33.110 v7]. For the CSIM Smartcard Profile, the Key Lifetime and Key Counter Limit parameter values will be defined in a future, relevant 3GPP2 specification. The Counter Limit parameter SHALL be used, as defined in [ETSI TS 102 484], to determine the maximum number of Master Sas that can be derived from the key Ks_local, the maximum number of Connection Sas that can be derived from a Master SA, and the maximum number of transactions that can be handled within a Connection SA.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

7. Short Term Key Message – Common Attributes

STKMs of the DRM Profile and the Smartcard Profile share a set of common attributes. These common attributes are introduced below.

Section 7.1 introduces the descriptors for access_criteria_descriptor_loop. Section 7.2 introduces used constant values. Section 7.3 introduces coding and semantics of the common STKM attributes.

7.1 Descriptors for access_criteria_descriptor_loop

Tag	8	uimsbf
Length	8	uimsbf
Value	8xlength	bit string

The Access Criteria Descriptor loop is an extension mechanism to allow the addition of new access criteria in the future versions of this specification. The device SHALL ignore Access Criteria Descriptors that it doesn't support. It is OPTIONAL for the BCAST Terminal to support Access Criteria Descriptors.

A single Access Criteria Descriptor can carry one or more access criteria.

The following Access Criteria Descriptors have been defined:

- Parental_rating
- Location_based_restriction

Note:

In case of Smartcard Profile, STKM can not contain more than one Parental_rating Access Criteria Descriptor

For both profiles, STKM can not contain more than one Location_based_restriction Access Criteria Descriptor

7.1.1 Parental_rating Descriptor

This descriptor is for the parental rating of the program. The descriptor tag for this descriptor is 1. The value for this descriptor is encoded as follows:

Table 34: parental_rating Access Criteria Descriptor

parental_rating descriptor	Length (in bits)	Type
rating_type	**7**	**uimsbf**
country_code_flag	**1**	**uimsbf**
rating_value	**8**	**uimsbf**
if (country_code_flag == TKM_FLAG_TRUE) {		
number_of_country_codes	**8**	**uimsbf**
for (I = 0; i < number_of_country_codes; i++) {		
country_code	**16**	**uimsbf**
}		
}		

The optional list of **country_code** specifies that the rating is for a specific list of one or more countries, which is analogous to the MPEG-7 definition of the ParentalGuidanceType. Each country code consists of two uppercase ASCII alpha characters and MUST be compliant with [ISO-3166].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The **rating_type** designates the content rating systems, and the **rating_value** is an integer with a meaning that is dependent on the rating_type. The rating values and rating types are registered in the OMA BCAST Parental Rating System Registry. The registry is available at [OMNA].

A special BCAST rating_type is specified in the OMA BCAST Parental Rating System Registry to allow the implementation of non-registered parental rating schemes. This scheme is called "BCAST-generic parental rating" and is defined as follows:

- the rating_type is 10
- the rating_value field can assume the values from 0 to 255, where 0 means "Not rated", and the degree of restrictiveness is growing monotonically between 1 and 255, i.e. 1 is the least restrictive value and 255 is the most restrictive value according to Section 6.7.3.11.1.

7.1.2 Location_based_restriction Descriptor

This descriptor is for the location-based restrictions on the rendering of content based on [BCAST10-SG].

An alternative service can be specified in the service guide [BCAST10-SG]. It is possible to specify an alternative service as a blank screen with a burnt-in text notifying the user of the blackout. In this case, the burnt-in text can be conveyed as a subtitle in 3GPP Timed Text format as described in [BCAST10-Services].

If a terminal supporting a location_based_restriction descriptor receives an STKM with this descriptor and the terminal is not able to obtain its current location or is not able to process the STKM, then the terminal MUST NOT decrypt the Traffic Key and possible Program Key contained in this STKM and MUST NOT decrypt the corresponding content. A terminal MAY be capable of determining at least its cell ID using a native bearer signalling mechanism. A terminal MAY in addition utilize a suitable position location protocol to determine its position. Examples are SUPL [OMA SUPL] or MLP [OMA MLP]. In the case that a terminal is not capable of determining location information other than a cell ID, additional location information (other than a list of blacked out cell IDs) provided in the location_based_restriction Access Criteria Descriptor MAY be ignored. In the case that the terminal is able to detect multiple cell IDs using native bearer signalling mechanisms, for the purpose of checking against a possible blackout it MAY select the same cell ID that is being used to receive the protected service.

The descriptor tag for this descriptor is 2. The value for this descriptor is encoded as follows:

Table 35: location_based_restriction Access Criteria Descriptor

location_based_restriction descriptor	Length	Type
version	32	Uimsbf
polarity	1	Uimsbf
override	1	Uimsbf
reserved_for_future_use	6	Bslbf
lev_conf	8	uimsbf
number_of_target_areas	8	uimsbf
for (i=0; i < number_of_target_areas; i++) {		
target_area_type	4	bslbf
reserved_for_future_use	4	bslbf
if (target_area_type == 0x1) {		
shape()		
}		
if (target_area_type == 0x2) {		
mobile_country_code	24	uimsbf
}		
if (target_area_type == 0x3) {		
name_area_length	8	uimsbf
name_area	8*name_area_length	bslbf
}		
if (target_area_type == 0x4) {		
zip_code_length	8	uimsbf
zip_code	8*zip_code_length	bslbf
}		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

if (target_area_type == 0x5) {		
cell_target_area()		
}		
}		

version – tells the terminal if the contents of this descriptor have changed since the last STKM. When the version number is the same as in previous STKMs and the terminal has already processed a descriptor with this same version number, it MAY ignore the contents of this descriptor and assume that geographical restrictions are the same as in previous STKMs with this same descriptor version number.

polarity – flag specifying how the restriction is interpreted. If set to 0 ("normal"), a terminal residing within the defined area may not render the associated media streams. If set to 1 ("spotbeam"), a terminal located outside of the defined area may not render the associated media streams.

override – flag specifying whether the location-based restriction may be ignored by a properly authorized terminal. If set to 0, the restriction must be obeyed. If set to 1, the restriction may be ignored if the terminal is able to obtain the necessary permissions (e.g., PPV Rights Object for the corresponding Program Key).

The override option allows the service provider to signal to the terminal that it may render restricted content regardless of its physical location, as long as the terminal has been pre-authorized to do so. This could also be used to notify unauthorized terminals of the ability to purchase rights to circumvent the restriction.

BCAST 1.0 does not specify how the terminal and Smartcard determine if the restriction can be ignored when the override flag is set to 1. Therefore, BCAST 1.0 terminals and Smartcards SHALL always interpret the override flag as if it were set to 0.

lev_conf – the target level of confidence that the terminal is indeed located within the indicated 'TargetArea' as defined in [OMA MLP]. Valid values are from 0 to 100, and 255. The value 255 indicates that lev_conf is undefined.

number_of_target_areas – the number of TargetAreas that define the geographical area. TargetArea is specified in OMA Service Guide for Mobile Broadcast Services [BCAST10-SG] as an XML element. It is adapted here with modifications such that it can be used for blackout restrictions.

target_area_type – specifies the type of area as specified in [BCAST10-SG]. The following values are possible:

0x1 = shapes used to represent a geographic area as defined by the shape descriptor below

0x2 = mobile country code, 3 ASCII digits, e.g. 276 for Albania as specified in [ITU-MCC]

0x3 = geopolitical name of area such as "Seoul" as specified in [OMA MLP]

0x4 = zip code

0x5 = a set of "cell_area_values" as defined by the cell_target_area descriptor below.

shape – adapted from shapes used to represent a geographical area specified in [OMA MLP]. The value for this descriptor is encoded as follows:

Table 36: shape Descriptor

shape descriptor	Length	Type
shape_type	4	bslbf
reserved_for_future_use	4	bslbf
if (shape_type == 0x3) {		
shape_polygon()		
}		
if (shape_type == 0x5) {		
shape_circular_area()		
}		
if (shape_type == 0x7) {		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

shape_elliptical_area()		
}		

shape_type–- specifies the type of shape as specified in [OMA MLP]. The following values are possible:

0x3 = a polygon

0x5 = a circular area

0x7 = an elliptical area

Note: Some shape types defined in [OMA MLP] are not included as they are not as applicable for use in blackout restriction.

shape_polygon – a polygon, which is a connected surface defined by an outer boundary and zero or more inner boundaries. The value for this descriptor is encoded as follows:

Table 37: shape_polygon Descriptor

shape_polygon descriptor	Length	Type
outerBoundarys		shape_linear_ring
number_of_innerBoundarys	8	uimsbf
for (i=0; i < number_of_innerBoundarys; i++) {		
innerBoundarys		shape_linear_ring
}		

Note: The shape_polygon, adapted from [OMA MLP], is a super-set of the polygon definition in [3GPP TS 23.032 V6].

outerBoundarys – the outer boundary of the polygon, defined by the shape_linear_ring type.

number_of_innerBoundarys – the number of inner boundaries, has to be larger than or equal to 0.

innerBoundarys – an inner boundary of the polygon, defined by the type shape_linear_ring.

shape_linear_ring – a closed, simple piece-wise linear path which is defined by a list of coordinates that are assumed to be connected by straight line segment. The value for this descriptor is encoded as follows:

Table 38: shape_linear_ring Descriptor

shape_linear_ring descriptor	Length	Type
number_of_coords	8	uimsbf
for (i=0; i < number_of_coords; i++) {		
coord()		
}		

number_of_coords – the number of coordinate points that define the linear ring. The number has to be larger than or equal to 3.

coord – a geographical coordinate. In [OMA MLP], a coordinate is specified by the tuple (x, y, z). The "z" component (specifying the altitude, if present) is not included as it is not applicable for blackout restriction use. For simplicity, the "x" and "y" components are represented as latitude and longitude as defined in [3GPP TS 23.032 V6]. The value for this descriptor is encoded as follows.

Table 39: coord Descriptor

coord descriptor	Length	Type
latitude	24	bslbf
longitude	24	bslbf

latitude – latitude coordinate, encoded as defined in [3GPP TS 23.032 V6].

longitude – longitude coordinate, encoded as defined in [3GPP TS 23.032 V6].

The definitions of latitude and longitude in [3GPP TS 23.032 V6] are quoted below for completeness:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- The latitude is coded with 24 bits: 1 bit of sign and a number between 0 and 2^{23}-1 coded in binary on 23 bits. The relation between the coded number N and the range of (absolute) latitudes X it encodes is the following (X in degrees):

$$N \leq \frac{2^{23}}{90} X < N+1$$

  except for N=2^{23}-1, for which the range is extended to include N+1.

- The longitude, expressed in the range -180°, +180°, is coded as a number between -2^{23} and 2^{23}-1, coded in 2's complement binary on 24 bits. The relation between the coded number N and the range of longitude X it encodes is the following (X in degrees):

$$N \leq \frac{2^{24}}{360} X < N+1$$

shape_circular_area – a set of points on the ellipsoid which are at a distance from the point of origin less than or equal to the radius. The value for this descriptor is encoded as follows:

Table 40: shape_circular_area Descriptor

shape_circular_area descriptor	Length	Type
origin		coord
radius	16	uimsbf
distance_unit	2	uimsbf
reserved_for_future_use	6	bslbf

Note: shape_circular_area corresponds to the type "ellipsoid point with uncertainty circle" specified in [3GPP TS 23.032 V6].

origin – specifies the coordinate of the origin.

radius – specifies the length of radius of the circular area.

distance_unit – specifies the distance unit used. The following values are possible:

0x0 = meter

0x1 = kilometer

0x2 = yard

0x3 = mile

shape_elliptical_area – a set of points on the ellipsoid, which fall within or on the boundary of an ellipse. The value for this descriptor is encoded as follows:

Table 41: shape_elliptical_area Descriptor

shape_elliptical_area descriptor	Length	Type
origin		coord
angle	10	uimsbf
semi_major	16	uimsbf
semi_minor	16	uimsbf
angular_unit	2	uimsbf
distance_unit	2	uimsbf

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

reserved_for_future_use	2	bslbf

Note: shape_elliptical_area corresponds to the type "ellipsoid point with uncertainty ellipse" specified in [3GPP TS 23.032 V6].

origin – specifies the coordinate of the origin.

angle – specifies the angle of the ellipse.

semi_major – specifies the length of the semi major.

semi_minor – specifies the length of the semi minor.

angular_unit – specifies the angular unit used. The following values are possible:

0x0 = degree

0x1 = grad

mobile_country_code – mobile country code, 3 ASCII digits, e.g. 276 for Albania as specified in [ITU-MCC].

name_area_length – number of bytes used to encode the name_area field.

name_area – a geopolitical name of area as specified in [OMA MLP].

zip_code_length – number of bytes used to encode the zip_code field.

zip_code – zip code represented by a character string.

cell_target_area – the target area defined by a set of cell IDs or other area identifiers. The value for this descriptor is encoded as follows:

Table 42: cell_target_area Descriptor

cell_target_area descriptor	Length	Type
cell_target_area_type	8	uimsbf
descriptor_length	20	uimsbf
reserved_for_future_use	4	bslbf
number_of_cell_area_values	16	uimsbf
for (i=0; i < number_of_cell_area_values; i++) {		
if (cell_target_area_type == 0x0) {		
cell_area_value_length	8	uimsbf
cell_area_value	8*cell_area_value_length	bslbf
}		
if (cell_target_area_type == 0x1) {		
3gpp_mcc	12	bslbf
3gpp_mnc	12	bslbf
3gpp_lac	16	bslbf
3gpp_ci	16	bslbf
}		
if (cell_target_area_type == 0x2) {		
3gpp_mcc	12	bslbf
3gpp_mnc	12	bslbf
3gpp_lac	16	bslbf
3gpp_rac	8	bslbf
}		
if (cell_target_area_type == 0x3) {		
3gpp_mcc	12	bslbf
3gpp_mnc	12	bslbf
3gpp_lac	16	bslbf
}		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

if (cell_target_area_type == 0x4) {		
3gpp_mcc	12	bslbf
3gpp_mnc	12	bslbf
3gpp_lac	16	bslbf
3gpp_sac	16	bslbf
}		
if (cell_target_area_type == 0x5) {		
mbms_sai	16	uimsbf
}		
if (cell_target_area_type == 0x6) {		
3gpp2_subnet_id_length	8	uimsbf
3gpp2_subnet_id	1*3gpp2_subnet_length	bslbf
padding_bits	(see definition below)	
}		
if (cell_target_area_type == 0x7) {		
reserved_for_future_use	1	bslbf
3gpp2_sid	15	bslbf
}		
if (cell_target_area_type == 0x8) {		
reserved_for_future_use	1	bslbf
3gpp2_sid	15	bslbf
3gpp2_nid	16	bslbf
}		
if (cell_target_area_type == 0x9) {		
reserved_for_future_use	1	bslbf
3gpp2_sid	15	bslbf
3gpp2_nid	16	bslbf
3gpp2_pzid	8	bslbf
}		
if (cell_target_area_type == 0xA) {		
reserved_for_future_use	1	bslbf
3gpp2_sid	15	bslbf
3gpp2_pzid	8	bslbf
}		
if (cell_target_area_type == 0xB) {		
dvbh_network_id	16	uimsbf
dvbh_cell_id	16	uimsbf
dvbh_hierarchy	2	bslbf
reserved_for_future_use	6	bslbf
number_of_subcell	8	uimsbf
for (j=0; j < number_of_subcell_id; j++) {		
dvbh_cell_id_extension	8	uimsbf
}		
}		
if (cell_target_area_type == 0x6) {		
number_of_3gpp2_cell_ids	16	uimsbf
for (j=0; j < number_of_3gpp2_cell_ids; j++) {		
3gpp2_cell_id	32	uimsbf
}		
}		
if (cell_target_area_type == 0x7 || 0x8 || 0x9 || 0xA) {		
number_of_3gpp2_cell_ids	16	uimsbf
for (j=0; j < number_of_3gpp2_cell_ids; j++) {		
3gpp2_cell_id	16	bslbf
}		
}		
}		

cell_target_area_type – specifies the cell_target_area type as defined in [BCAST10-SG]. The following values are possible:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

0x0 = Unspecified

0x1 = 3GPP Cell Global Identifier as defined in [3GPP TS 23.003 v6]

0x2 = 3GPP Routing Area Identifier as defined in [3GPP TS 23.003 v6]

0x3 = 3GPP Location Area Identifier as defined in [3GPP TS 23.003 v6]

0x4 = 3GPP Service Area Identifier (SAI) as defined in [3GPP TS 23.003 v6]

0x5 = 3GPP MBMS Service Area Identity (MBMS SAI) as defined in [3GPP TS 23.003 v6]

0x6 = 3GPP2 Subnet ID as defined in [3GPP2 X.S0022-A]

0x7 = 3GPP2 SID as defined in [3GPP2 C.S0005-D]

0x8 = 3GPP2 SID+NID as defined in [3GPP2 C.S0005-D]

0x9 – 3GPP2 SID+NID+PZID as defined in [3GPP2 C.S0005-D]

0xA = 3GPP2 SID+PZID as defined in [3GPP2 C.S0005-D]

0xB = DVB-H Cell ID (specified in section 6.3.4.1 of [BCAST10-DVBH-IPDC-Adaptation])

descriptor_length – the length of the descriptor, in bytes.

number_of_cell_area_values – specifies the number of cell_area_value included in the target area.

cell_area_value_length – specifies the length (in bytes) of the cell_area_value field.

cell_area_value – identifies a generic cell area used when the cell_target_area_type is 0x0. The format of this value is not defined.

3gpp_mcc – Mobile Country Code used for 3GPP networks that identifies the country in which the GSM PLMN is located as defined in [3GPP TS 23.003 v6]. Coding of this field is defined in [3GPP TS 23.003 v6].

3gpp_mnc – Mobile Network Code used for 3GPP networks that identifies the GSM PLMN in the country defined by 3gpp-mcc as defined in [3GPP TS 23.003 v6]. Coding of this field is defined in [3GPP TS 23.003 v6].

3gpp_lac – Location Area Code used for 3GPP networks that identifies a location area within a PLMN as defined in [3GPP TS 23.003 v6].

3gpp_ci – Cell Identity used for 3GPP networks as defined in [3GPP TS 23.003 v6].

3gpp_rac – Routing Area Code used for 3GPP networks as defined in [3GPP TS 23.003 v6].

3gpp_sac – Service Area Code used for 3GPP networks as defined in [3GPP TS 23.003 v6].

mbms_sai – MBMS Service Area Identities used for 3GPP networks that identifies a group of cells within a PLMN as defined in [3GPP TS 23.003 v6].

3gpp2_subnet_id_length – number of bit of the 3gpp2_subnet_id field.

3gpp2_subnet_id – binary representation of the subnet value for the subnet. This field is 128 bits at most as defined in [3GPP2 X.S0022-A].

padding_bits – these bits ensure the descriptor is byte-aligned and are set to 0. Length of this field, in bits, is given by the formula "8–- mod(3gpp2_subnet_id_length, 8)", where mod(*a,b*) gives the remainder on the division of *a* by *b*.

3gpp2_sid – System Identification; number that uniquely identify the 3GPP2 wireless system as defined in [3GPP2 C.S00005-D].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

3gpp2_nid – Network Identification; uniquely identifies a network which is subset of base stations within the wireless system as defined in [3GPP2 C.S00005-D].

3gpp2_pzid – Packet data services zone identifier of the base station as defined in [3GPP2 C.S00005-D].

dvbh_network_id – Network Identifier of the DVB-H system as defined in [ETSI EN 300 468 V1.6.1]. This Network Identifier is transmitted in the Network Information Table (NIT) according to [ETSI EN 300 468 V1.6.1].

dvbh_cell_id – Cell Identifier of the DVB-H system as defined in [ETSI EN 300 468 V1.6.1]. This Cell Identifier is transmitted in the TPS bits of the DVB-H signal according to [ETSI EN 302 304 V1.1.1].

dvbh_hierarchy – defines the logical channel ("lp" for "low priority" or "hp" for "high priority") that is selected for reception when hierarchical modulation is used. Coding of Hierarchy field is:

0x0 = Not defined

0x1= low priority

0x2= high priority

dvbh_cell_id_extension – Cell Identifier extension defined in [ETSI EN 300 468 V1.6.1] and transmitted in the Network Information Table (NIT) according to [ETSI EN 300 468 V1.6.1].

number_of_3gpp2_cell_ids – specifies the number of 3gpp2_cell_id fields included in the following loop.

3gpp2_cell_id – If cell_target_area_type = 6, then the value is Sector_ID as defined in [3GPP2 C.S0024-A]. If cell_target_area_type = 7, 8, 9, or A, then the value is BASE ID as defined in [3GPP2 C.S0002-0].

7.2 Constant Values

TKM_ALGO_IPSEC 0

TKM_ALGO_SRTP 1

TKM_ALGO_ISMACRYP 2

TKM_ALGO_DCF 3

TKM_FLAG_FALSE 0

TKM_FLAG_TRUE 1

7.3 Coding and Semantics of Attributes

protocol_version – indicates the protocol version of this STKM.

The device SHALL ignore messages that have a protocol_version number it doesn't support. The value of the protocol_version of this message is set to 0x0 (i.e. the original format).

Note: If set to 0x0 the format specified in this version of the specification is used. If set to anything else than 0x0, then the format is beyond the scope of this version of the specification.

protection_after_reception – 2-bit field defining the required protection after the removal of the service protection, according to the following table:

Table 43: Protection_after_Reception Values

Value	Description DRM Profile	Description Smartcard Profile

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

0x00	**Content Protection** Content only available to terminals with the Content Protection function. Device has to protect all content against access in the clear, unless such access is explicitly permitted by the GRO / permissions_category. Only the explicitly allowed types of consumption as defined in Generalized Rights Objects (GROs) that the device has for this service or program are permitted (taking also into account the impact of permissions_category value, if included in the STKM). An example permission in GROs is''Acces'' for the immediate rendering of the service or program. If a GRO has explicit permissions / constraints, then these SHALL be respected, without taking into account the protection_after_reception value.	same as 0x01 described below
0x01	**Content Protection with Implicit Direct Rendering Permission** Content only available to terminals with the Content Protection function. Device has to protect all content against access in the clear, unless such access is explicitly permitted by the GRO / permissions_category, but: • Direct rendering is implicitly allowed. No Generalized Rights Object is required in the device for direct rendering; a GRO with only the service or program key but without any permissions is sufficient. The device needs to have an GRO with the appropriate permissions (and possibly constraints) for any other type of consumption. If a GRO has explicit permissions / constraints, then these SHALL be respected, without taking into account the protection_after_reception value.	**Content Protection with Implicit Direct Rendering Permission** Content only available to terminals with the Content Protection function. Device has to protect all content against access in the clear, but: • Direct rendering is implicitly allowed. LTKMs provide keys for access to live content, broadcast files and recordings. When using GBA_U, recordings SHALL include STKMs. These SHALL be sent to the Smartcard for processing during playback.
0x02	**Content Protection with Implicit Direct Rendering Permission and Playback of Protected Recording** Content only available to terminals with the Content Protection function. Device has to protect all content against access in the clear, unless such access is explicitly permitted by the GRO / permissions_category,	**Content Protection with Implicit Direct Rendering Permission and Playback of Protected Recording** Content only available to terminals with the Content Protection function. Device has to protect all content against access in the clear, but implicitly, two types of consumption are allowed: • Direct rendering, and

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	but implicitly, two types of consumption are allowed: • Direct rendering, and • Unlimited play back of protected recordings of this service or program or protected files The above two types of consumption may also be made available over appropriately protected digital output links (see Appendix D for examples). If the protection_after_reception flags are not available for a protected recording, the device SHALL assume that they have the value 0x1 for that recording. If a GRO has explicit permissions / constraints, then these SHALL be respected, without taking into account the protection_after_reception value.	• Unlimited playback of protected recordings of this service or program or protected files. The above two types of consumption may also be made available over appropriately protected digital output links (see Appendix D for examples). LTKMs provide keys for access to live content, broadcast files and recordings. When using GBA_U, recordings SHALL include STKMs. These SHALL be sent to the Smartcard for processing during playback.
0x03	**Service Protection** Content available to terminals with the Service Protection or Content Protection function. This specification does not impose any protection measures for the content after the removal of service protection. If a permissions_category value is included in the STKM, it SHALL be set to 0xFF to allow exporting in plaintext. Note that for e.g. legal or other reasons, the device still might have to protect the content in some way. GROs provide keys for access to live content and broadcast files.	**Service Protection** Content available to terminals with the Service Protection or Content Protection function. This specification does not impose any protection measures for the content after the removal of service protection. Note that for e.g. legal or other reasons, the device still might have to protect the content in some way. LTKMs provide keys for access to live content and broadcast files.
Note: the creation of protected recordings, except for the protected format specified by the SAVE permission for the DRM Profile, is always allowed, because the play-back (or any consumption in general) is governed by GROs or LTKMs, or by the implicit play-back of protected recordings right when the protection_after_reception field has the value 0x02.		

traffic_protection_protocol – defines the protocol used for the encryption and authentication of traffic:

TKM_ALGO_IPSEC	IPsec ESP (transport mode; encryption: AES-128-CBC [key length 128]; authentication: HMAC-SHA1-96 [key length 160] or NULL).
TKM_ALGO_SRTP	SRTP (encryption: AES_128_CTR [key length 128]; authentication: HMAC-SHA1-80 [key length 160] or NULL).
TKM_ALGO_ISMACRYP	AU encryption (encryption: AES_128_BYTE_CTR [key length 128] (refer to [XBS DRM extensions-v1.0] for details); SRTP authentication: HMAC-SHA1-80 [key length 160] or NULL).
TKM_ALGO_DCF	DCF encryption (encryption: AES-128-CBC [key length 128]; authentication: HMAC-SHA1-80 [key length 160])

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Other values	Reserved for future use

Whether or not authentication is used depends on <traffic_authentication_flag>.

traffic_authentication_flag – defines whether or not the traffic is authenticated:

TKM_FLAG_FALSE	Traffic authentication is not used.
TKM_FLAG_TRUE	Traffic authentication is used, and the algorithm depends on <traffic_protection_protocol>.

access_criteria_flag – indicates whether or not access criteria are defined for the program:

TKM_FLAG_FALSE	No access criteria are defined, implying that the terminal is allowed to access program without further restrictions (provided the necessary keys are available to the terminal).
TKM_FLAG_TRUE	Access criteria are defined, implying that the terminal is allowed to access the program only if the specified access criteria are met.

Access criteria cannot change during a program, i.e. as long a PEK is valid.

traffic_key_lifetime – is the lifetime of the Traffic Encryption Key, relative to the first occurrence of an SPI or MKI.

If <traffic_key_lifetime> is ***n***, then the actual lifetime is 2^{n} seconds.

Note: Although the allowed values for the traffic_key_lifetime span from seconds to hours, service providers should not use TKM key material to realize long term key functionality. The TKM messages should be considered and used strictly for short-term key signalling. Also, the lifetime of traffic keys should be considerably shorter than the lifetime of service keys and program keys, to avoid users receiving the service or PPV event (encrypted with traffic keys) even after their service key or current program key has expired.

The following scenario may help in explaining the note. The field "next_encrypted_traffic_key_materia" maybe present in the STKM. The field is encrypted with the current Service Key or current Program Key. If someone subscribes to a service, or someone purchases a PPV event, then the person obtains both the current TEK and the next TEK. At the end of the service period, or the end of a PPV event, this means that the person has also a TEK for the next service period or the next PPV event. If the person stops subscription at the end of the current service period or the end of the current PPV event, then the person still has access to the first TEK of the next service period or next PPV event. When the maximum TEK lifetime is 1.5 minutes, a subscriber can at most have 1.5 minutes of unauthorized content, which may not be considered to be excessive. If the traffic_key_lifetime becomes 2 hours, then the subscriber may have excessive access to unauthorized content, especially in the case of PPV events, because the person now may have 2 hours of unauthorized content.

The TEK can be changed frequently to mitigate the risk of end-users posting the key via the interactive channel so that non-members can download that key. The cost of the attack, i.e., extracting the key, and uploading and downloading the key should be made to be more expensive than the cost of BCAST service/content. The frequency of change depends on the value of the BCAST service/content. For high-value PPV content, the TEK SHOULD be changed frequently whereas for low-value content, the TEK MAY be changed infrequently. The exact frequency is a configurable value and does not have impact on interoperability. The option to include two consecutive keys into one STKM, using next_encrypted_traffic_key_material, should be executed with care, since it allows the end user in any case to access service for 2*traffic_key_lifetime.

In the case when a Program Event is available either through subscription or as a PPV event, a STKM containing the next TEK at the end of a PPV program would allow a PPV user to view part of the next PPV event that corresponds to the next TEK. In this case, if next_encrypted_traffic_key_material is used, it SHOULD be utilized with sufficiently short Traffic Key lifetimes so as not to provide PPV users with free access to a PPV event that has not yet been purchased.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The actual duration of the crypto period SHALL be strictly shorter than the defined lifetime of the traffic key material. Typically, an SPI or MKI appears for the first time implicitly, when the “next” traffic key material is included in a STKM. Any safety margins to cope with network and transmission delays SHALL be added by the network. A typical value for the lifetime could be three times the crypto period.

The maximal value for the crypto period duration is in practice slightly shorter than the TEK lifetime, because the TKM will include the “current” and “next” traffic key material before a change of crypto period, to allow the devices to set up the security associations.

After the lifetime has expired, the security association containing the TEK can be safely deleted by the terminal. This may help managing the security association database in the terminal or enable other optimizations.

The maximum value for the TEK lifetime is defined mainly in order to have a strict upper bound for the effect of the “sneak post view” problem: the next traffic key material is distributed under the current PEK, and allows viewers to view a program during the next crypto period. Should this possibility still be of a concern, the network MAY choose a shorter crypto period than the maximum value, or, during the crypto period where the current program ends and a new program starts, choose to distribute the current and the next traffic key material in separate STKMs, encrypted with their respective PEKs.

number_of_access_criteria_descriptors – indicates the number of Access Criteria Descriptors.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

8. Recording

8.1 Recording of Protected Streams

Service protection, whether it is provided using the DRM Profile or the Smartcard Profile, is an access-control mechanism only, i.e. once the SEK or PEK has been delivered to the user, access to a given broadcast stream is typically unrestricted.

However, certain broadcast content may have premium value and recording may be allowed only in protected form. This is achieved by using the protection_after_reception parameter of the STKM, as explained in Section 7.3. Both cases are explained below.

Recording can be governed by different flags. Depending on the profile, not all flags are considered to allow recording.

- The permissions associated with broadcast RTP streams, defined in the OMA DRM v2.0 Extensions for Broadcast Support document [XBS DRM extensions-v1.0], are sent in rOs for the DRM Profile. The value of the permissions_flag and the permissions_category (Section 10.1.5 of [XBS DRM extensions-v1.0]) for a programme that is part of the STKM must also be considered.
- Protection_after_reception values in the STKM define the type of protection provided for the recorded content. These are applicable to both DRM and Smartcard Profiles.

Depending on the above, content may be recorded in the clear or in protected form, as explained below.

8.2 Recording in the Clear

If recording in the clear is allowed, this SHALL be signalled in the Short Term Key Messages by setting the protection_after_reception to 0x03.

In this case, recording of content (unencrypted AUs) is possible in the clear, using appropriate file formats (provided by the BDS specifications, from other standards bodies or using proprietary formats). For BCAST, the existing DCF or PDCF file formats as defined in OMA DRM 2.0 [XBS DRM extensions-v1.0] MAY be used for recording in the clear [DRMCF-v2.0]. Other similar formats such as ISO or 3GPP can be used.

8.3 Recording in Protected Form Only

If recording in protected form only is allowed, this SHALL be signalled by setting protection_after_reception to 0x00, 0x01 or 0x02. In such cases, recording MUST be protected against access in the clear. This MAY be done by encrypting the content and protecting the decryption key(s) against access in the clear. This MAY be done using other means to protect content against access in the clear.

Access to the recorded content depends on the value of the protection_after_reception parameter. See Section 7.3.

The broadcast stream can be encrypted at transport level (IPsec or SRTP) or content level (ISMACryp) as described in Section 9.

If the broadcast stream in encrypted at content level using ISMACryp, recording in encrypted format may be achieved by recording the encrypted AUs without decryption in the adapted PDCF file format together with the TEK stream as explained in [XBS DRM extensions-v1.0]. Other methods and file formats may also be used. Note that recording of encrypted broadcast streams is possible without having the appropriate service protection rights (i.e. SEK or PEK) when using ISMACryp. These can be acquired at a later stage using the information stored in the KeyInfo box. This allows automatic recording of programmes based on user profiles, for example, or pricing models based on the time at which rights are acquired for service protection, i.e. the value of recorded content reduces as time goes by.

If the broadcast stream is encrypted at transport level using IPsec or SRTP, then the recording may require first decryption of the content and then re-encryption in an appropriate file format. This method is only applicable in the case of DRM Profile.

Recommendations for dealing with changes in rights are given in Section 8.4.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

8.3.1 Recording of Streamed Content using (P)DCF File Format

Streamed protected content MAY be stored using the DCF file format [DRMCF-v2.0]. If the PDCF file format is used instead, the protected file MAY be stored using this file format. Both file formats are defined in OMA DRM 2.0 [DRMCF-v2.0].

Recording of super-distributable OMA assets containing a recording of broadcast content that is suitable for standard DRMv2 devices is described in section 7.4 of [XBS DRM extensions-v1.0]. This involves re-encryption with a single key and hence does not require recording of the key stream.

8.3.2 Recording of ISMACryp Protected Streamed Content using Adapted PDCF File Format

When recording content from a real-time delivery service using ISMACryp, the file MAY be created according to a modified version of OMA DRM PDCF 2.0 that allows usage of multiple encryption keys (TEKs) for content encryption in a single file [XBS DRM extensions-v1.0]. This is achieved by using the Access Unit header OMABCASTAUHeader, which signals AU encryption and provides storage for the Key Indicator and IV. The Key Indicator identifies the TEK key used to encrypt Access Unit and the IV is used for the Counter mode of AES. The elements of the ISMACrypContextAU (as defined in [ISMACRYP11] and [ISMACRYP20]) are mapped to the OMABCASTAUHeader defined in [XBS DRM extensions-v1.0] as follows:

Table 44: Mapping of Elements of ISMACrypContextAU to OMABCASTAUHeader

ISMACrypContextAU field	OMABCASTAUHeader field
AU_is_encrypted	EncryptedAU
initial_IV, delta_IV	IV
key_indicator	KeyIndicator

Note: The *IV* is computed for each AU from the *initial_IV* and *delta_IV* as specified in [ISMACRYP11] and [ISMACRYP20].

The STKMs are recorded in a STKM track. Note that repeated STKMs can be ignored i.e. if the same STKM is received as one already recorded, it SHOULD not be recorded. The type of STKM is indicated in the adapted PDCF.

The Table below shows the appropriate location for parameters that need to be stored in the adapted PDCF file.

Table 45: Mapping of Broadcast Parameters to PDCF Parameters

Parameter	Source Location	Destination Location
PermissionsIssuerURI	SG Access Fragment	RightsIssuerURL in CommonHeadersBox or KeyIssuerURL in KeyInfoBox
Service_BCI or Service_CID or Program_BCI or Program_CID	SG Access Fragment and STKM	ContentID in CommonHeadersBox
STKMs	STKM stream	OMAKeySample in STKM track
STKM type indication	SDP	sample_type in OMAKeySampleDescriptionEntry
TerminalBindingKeyID (if TBK is used)	SG Access Fragment	entry in TerminalBindingFlagInSTKM and KeyInfoBox
tbkPermissionsIssuerURI	SG Access Fragment	entry in TerminalBindingFlagInSTKM and KeyInfoBox

This applies to both DRM and Smartcard profiles.

The Table below shows the content of the CommonHeadersBox fields when using the adapted PDCF. The equivalent table when using re-encryption with a single key for a DRMv2 format can be obtained from section 7.4 in [XBS DRM extensions-v1.0]. The Table shows which parameters are used for DRM and Smartcard Profiles.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 46: CommonHeaders Box Fields for Adapted PDCF

Field	Contents DRM Profile	Content Smartcard Profile
EncryptionMethod	NULL (0x00) if no encryption. AES_128__BYTE_CTR (0x03) for ISMACryp encryption with TEKs.	same
PaddingScheme	Determined by the recording device.	same
PlaintextLength	Determined by the length of the recorded asset, calculated by the recording device.	same
ContentIDLength ContentID[]	N/A	N/A
RightsIssuerURLLength RightsIssuerURL[]	RightsIssuerURI if KeyIssuerURL not used in KeyInfoBox.	N/A
TextualHeadersLength TextualHeaders[]	Determined by context information (original asset, service guide, session description protocol).	N/A
ExtendedHeaders[]	Empty.	same

In the definition of these fields, the base64() operation is defined by [RFC2045].

The Table below shows the content of the KeyIDBox fields when using the adapted PDCF. The Table shows which parameters are used for DRM and Smartcard Profiles.

Table 47: KeyInfo Box Fields for Adapted PDCF

Field	Contents DRM Profile	Content Smartcard Profile
KeyID Type	0x00	0x01
KeyID	base64Binary(Service_BCI) for recording of stream protected by SEK base64Binary(Program_BCI) for recording of stream protected by PEK	base64Binary(Key Domain ID|MSK ID)
KeyIssuerURL	RightsIssuerURI	PermissionsIssuerURI
TBK_ID	N/A	TerminalBindingKeyID
TBKIssuerURL	N/A	tbkPermissionsIssuerURI
STKM[]	N/A as have STKM track	same

The following section provides recommendations for how change of rights is handled when recording.

8.4 Change of Rights and Recommendations for Recording

The following rules SHALL be observed when recording streamed content in a PDCF:

1. If the user has a valid Rights and the end of a program / event is reached, a new track MAY be created for the new program / event. Alternatively, a new file MAY be created for the new program / event, rather than using the same file.

2. If the user has a valid Service Rights and PEKs are used to protect TEKs, then new tracks or files MAY be created when PEKs change, rather than using the same track.

3. If a program / event is being recorded for which the user has the appropriate Rights and a new program / event starts for which the user has NO valid Rights, a new track or a new file SHOULD be created, rather than using the same track.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

4. If a program / event is being recorded for which the user has no Rights, a new track or file MAY be created for a new program / event, rather than using the same track, if the user still has no valid Rights for the new program / event.

5. If the user has valid Rights for the new program / event, a new track or file SHOULD be created, rather than using the same track.

6. In all cases, if different rights or a different GRO is required, a different track or file SHALL be used.

8.5 Signalling of Recording to the Smartcard in Smartcard Profile

In order to allow the efficient management of the SEKs/PEKs and SPE stored on the Smartcard, the terminal SHALL signal to the Smartcard the recording of streamed content or storage of downloaded content accessible with the use of a specific SEK/PEK or SPE to the Smartcard using the OMA BCAST command in Record Signalling mode (AppendixE: E.3.3). The terminal SHALL send this OMA BCAST command during the recording for each SEK/PEK Key number part involved in the protection of the recorded content.

The terminal introduces in the command the terminal identifier (specified by its terminal identifier type) and a Content_ID, as a unique identifier of the content. This content identifier is terminal specific and then its coding is implementation specific. This Content_ID could be for example the hash of the SEK ID, TEK ID and TS value of the first STKM in the key track along with some other unique identifier from the content file. Alternatively the terminal could simply create its own unique identifier. The terminal MAY send the same Content_ID for several Record Signalling command if these commands concern the same recorded content but for different SEK/PEK Key number part. This is the case when the recorded content covers several SEK/PEK IDs.

At the reception of the OMA BCAST Command in Record Signalling mode, the SPE used to access the content is flagged as being required for the playback of recorded (or stored) protected content. This flag (UsedForRecording flag) MAY then be used to inform key deletion/management policies (see Section 6.7.3.10). In addition, the flag MAY be used to enforce a limit on the number of SEKs/PEKs and SPE that can be stored on the Smartcard to allow access to recorded/stored content, thereby ensuring that a certain number of SEKs/PEKs and SPE can be stored for access to live content. The Smartcard stores the terminal identifier and the Content_ID received in the command and associates them to the SPEs used for the protection of the content and flagged internally in the Smartcard during the execution of this command. A description of this command may be found in AppendixE: E.3.3.

In the response of the Record Signalling command, the terminal discovers the number of remaining SPE records available for SPEs instance required for the playback of recorded/stored content after the execution of this command. The terminal receives also the description of SPEs flagged internally in the Smartcard.

In case the recorded content or a part of the recorded content described in the input parameters of the command is not covered by an SPE in the Smartcard, the command fails, the status word ‘6A88’ (Referenced data not found) is returned, and none of the SPE is flagged as a SPE Used For Recording as described in E.2.3. In this case the terminal MAY ask the user if she wants to acquire the rights for the part of the content covered by the failed Record Signalling and then a LTKM request SHOULD be sent to the BSM if applicable. The terminal SHOULD then re-send the Record Signalling command after reception of associated LTKM message until the Record signalling command ends successfully.

The terminal MAY use the OMA BCAST command in Recording Audit mode, to retrieve all content identifiers with the associated flagged SPEs stored in the Smartcard. A description of this command may be found in AppendixE: E.3.4.

When a content is erased by the user in the terminal, the Terminal MAY use the Authenticate command for the MBMS security context in OMA BCAST operation and in Recording Deletion Mode (AppendixE: E.2.3.2) to delete the content identifier in the Smartcard and its association to the flagged SPEs.

NOTE: This signalling of recording is applicable for recorded contents only and not for time-shifted contents. In this latter case On-Live SPEs are used and not PLAYBACK SPEs.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

9. Encryption Protocols

This section deals with the "Traffic Encryption Layer" (Layer 4) in the 4-layer model. The encryption protocols discussed below are optional on the Network (server) side.

9.1 IPsec

IPsec [RFC4301] fulfills both the criterion to be bearer-agnostic and to be universally usable for all types of IP-based services. The Broadcast System MAY use IPsec to protect Broadcast Services. Broadcast Terminals MAY support IPsec.

The IPsec implementation in the device can be such that it does not interfere with the usage of IPsec for other applications than OMA BCAST. This implies that the SPI allocation and security association lookups can be implemented in such a way that they interoperate with existing IPsec implementations.

An IPsec Security Association (SA) consists of a tuple of the following parameters.

- Selectors (IP protocol version, source IP address, destination IP address, protocol, source port and destination port)
- SPI
- Destination IP address
- Security protocol, security protocol mode and security protocol parameters
- Algorithms and algorithm parameters
- Key material

An IPsec SA is uniquely identified by a destination IP address and SPI pair.

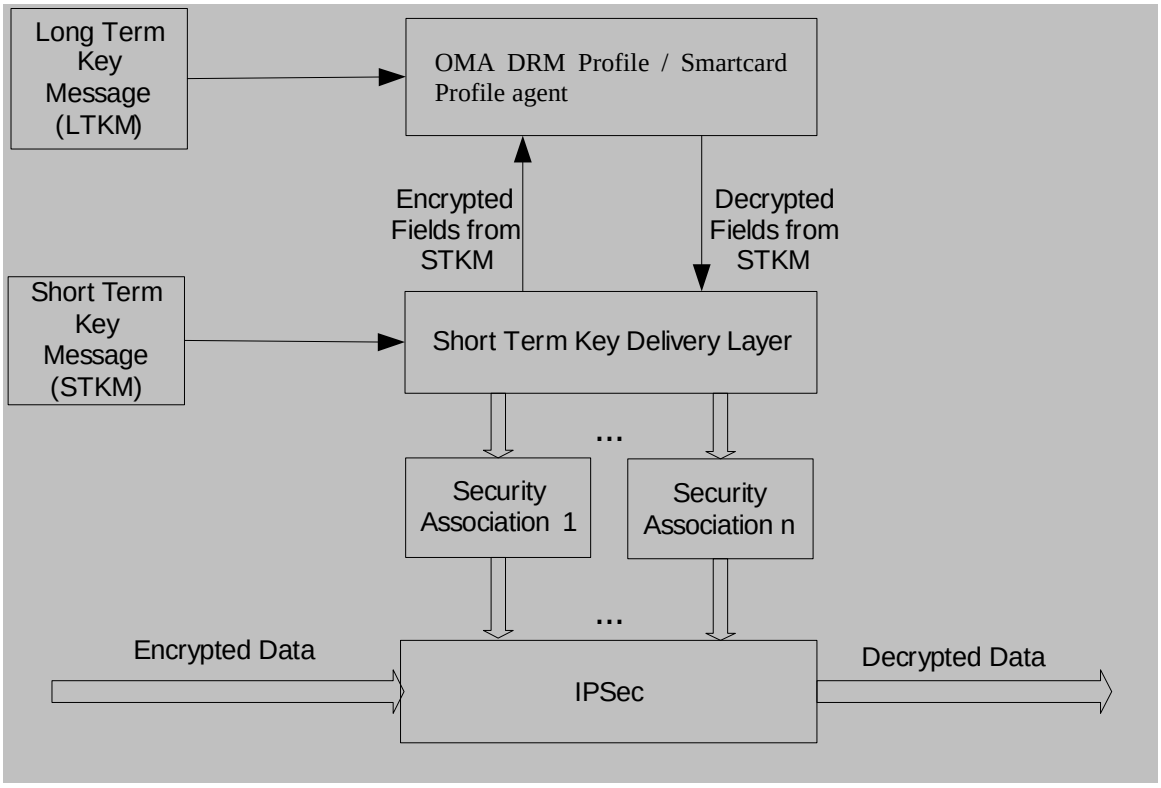


Figure 8 – IPsec Security Association Elements

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Figure 8 shows the different objects and elements involved in instantiating IPsec security associations. The instantiation of security associations is performed by Layer 2 and Layer 3 messages. Given a Layer 3 message, Layer 3 extracts the encrypted fields from the message. Layer 3 passes these and other relevant fields (service-CID_extension and program_CID_extension) for Layer 2 message identification to Layer 2. For all Layer 2 messages on the device, Layer 2 examines them to see if one would be able to decrypt the fields in the Layer 3 message. If Layer 2 does find suitable Layer 2 messages, then it uses Layer 2 keys (SEK or PEK) in these messages to decrypt Layer 3 message fields. The decrypted fields are provided back to Layer 3, which based on the Layer 3 message and the decrypted fields instantiates a set of security associations. If Layer 2 does not find a suitable Layer 2 messages in the device, then the Layer 3 message SHOULD be silently dropped.

Selectors

Selectors are provided by the Layer 3 messages. The selectors can contain wildcards, ranges or point values, but all the other parameters SHALL be exactly defined. For transport mode all address selectors SHALL be point values and the destination address selector SHALL match the destination IP address of the SA.

Encapsulation Protocol and Mode

If IPsec is used for encryption of broadcast services, the protocol and mode SHALL be ESP in Transport Mode, according to [RFC2401] and [RFC2406]. Other IPsec encapsulation protocols or modes SHALL NOT be used.

Encryption Algorithm

The encryption algorithm for IPsec ESP SHALL be AES-128-CBC with explicit IV in each IP packet, as defined in [RFC2451] and [RFC3602]. Other encryption algorithms or key sizes or chaining modes SHALL NOT be used.

Authentication Algorithm

The authentication algorithm for IPsec ESP SHALL be HMAC-SHA-1-96, as defined in [RFC2104] and [RFC2404]. Other authentication algorithms or truncations SHALL NOT be used.

Support for the authentication algorithm as specified above is MANDATORY for both the terminal and the broadcast system. If no authentication is desired, the NULL authentication algorithm SHALL be specified. In this case, replay protection SHALL NOT be performed by the terminal.

The traffic_authentication_flag field in STKM indicates whether security transform includes integrity protection.

SA Management

The STKM Layer defines how often the IPsec encryption keys are rekeyed. This sets the following requirements:

- The IPsec encryption key SHALL be used as the key for the ESP encryption.
- The IPsec encryption key SHALL be derived from the key material contained within the STKM as follows:
  - In case of DRM Profile, the IPsec encryption key SHALL be the first 128 bits of the decrypted traffic key material. See Section 5.5.1.
  - In case of Smartcard Profile, the IPsec encryption key SHALL be obtained as described in Section 6.7.4.
  - NOTE: In Smartcard Profile, the MBMS MTK is called TEK, from which the actual encryption and authentication keys for IPsec are derived. To avoid confusion, the key for the ESP encryption is called “IPsec encryption key”.
- The IPsec authentication key TAK, which is derived from the key material in the STKM, refer to “Authentication for IPsec” below, SHALL be used as the key for the ESP message integrity code if authentication is used.
- The IPsec implementation SHALL be able to manage security associations relating to the key stream messages separately from those managed manually or by any other protocol such as IKE. This implies the ability to identify whether an SA is relating to key stream messages.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Security associations relating to STKMs SHALL be prioritized lower than those security associations that have a locally defined policy or a policy that is provided by a trustworthy party.
- Security associations relating to STKMs are simplex and SHALL be applied only to inbound traffic on the recipient side.
- An implementation SHALL be able to keep alive the security associations for at least two crypto periods (crypto period is the time span during which a specific traffic key is authorized) of the key stream.

The rekeying of existing IPsec sAs by Layer 3 SHOULD be managed on a resource basis by the Traffic Encryption Layer according to the following recommendations:

- The IPsec implementation SHOULD be able to keep alive at least the two most recently instantiated IPsec security associations for a specified set of selectors.
- The IPsec implementation SHOULD provide a least-recently-instantiated mechanism for destroying security associations as resources reserved for OMA BCAST IPsec security associations are exhausted.
- The amount of IPsec sAs required to exhaust the resources such that the cleanup mechanism is triggered SHOULD be 3 per SEK per set of IP selectors.

Authentication for IPsec

IPsec can be used with authentication. In case of authentication with IPsec the authentication data SHALL carry the TAS. The authentication mechanism SHALL create the TAK from the TAS.

For the DRM Profile, to obtain the encrypted traffic key material from the STKM the encrypted traffic key material SHALL be decrypted with the SEK or PEK:

$$TAS = D\{SEK\}(traffic\_key\_material)$$

or

$$TAS = D\{PEK\}(traffic\_key\_material)$$

For Smartcard Profile, the authentication seed TAS SHALL be obtained as described in Section 6.7.4.

The authentication key SHALL be generated from the authentication seed as follows:

$$TAK = f_{auth}\{TAS\}(CONSTANT\_KSM)$$

where:

CONSTANT_KSM = 0x040404040404040404040404040404 (120 bit)

Refer to Section 5.5.2 for details on f_{auth}.

The TAK SHALL be used in the MAC generation / verification of the IPsec data. Refer to [RFC 2406] for details.

9.2 SRTP

The Broadcast System MAY use SRTP [RFC3711] to protect Broadcast Services. Broadcast Terminals SHALL support SRTP.

An SRTP session is defined as a cryptographic context in the terminology of SRTP. A cryptographic context for SRTP, when used for service protection in OMA BCAST, consists of the following elements:

- Roll-over counter (ROC)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Receiving sequence number
- Cipher and mode definition
- MAC method definition
- List of received packets
- MKI indicator bit
- Length of the MKI field
- Value of currently active MKI
- Array of secret master keys (MK)
- Array of counter of processed packets for each master key
- Length of encryption and authentication keys
- Master salt
- Context id

A cryptographic context is uniquely identified by its context id. The context id consists of the SSRC, destination network address and destination transport port number, as defined in [RFC3711].

Figure 9 shows a general case of key management for SRTP. Figure 10 shows a special case where Layer 3 is omitted and the necessary data is received from MKI to derive TEK (see [3GPP2 X.S0022-A] and [3GPP2 S.S0083-A]).

The instantiation of a cryptographic context is performed via the STKM and is driven by STKM and LTKMs. In the case of SRTP, an STKM includes an MKI (Master Key Index) length and MKI that is necessary to identify an SRTP cryptographic context. STKMs may also carry Master Salt (MS) depending upon the underlying BDS (See below under "key management" for further discussions). The ROC (Roll-Over-Counter) values are carried in band in SRTP packets. STKM also carries the length of the encrypted key material (Master Key (MK) for SRTP).

The SRTP Master Key is extracted from the STKM. The traffic encryption layer then creates an SRTP session with decryption and (optionally) authentication keys that are derived from the Master Key as required by SRTP.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

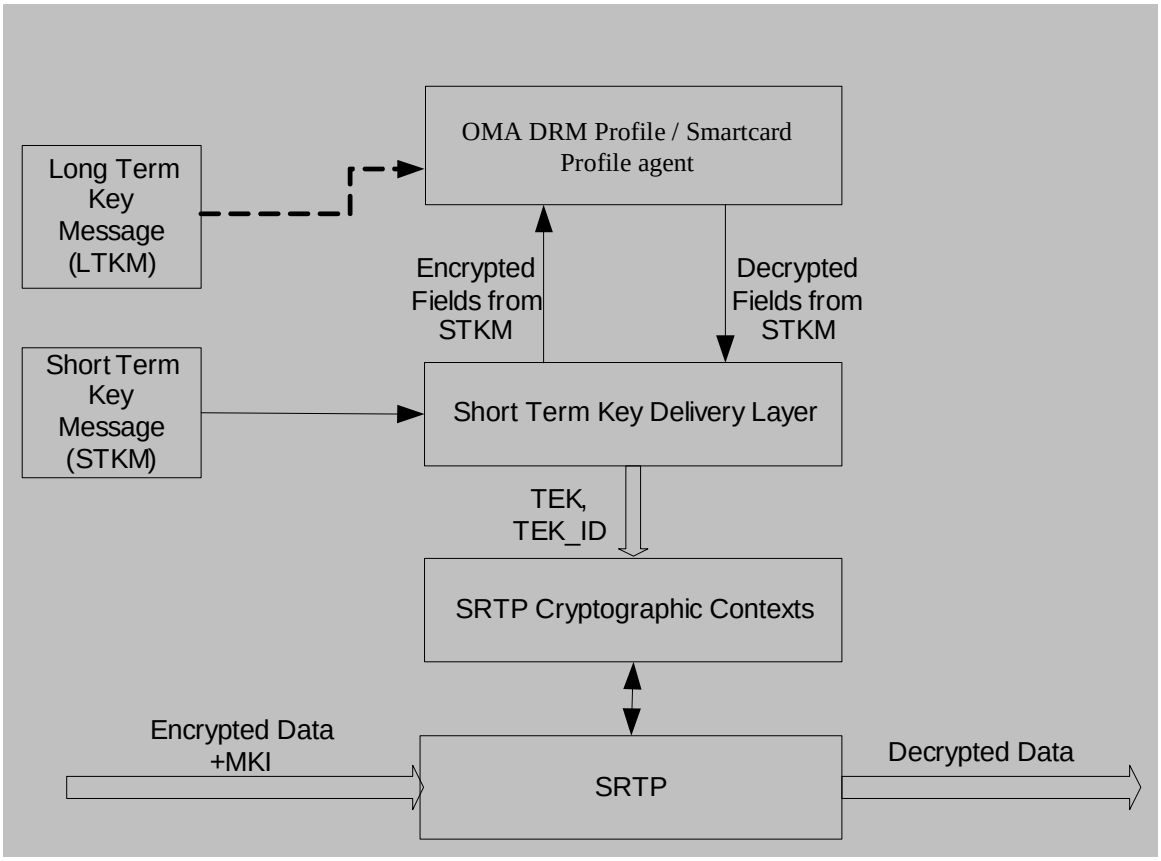


Figure 9 – SRTP Cryptographic Context Management (General Case)

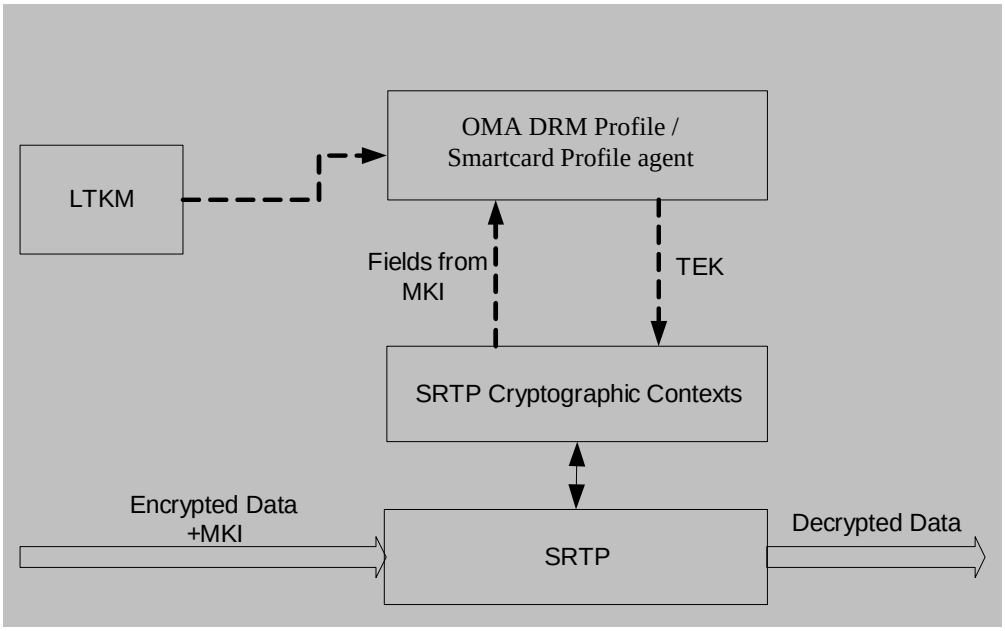


Figure 10 – SRTP Cryptographic Context Management (No Short Term Key Delivery Layer)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Key Management

The OMA BCAST SRTP application SHALL use the MKI value for looking up decryption keys. This means that a cryptographic context SHALL have the MKI indicator bit set to 1. The <From, To> value method of key lookup SHALL NOT be used.

The Master Salt (MS) MAY be used. The MS SHALL NOT be longer than 112 bits.

For the DRM Profile used independently, a NULL MS is used. A non-NULL 112 bit MS may be used for interoperability with the underlying BDS or the Smartcard Profile (see Section 11.3 for further details).

For the (U)SIM, a NULL 112 bit MS may be used for interoperability with the DRM Profile. Otherwise a non-NULL 112 bit MS is used. For interoperability with the (R-)UIM/CSIM Smartcard Profile, the 112 bit MS may correspond to the 32 bit SK_RAND with zero bit padding for the remaining 80 bits. See Section 11.3 for further details.

For the (R-)UIM/CSIM, the MS SHALL be the 32 bit SK_RAND value with zero bit padding for the remaining 80 bits. A NULL MS SHALL NOT be used as this results in a fixed encryption key for the lifetime of the SEK or PEK (see Section 11.3 for further details).

The TEK contained in the STKM SHALL be used as the SRTP master key.

The key derivation rate SHALL be 0 except for interoperability with 3GPP MBMS terminals, where the key derivation rate MAY be zero (see Section 11.3 and [BCAST10-MBMS-Adaptation]).

Layer 3 SHALL provide and update the cryptographic contexts to the SRTP implementation (excluding the ROC). Note that some fields are initialized or managed internally, such as the list of received packets used in replay protection, receiving sequence number, and the ROC.

The ROC SHALL be transferred in every R-th packet according to [RFC4771].

Because the SRTP key-deriviation rate is not used and the <From,To> values are also not used, the SRTP crypto context will be rekeyed by Layer 3.

Encryption Algorithm

The encryption algorithm for SRTP packets SHALL be AES_128_CTR, as defined in [RFC3711]. Other encryption algorithms or key sizes or chaining modes SHALL NOT be used.

Authentication Algorithm

The authentication algorithm for SRTP SHALL be as defined in [RFC4771], based on HMAC-SHA-1-80 as defined in [RFC2104] and [RFC3711]. Other authentication algorithms or truncations SHALL NOT be used.

Support of the authentication algorithm for SRTP as specified above shall be OPTIONAL for both the terminal and the broadcast system. If no authentication is desired, the NULL authentication algorithm SHALL be specified. In this case, also replay protection SHALL NOT be performed by the terminal.

Note that there must be a secure way of notifying whether a security transform includes integrity protection. This should be handled as part of the mechanism for negotiating SRTP security parameters e.g. MIKEY.

For the DRM Profile, the traffic_authentication_flag field in STKM indicates whether security transform includes integrity protection. For the Smartcard Profile, this can be handled by the corresponding Layer 3 message or some other mechanism for negotiating SRTP security parameters.

Some specific points of the implementation need to be specified to be able to share protected stream(s) between operators. Section 11.3 introduces how to be able to share a media stream among operators that implement different key management mechanisms, Section 11.3.1 with respect to 3GPP-MBMS bearer features using SRTP, and Section 11.3.2 considering access limited to BCAST terminals.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

9.3 ISMACryp

For content encryption of RTP streams, content that is part of a real-time delivery service MAY be protected using ISMACryp as explained in this specification, i.e. by encrypting elementary audio video samples called Access Units (AUs). Individual AUs are encrypted using AES_128_BYTE_CTR mode. Each encrypted AU has an ISMACrypContextAU defined in [ISMACRYP11] and [ISMACRYP20].

BCAST terminals MAY support content encryption using ISMACryp as specified in this section.

9.3.1 Encryption Algorithm

The encryption algorithm SHALL be AES_128_BYTE_CTR. Refer to [ISMACRYP11], [XBS DRM extensions-v1.0], or [ISMACRYP20] for further details. Other encryption algorithms or key sizes or chaining modes SHALL NOT be used.

The TEK is sent in STKMs, the IV is in the ISMACrypContextAU preceding the encrypted data, the salt key k_s is signalled in the SDP file and the use of the counter is described in [ISMACRYP11] or [ISMACRYP20].

The Table below shows BCAST parameters and equivalent ISMACryp parameters.

Table 48: Equivalent BCAST and ISMACryp parameter names

OMA BCAST parameters	Equivalent ISMACryp parameters [ISMACRYP11, ISMACRYP20]
TEK	key_k (encryption key)
IV	IV
k_s	k_s (salt key)

9.3.2 Authentication Algorithm

The default authentication algorithm is SRTP with an HMAC-SHA1 with an 80-bit output tag and a 160-bit key [RFC3711]. Other authentication algorithms or truncations SHALL NOT be used. Support of the authentication algorithm for ISMACryp shall be OPTIONAL. The authentication key to be used is derived as per SRTP using the 128 bit MK and 112 bit MS sent in STKMs. SRTP authentication is signalled using SDP security descriptions [RFC4568]. ROC is signalled as per SRTP described in Section 9.2, key indicator (see Section 5.5) is used as MKI for DRM Profile and MTK ID is used as MKI for Smartcard Profile. MKI and Authentication tag is delivered over SRTP packet according to [RFC3711].

9.3.3 RTP Transport of Encrypted AUs (ISMACryp)

Content encryption modifies data before packetization of RTP packets, thus the various RFCs defining ways to encapsulate audio and video data do not apply. In addition, some signalling is necessary in the SDP in order to enable the decryption of the data. ISMACryp 1.1 [ISMACRYP11] has defined encapsulation for some MPEG-4 codecs [ISO-14496-2, ISO-14496-3, ISO-14496-10]. For these codecs, the encapsulation as defined in [ISMACRYP11] SHALL be used. For any other encrypted media that has a defined mapping to the ISO Media File Format ([ISO-14496-12]), the encapsulation as defined in section 7 of [ISMACRYP20] SHALL be used.

9.4 (P)DCF Encryption with TEK

This section describes how (P)DCF files can be protected over the broadcast channel by encrypting individual files with individual TEKs. This technique is based on material from MBMS text in [3GPP TS 33.246 v7]. The mechanism described in this section was adopted from [3GPP TS 33.246 v7] and adapted to BCAST needs.

Protection of download data uses DCF as a container for ciphered file data. The DCF container also identifies the key used in protecting the data. In this case the encryption key is a single TEK. Usage of DCF is independent of the KMS type. The same principle applies to the PDCF format for audio video data.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Data that belongs to a download Service is decrypted as soon as possible by the terminal, if the SEK or PEK needed to provide the relevant TEK are already available on the terminal or Smartcard. Storage of the STKM containing the TEK is also allowed in BCAST.

The following method is compatible with the OMA DRMv2 DCF file format as defined by [DRMCF-v2.0] as it uses the Key Info box defined in [XBS DRM extensions-v1.0] in the Extended Headers field, which is ignored by OMA DRMv2 terminals.

Access to the file SHALL respect the protection_after_reception values defined in the STKM message.

9.4.1 Integrity Protection using OMADRMSignature Box

When it is required to protect BCAST download data, OMA DRM V2.0 DCF as defined in reference [DRMCF-v2.0] shall be used. However, encryption and authentication keys are generated from TEK. For integrity protection, an OMADRMSignature as specified below MAY be attached inside the optional Mutable DRM information box ‘'mdr’') of the (P)DCF.

The OMADRMSignature Box is an extension to OMA DRM V2.0 DCF for use by OMA BAC BCAST, and is defined by 3GPP as follows:

Table 49: OMA DRM Signature Box

```
aligned(8) class OMADRMSignature extends Fullbox('odfs', version, flags) {
   Unsigned int(8)   SignatureMethod;     // Signature Method
   Char           Signature[];            // Actual Signature
}

SignatureMethod Field:
NULL  0x00
HMAC-SHA1   0x01
```

The range of data for the HMAC calculation shall be according to section 5.3 of reference [DRMCF-v2.0].

9.4.2 Use of OMABCAST Key Info Box

BCAST has defined a specific box [XBS DRM extensions-v1.0] to provide key management information for both DRM Profile and Smartcard Profile.

The OMABCASTKeyInfo box allows the following information to be stored for the DRM Profile and / or Smartcard Profile:

- KeyID: SEK / PEK ID & TEK ID used for decrypting the (P)DCF
- KeyIssuerURL: PermissionsIssuer URL used to acquire the appropriate permissions
- TBK_ID: TerminalBindingKey ID and URL if used
- STKM containing the TEK used to decrypt the content

In order to ensure key material can be acquired, the KeyIssuerURL in the Key Info box MAY be used. If the Terminal does not have the SEK or PEK required to decrypt the TEK within the STKM, it may request it by sending the Service request described in [BCAST10-Services] to the KeyIssuerURL with the corresponding SEK or PEK ID. If the KeyIssuerURL is not present, the RightsIssuerURL in the OMADRMCommonHeader box MAY be used instead.

The STKM containing the TEK used to decrypt the DCF MAY be stored inside the OMABCASTKeyInfo box in the STKM field. Note that as the OMABCASTKeyInfo box is part of the HMAC calculation, if the OMADRMSignature box is included but the STKM is not delivered within the OMABCASTKeyInfo box, subsequently adding the STKM to the (P)DCF invalidates the hash. A terminal doing this would typically remove the OMADRMSignature box.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

9.4.3 FDT Protection within DCF

In case the FDT of the FLUTE protocol needs to be protected, the FDT MAY also be wrapped in a different DCF. Confidentiality or integrity protection of FDT can be provided this way.

9.4.4 Support of OMA DRM v2 Boxes

The OMA BCAST DCF format SHALL support the following boxes specified in OMA DRM V2.0 DCF [DRMCF-v2.0]:

- Fixed DCF header;
- Mutable DRM information Box;
- OMA DRM Container Box.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

10. Signalling

Access to key streams is provided in SDP.

10.1 Protection Signalling in SDP

10.1.1 Description

SDP information is used to specify streaming sessions according to [RFC4566].

Additional information is required to identify parameters relative to key management: STKM streams, KMS versions, etc. These are defined below and SHALL be used to describe encrypted media streams and key streams (STKM and LTKM). Note that, in the case of MBMS, such information can be signalled in the MBMS security description as per [3GPP TS 26.346 v7].

The table below defines the <field values> to be used for signal protection information. These parameters are used for the signalling of media, short-term key message (STKM) and long-term key message (LTKM) streams. Their usage for the different streams will be explained in the following sections. A media stream can be protected by one or more STKM streams. Some other optional and stream specific parameters are introduced in the relevant sections.

Table 50: Protection Signalling in SDP

Field name	Category	Type	Purpose
kmstype	NM/TM	String	Identifies the Key Management system (KMS) used. (See Table 51 for supported KMSs)
bcastversion	NM/TM	Decimal x.y	Identifies the BCAST version as defined in [RFC5159]. Although [RFC5159] defines the bcastversion attribute for session level only, the OMA BCAST SPCP specification [BCAST10-ServContProt] also makes use of this attribute at media level. Both the Network and Terminal SHALL support the attribute bcastversion both at session and media level in SDP. (See also informative Appendix I.) It is possible to associate different BCAST versions to different STKM streams.
serviceproviders	NM/TM	String	Identifies the service providers using the key stream, by referencing one or more BSMSelectors as declared in the SGDD in the SG [BCAST10_SG] or by referencing one or more <X>/ServiceProviders/<X>/ID nodes as specified in [BCAST10-Services]. (See Table 53 for the syntax and semantics.)
streamid	NM/TM	UnsignedShort	Unique positive integer identifying a particular key stream. Numbers are unique within a particular SDP session i.e. no global numbering is required. Used to indicate which media stream is protected by the actual STKM stream.
baseCID	NO/TO	AnyURI	For the DRM Profile, part of the Service or Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. The Service or Program CID is obtained from the BaseCID as described in Section 5.5.1. The network and terminal SHALL support this field in case the DRM Profile is supported.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

srvCIDExt	NO/TO	unsignedByte	For the DRM Profile, part of the Service CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. The Service is obtained from the service_CID_extension as described in Section 5.5.1. This parameter SHALL be provided if the protected media stream has multiple DRM Profile STKM streams for a service provider with different values of service_CID_extension. Within each service provider, for each protected service or media stream, the value of this parameter SHALL be unique for each STKM stream. The Terminal SHOULD use the STKM stream with the matching srvCIDExt. The Terminal MAY use any STKM stream for which it has the correct SEK, as indicated by matching srvCIDExt in the GRO and the STKM. The network and terminal SHALL support this field in case the DRM Profile is supported.
prgCIDExt	NO/TO	unsignedByte	For the DRM Profile, part of the Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. The Program CID is obtained from the program_CID_extension as described in Section 5.5.1. This parameter SHALL be provided if the protected media stream has multiple DRM Profile STKM streams for a service provider with different values of program_CID_extension. Within each service provider, for each protected service or media stream, the value of this parameter SHALL be unique for each STKM stream. The Terminal SHOULD use the STKM stream with the matching prgCIDExt. The Terminal MAY use any STKM stream for which it has the correct PEK, as indicated by matching prgCIDExt in the GRO and the STKM. The network and terminal SHALL support this field in case the DRM Profile is supported.
srvKEYList	NO/TO	See Table 58	For the Smartcard Profile, the srvKEYList is a list of so-called srvKEY values. Each srvKEY value is a concatenation of the Key Domain ID with the Key Group part of a SEK/PEK associated to the related STKM stream. This parameter SHALL be provided if the protected media stream has multiple Smartcard Profile STKM streams for a service provider with different values of srvKEYList. Within each service provider, for each protected service or media stream, the value of this parameter SHALL be unique for each STKM stream. Within each service provider, for each protected service or media stream, there SHALL NOT be identical srvKEY value(s) between STKM streams

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

			having the same "bcastversion" parameter value. The Terminal SHOULD use the STKM stream with one matching srvKEY. In the case there are more than one matching STKM streams, the Terminal MAY use any of those STKM streams: it's out of the scope of the current specification to further detail rules regarding the selection process on the terminal side. Hence, to avoid heterogeneous behaviors on the Terminal side, the basic assumption is that the Terminal is provisioned with SEKs so that there is only one STKM stream to match at a given time, and the BSM is RECOMMENDED to enforce such provisioning policy. The network and terminal SHALL support this field in case the Smartcard Profile is supported.

where, NM=Mandatory for network to support; NO=Optional for network to support; TM=Mandatory for terminal to support; TO=Optional for terminal to support.

The tables below shows the corresponding <field values> for the <field names>:

Table 51: kmstype Values

Value (String)	Semantics
oma-bcast-drm-pki	DRM Profile Key Management System
oma-bcast-gba_u-mbms	Smartcard Profile Key Management System, using 3GPP GBA_U to establish Layer 1 keys
oma-bcast-gba_me-mbms	Smartcard Profile Key Management System, using either 3GPP GBA_ME or 3GPP GBA_U to establish Layer 1 keys
oma-bcast-prov-bcmcs	Smartcard Profile Key Management System, using provisioned 3GPP2 BCMCS Symmetric Key Infrastructure

Table 52: bcastversion Values

Value (Decimal x.y)	Semantics
1.0	Current version in this specification is 1.0

Table 53: serviceproviders Syntax and Semantics

Semantics
The value of the 'serviceproviders' SDP parameter is a list of URIs that reference the terminal's affiliated BSM or service providers. The syntax of the 'serviceproviders' value is defined as follows: "<uri>("|"<uri>)*" , where uri is defined as in [RFC3986]. The terminal's affiliated BSM(s) are represented within the terminal as Management Objects with identifier '<X>/ BSMFilterCode' or as codes on the Smartcard as defined by [3GPP TS 22.022], [3GPP2 C.S0068-0], [3GPP TS 31.102 v7], [3GPP2 C.S0023-C], or [3GPP2 C.S0065-0]. The terminals' affiliated service provider(s) are represented within the terminal as Management Objects with identifier '<X>/ ServiceProvider'. Either all or none of the STKM streams SHALL have a 'serviceproviders' parameter instantiated in the SDP. Either all or none of the LTKM streams SHALL have the 'serviceproviders' parameter instantiated in the SDP.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

If all of the STKM streams or LTKM streams have the 'serviceproviders' parameter instantiated in the SDP, the terminal SHALL only use STKM or LTKM streams for which either:

- the 'serviceproviders' parameter contains a URI referencing to the identifier of a BSMSelector as declared within the''BSMLis'' of the SGDD of the SG [BCAST10_SG] with a BSMFilterCode that does match to any of the '<X>/BSMFilterCode' entries within the terminal or any of the codes on the Smartcard
- or the 'serviceproviders' parameter contains a URI that does match with any of the <X>/Serviceproviders/<X>/ID entries within the terminal.

If none of the STKM streams or LTKM streams have a 'serviceproviders' parameter instantiated in the SDP, the terminal MAY use any of the streams.

The network SHALL ensure that the sets of BSMSelector IDs and <X>/ServiceProviders/<X>/ID values are disjoint.

Table 54: streamid Values

Value (Decimal)	Semantics
1, 2, 3, etc.	Provides identification of media streams which can be used to associate an encrypted media stream with the corresponding STKM. Each stream declared in the SDP will be uniquely numbered. Only positive integers are acceptable. It is RECOMMENDED that streams are numbered in increasing order. Duplicate streamids SHALL be ignored, i.e. only the first one SHALL be used.

Table 55: BaseCID Values

Value (String)	Semantics
<BaseCID>	<BaseCID> is part of the Service or Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. Upon reception of a STKM, the terminal can assemble the service_CID/program_CID/BCI and look up the SEK or PEK (wrapped inside a LTKM) as described in Section 5.5.1.

Table 56: srvCIDExt Values

Value (unsignedByte)	Semantics
<srvCIDExt>	service_CID_extension is part of the Service CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. Upon reception of a STKM, the terminal can assemble the service_CID/BCI and look up the SEK (wrapped inside a LTKM) as described in Section 5.5.1. This parameter provides the value of the most significant byte of the service_CID_extension in the corresponding STKM stream.

Table 57: prgCIDExt Values

Value (unsignedByte)	Semantics
<prgCIDExt>	program_CID_extension is part of the Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. Upon reception of a STKM, the terminal can assemble the program_CID/BCI and look up the PEK (wrapped inside a LTKM) as described in Section 5.5.1. This parameter provides the value of the most significant byte of the program_CID_extension in the corresponding STKM stream.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 58: srvKEYList Values

Value (unsignedByte)	Semantics
<srvKEYList>	The value of the 'srvKEYList' parameter is a list of so-called 'srvKEY' Base64 encoded data. Each of those 'srvKEY' value corresponds to the concatenation of the Key Domain ID with the Key Group part of a SEK/PEK which applies to the related Smartcard Profile STKM stream. The syntax of the 'srvKEYList' value is defined as follows: "<base64>("|"<base64>)*", where base64 is defined as in [RFC3548].

10.1.2 Short-Term Key Message Streams (STKM)

This section specifies descriptions of short-term key message (STKM) streams using SDP.

10.1.2.1 Description

To support efficient STKM carriage, each STKM Stream is carried in its own UDP stream. The MIME type application/vnd.oma.bcast stkm is defined to signal an STKM Stream.

The location of an STKM stream is signaled within the SDP file used to describe the delivery parameters for a given service. The SDP file describing the service typically contains a media announcement entry for the video and one for the audio. In addition, to signal the associated STKM streams, one or two additional stream announcements are added.

An STKM stream is signaled in the following way:

m= application <port> udp vnd.oma.bcast.stkm.

MIME type parameters are signaled in the "a=fmtp:" line. MIME type parameters for STKM as defined in Table 59 SHALL be supported.

Table 59: Parameters of the MIME Type application/vnd.oma.bcast.stkm

Parameter	Terminal support	Server support	Purpose
streamid	Mandatory	Mandatory	See Table 50
kmstype	Mandatory	Mandatory	See Table 50
serviceproviders	Mandatory	Mandatory	See Table 50
baseCID	Optional	Optional	See Table 50
srvCIDExt	Optional	Optional	See Table 50
prgCIDExt	Optional	Optional	See Table 50
srvKEYList	Optional	Optional	See Table 50

10.1.2.2 SDP Example for Short –Term Key Message Streams

This Section gives an example of SDP descriptions of short term key streams:

m= application 49230 udp vnd.oma.bcast.stkm

c= IP4 224.2.17.12/127

a=fmtp:vnd.oma.bcast.stkm streamid=10; serviceprovider=DiscountBcast; kmstype=oma-bcast-drm-pki

10.1.3 Short-Term Key Message (STKM) Streams Binding

The signalling described below allows the terminal to clearly identify which STKM streams are relevant for each media stream. Several media streams may reference the same STKM stream, thereby sharing the same Traffic Encryption Keys, but

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

each media stream may also reference a different STKM stream. An encrypted media stream must reference one or more STKM streams, each providing secure delivery of the same Traffic Encryption Keys (TEKs):

- In the case of the Smartcard Profile, one or more STKM streams carry the STKM as specified in Section 6.7.
- In the case of the DRM Profile, one or more STKM streams carry the STKM as specified in Section 5.5.

AES in counter mode requires that the same key stream is never reused. In the case that the same STKM stream is shared among several media streams – a distinct IV must be provided for each such media stream. This is already the case for SRTP-based encryption (where each IV is based on the SSRC value in the RTP header).

In the case of IPsec, only AES in CBC mode is currently supported.

In the case of ISMACryp the effective IV value is based on the salting key k_s that can be made different for each media stream. To ensure that ISMACryp can safely allow sharing of the same STKM stream between multiple media streams, each such media stream MUST have a unique salting key k_s specified in the SDP file.

Example: A service comprising a video stream and an audio stream, both encrypted with the same Traffic Encryption Keys, and protected by two different KMSs will make use of 4 streams: one for the video, one for the audio, one for KMS#1 (supporting DRM Profile) STKM stream and one for KMS#2 (supporting Smartcard Profile) STKM stream.

This way, the terminal will only listen to and process the STKM stream coming on the relevant IP connection. SDP [RFC4566] is used to describe the STKM stream(s) associated with each media stream. The following attribute is defined for mapping STKM streams to media streams in the SDP:

Table 60: Definition of STKM Stream SDP Attribute

Attribute	Terminal support	Server support	Type	Purpose
stkmstream	Mandatory	Mandatory	Stream Reference	Reference to the ID of the STKM stream (assigned by the parameter "streamid") indicating which STKM stream applies to this media stream .

The attribute can be at session level, in which case it applies to all media streams, or the attribute can be at media level, in which case it only applies to the specified media and would override possible session level attributes.

The SDP attribute stkmstream as defined in Table 60 SHALL be supported. (See also informative Appendix I.)

Each session or media stream can have multiple stkmstream attributes. Using this attribute the terminal can lookup the corresponding STKM stream announcements and figure out which one to listen to and process. We note that this attribute is optional and hence would not be there for unencrypted media streams.

10.1.3.1 STKM Streams Binding Example

Below is an example where two STKM streams (10 and 11) are associated on session level with the media streams, however two other STKM streams (13 and 14) are associated to a second audio track. The stkmstream attribute on media level overrides the stkmstream attribute on session level for that particular media stream. In this example, to decrypt the Spanish audio track, STKM stream 13 or 14 can be used.

```
v=0
o=BCAST 2890844526 2890842807 IN IP4 126.16.64.4
s=A protected Bcast stream
c=IN IP4 224.2.17.12/127
t=2873397496 2873404696
a=recvonly
a=stkmstream:10
a=stkmstream:11
```

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
m=audio 49170 RTP/AVP 0
a=lang:en

m=audio 52002 RTP/AVP 0
a=lang:ES
a=stkmstream:13
a=stkmstream:14
```

In the case of English language audio track, this signalling announces that to gain access to the English audio stream, the terminal may use either the STKM with streamid=10, or the one with streamid=11. The terminals can then look up in the same SDP file for both two STKM streams (identified by their streamid), and to identify the KMS and the operator each is associated with. Similarly Spanish language audio track can be decrypted using STKM streams with id 13 or 14 in the same SDP file. Then, on the basis of this information and depending on which KMS it is supporting, the terminal can decide which stream it needs to listen to in order to get the short-term key message (STKM) stream it requires.

10.1.4 Long-Term Key management Message (LTKM) Stream

This section defines the description of LTKM stream using SDP. The signalling specified in this sub-section SHALL be used for LTKM streams carried over the broadcast channel (DRM Profile BCROs), and it SHALL NOT be used for LTKMs carried over interactive channel. (DRM Profile rOs and Smartcard Profile LTKMs).

10.1.4.1 Description

The MIMEe type for long-term key management message (LTKM) streams (e.g. stream carrying rights objects/entitlements) is application/vnd.oma.bcast.ltkm.

A long term key management message stream is signaled in the following way:

m=application <port> udp vnd.oma.bcast.ltkm.

The actual format of the key management message stream is given by the kmstype in the “a=fmtp” line. Every such line may contain a parameter streamid which identifies the particular LTKM stream.

MIME type parameters are signalled in the “a=fmtp” line. MIME type parameters for LTKM as defined in Table 61 SHALL be supported.

Table 61: Parameters of the MIME Type bcast-ltkm

Parameter	Terminal support	Server support	Purpose
kmstype	Mandatory	Mandatory	See Table 50
serviceproviders	Mandatory	Mandatory	See Table 50

10.1.4.2 SDP Example for LTKM Stream

m=application 49230 udp vnd.oma.bcast.ltkm

c=IN IP4 224.2.17.12/127

a=fmtp:vnd.oma.bcast.ltkm kmstype=oma-bcast-drm-pki; serviceprovider=SOMEID

10.1.5 SDP Entry Examples (Informative)

This section provides several examples illustrating how the parameters defined above are signalled in an SDP file. Note that these are simplified example i.e. lots of parameters are missing, but these have been omitted for clarity.

Example 1: This example shows a video and audio stream protected by both Long Term and Short Term Key Message streams using DRM Profile.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

```
m=video 49168 RTP/AVP 96i=video
c=IN IP6 FF15:0:0:0:0:0:81:1BC
a=rtpmap:96 H264/90000
a=fmtp:96 <rtp_param>
a=stkmstream:3

m=audio 49170 RTP/AVP 97
i=audio
c=IN IP6 FF15:0:0:0:0:0:81:1BC
a=rtpmap:97 mpeg4-generic/32000
a=fmtp:97 <rtp_param>
a=stkmstream:3

m=application 49172 udp vnd.oma.bcast.stkm
c=IN IP6 FF15:0:0:0:0:0:81:1BC
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm  streamid=3; kmstype=oma-bcast-drm-pki; ↵
      serviceproviders=DiscountBCAST

m=application 49173 udp vnd.oma.bcast.ltkm
c=IN FF15:0:0:0:0:0:81:1BC
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.ltkm kmstype=oma-bcast-drm-pki; ↵
      serviceproviders=DiscountBCAST
```

Example 2: This example shows a video and audio stream protected by Short Term Key Message streams using GBA_ME MBMS.

```
m=video 49168 RTP/AVP 96
i=video
c=IN IP4 224.2.1.1
a=rtpmap:96 H264/90000
a=fmtp:96 <rtp_param>
a=stkmstream:3

m=audio 49170 RTP/AVP 97
i=audio
c=IN IP4 224.2.1.1
a=rtpmap:97 mpeg4-generic/32000
a=fmtp:97 <rtp_param>
a=stkmstream:3

m=application 49172 udp vnd.oma.bcast.stkm
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=3; kmstype=oma-bcast-gba_me-mbms; ↵
      serviceproviders=DiscountBCAST
```

Example 3: This example shows two audio streams, each protected by a different key stream

```
m=audio 49170 RTP/AVP 96
```

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
i=audio_english protected by stkm with id 3
c=IN IP4 224.2.1.1
a=rtpmap:96 mpeg4-generic/32000
a=fmtp:96 <rtp_param>
a=stkmstream:3

m=audio 49172 RTP/AVP 97
i=audio_spanish protected by stkm with id 3
c=IN IP4 224.2.1.1
a=rtpmap:97 mpeg4-generic/32000
a=fmtp:97 <rtp_param>
a=stkmstream:4

m= application 49174 udp vnd.oma.bcast.stkm
i=short term key messages
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=3; kmstype=oma-bcast-drm-pki; ↵
      serviceproviders=DiscountBCAST1|supertv.tv

m=application 49175 udp vnd.oma.bcast.stkm
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=4; kmstype=oma-bcast-gba_u-mbms; ↵
      serviceproviders=DiscountBCAST2|supertv.tv
```

Example 4: This example shows how two separate providers can use different key streams to give access to the same video stream (audio stream left out for brevity). The different key streams carry the same keys. The second service provider uses two key streams with different service_CID_extension values.

```
m=video 49168 RTP/AVP 96
i=video
c=IN IP4 224.2.1.1
a=rtpmap:96 H264/90000
a=fmtp:96 <rtp_param>
a=stkmstream:2
a=stkmstream:3
a=stkmstream:4

m=application 49171 udp vnd.oma.bcast.stkm
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=2; kmstype=oma-bcast-gba_me-mbms; ↵
      serviceproviders=supertv.tv

m=application 49190 udp vnd.oma.bcast.stkm
i=short term key messages
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=3; kmstype=oma-bcast-drm-pki; ↵
      serviceproviders=bargain.tv; srvCIDExt=2
```

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

```
m=application 49191 udp vnd.oma.bcast.stkm
i=short term key messages
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=4; kmstype=oma-bcast-drm-pki; ↵
      serviceproviders=bargain.tv; srvCIDExt=8
```

Example 5: This example shows how a service provider, using the Smartcard profile, can be associated to different key streams that give access to the same video stream (audio stream left out for brevity). The different key streams carry the same keys. The service provider uses two key streams with different "srvKEYLis" values.

```
m=video 49168 RTP/AVP 96
i=video
c=IN IP4 224.2.1.1
a=rtpmap:96 H264/90000
a=fmtp:96 <rtp_param>
a=stkmstream:3
a=stkmstream:4

m=application 49190 udp vnd.oma.bcast.stkm
i=short term key messages
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=3; kmstype=oma-bcast-gba_u-mbms; ↵
      serviceproviders=bargain.mo; srvKEYList=ggABAAI=|ggABAAJ=

m=application 49191 udp vnd.oma.bcast.stkm
i=short term key messages
c=IN IP4 224.2.1.1
a=bcastversion:1.0
a=fmtp:vnd.oma.bcast.stkm streamid=4; kmstype=oma-bcast-gba_u-mbms; ↵
      serviceproviders=bargain.mo; srvKEYList=ggABAAQ=
```

10.2 SDP Signalling of ISMACryp

SDP signalling of streams encrypted with ISMACryp SHALL be done as described in [ISMACRYP11] or section 8 of [ISMACRYP20], as applicable. Terminals that support ISMACryp SHALL also support this signalling with the following extension for the DRM profile:

MasterSaltKey

For the DRM Profile, when SRTP authentication is used, the 112-bit Master Salt (MS) MAY be signalled as follows in the "a=fmtp" line of the SDP, if it is not sent in the STKM:

MasterSaltKey=MS where MS is the 112-bit master key used for the derivation, base64 encoded.

10.3 Service Guide Signalling

Session Description information is contained or referenced in Access fragment of the Service Guide [BCAST10-SG].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

10.4 SDP Signalling of SRTP

When SRTP is used, the following specific paramaters SHALL be included into the SDP; (see also informative Appendix I):

- a=SRTPAuthentication:n

where n is the SRTP authentication algorithm value for the authentication algorithm to use. Only values specified in [RFC4771] are allowed, i.e. values 0 and 1 representing NULL and HMAC-SHA-1-160 are not allowed.

- a=SRTPROCTxRate:R

where R is the value of the ROC transmission rate parameter, an integer between 1 and 65535 inclusive, as specified in [RFC4771].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

11. Common Keys / Sharing Streams for DRM Profile and Smartcard Profile

This section explains how different keys are mapped between the DRM Profile and the Smartcard Profile. It also explains how a protected data stream can be shared between different operators using both DRM and Smartcard Profiles.

11.1 Service and Program Encryption Keys

For the DRM Profile, Service Encryption Keys (SEKs) and Program Encryption Keys (PEKs) are as described in Section 5.4.

For the Smartcard profile the PEK and SEK map to the same key and the differentiation is based on the Key Validity data, e.g. the PEK Key Validity data will define a shorter validity period than the SEK Key Validity data. This enables the same key to be used for both subscription and Pay-Per-View service offerings.

The mapping between Smartcard Profile keys and MBMS keys is described in Section 6.2.

11.1.1 Mapping of Encryption and Authentication Keys

The SEK/PEK used within the Smartcard Profile is not used directly to secure the delivery of Traffic Encryption Keys (TEKs). Instead the MIKEY protocol [RFC3830] uses the SEK/PEK to derive an integrity key (auth_key) and encryption key (encr_key).

The DRM Profile similarly utilizes separate encryption and authentication keys to encrypt the Traffic Keys and to authenticate STKMs. However, in the case of the DRM Profile the Service Encryption Key (SEK) and the Service Authentication Key (SAK) are not derived from the same key. Likewise, the PEK and the PAK are not derived from the same key.

A more detailed mapping of the encryption and authentication keys between the DRM and Smartcard profiles is provided in the following table:

Table 62: BCAST Encryption and Authentication Key Mapping

Purpose of Key	Service or Program Key	DRM Profile Key	Smartcard Profile Key
Encryption of STM	Service	SEK (128 bits)	MIKEY encr_key (128 bits derived from SEK)
Athentication of STKM	Service	SAK (160 bits derived from 128 bit SAS)	MIKEY auth_key (160 bits derived from SEK)
Encryption of STM	Program	PEK (128 bits)	MIKEY encr_key (128 bits derived from PEK)
Athentication of STKM	Program	PAK (160 bits derived from 128 bit PAS)	MIKEY auth_key (160 bits derived from PEK)

11.2 SEK, PEK and TEK Key IDs in STKM

The table below decribes the mapping between key identifiers used in the Smartcard Profile and DRM profile:

Table 63: Mapping between Key Identifiers Used in the Smartcard Profile and DRM Profile

Key to locate	DRM Profile Identifier (contained in STKM)	Smartcard Profile Identifier (contained in STKM)
SEK	service_CID_extension (32 bit)	Key Domain ID ‖ SEK ID (3+4=7 bytes)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

PEK	program_CID_extension (32 bit)	Key Domain ID ‖ PEK ID (3+4=7 bytes)
TEK for IPsec	SPI (32 bits)	SPI (32 bits) = 0x0001 ‖ MTK ID (2 bytes)
TEK for SRTP	MKI (8*key_indicator_length bits) Note that if compatibility with the Smartcard Profile is required MKI must be 2 bytes long	MKI = TEK ID (2 bytes) for SRTP implemenations aimed only at BCAST terminals. MKI = (SEK/PEK ID) ‖ TEK ID (6 bytes) for for SRTP implemenations that are MBMS compatibile
TEK for ISMACryp	key_indicator (8*key_indicator_length bits)	TEK ID (2 bytes)

The terminal MUST use the SDP to locate the relevant STKM stream for the encrypted traffic stream it needs to decrypt.

For information on how the parameteres in the above table should be specified for the case where both DRM Profile and Smartcard Profile provide access to the same data stream, please refer to Section 11.3 below.

11.3 Sharing SRTP Protected Data Streams

This section describes how a protected data stream can be shared between different operators using both DRM and Smartcard Profiles. Two solutions for sharing protected data streams are described below:

- Section 11.3.1 introduces a solution that is compatible with the 3GPP MBMS key management specification.
- Section 11.3.2 introduces a solution aimed at BCAST terminals only.

Section 11.3.3 discusses the properties of these solutions.

11.3.1 Sharing 3GPP-MBMS Compatible SRTP Protected Media Streams

The way in which key identifiers are used by the SRTP implementation is based on the MBMS specification (cf. SRTP). Specification details related to this section are described in [BCAST10-MBMS-Adaptation] in the section titled “Sharing 3GPP-MBMS compatible SRTP protected media streams”.

11.3.2 Sharing a Protected Media Stream where Content is Aimed only at BCAST Terminals

Compared with Section 11.3.1, this section outlines how to handle the sharing protected stream(s) between different broadcast service providers none of which are using MBMS service protection.

The management and use of key identifiers for the protected media stream is based on the BCAST specification. It simplifies the way in which the MKI is constructed allowing the use of SRTP and ISMACryp at the content encryption layer. Note that the solution is not compatible with the 3GPP MBMS key management solution.

A shared TEK ID enables the retrieval of the correct TEK required to decrypt the protected media stream. This necessitates that:

- A single TEK ID SHALL be used to enable access to the corresponding shared protected stream among different broadcast service providers.
- The TEK ID SHALL be synchronised for all broadcast service providers.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- The same TEK material SHALL be used by all broadcast service providers. This means:
  - the same 128 bit Master Key (MK) SHALL be used.
  - the same 112 bit Master Salt (MS) SHALL be used. The MS MAY be NULL.
  - the key derivation rate SHALL be zero for the DRM Profile. For the Smartcard Profile the key derivation rate may be zero or non-null.

SRTP key management parameters used when an SRTP stream is to be shared between DRM Profile and Smartcard Profile Terminals are summarised in the table below.

Table 64: SRTP Parameters – to Enable Sharing Common Stream

SRTP Parameter	DRM Profile value	Smartcard Profile value
MKI	same as Smartcard	TEK ID (2 bytes)
MK	same as Smartcard	TEK (random 128 bits)
MS	same as Smartcard	random 112 bits or NULL

The figures and section below explain the MKI format used to allow DRM Profile and Smartcard Profile to be shared between different broadcast service providers.

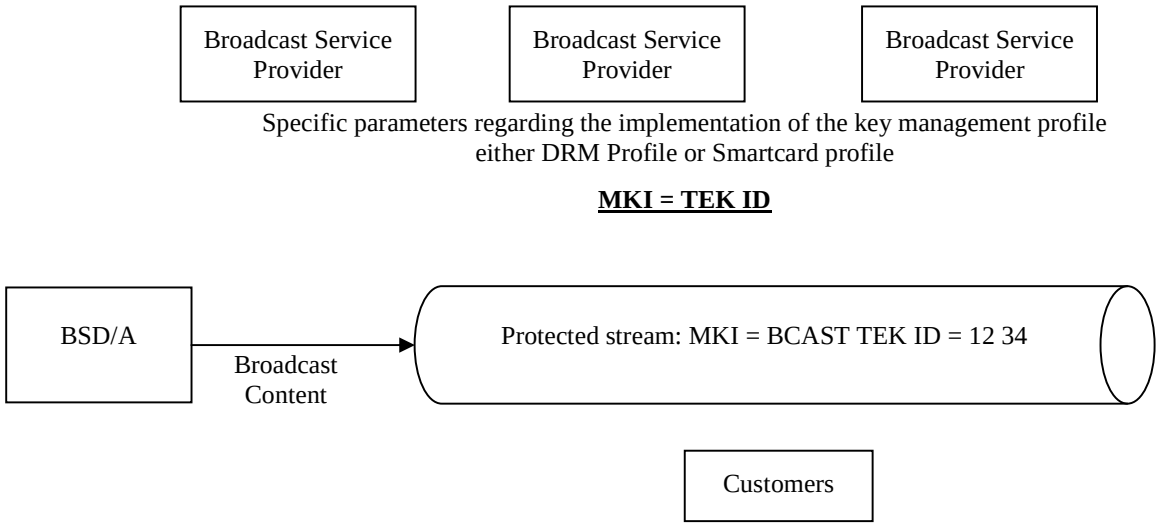


Figure 11 – Sharing a Single Protected Media Stream between Several Broadcast Service Providers using the Smartcard Profile and the DRM Profile, where there is no Requirement to also Share the Protected Stream with MBMS only Terminals

Figure 11 illustrates how a single broadcast content distributed by the Service Provider (BSD/A) is shared between three broadcast service providers A, B and C. Broadcast service providers A, B and C implement either the DRM Profile or the Smartcard Profile.

The TEK material and the corresponding TEK ID must be shared among broadcast service providers. Broadcast service providers A, B and C generate and use their own SEK/PEK material to protect the shared TEKs. Each broadcast service provider constructs and broadcasts their own STKMs to their subscribers, containing the common TEK protected with the service provider specific SEK/PEK. There is no need to synchronise any key material or key identifiers above the TEK level.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The service provider can then broadcast the content encrypted with this common TEK by using SRTP. Upon reception the terminal retrieves the TEK based on the MKI where:

MKI = TEK ID (2bytes)

Each BCAST terminal can then retrieve the TEK needed to decrypt the shared protected stream from the STKM provided by their broadcast service provider.

The following can then be considered:

- A subscriber from broadcast service provider A has access to media streams 1, 2 and 3, using
  - ◊ SEK ID = SEK _IDA, key material = SEK A, and has a key validity period = 1 week. SEK A is transmitted by broadcast service provider A.
  - ◊ TEK ID = TEK _ID, key material = TEK, and has a key validity period = TEK _period, transmitted with broadcasted content.
- A subscriber from broadcast service provider B has access to media streams 1, 2 and 4, using
  - ◊ SEK ID = SEK _IDB, key material = SEK B, and has a key validity period = 1 month. SEK B is transmitted by broadcast service provider B.
  - ◊ TEK ID = TEK _ID, key material = TEK, and has a key validity period = TEK_period, transmitted with broadcasted content.

The value of the SEK/PEK ID is not shared and is specific to each broadcast service provider. The frequency of the update of SEK is up to each broadcast service provider.

In contrast, the value for TEK ID and TEK material have to be synchronised and coordinated for all broadcast service providers.

Summary:

In summary, for the Smartcard Profile to share protected media streams with DRM profile terminals, when the broadcast media is protected using SRTP, it must deviate from the MBMS specifications [TS 3GPP 33.246]. As described in Section 11.3.1, when SRTP is used in MBMS the MKI value is constructed as a concatenation of MSK ID (SEK/PEK ID) and MTK ID (TEK ID):

MKI = (MSK ID || MTK ID)

where MSK ID is 4 bytes long and MTK ID is 2 bytes long resulting in an MKI length of 6 bytes.

In contrast, when following the scheme described in this section, the MKI value is constructed using only the TEK ID (equivalent to the MTK ID in MBMS):

MKI = TEK ID = MTK ID

where TEK ID is 2 bytes long.

Restructuring the way that the MKI is formatted by omitting the SEK/PEK ID (equivalent to the MSK ID in MBMS) removes the need for broadcast service providers to synchronise the SEK/PEK ID and SEK update period as described in Section 11.3.1. The deviation from the MBMS specification means that it is not possible for a media stream protected in the manner described in this section to also be shared by MBMS only terminals.

It is also necessary for terminals implementing the Smartcard Profile to recognise whether the MKI in the SRTP stream they are trying to decrypt is constructed in the way described in Section 11.3.1 or 11.3.2 in order to find the required TEK. This is achieved by looking at the MKI length: if it is 2 bytes long then the MKI corresponds to the TEK ID transported in the STKM; if the MKI length is 6 bytes long, the MKI corresponds to the SEK/PEK ID || TEK ID, both of which are transported in the STKM. Note that in the first case the MKI signalled in SRTP does not contain the SEK/PEK ID, but the SEK needed to decrypt the STKMs is still signalled in the STKM, i.e in the EXT MBMS payload.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

11.3.3 Properties of the above Solutions

For the solution that is compatible with [3GPP TS 33.246 v7] the following can be stated:

◊ It is possible to share a protected media stream between a broadcast service provider using MBMS and a broadcast service provider using another broadcast bearer with either the Smartcard or DRM Profile.

◊ The requirement for all broadcast service providers that are sharing the same protected stream, regardless of the profile that they are using, is to use the same SEK/PEK ID. Using the same SEK/PEK ID further necessitates that the update frequency of the SEK and SEK/PEK ID must also be coordinated.

For the solution that aims protected media stream sharing at BCAST terminals only, the following can be stated:

◊ There is no need for broadcast service providers to synchronise key identifiers or key update periods above the TEK layer

◊ Broadcast service providers using MBMS service protection (following [3GPP TS 33.246 v7]) cannot share streams with broadcast service providers using the OMA BCAST service protection not relying on SRTP encryption protocol.

In summary, both solutions are possible, and signalling information within the STKM might be necessary.

11.4 Sharing Streams using ISMACryp

This section explains how a single protected media stream encrypted using ISMACryp can be accessed by different broadcast service providers.

As explained in Section 9.3, the ISMACrypContextAU indicates the KeyIndicator (TEK ID) in the protected content stream. This KeyIndicator is used to find the relevant TEK used to decrypt the content in the STKM stream. In the head-end, the TEK and TEK ID are used during STKM generation by BSM / BSDA so that broadcast service providers can generate their own STKM streams (using DRM Profile or Smartcard Profile). These are broadcast together with the protected content stream.

The KeyIndicator can be found in the STKMs for the DRM Profile and for the Smartcard Profile. SDP signalling provides information on the relevant STKM streams (see Section 10.1) indicating whether the STKM stream is a DRM Profile stream or a Smartcard Profile stream. Furthermore, the "serviceprovider" string allows individual broadcast service providers to use their own STKM stream.

Once the correct STKM stream has been identified, the terminal can obtain the correct TEK and KeyIndicator, and hence match it to the TEK needed to decrypt the content.

To summarise:

- Content is encrypted using a key identified by KeyIndicator (TEK ID) in ISMACrypContextAU
- STKMs are identified in SDP
- STKMs contain the TEK and associated TEK ID (KeyIndicator)
- The correct TEK is used to decrypt protected content

11.5 Sharing (P)DCF File Delivery Protection using TEK (Informative)

Section 9.4 explains how (P)DCF files can be protected by individual TEKs during file delivery.

This section highlights how the OMABCASTKeyInfo box can be used to allow both DRM Profile and Smartcard Profile signalling to be provided, allowing both to operate in parallel. Indeed, the Key Info box allows multiple "KeyInfo" entries to be present.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

11.5.1 Use of OMABCASTKeyInfo Box

For the DRM Profile, the KeyIDType would indicate that information on KeyID corresponds to the DRM Profile.

For the Smartcard Profile, the KeyIDType would indicate information on KeyID corresponds to the Smartcard Profile. If multiple service providers are using the Smartcard Profile, multiple KeyInfo entries can be provided for multiple KeyIDs.

Note also that the use of the OMABCASTKeyInfo box allows OMA DRMv2 compatibility to be maintained, potentially allowing a (P)DCF to be provided that is compatible for OMA DRMv2, OMA BCAST DRM Profile and OMA BCAST Smartcard Profile.

Note also that 3GPP MBMS also uses the same approach, so that 3GPP MBMS signalling can also be provided in parallel. This allows interoperability across OMA DRM, OMA BCAST and 3GPP MBMS domains.

The OMABCASTKeyInfo box allows the STKM used to provide the TEK for decryption of DCF files to be stored or delivered within the DCF. Again, as this is contained within a single "KeyInfo" entry, STKMs for different key management systems can be provided in parallel.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

12. Terminal Binding Key

In case of Smartcard Profile, a Permissions Issuer MAY elect to bind some or all of the content being broadcasted to valid terminals by the use of a Terminal Binding Key (TBK). This binding is in addition to the UICC binding provided by the Smartcard Profile. The binding is signalled in the SG and in the STKM and LTKM for the Smartcard Profile. TBK is not applicable for the DRM profile.

The following section and subsections are MANDATORY to support for Terminals with the Smartcard Profile for content protection. In all other cases, the sections are OPTIONAL for both server and terminal to support.

12.1 TBK Generation

If Terminal Binding is desired for any of the content being broadcasted, the Permissions Issuer will define the TBK to be a randomly, or pseudo-randomly, generated key of 128 bits. This key will be shared by all compliant non-revoked devices. For each TBK generated, the PI will issue a unique 32 bit TerminalBindingKeyID.

The TBK can be changed by the PI at will, such as when devices need to be revoked. The TBK change can occur as seldom as never once it was set, or as frequently as desired.

A single TBK can be set for the PI to use with all terminal-bounded content, or a separate TBK may be set for contents related to each SG entry. The scope and lifetime of the TBK are implementation specific.

12.2 Encrypting of TEKs with TBK

The TBK used to protect TEKs is used as follows:

Upon generation of each TEK, the PI determines if it would like to bind the TEK also to the terminal. If not, the TEK is processed as usual (encrypted by SEK/PEK). If terminal binding is desired, a TBK has already been generated, given an ID (TerminalBindingKeyID), and this ID was added to the SG entry. For each TEK generated while terminal binding is on, an *Encrypted_TEK* is computed as follows:

$$Encrypted\_TEK = \text{AES-ECB}_{TBK}(TEK)$$

Where *TBK* is fed as the 128-bit key that is used (referred to as *KEK* in AES-ECB), and *TEK* is fed as the key to be encrypted (referred to as *plaintext* in AES-ECB). The resulting *encrypted_TEK* is processed from that point onwards instead of the original, plaintext, *TEK*.

12.3 Decrypting of TEKs with TBK

When content is selected to be processed from the SG, the terminal will note the ID of the TBK that is being used with that content, if at all. If a TBK of the specified ID is not available in the terminal cache, the terminal MAY attempt to obtain it, as described in Section 12.4.

When processing a STKM, if the terminal binding flag bit is set, the terminal will fetch from its cache the correct TBK, according to the TerminalBindingKeyID specified in the SG. This fetch may occur once when processing the LTKM to avoid repeatedly retrieving the same value from the cache. The terminal will use this TBK to decrypt, using AES-ECB, encrypted TEKs that are received from the UICC, before these are used for content decryption.

The effect of this additional decryption, that is required when terminal binding is on, is that an unapproved terminal, which does not possess the correct TBK, is unable to utilize the output of the UICC to deduce meaningful TEK values. It is perceived as infeasible to obtain the correct TEK values from $\text{AES-ECB}_{TBK}(TEK)$ without knowledge of TBK.

12.4 TBK Acquisition

The PI SHALL deliver any requested TBK value to any requesting Terminal, as long as the Terminal was successfully authenticated and was positively identified as a Terminal that has not been revoked.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The protocol by which TBK values are delivered is initiated by the Terminal at any time, typically when an SG entry indicates the requirement for a TBK that is not cached by the device.

To obtain a TBK value, the Terminal starts an HTTPS session with the PI server (see Section 6.11.2 for the Smartcard Profile). The HTTPS session SHALL be based on mutual authentication using both client and server certificates. The server SHALL verify the authenticity and the validity of the client certificate and SHALL consider the identity of the Terminal to be the one indicated by the certificate.

Following the HTTPS session establishment, the Terminal SHALL send the *BCAST_Client_ID* (see Section 6.11.2). The PI server MAY use this ID information, but if doing so it SHALL assure that the identity of the terminal as reflected in the *BCAST_Client_ID* matches the identity indicated by the client certificate mentioned above.

If the terminal ID that is supplied in the *BCAST_Client_ID* does not match the ID indicated by the client certificate, or if the ID reflects a device that has been revoked, or if the identification failed, or if the HTTPS session failed, then the PI server SHALL close the connection without providing the requested TBK but while returning a "*Forbidden*" error instead.

If the version number sent in the *BCAST_Client_ID* reflects an inadequately old version, the PI server SHALL close the connection without delivering the requested TBK, and MAY indicate the URI at which an update or further information can be found (see response table).

If none of the above conditions was met, then the PI server SHALL return the required TBK over the secure connection and close the connection.

Upon reception of the requested TBK, the terminal MAY cache it. The policy and size of this cache is implementation specific.

The Figure below illustrates the steps explained above.

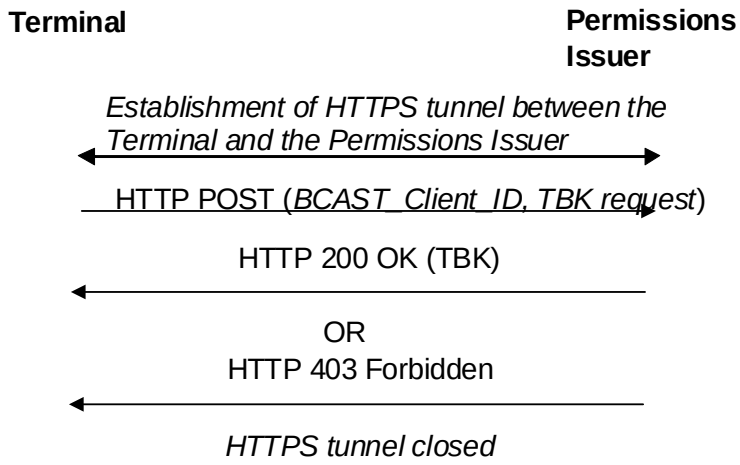


Figure 12 – Mutual Authentication, sending BCAST_Client_ID and TBK Exchange

The BCAST_Client_ID and TBK request SHALL be sent using the following notation:

```
POST / HTTP/1.1
User-Agent:BCAST_Client_ID=BCAST_Client_ID
TBK_request=TerminalBindingKeyID
```

Where: "BCAST_Client_ID" is text allowing the server to identify the BCAST client ID and *BCAST_Client_ID* is the actual value.

BCAST_Client_ID is Base64 encoded. "TBK_request" is text allowing the server to identify the TBK request and TerminalBindingKeyID is the ID of the TBK key being requested.

TerminalBindingKeyID is Base64 encoded.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The Permissions Issuer response, if successful SHALL be sent using the following notation:

```
HTTP/1.1 200 OK
Server: BCAST Permissions Issuer
Date: Thu, 08 Jan 2004 10:13:18 GMT
TBK=TBK
```

Where: "TBK" is text indicating the TBK follows.

TBK is the actual Terminal Binding Key

TBK is Base64 encoded.

If the Permissions Issuer refuses to issue the TBK it SHALL send the following response:

```
HTTP/1.1 403 Not acceptable
Server: BCAST Permissions Issuer
Date: Thu, 08 Jan 2004 10:13: GMT
```

Or:

```
HTTP/1.1 403 Not acceptable
Server: BCAST Permissions Issuer
Date: Thu, 08 Jan 2004 10:13:18 GMT
Update_URI=Update_URI
```

Where: "Update_URI" is text indicating that the URI where update or further information can be obtained, follows.

Update_URI is the URI where an update or further information can be obtained.

Update_URI is Base64 encoded.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13. Server Side Interfaces and Messages

Message flows can be found in the OMA BCAST AD [BCAST10-Architecture].

13.1 Interface SP-4

Interface SP-4 has three functions:

1) To deliver Service and Program key material from SP-M in the BSM to the SP-KD in the BSD/A for the service and content protection.

2) To deliver the LTKM or Registration key material from SP-M in the BSM to SP-KD in the BSD/A, for subsequent broadcast distribution of these data..

3) To deliver of the STKM's from the BSM to the BSD/A.

A BSM that support service or content protection SHALL support Interface SP-4. A BSD/A that support service or content protection SHALL support Interface SP-4.

Two options are given for Interface SP-4:

- Using DVB Simulcrypt based interfaces
- Using BCAST specific interfaces

Interface SP-4 MAY support DVB Simulcrypt as specified in Section 13.1.1.

Interface SP-4 MAY support OMA BCAST specific signalling as specified in Section 13.1.2.

Interface SP-4 SHALL support either DVB Simulcrypt as specified in Section 13.1.1 or OMA BCAST specific signalling as specified in Section 13.1.2.

13.1.1 Interface SP-4: Adaptation of DVB Simulcrypt Head-End Interfaces to the OMA BCAST Environment

This section defines the use of a DVB Simulcrypt interface for SP-4. All normative text in this Section (13.1.1 and sub-sections) applies if DVB-H Simulcrypt is supported over interface SP-4. In this case, support of the interfaces defined in Section 13.1.1.2 is MANDATORY.

Simulcrypting, from a device perspective, means that signalling is available that allows the device to acquire the necessary information based on the supported Key Management Systems (KMS), either the DRM Profile or the Smartcard Profile. Such signalling is obtained from the Service Guide descriptors or the SDP. It also allows multiple Service Providers to generate STKMs and LTKMs i.e. multiple BSMs in the BCAST architecture.

At head-end side, supporting both profiles has implication on the architecture. The final data-cast shall include all necessary information and some data, such as the TEK, have to be shared among the various KMS to ensure well formed STKM. Simulcrypt also defines interfaces between various entities in the head-end. The original Simulcrypt specifications have been extended by DVB to allow a full support of broadcast over IP. This section describes how the DVB Simulcrypt head-end interfaces are adapted to the OMA BCAST environment, where the system is considering IP transport and encryption with IPsec, SRTP or ISMACryp.

13.1.1.1 Reference DVB Head-end Architecture

Figure 13 (extracted from [SIMULCRYPT]) illustrates the DVB Head-end reference architecture as described in [SIMULCRYPT]. For further information on the Simulcrypt system in a DVB environment, refer to this specification.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

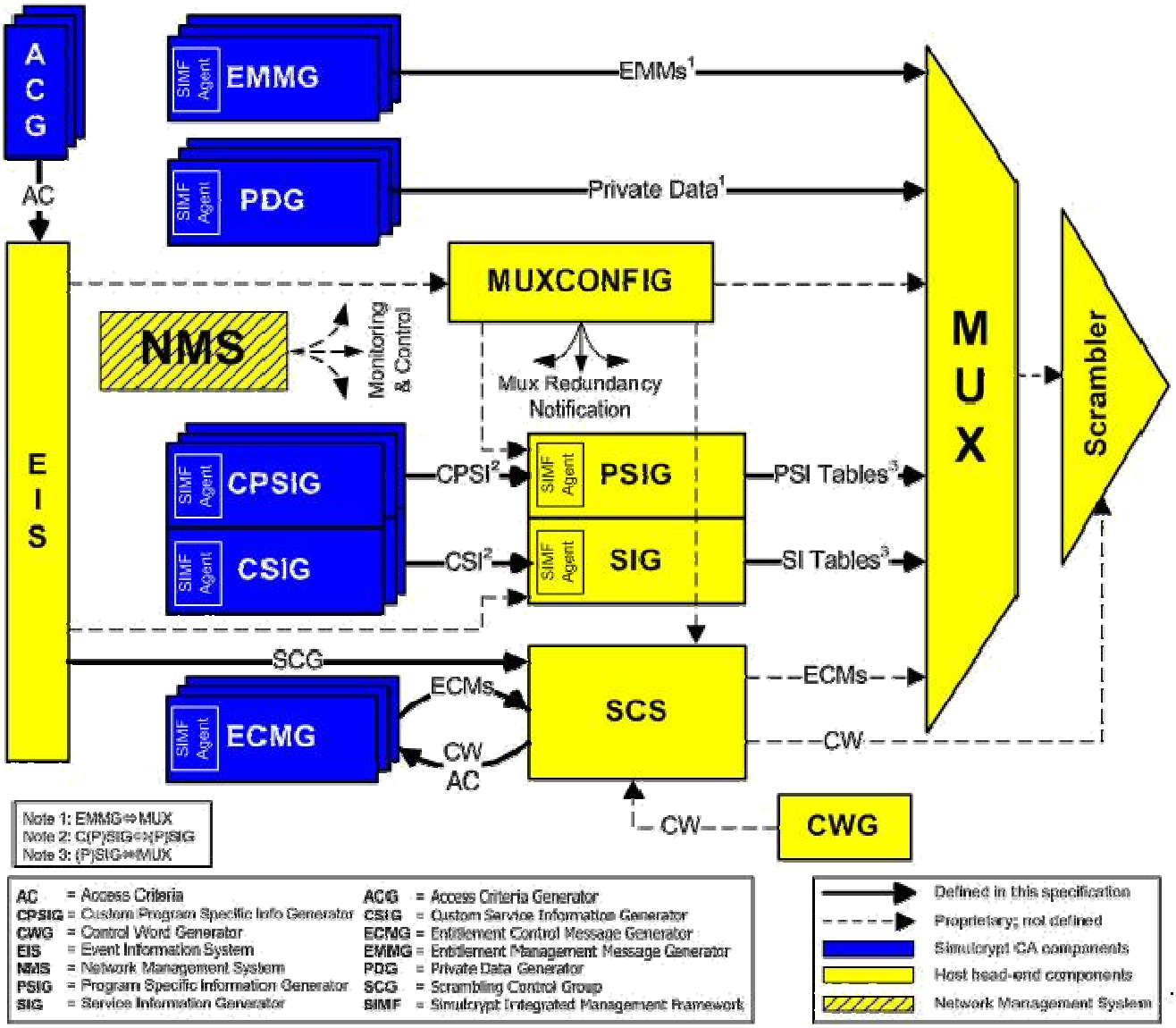


Figure 13 – Reference DVB Head-end Architecture

13.1.1.2 OMA BCAST Head-end Architecture and Interfaces

From Figure 13, the elements to take into account in the OMA BCAST head-end system architecture are (as described in [SIMULCRYPT]). Note that an STKM corresponds to an ECM in Simulcrypt:

- SCS (SimulCrypt Synchroniser). The role of the Simulcrypt Synchronizer is to:
    - Establish TCP connections with ECMGs (STKM generator) and setup one channel per connection;
    - Setup streams within channels as needed and allocate ECM_stream_id values;
    - Get the Control Words (TEK) from the CWG (TEK generator);
    - Supply the CWs to the relevant ECMGs on the relevant streams, as well as any KMS specific information, if any.
    - Acquire ECMs (STKM) from the ECMGs;

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Synchronize the ECMs (STKMs) with their associated Crypto periods according to channel parameters;
- Submit these ECMs (STKMs) to the multiplexer and request their repetition according to the channel parameters;
- Supply the CW (TEK) to the scrambler for use in the specified Crypto period.

- CWG (Control Word Generator–- TEK generator). This component is responsible for generating control words used in scrambler initialization stream. The exact functionality of the Scrambler is implementer specific. The Control Word Generator shall be able to provide the SCS with control words.

- ECMG (Entitlement Control Message Generator) or BCAST STKM Generator (STKMG). The ECMG shall receive CWs in a CW provision message as well as access criteria and shall reply with an ECM or an error message. The ECMG does not support ECM repetition. This corresponds to STKM generation in BCAST.

- EMMG (Entitlement Management Message Generator) or BCAST LTKM Generator (LTKMG). This component, supplied by the KMS system provider shall interface over a specified interface to the multiplexer. The EMMG initiates connections to the multiplexer. This corresponds to LTKM generation in BCAST.

Mapping on the OMA BCAST head-end architecture with these elements is shown in Figure 14. For SP-4 to be applicable in this diagram, the entity containing the ECMG/STKMG must be the BSM.

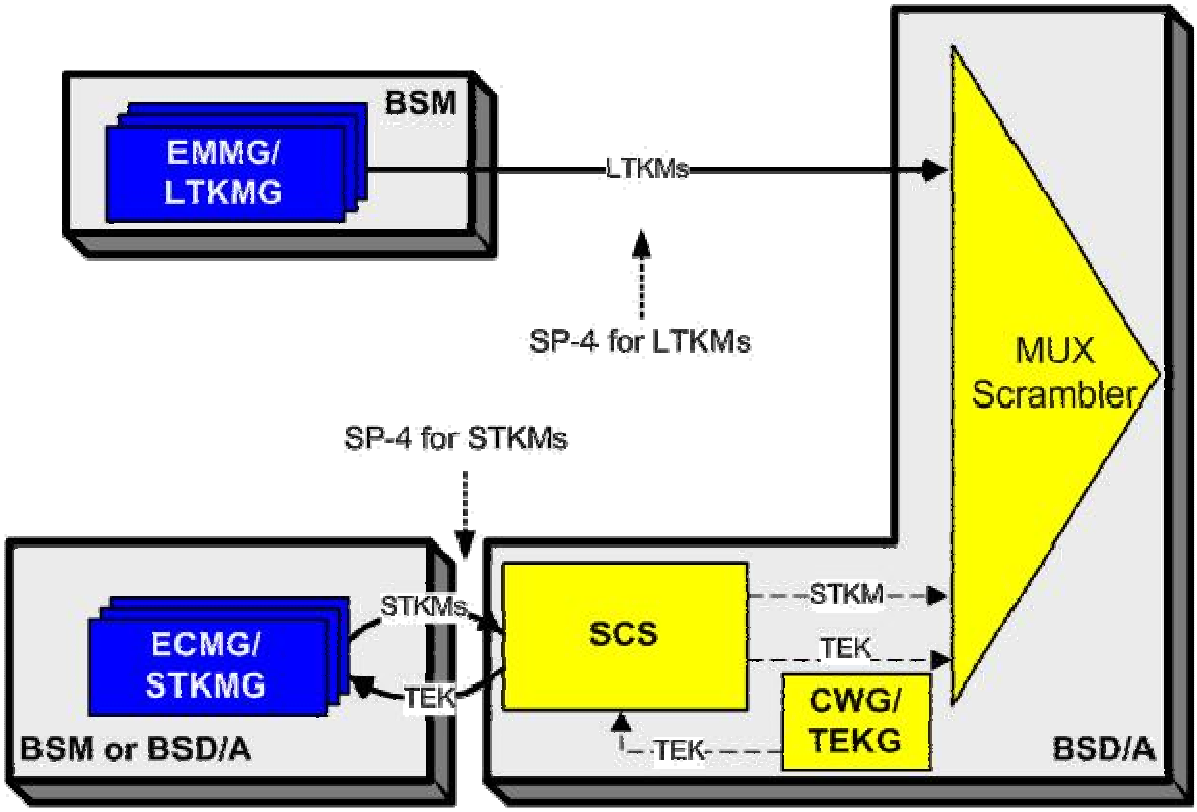


Figure 14 – OMA BCAST Head-end Architecture

The interfaces between the different elements are then as follows. The reference architecture can be mapped to BCAST head-end architecture as follows:

1. Interface ECMG/STKMG ⇔ SCS SHALL be implemented according to [SIMULCRYPT], Section 5 and the modifications in this specification, Section 13.1.1.3.1.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2. Interface EMMG/LTKMG ⇔ Multiplex MAY be implemented according to [SIMULCRYPT], Section 6 and the modifications in this specification, Section 13.1.1.3.2.

Any other interface is out of the scope of this specification and may be proprietary.

Encryption parameters for IPsec, ISMACryp or SRTP are generated by the TEK generator and sent to the SCS. The SCS transfers them to the STKMG as defined in [SIMULCRYPT], allowing the STKMG to generate the appropriate STKMs for DRM Profile and / or Smartcard Profile.

13.1.1.3 Adaptation of Simulcrypt Interfaces to OMA BCAST

13.1.1.3.1 Interface ECMG/STKMG ⇔ SCS

This interface allows a KMS system to provide a SCS with ECMs/STKMs under the control of this SCS. The following provides adaptations of some parameters on this interface. The combination Super_CAS_id + ECM_id identifies uniquely an STKM stream in the whole system. On this interface, the SCS sends TEKs to an ECMG to allow it to generate STKMs. The SCS knows which ECMG it contacts by using the Super_CAS_id. The SCS indicates to the ECMG for which STKM stream the TEKs are used for STKM generation by using the ECM_id, which links to the stkmstream value (also used in the SDP). This allows the ECMG to make the link with the Service Guide information related to the protected stream.

The Super_CAS_id is a 4-byte identifier that uniquely identifies a KMS system provider and BSM. The first 2 bytes of the Super_CAS_id for the DRM Profile are 0x01. The first 2 bytes of the Super_CAS_id for the Smartcard Profile are 0x02. The last 2 bytes are defined by the user (as an example, they can allow to identify the “serviceprovider” or BSM)

The ECM_id is a 2-byte identifier internal to a given ECMG. It is used by the MUX to map the ECM to the correct IP address and port. It is equal to the stkmstream value, as defined in Section 10.1.3 (in this case, the stkmvalue has to be unique across all SDP for a given KMS in a BSM) or it is an ID valid in the head-end only that allows both entities to uniquely identify STKM streams.

The ECM_datagram format is extended for BCAST and can also be formatted as STKM. As a consequence, the section_TSpkt_flag is equal to:

0x02	**The ECMs carried on the interface are in binary STKM format as defined for the DRM Profile or the Smartcard Profile.**
All other values are DVB reserved and SHALL not be used.	

The ECM_datagram is the actual STKM to be sent by the SCS to the MUX. Depending of the value of the Super_CAS_id, it is either a DRM Profile STKM as defined in Section 5.5 or a Smartcard Profile STKM as defined in Section 6.7.

The CP_CW_combination is the concatenation of the Crypto period number, the Key System Information (KSI) and the Traffic Key Material (TKM) from which the TEK and the optional Traffic Authentication Key (TAK) are derived. The meaning and length of the KSI and TEK are defined in the Simulcrypt specification for IPsec, ISMACryp and SRTP. BCAST has extended these for IPsec and for DCF_algo, as shown below:

IPsec Option 1, (as specified in [SIMULCRYPT], can be used with DRM Profile):

IPsec:

- The KSI = SPI, and KSI length = 4 bytes
- IPsec (no auth): TKM = TEK, TKM length = 16 bytes
  - Note: TEK = IPsec encryption key (see Section 5.5.1 and Section 9.1)
- IPsec (auth): TKM = TEK || TAS, TKM length 32 bytes
  - Note: TEK = IPsec encryption key, TAS = IPsec authentication seed (see Section 5.5.1 and Section 9.1)

IPsec Option 2 (can be used with Smartcard Profile):

- The KSI = SPI, and KSI length = 4 bytes
- TKM = TEK, TKM length = 16 bytes

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Note: IPsec encryption key and authentication seed are derived from TEK using MIKEY PRF (see Section 6.7.4 and Section 9.1).

- Note: The Scrambler derives the encryption key and possible the authentication key from TEK, using RAND and CSB ID. How RAND and CSB ID are shared between STKMG and scrambler is outside of the scope of the current specification.

DCF_algo for DRM Profile:

- The KSI is the Key Identifier, and 1<= KSI length =< 255 bytes.
- The TKM is identical to the TEK, and the TKM length = 16 bytes.

DCF_algo for Smartcard Profile:

- The KSI is the TEK ID, and KSI length = 2 bytes.
- The TKM is identical to the TEK, and the TKM length = 16 bytes.

13.1.1.3.2 Error messages on ECMG/STKMG ⇔ SCS

Errors messages defined in section 5.6 of [SIMULCRYPT] are supported. Errors messages defined in 0 are also supported and are in the range 0x8000 to 0xFFFF, i.e., for example, error message 000 defined in 0 is coded as 0x8000.

13.1.1.3.3 Using ECMG/STKMG ⇔ SCS in BCAST (Informative)

This section shows a possible method to use the ECMG/STKMG ⇔ SCS interface in the scope of BCAST. This is provided as an example only.

There is one instantiation of the SCS for each encrypted service in the BSDA. The connection between a SCS and an ECMG/STKM is established in three steps:

- A TCP connection is established as described in section 5.1.2 of [SIMULCRYPT].

- A "Channel" is set-up on top of this TCP connection. The SCS connects to an ECMG/STKMG in the BSM over a TCP connection and assigns to this connection an ECM_channel_id value. This value allows the SCS to uniquely identify the channel to the ECMG/STKMG. This first message contains the super_CAS_id that states which KSM and which service provider are considered. The ECMG/STKMG replies with a status message that contains some information, including its optimal crypto-period, the maximum number of STKM streams it can concurrently create, and the section_TSpkt_flag that defines the format of the STKM it will generate (full detail of this message is given in the channel_status message of section 5.4.3 in [SIMULCRYPT]). It can also reply with a channel_error message.

- On top of this Channel, for each STKM stream that has to be created for the service, a "Stream" is created. The SCS sends a set-up message to the ECMG/STKMG that contains the effective crypto-period duration to use for this STKM stream, the ECM_stream_id equivalent to the ECM_channel_id, and the ECM_id equal to the stkmstream. The ECMG/STKMG replies with a stream_status message or a stream_error message.

Once these three steps are completed, the ECMG/STKMG can provision the SCS with STKMs on the established stream. The CW_provision message allows the SCS to send to the ECMG/STKMG the necessary material (depending on the selected scrambling algorithm, i.e. IPsec, SRTP, or ISMACryp) to create STKMs that are send back in ECM_message response. Sending of TEKs in the CW_provision message can be optionally encrypted with a selectable algorithm.

13.1.1.3.4 Interface EMMG/LTMKG ⇔ Multiplex

This interface allows a BSM to provide a Multiplex with EMMs (LTKMs) under the control of the BSM. This interface applies only if the BSM is creating LTKMs as described for the DRM Profile. LTKMs created for the Smartcard Profile are not broadcasted, but sent over the unicast link the device. The combination {data_type + client_id + data_id} identifies uniquely this new LTKM data stream in the whole system.

The data_type is equal to 0x00, i.e. for EMM/LTKM data which is equivalent to LTKM in OMA BCAST.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

The client_id is a 4-byte identifier uniquely identifying a BSM. The first 2 bytes of the client_id are 0x01 (value for the DRM Profile). The last 2 bytes are defined by the user (as an example, they can allow to identify the "serviceprovider" or BSM).

The data_id is a 2-byte identifier correspond to the ECM_id for the ECMG/STKMG, i.e. it is an internal identifier to the EMMG/LTKMG allowing the MUX to map the EMM/LTKM to a given IP address and port.

The datagram format is extended and can also be formatted as LTKM. As a consequence, the section_TSpkt_flag is equal to:

0x02	**The EMMs carried on the interface are in binary LTKM format as defined for the DRM Profile.**
All other values are DVB reserved and SHALL not be used.	

The datagram is the actual LTKM sent to the MUX.

13.1.1.3.5 Error messages on EMMG/LTMKG ⇔ Multiplex

Errors messages defined in [SIMULCRYPT], section 6.2.6, are supported. Errors messages defined in 0 are also supported and are in the range 0x8000 to 0xFFFF, i.e., for example, error message 000 defined in 0 is coded as 0x8000.

13.1.1.3.6 Using EMMG/LTMKG ⇔ Multiplex in BCAST

This section shows a possible method to use the EMMG/LTKMG ⇔ Multiplex interface in the scope of BCAST. This is provided as an example only.

A channel and then a stream are created over a TCP connection. This follows the same principle than the one presented in Section 13.1.1.3.3 for the ECMG/STKMG ⇔ SCS. Each creation level allows negotiating parameters related to this LTKM exchange, parameters are, among others, the KMS and service provider, the format of data (LTKM in this case).

A first difference appears in the optional possibility to negotiate bandwidth allocation for the LKTM stream. The EMMG/LTKMG can request a specific bandwidth, the Multiplex replies with the effective allocated bandwidth.

A second difference is that the LTKM stream can be provided to the Multiplex over UDP instead of TCP.

13.1.2 BCAST Specific Interface

This section defines a BCAST specific interface that is not compatible with the DVB Simulcrypt interface described above. All normative text below applies if interface SP-4 follows this section.

13.1.2.1 Protocol Stacks

The following protocol stack SHALL be used for messages between the BSD/A and the BSM connected via interface SP-4.

Figure 15 – Protocol Stack for SP-4-1

HTTPS that SHALL be based on .SSL3.0 [SSL30] and TLS 1.0 [RFC2246] SHALL be used to secure the interface between the BSD/A the BSM. All the messages defined over the SP-4 interface are XML documents. The XML schema definition is specified in [BCAST10-XMLSchema-SPCP-Backend].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13.1.2.2 Service and Program key material delivery

For the DRM Profile, SEAK or PEAK is sent from the BSM to the BSD/A.

For the Smartcard Profile, SEK or PEK is sent from the BSM to the BSD/A.

As these messages allow the delivery of high-level key material from BSM to BSD/A, the BSM MAY decide not to do so. This means the BSM MAY decide not to send Key_Delivery messages or MAY send an empty Key_Delivery_Confirmation message with Status code 011 "Operation not Permitted". If the BSD/A receives such a reply then the STKM Delivery section applies (see Section 13.1.2.4).

13.1.2.2.1 Message flows

Tags are defined in the following table to identify the type of each message. There are two cases for the delivery of SEAK or SEK or PEAK or PEK.

1. In the first case the BSD/A sends a Key_Request message to the BSM. The BSM then sends a Key_Request_Response message to the BSD/A.

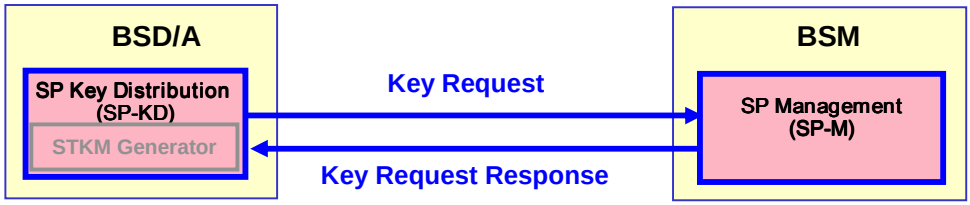


Figure 16 – Message Flow Between BSD/A and BSM for Delivery of Service and Program Key Material

2. In the second case the BSM sends a Key_Delivery message to the BSD/A. The BSD/A then sends a Key_Delivery_Confirmation message to the BSM.

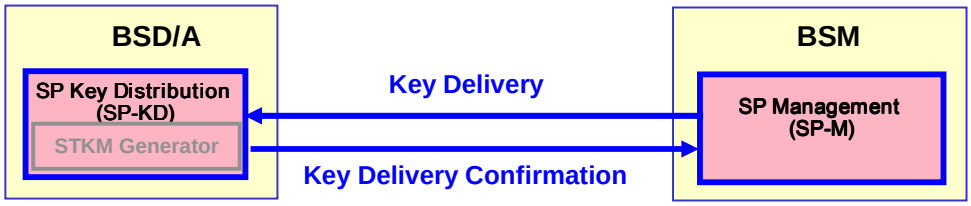


Figure 17 – Alternative Message Flow Between BSD/A and BSM for Delivery of Service and Program Key

Tag	Message Type	Key
1	Key Request	SEAK/PEAK or SEK/PEK
2	Key Request Response	SEAK/PEAK or SEK/PEK
3	Key Delivery (same as Key Request Response)	SEAK/PEAK or SEK/PEK
4	Key Delivery Confirmation	SEAK/PEAK or SEK/PEK

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13.1.2.2.1.1. Key Request

This message is sent from the BSD/A to the BSM for the acquisition of SEAK/PEAK or SEK/PEK, which in turn enables BSD/A to generate Short Term Key Messages (STKMs)

Name	Type	Category	Cardinality	Description	Data Type
KeyRequest	E			Key Request Message Contains the following attributes - tag - version - messageID - destination - source - time Contains the following elements - GlobalServiceID - GlobalContentID - ScheduleID - KeyStartTime - KeyEndTime	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	This message ID	string
destination	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. Used if service protection is program based. Only GlobalContent ID which is related to the GlobalService ID is allowed.	anyURI
ScheduleID	E1	O	0..1	Identieifer of the schedule that is protected. Used if service protection is program based. Only Schedule ID which is related to the GlobalService ID is allowed.	anyURI
KeyStartTim	E1	M	1	Provides the start time of the period for which the BSD/A requires a SEAK or SEK and/or PEAK or PEK	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

e				for creating secured STKMs This field expressed as the first 32bits integer part of NTP timestamps.	
KeyEndTime	E1	M	1	Provides the end time of the period for which the BSD/A requires a SEAK or SEK and/or PEAK or PEK for creating secured STKMs. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt

13.1.2.2.1.2. Key Request Response

After the reception of the Key_Request message, the BSM sends this message to the BSD/A for the delivery of SEAK/PEAK or SEK/PEK. In case a SEAK or SEK is used for Service Protection, the use of the SEAK or SEK is bound by its start and end-times. During the lifetime of the Service, the SEAK can be changed periodically. In case a PEAK is used for Service Protection, the PEAK is used throughout the total lifetime of the Program. If both PEAK and SEAK are used in parallel, then the TEK encrypted with the PEK and the PEK encrypted with the SEK SHALL be present in the STKM. When only the PEAK is provided, the STKM should only include the TEK encrypted with the PEK.

Name	Type	Category	Cardinality	Description	Data Type
KeyRequest Response	E			Response to the Key Request message Contains the following attributes: tag version messageID destination source status time secureChannelFlag Contains the following elements: GlobalServiceID GlobalContentID ScheduleID SPPType ServiceKey ProgramKey AccessCriteriaDescriptor ProtectionAfterReceptionFlag	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				TerminalBindingKey	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
message ID	A	M	1	Key Request Message ID	string
destination	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
status	A	M	1	Indication of the reception status of Key Request Message. Global Status codes are used as specified in 0.	unsignedByte
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
secureChannelFlag	A	O	0..1	Indicates whether SAC should be used when transporting TEKs between the SIM and terminal. The absence of this attribute indicates that no SAC is used.	boolean
GlobalService ID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. This field is mandatory if GlobalContent ID was provided in the key request message.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. This field is mandatory if schedule ID was provided in the key request message.	anyURI
SPPType	E1	M	1	This specifies the type of the Service protection profile used by the BSM. 0 if service protection profile == DRM Profile 1 if service protection profile == Smartcard Profile 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
ServiceKey	E1	O	0..N	It specifies the SEAK or SEK Contains the following attribute: keyIdentifier value rand Contains the following elements: ServiceKeyStart	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				ServiceKeyEnd ServiceKeyMTKStart ServiceKeyMTKEnd	
keyIdentifier	A	M	1	Provides the identifier of the SEAK/SEK. The SEAK/SEK identifier is the same as the one provided to the terminal the LTKM and is included with the STKM generated by the BSDA The SEAK/SEK identifiers are as defined for each profile in this specification.	hexBinary
value	A	M	1	This field contains the SEAK if SPP type == 0 This field contains the SEK if SPP type == 1	hexBinary
rand	A	M	1	This field contains the RAND of the LTKM, used for calculating the STKM encryption and authentication keys.	hexBinary
ServiceKeyStart	E2	M	1	Provides the start time of the period in which the SEAK or SEK provided can be used by the BSD/A in creating secured STKMs. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt
ServiceKeyEnd	E2	M	1	Provides the end time of the period in which the SEAK or SEK provided can be used by the BSD/A in creating secured STKMs This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt
ServiceKeyMTKStart	E2	O	0..1	TEK ID start value for SEK/PEK validity This field is mandatory if SDPP type ==1	hexBinary
ServiceKeyMTKEnd	E2	O	0..1	TEK ID end value for SEK/PEK validity This field is mandatory if SPP type ==1	hexBinary
ProgramKey	E1	O	0..1	This field contains the PEAK if SPP_type == 0 and is only applicable to the DRM Profile. This field SHALL NOT be used for the Smartcard Profile. In the Smartcard Profile there is no service key / program key hierarchy available. For the Smartcard Profile the PEK is send using Service Kery fields as described above. Note: Either Service Key, Program Key or both SHALL	hexBinary

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				be included for the DRM Profile. Contains attribute: - keyIdentifier - value	
keyIdentifier	A	M	1	Provides the identifier of the PEAK/PEK. The PEAK/PEK identifier is the same as the one provided to the terminal with the LTKM.	hexBinary
value	A	M	1	This field contains the PEAK if SPP type == 0 This field contains the PEK if SPP type == 1	hexBinary
AccessCriteria Descriptor	E1	O	0..N	The Access Criteria Descriptor to be included in the STKM. Whenever access criteria are defined for a piece of Content, then these access criteria take precendence over the access criteria which where defined for the service to which the content item is related.	hexBinary
ProtectionAfterReceptionFlag	E1	M	1	2 bit field defining the required protection after the removal of the service protection, as specified paragraph 6.3.1	unsignedByte
TerminalBindingKey	E1	O	0..1	An element indicating that a terminal binding key must be used. It contains the following attributes: - keyIdentifier - value	hexBinary
keyIdentifier	A	M	1	Number identifying the Terminal Binding Key ID (TBK ID) that is needed to access the service. This element is only present if the TerminalBindingKey is present.	hexBinary
value	A	M	1	The value of the terminal binding key	hexBinary

13.1.2.2.1.3. Key Delivery Message

This message is sent from the BSM to the BSD/A for the delivery of SEAK/PEAK or SEK/pEk without a request from the BSD/A. If the BSD/A receives this message, then the BSD/A replies to the BSM with Key Delivery Confirmation message. In case a SEK is used for Service Protection, the use of the SEAK or SEK is bound by its start and end-times. During the lifetime of the Service, the SEK can be changed periodically. In case a PEK is used for Service Protection, the PEK is used throughout the total lifetime of the program. If both PEAK and SEAK are used in parallel, then the TEK encrypted with the PEK and the PEK encrypted with the SEK SHALL be present in the STKM. When only the PEAK or PEK is provided, the STKM should only include the TEK encrypted with the PEK.

The message is the same as the Key Request Response message defined above in Section 13.1.2.2.1.2. The root element of the associated XML schema for this message SHALL have the name “KeyDelivery” instead of “KeyRequestResponse”. Status can be set to any value and SHALL be ignored by BSD/A

13.1.2.2.1.4. Key Delivery Confirmation

This message is sent from the BSD/A to the BSM to acknowledge the reception of Key Delivery Message.

Name	Type	Category	Cardinality	Description	Data Type
KeyDeliveryConfirmatio	E			Confirmation to the Key Delivery	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

n				Contains the following attributes: tag version messageID destination source status time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	Key Delivery Message ID	string
destination	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
status	A	M	1	Indication of the reception status of Key Delivery Message. Global Status codes are used as specified in 0.	unsignedByte
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected.. This field is mandatory if GlobalContent ID was provided in the key delivery message	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. This field is mandatory if schedule ID was provided in the key delivery message.	anyURI

13.1.2.3 LTKM and Registration Key Material Delivery

This paragraph describes the delivery of the LTKM or Registration key material are from the BSM to the BSD/A over interface SP-4, for subsequent broadcast distribution. Note that delivery of LTKM or registration key material applies to the DRM Profile only.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13.1.2.3.1 Message Flows

Tags are defined in the following table to identify each message. There are two cases for delivery of LTKM material for broadcast distribution of LTKM's or Registration Key Material for subsequent broadcast distribution.

1. In the first case the BSD/A sends a Key_Request message to the BSM. the BSM then sends a Key_Request_Response message to the BSD/A. The Key Request Response should contain all LTKMs or registration key material that needs to be broadcast in the requested time period.

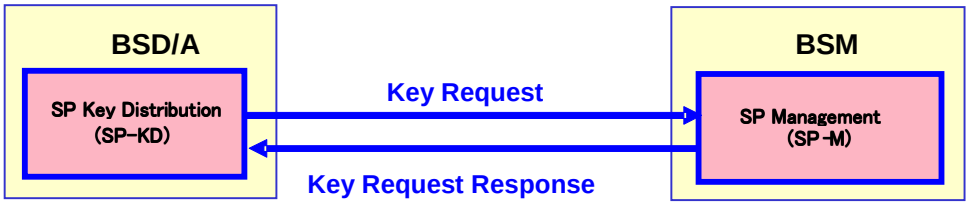


Figure 18 – Message Flow Between BSD/A and BSM for Delivery of LTKM and Registration Key Material

2. In the second case the BSM sends a Key_Delivery message to the BSD/A. The BSD/A then sends a Key_Delivery_Confirmation message to the BSM.

.

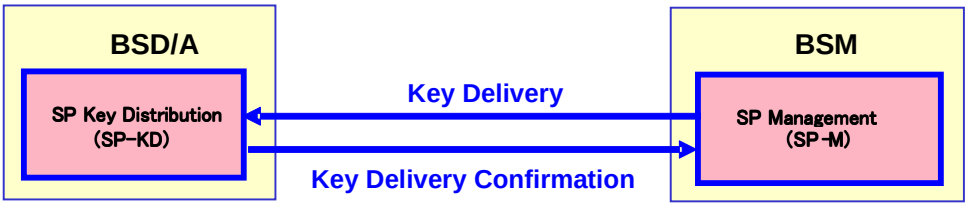


Figure 19 – Alternative Message Flow Between BSD/A and BSM for Delivery of LTKM and Registration Key Material

Tag	Message Type	Key
5	Key_Request	Long-Term Key Message
6	Key_Request_Response	Long-Term Key Message
7	Key_Delivery (same as Key_Request_Response)	Long-Term Key Message
8	Key_Delivery_Confirmation	Long-Term Key Message
9	Key_Request	Key Material for Registration
10	Key_Request_Response	Key Material for Registration
11	Key_Delivery (same as Key Request Response)	Key Material for Registration
12	Key_Delivery_Confirmation	Key Material for Registration

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13.1.2.3.1.1. LTKM or Registration Key Material Request

This message is sent from the BSD/A to the BSM for the request for the delivery of LTKM or Registration Key material for broadcast distribution of LTKM's or Registration Key Material.

Name	Type	Category	Cardinality	Description	Data Type
LTKMorReg Request				Key Request Message Contains the following attributes: tag version messageID destination source time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID DistributionStart DistributionEnd	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	This message ID	string
destination	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
sourceID	A	M	1	BSD/A ID (Note: To be independent of underlying network protocols, Source is included in the message.)	string
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the target service	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. Only GlobalContent ID which is related to the GlobalService ID is allowed.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. Only Schedule ID which is related to the GlobalService ID is allowed.	anyURI
	E1	M	1	This field is mandatory if LTKM or Registration Key material is provided. Provides the start time of the period	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Distribution Start				in which the LTKM or Registration Key material should be distributed by the BSD/A. This field expressed as the first 32bits integer part of NTP timestamps.	
Distribution End	E1	M	1	This field is mandatory if LTKM or Registration Key material is provided. Provides the end time of the period in which the LTKM or Registration Key material should be distributed by the BSD/A. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt

13.1.2.3.1.2. LTKM or Registration Key Material Request Response

After the reception of the Key Request Message, the BSM sends this message to the BSD/A for the delivery of LTKM or Registration Key material for broadcast distribution of LTKM's or Registration Key Material.

Name	Type	Category	Cardinality	Description	Data Type
LTKMorRegRequestResponse				Key Request Response Contains the following attributes: tag version messageID destination source status time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID Data DistributionStart DistributionEnd	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	Key Request Message ID	string

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

destination	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSM ID (Note: To be independent of underlying network protocols, Source is included in the message.)	string
status	A	M	1	Indication of the reception status of Key Request Message. Global Status codes are used as specified in 0.	unsignedByte
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the target service	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. This field is mandatory if GlobalContent ID was provided in the Key request message.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. This field is mandatory if schedule ID was provided in the Key request message.	anyURI
Data	E1	O	0..N	LTKM material for broadcast distribution of LTKM's or Registration Key Material for subsequent broadcast distribution. For LTKM material a single data element carries a single BCRO. • For Registration Key material a single data element carries a single message of either one of the messages below: - device_registration_response(), - update_ri_certificate_msg(), - update_drmtime_msg(), - update_contact_number_msg(), - re_register_msg(), - token_delivery_response(), - domain_registration_response(), - domain_update_response(), - join_domain_msg(), - leave_domain_msg(), as specified in section 7 of [XBS DRM extensions-v1.0]	hexBinary
Distribution Start	E1	O	0..1	This field is mandatory if LTKM or Registration Key material is provided. Provides the start time of the period in which the LTKM or Registration Key material should be distributed by the BSD/A. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt
Distribution End	E1	O	0..1	This field is mandatory if LTKM or Registration Key material is provided. Provides the end time of the period in which the LTKM or Registration Key material should be distributed by the BSD/A.	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				This field expressed as the first 32bits integer part of NTP timestamps.	

13.1.2.3.1.3. LTKM or Registration Key Material Delivery

This message is sent from the BSM to the BSD/A for the delivery of LTKM material for broadcast distribution of LTKM's or Registration Key Material without a request from the BSD/A. If the BSD/A receives this message, then the BSD/A replies to the BSM with Key_Delivery_Confirmation message.

This message is the same as the LTKM or Registration Key Material Request Response message defined above in Section 13.1.2.3.1.2. The root element of the associated XML schema for this message SHALL have the name "LTKMorRegDelivery" instead of "LTKMorRegRequestResponse". Status can be set to any value and SHALL be ignored by BSD/A.

13.1.2.3.1.4. LTKM or Registration Key Material Key Delivery Confirmation

This message is sent from the BSD/A to the BSM to acknowledge the receipt of the LTKM or Registration Key Material Delivery message.

This message is the same as the Key Delivery Confirmation message defined above in Section 13.1.2.2.1.4.

13.1.2.4 STKM Delivery

This paragraph describes the delivery of STKM's from the BSM to the BSD/A or from the BSD/A to the BSM over interface SP-4. The STKM delivered from the BSM to the BSD/A can be sent to Terminal using broadcast channel. The STKM delivered from the BSD/A to the BSM can be sent to Terminal using interaction channel.

13.1.2.4.1 Message Flows from BSM to BSD/A

Tags are defined in the following table to identify a type of each message. There are two cases for delivery of STKM to the BSD/A when STKM generation is done by BSM.

1. The first case consists of the STKM Request message by the BSDA and the Response with the Delivery of the STKM data by the BSM, i.e. BSD/A initiated STKM request.

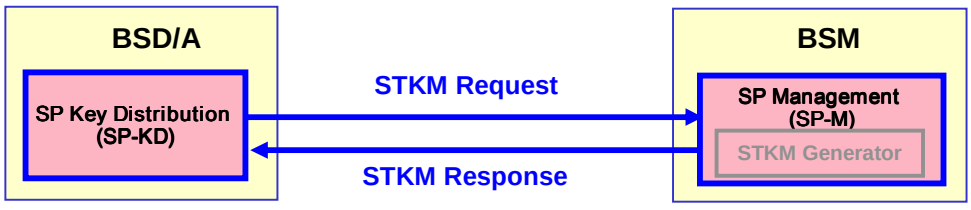


Figure 20 – Message Flow between BSD/A and BSM for Delivery STKMs

2. The second case is BSM initiated. In this case the BSM requests a set of TEK's from the BSDA which it will use during a specific time period to encrypt the service or program. In response, the BSDA delivers the TEKs and the associated security protocol parameters. With this data, the BSM can send an STKM delivery message to the BSD/A. The BSD/A confirms this delivery message.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

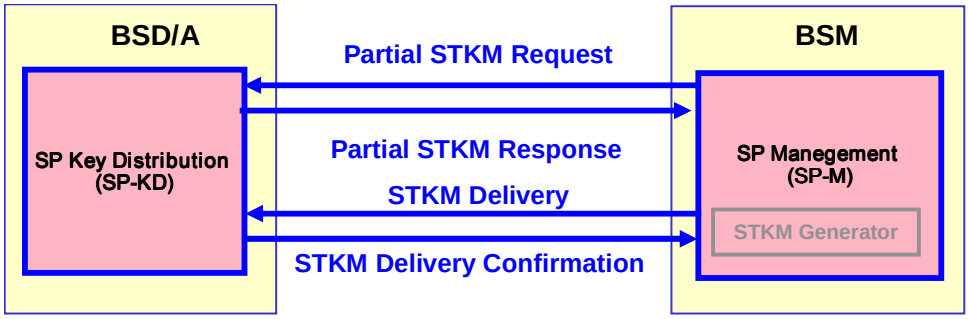


Figure 21 – Alternative Message Flow Between BSD/A and BSM for Delivery STKMs

Services can be shared between multiple BSM's, where each BSM uses it's own SEK and it's own SEK validity time. Furthermore the validity of the SEK is coupled to the TEK ID for anti-replaySo when the TEK ID is wrapped around, the TEK has to be encrypted with the new SEK. . E.g. in the case of protection at the RTSP layer with the Smartcard Profile the amount of TEK ID's are limited to 16 bits. Therefore wrap around time and wrap around indicators are included as attributes of each TEK. These TEK wrap around times cannot be chosen randomly, but should be coordinated with the subscription periods of the BSM, e.g. at the end of each month or week

Tag	Message Type	Key
23	STKM_Request	TEK
24	STKM_Response	STKM
25	Partial_STKM_Request	Partial STKM including TEK
26	Partial_STKM_Response	Partial STKM including TEK
27	STKM_Delivery (same as STKM_Response)	STKM
28	STKM_Delivery_Confirmation	STKM

13.1.2.4.1.1. STKM Request

This message is sent from the BSD/A to the BSM for the acquisition of the Short Term Key Messages.

Name	Type	Category	Cardinality	Description	Data Type
STKMRequest				Request message for STKMs Contains the following attributes tag version messageID destination source time	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				Contains the following elements: GlobalServiceID GlobalContentID ScheduleID SPPType KeyMaterial	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	This message ID	string
destination	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. Used if service protection is program based. Only GlobalContent ID which is related to the GlobalService ID is allowed.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. Only Schedule ID which is related to the GlobalService ID is allowed.	anyURI
SPPType	E1	M	1	This specifies the type of the Service protection profile used by the BSM. 0 if service protection profile == DRM Profile 1 if service protection profile == Smartcard Profile 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
KeyMaterial	E1	M	1..N	The key material used to encrypt the service or program Contains the following attributes: masterKey masterSalt type traffic_authentication_flag	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				validityTime cryptoPeriod wrapAroundTime wraAaroundInidcator Contains the following element TrafficProtectionProtocolParameters NextTrafficKey	
masterKey	A	M	1	The master key used for traffic and content encryption	hexBinary
masterSalt	A	M	1	The master Salt used for traffic and content encryption	hexBinary
type	A	M	1	The traffic protection protocol used. This attribute can have the following values, as specified in the STKM in Section 7.2. Allowed values are: 0 if traffic_protection_protocol == TKM_ALGO_IPSEC 1 if traffic_protection_protocol == TKM_ALGO_SRTP 2 if traffic_protection_protocol == TKM_ALGO_AUENCRYP 3 if traffic_protection_protocol == TKM_ALGO_DCF 4-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
traffic_authentication_flag	A	M	1	True if the traffic_authentication_flag in the STKM should be set to TKM_FLAG_TRUE (authentication will be used). False otherwise.	boolean
validityTime	A	M	1	NTP time when the traffic encryption key is used to encrypt the service or program. This value indicates to the BSM which Service Key it needs to use to encrypt the traffic encryption key. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt
cryptoPeriod	A	M	1	The crypto period used for service protection. The Validity-Time of the next TEK SHOULD be 1 crypto period later than the Validity Time of this TEK.	unsignedInt
wrapAroundTime	A	M	1	This indicates the wrap around time of the TEK sequence to which this TEK belongs. After the TEK wrap around time the TEK key indicator, master key index or security parameter index is reset. This field is used to indicate that the first TEK after the Wrap around time SHOULD be encrypted with a new SEK. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

wrapAround Iindicator	A	M	1	This field is set to "true" for the first TEK after the Wrap around time has passed. It is used to indicate that this and subsequent TEKs SHOULD be encrypted with a new SEK.	boolean
TrafficProte ctionProtoco lParameters	E2	M	1	This specifies the data related to the traffic protection protocol for the STKM, as defined in the STKM in Section 5.5. Contains the following elements: SPI MKI KeyIndicator KeyIdentifier Note the following: • SPI is mandatory in case 'type' of 'KeyMaterial' is 0 • MKI is mandatory in case 'type' of 'KeyMaterial' is 1 • KeyIndicator is mandatory in case 'type' of 'KeyMaterial' is 2 • KeyIdentifier is mandatory in case 'type' of 'KeyMaterial' is 3 • SynchronisationSource is mandatory in case 'type' of 'KeyMaterial' is 1 and is optional in other cases This constraint is expressed by using the <choice> element in XML Schema	
SPI	E3	O	0..1	Security Parameter Index. Contains the following attributes: spi nextSpi	
spi	A	M	1	security_parameter_index	unsignedInt
nextSpi	A	M	1	next_security_parameter_index	unsignedInt
MKI	E3	O	0..1	Master Key Index Contains the following attributes: mkilength mki mediaFlows	
mkiLength	A	M	1	master_key_index_length	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

mki	A	M	1	master_key_index	unsignedInt
mediaFlows	A	M	1	number_of_media_flows	int
KeyIndicator	E3	O	0..1	Key Indicator Contains the following attributes: keyIndicatorLength keyIndicator	
keyIndicator Length	A	M	1	key_indicator_length This field is mandatory in case 'Type' of 'Key material' is 2	unsignedInt
keyIndicator	A	M	1	key_indicator	unsignedInt
KeyIdentifier	E3	O	0..1	Key Identifier Contains the following attributes: keyIdentifierLength keyIdentifier	
keyIdentifier Length	A	M	1	key_identifier_length	unsignedInt
keyIdentifier	A	M	1	The key identifier	hexBinary
NextTraffic Key	E2	O	0..1	Flag for indication of the next traffic key in an STKM. If true, the STKM SHALL also include the next encrypted traffic key. Note that this field is only relevant to DRM profile STKMs. Next traffic key has the following attributes, only if Next traffic key is set to "true": masterKey masterSalt	boolean
masterKey	A	O	0..1	The next master key used to encrypt the service or program. This field is mandatory if next traffic key is "true"	hexBinary
masterSalt	A	O	0..1	The next Master Salt used to encrypt the service or program. This field is mandatory if next traffic key is "true"	hexBinary

13.1.2.4.1.2. STKM Response

After the reception of the STKM Request message, the BSM sends this message to the BSD/A for the delivery of STKM.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Name	Type	Category	Cardinality	Description	Data Type
STKMResponse	E			This message is the response to the STKM Request message. Contains the following attributes tag version messageID destination source status time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID STKM	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	Key Request Message ID	string
destination	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
status	A	M	1	Indication of the reception status of STKM Request Message. Global Status codes are used as specified in 0.	unsignedByte
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. This field is mandatory if GlobalContent ID was provided in the STKM request message.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. This field is mandatory if schedule ID was provided in the STKM request message.	anyURI
STKM	E1	M	1..N	The STKM	hexBinary

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				STKM has the following attribute: - validityTime	
validityTime	A	M	1	This validityTime attribute is used to associate the stkm with the TEK. The validityTime of the STKM SHALL be the same as the validityTime of the TEK to which this stkm is associated. This field is expressed as the first 32bits integer part of NTP timestamps.	unsignedInt

13.1.2.4.1.3. Partial STKM Request Message

The Partial STKM Request message is used by the BSM to request a set of TEK's from the BSD/A to deliver a set of TEK's to be used for the encryption of the service or program. The set of TEK's to be delivered are indicated by a start time and an end time in the request message.

Name	Type	Category	Cardinality	Description	Data Type
PartialSTKMRequest				Partial STKM Request Message Contains the following attributes tag version messageID destination source time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID TEKStartTime TEKEndTime	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	This message ID	string
destination	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. Used if service protection is program based. Only GlobalContent ID which is related to the GlobalService ID is allowed.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. Only Schedule ID which are related to the GlobalService ID is allowed.	anyURI
TEKStartTime	E1	M	1	This is the start time of the TEKs that are used for the encryption of the service or program. This field expressed as the first 32bits integer part of NTP timestamps.	unsignedInt
TEKEndTime	E1	M	1	This is the end time of the TEK that are used for the encryption of the service or program. This field expressed as the first 32bits integer part of NTP timestamps..	unsignedInt

13.1.2.4.1.4. Partial STKM Response Message

The Partial STKM Response message is used by the BSD/A to deliver the TEK's and the associated traffic protection protocol parameters to the BSM.

Name	Type	Category	Cardinality	Description	Data Type
PartialSTKMResponse	E			Partial STKM Response Message Contains the following attributes tag version messageID destination source status time Contains the following elements: GlobalServiceID GlobalContentID ScheduleID	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				SPPType KeyMaterial	
tag	A	M	1	Identifier for the message type	unsignedByte
version	A	O	0..1	BCAST enabler version supported by this message	unsignedInt
messageID	A	M	1	This message ID	string
destination	A	M	1	BSM ID (Note: To be independent of the underlying network protocols, Destination is included in the message.)	string
source	A	M	1	BSD/A ID (Note: To be independent of the underlying network protocols, Source is included in the message.)	string
status	A	M	1	Indication of the reception status of TEK Request Message. Global Status codes are used as specified in 0.	unsignedByte
time	A	O	0..1	The time when this message is sent. This field contains the 32bits integer part of an NTP time stamp.	unsignedInt
GlobalServiceID	E1	M	1	Identifier of the service to be encrypted	anyURI
GlobalContentID	E1	O	0..1	Identifier of the content that is protected. Used if service protection is program based. This field is mandatory if Global Content ID was provided in the TEK request message.	anyURI
ScheduleID	E1	O	0..1	Identifier of the schedule that is protected. This field is mandatory if schedule ID was provided in the TEK request message.	anyURI
SPPType	E1	M	1	This specifies the type of the Service protection profile used by the BSM. 0 if service protection profile == DRM Profile 1 if service protection profile == Smartcard Profile 2-127 reserved for future use 128-255 reserved for proprietary use	unsignedByte
KeyMaterial	E1	M	1..N	The key material used to encrypt the service or program KeyMaterial has the following attributes: - masterKey - masterSalt - type - traffic_authentication_flag -validityTime - cryptoPeriod - wrapAroundTime	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				- wrapAroundInidcator KeyMaterial contains the following elements - TrafficProtectionProtocolParameters - NextTrafficKey	
masterKey	A	M	1	The master key used for traffic and content encryption	hexBinary
masterSalt	A	M	1	The master Salt used for traffic and content encryption	hexBinary
type	A	M	1	The traffic protection protocol used. This attribute can have the following values, as specified in the STKM in Section 7.2: - 0 if traffic_protection_protocol == TKM_ALGO_IPSEC - 1 if traffic_protection_protocol == TKM_ALGO_SRTP - 2 if traffic_protection_protocol == TKM_ALGO_AUENCRYP - 3 if traffic_protection_protocol == TKM_ALGO_DCF - 4-127 reserved for future use - 128-255 reserved for proprietary use	unsignedByte
traffic_authentication_flag	A	M	1	True if the traffic_authentication_flag in the STKM should be set to TKM_FLAG_TRUE (authentication will be used). False otherwise.	boolean
validityTime	A	M	1	NTP time when the traffic encryption key is used to encrypt the service or program. This value indicates to the BSM which Service Key it needs to use to encrypt the traffic encryption key. This field expressed as the first 32bits integer part of NTP timestamps. The NTP value SHALL be bound by the start and end-times as indicated in the TEK request message.	unsignedInt
cryptoPeriod	A	M	1	The crypto period used for service protection. The Validity-Time of the next TEK SHOULD be 1 crypto period later than the Validity Time of this TEK.	unsignedInt
wrapAroundTime	A	M	1	This indicates the wrap around time of the TEK sequence to which this TEK belongs. After the TEK wrap around time the TEK key indicator, master key index or security parameter index is reset. This field is used to indicate that the first TEK after the Wrap around time SHOULD be encrypted with a new SEK. This field expressed as the first 32bits integer part of	unsignedInt

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

				NTP timestamps.	
wrapAround Indicator	A	M	1	This field is set to "true" for the first TEK after the Wrap around time has passed. It is used to indicate that this and subsequent TEKs SHOULD be encrypted with a new SEK.	boolean
TrafficProtectionProtocolParameters	E2	M	1	This specifies the data related to the traffic protection protocol for the STKM, as defined in the STKM in Section 5.5. Contains the following elements: SPI MKI KeyIndicator KeyIdentifier Note the following: • SPI is mandatory in case 'type' of 'KeyMaterial' is 0 • MKI is mandatory in case 'type' of 'KeyMaterial' is 1 • KeyIndicator is mandatory in case 'type' of 'KeyMaterial' is 2 • KeyIdentifier is mandatory in case 'type' of 'KeyMaterial' is 3	
SPI	E3	O	0..1	Security Parameter Index. Contains the following attributes: spi nextSpi	
spi	A	M	1	security_parameter_index	unsignedInt
nextSpi	A	M	1	next_security_parameter_index	unsignedInt
MKI	E3	O	0..1	Master Key Index Contains the following attributes: mkilength mki mediaFlows	
mkiLength	A	M	1	master_key_index_length	
mki	A	M	1	master_key_index	unsignedInt
mediaFlows	A	M	1	number_of_media_flows	int

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

KeyIndicator	E3	O	0..1	Key Indicator Contains the following attributes: keyIndicatorLength keyIndicator	
keyIndicator Length	A	M	1	key_indicator_length	unsignedInt
keyIndicator	A	M	1	key_indicator	unsignedInt
KeyIdentifier	E3	O	0..1	Key Identifier Contains the following attributes: keyIdentifierLength keyIdentifier	
keyIdentifier Length	A	M	1	key_identifier_length	unsignedInt
keyIdentifier	A	M	1	key_identifier	hexBinary
NextTraffic Key	E2	O	0..1	Flag for indication of the next traffic key in an STKM. If "true", the STKM SHALL also include the next encrypted traffic key. Note that this field is only relevant to DRM profile STKMs. Next traffic key has the following attribute, only if Next traffic key is set to "true": - masterKey - masterSalt	boolean
nextTrafficKey	A	O	0..1	The next traffic encryption key used to encrypt the service or program. This field is mandatory if next traffic key is "TRUE"	hexBinary
masterKey	A	O	0..1	The next master key used to encrypt the service or program. This field is mandatory if NextTrafficKey is "true"	hexBinary
masterSalt	A	O	0..1	The next Master Salt used to encrypt the service or program. This field is mandatory if NextTrafficKey is "true"	hexBinary

13.1.2.4.1.5. STKM Delivery

This message is used by the BSM to deliver the STKM to the BSDA.

This message is the same as the STKM Response message defined above in Section 13.1.2.4.1.2. The root element of the associated XML schema for this message SHALL have the name "STKMDelivery" instead of "STKMResponse". Status can be set to any value and SHALL be ignored by BSD/A.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

13.1.2.4.1.6. STKM Delivery Confirmation

This message is used by the BSD/A to confirm the reception of the STKM delivery message.

This message is the same as the Key Delivery Confirmation message defined above in Section 13.1.2.2.1.4. The root element of the associated XML schema for this message SHALL have the name "STKMDeliveryConfirmation" instead of "KeyDeliveryConfirmation".

13.1.2.4.2 Message Flows from BSD/A to BSM

Tags are defined in the following table to identify a type of each message. There are two cases for delivery of STKM to the BSD/A when STKM generation is done by the BSD/A.

1. The first case consists of the STKM Request message by the BSM and the Response with the Delivery of the STKM data by the BSD/A, i.e. BSM initiated STKM request.

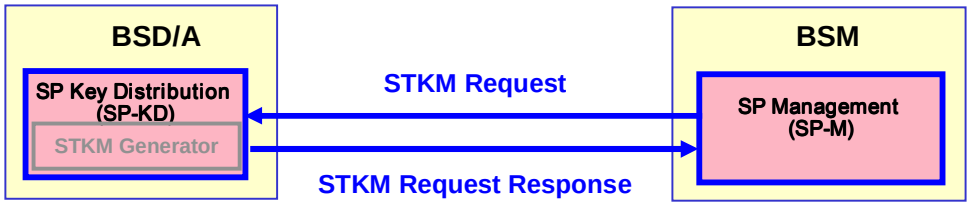


Figure 22 – Message Flow between BSM and BSD/A for Delivery STKMs

2. The second case is BSD/A initiated. In this case the BSD/A sends an STKM delivery message to the BSM. The BSM confirms this delivery message.

Figure 23 – Alternative Message Flow between BSM and BSD/A for Delivery STKMs

Tag	Message Type	Key
29	STKM_Request	STKM
30	STKM_Response	STKM
31	STKM_Delivery (same as STKM_Response)	STKM
32	STKM_Delivery_Confirmation	STKM

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

13.1.2.4.2.1. STKM Request

This message is send from the BSM to the BSD/A for the acquisition of the STKM. This message is the same as the Key Request message defined in Section 13.1.2.3.1.1. The root element of the associated XML schema for this message SHALL have the name "STKMRequest" instead of "KeyRequest".

13.1.2.4.2.2. STKM Request Response

This message is used by the BSD/A to delivery the STKM to the BSM. This message is the same as the STKM Response message defined above in Section 13.1.2.4.1.2.

13.1.2.4.2.3. STKM Delivery

This message is used by the BSD/A to deliver the STKM to the BSM.

This message is the same as the STKM Response message defined above in Section 13.1.2.4.1.2. The root element of the associated XML schema for this message SHALL have the name "STKMDelivery" instead of "STKMResponse". Status can be set to any value and SHALL be ignored by BSD/A.

13.1.2.4.2.4. STKM Delivery Confirmation

This message is used by the BSM to confirm the reception of the STKM delivery message.

This message is the same as the Key Delivery Confirmation message defined above in Section 13.1.2.2.1.4. The root element of the associated XML schema for this message SHALL have the name "STKMDeliveryConfirmation" instead of "KeyDeliveryConfirmation".

13.2 Interface CP-4

The interface CP-4 can be used for up to three different functions:

1) Delivery of the Service and Program key material from the SP-M in the BSM to the SP-KD in the BSD/A for content protection.

2) Delivery of the LTKM or Registration key material from the SP-M in the BSM to SP-KD in the BSD/A, for subsequent broadcast distribution of these data..

3) Delivery of the STKMs from the BSM to the BSD/A for subsequent broadcast distribution.

A BSM that supports Content Protection SHALL support the interface CP-4. A BSD/A that supports Content Protection SHALL support the interface CP-4.

The message flows for the interface CP-4 are the same as the message flows for SP-4. Therefore the same protocol stack and messages as described for SP-4 SHALL be used for CP-4. This means that two options are given for the interface CP-4:

- Using DVB Simulcrypt based interfaces
- Using BCAST specific interfaces

The interface CP-4 MAY support DVB Simulcrypt as specified in Section 13.1.1. The interface CP-4 MAY support OMA BCAST specific signalling as specified in Section 13.1.2. The interface CP-4 SHALL support either DVB Simulcrypt as specified in Section 13.1.1 or OMA BCAST specific signalling as specified in Section 13.1.2.

The technical difference between SP-4 and CP-4 is the value of protection_after_reception. In the case of Content Protection, the BSM sets protection_after_reception to value 0x00, 0x01 or 0x02 as defined in Section 7.3. Setting protection_after_reception to 0x03 is possible only in the case of Service Protection. This is valid for both streams and files.

Files can be DCF-protected. DCF files can be offered by the BSA or the Content Creator. In this case the file is transported as any other file to the terminal. In case Content Protection is required for a file, the file has to be protected using DCF by the BSD/A. Therefore, a BSD/A that supports Content Protection SHALL support DCF protection of files. Note, that DCF

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

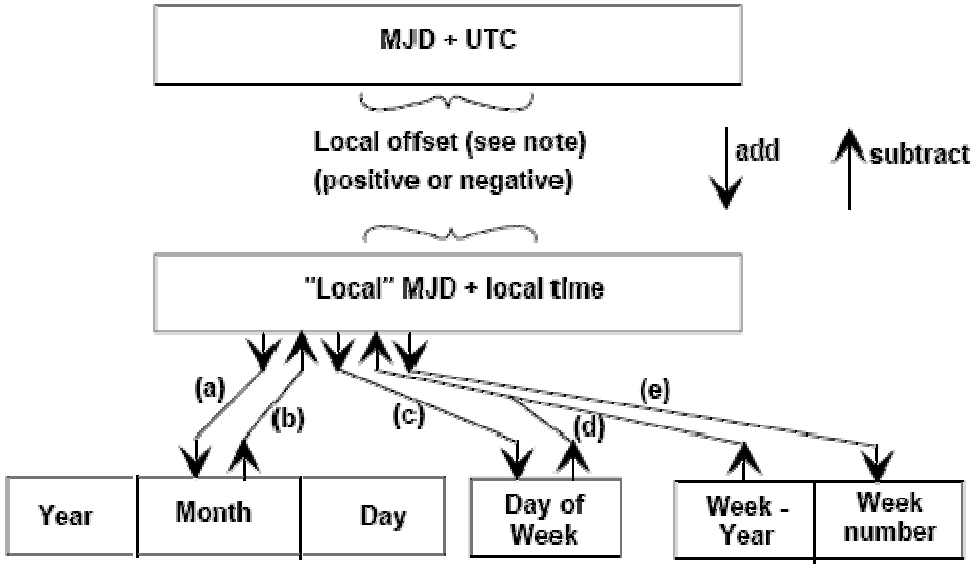
files, which were protected by the BSA or the Content Creator can be additionally encapsulated in another DCF for additional Content Protection, i.e. DCF inside a DCF. The usage rights of the ‘outer’ DCF file is dictated by the protection_after_reception. The usage rights of the ‘inner’ DCF is dictated by the Rights Objects that the device has for the file, that were provided by the BSA or the Content Creator.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

14. Conversion between Time and Date Conventions

(Note: this text is modified from [ETSI EN 300 468 V1.6.1] for BCAST purposes. In particular, the version in this text maintains byte alignment everywhere.)

The types of conversion which may be required are summarized in Figure 24.



NOTE: Offsets are positive for Longitudes East of Greenwich and negative for Longitudes West of Greenwich.

Figure 24 – Conversion routes between Modified Julian Date (MJD) and Co-ordinated Universal Time (UTC)

The conversion between MJD + UTC and the ”local” MJD + local time is simply a matter of adding or subtracting the local offset. This process may, of course, involve a “carry” or “borrow” from the UTC affecting the MJD. The other five conversion routes shown on the diagram are detailed in the formulas below:

Symbols used:

- D: Day of month from 1 to 31
- Int: Integer part, ignoring remainder
- K, L ,'', W, '': Intermediate variables
- M: Month from January (= 1) to December (= 12)
- MJD: Modified Julian Date
- MN: Week number according to ISO 2015 [21]
- mod 7: Remainder (0-6) after dividing integer by 7
- UTC: Universal Time, Co-ordinated
- WD: Day of week from Monday (= 1) to Sunday (= 7)
- WY: “Week number” Year from 1900
- X: Multiplication

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- Y: Year from 1900 (e.g. for 2003, Y = 103)

To find Y, M, D from MJD

- ’' = int [(MJD–- 15 078,2) / 365,25]
- ’' = int { [MJD–- 14 956,1–- int (’' × 365,25)] / 30,6001 }
- D = MJD–- 14 956–- int (’' × 365,25)–- int (’' × 30,6001)
- If ’' = 14 or ’' = 15, then K = 1; else K = 0
- Y = ’' + K
- M = ’'–- 1–- K × 12

To find MJD from Y, M, D

- If M = 1 or M = 2, then L = 1; else L = 0
- MJD = 14 956 + D + int [(Y–- L) × 365,25] + int [(M + 1 + L × 12) × 30,6001]

To find WD from MJD

- WD = [(MJD + 2) mod 7] + 1

To find MJD from WY, WN, WD

- MJD = 15 012 + WD + 7 × { WN + int [(WY × 1 461 / 28) + 0,41] }

To find WY, WN from MJD

- W = int [(MJD / 7)–- 2 144,64]
- WY = int [(W × 28 / 1 461)–- 0,0079]
- WN = W–- int [(WY × 1 461 / 28) + 0,41]
- EXAMPLE: MJD = 45 218 W = 4 315
- Y = (19)82 WY = (19)82
- M = 9 (September) N = 36
- D = 6 WD = 1 (Monday)

NOTE: These formulas are applicable between the inclusive dates 1900 March 1 to 2100 February 28.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

14.1 Local Time Offset

This 16-bit field contains the current offset time from UTC in the range between –12 hours and +13 hours at the area which is indicated by the combination of country_code and country_region_id in advance. These 16 bits are coded as 4 digits in 4-bit BCD in the order hour tens, hour, minute tens, and minutes.

The positive or negative offset from the UTC is indicated with the 1 bit local_time_offset_polarity. If this bit is set to “0” the polarity is positive and the local time is advanced to UTC. (Usually east direction from Greenwich). If this bit is set to “1” the polarity is negative and the local time is behind UTC. Please note that the local_time_offset_polarity is represented by the first bit of the first nibble representing the hour tens field. The first nibble of the local_time_offset is therefore encoded as follows:

Table 65: Local Time Offset Coding

local_time_offset_polarity	offset hour tens	first nibble
0 (i.e. “+”)	0	0000
0 (i.e. “+”)	1	0001
1 (i.e. “-”)	0	1000
1 (i.e. “-”)	1	1001

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

15. Interfacing to underlying BDSes

15.1 BCMCS

Interfacing to underlying BCMCS BDS SHALL be as specified in the BCMCS adaptation specifications [BCAST10-BCMCS-Adaptation].

15.2 MBMS

Interfacing to underlying MBMS BDS SHALL be as specified in the MBMS adaptation specifications [BCAST10-MBMS-Adaptation].

15.3 IPDC over DVB-H

Interfacing to underlying DVB BDS SHALL be as specified in the DVB adaptation specifications [BCAST10-DVBH-IPDC-Adaptation].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

16. Broadcast Roaming – Roaming at Service Provider Level (Informative)

Section 5.7 “Broadcast Roaming” in [BCAST10-Services] describes roaming across different Service Providers. BCAST provides messages that allow the Terminal to acquire relevant roaming rules, allowing it to discover available services based on service provider roaming agreements.

This chapter provides additional information relating to Service Protection and Content Protection aspects that must also be taken into account when considering broadcast roaming (across different service providers) when content / services are also protected.

16.1 Broadcast Roaming –DRM Profile

A terminal can have access to content / services protected using the DRM Profile provided:

1. BCAST service provisioning or roaming message exchange with the visited Service Provider is successful (see [BCAST10-Services]).
2. Registration with the Rights Issuer has been completed (see Section 5.3). This means the Rights Issuer has to authorise delivery of GROs the terminal, i.e. the Terminal and Rights Issuer must mutually accept each othe’’s certificates.

Note that Step 1 can require the appropriate broadcast/ service roaming agreements to be in place between home and visited Service Provider.

16.2 Broadcast Roaming – Smartcard Profile

A terminal can have access to content / services protected using the Smartcard Profile provided:

1. BCAST service provisioning and/ or roaming message exchange with the visited Service Provider is successful (see [BCAST10-Services]).
2. Subscriber Key Establishment has been completed (see Section 6.5), allowing SMK/SRK to be derived. For the (U)SIM Smartcard Profile this means the visited network BSM must be able to obtain the SMK/SRK from the home network BSM with which GBA bootstrapping is accomplished. If no agreement exists between home and visited BSM, LTKMs will not be able to be delivered to the Smartcard / Terminal, preventing access to protected / content available through the visited BSM. In the case of the (R-)UIM/CSIM Smartcard Profile, the Visited BSM relies on the Home BSM to perform authentication of the subscriber. Upon successful authentication, the Home BSM provides the SMK (i.e. the TK) to the Visited BSM. Subsequently, LTKMs can be delivered from the Visited BSM to the BCAST Terminal, assuming the existence of roaming agreement between the Visited and Home Service Providers.

Note that Step 1 can require the appropriate broadcast/ service roaming agreements to be in place between home and visited Service Provider.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-TS-BCAST_SvcCntProtection-V1_0	12 Feb 2009	Approved by TP TP ref# OMA-TP-2009-0071-INP_BCAST_V1_0_ERP_for_Notification_and_Final_Approval
OMA-TS-BCAST_SvcCntProtection-V1_0_1	09 Jan 2013	Incorporated agreed Class 3 CRs: "OMA-BCAST-2009-0124R01-CR_SecureChannelFlag_SP4", " OMA-BCAST-2010-0028-CR_Move_section_6.7.3.1", "OMA-BCAST-2010-0027-CR_event_signalling_description" and agreed Class 2 CR: OMA-BCAST-2010-0072R01-CR_Change_references_from_3GPP_to_ETSI Notified to TP see TP ref# OMA-TP-2013-0001-INP_BCAST_V1_0_1_ERP_for_notification

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [IOPPROC].

Note: BCAST adaptation specifications, such as [BCAST10-BCMCS-Adaptation], [BCAST10-DVBH-IPDC-Adaptation], and [BCAST10-MBMS-Adaptation], in which it is specified how the BCAST 1.0 enabler is implemented over a specific BDS (Broadcast Distribution System), may overrule or adapt requirements from this SCR or provide additional requirements.

B.1 SCR for Clients

Item	Function	Reference	Status	Requirement
BCAST-SPCP-C-001	Support for Service Protection	4.1	O	
BCAST-SPCP-C-002	Support for Content Protection	4.1	O	
BCAST-TerminalCapability-C-003	Terminal with cellular radio interface and with Smartcard supporting Service Protection	4.1	O	BCAST-SPCP-C-005
BCAST-TerminalCapability-C-004	Terminal without cellular radio interface or without Smartcard supporting Service Protection	4.1	O	BCAST-SPCP-C-006
BCAST-SPCP-C-005	Support for Smartcard Profile for Service Protection	4.1	O	BCAST-ContentLayer-C-007 AND BCAST-STKM_SC-C-010 AND BCAST-SDP-C-014 AND BCAST-LTKM_SC-C-015 AND BCAST-KeyManagement-C-016 AND BCAST-SC_ParentalControl-C-033 AND BCAST-SC_LocationRestriction-C-034
BCAST-SPCP-C-006	Support for DRM Profile for Service Protection	4.1	O	BCAST-ContentLayer-C-007 AND BCAST-STKM_DRM-C-011 AND BCAST-LTKM_DRM-C-013 AND BCAST-SDP-C-014
BCAST-ContentLayer-C-007	Processing of Content Layer encryption - SRTP	9.2	O	BCAST-SRTPsignal-C-030
BCAST-ContentLayer-C-008	Processing of Content Layer encryption - IPsec	9.1	O	
BCAST-ContentLayer-C-009	Processing of Content Layer encryption - ISMACRYP	9.3	O	
BCAST-STKM_SC-C-010	STKM for Smartcard Profile	6.7	O	BCAST-SCCommands-C-035
BCAST-STKM_DRM-C-011	STKM for DRM Profile	5.5	O	
BCAST-SC_Binding-C-012	Binding of STKM to Terminal with Smartcard	12	O	
BCAST-LTKM_DRM-C-013	LTKM for DRM Profile	5.4	O	
BCAST-SDP-C-014	Protection Signalling in SDP	10.1	O	
BCAST-LTKM_SC-C-015	LTKM for Smartcard Profile	6.6	O	BCAST-SCCommands-C-036
BCAST-KeyManagement-C-016	Service Access for Terminal with Smartcard	6.10	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-Recording-C-017	Recording	8	O	
BCAST-CPFilesDRM-C-018	Content Protection for Downloading Files in Terminals with no Smartcard	5.6.2.2	O	
BCAST-CP_RTP_DRM-C-019	Content Protection for Streaming in Terminals with no Smartcard	5.6.1.2	O	BCAST-CP_Form-C-023
BCAST-CPFilesSC-C-020	Content Protection for Downloading Files in Terminals with Smartcard	6.8.2.2	O	BCAST-ClientID-C-027
BCAST-CP_RTP_SC-C-021	Content Protection for Streaming in Terminals with Smartcard	6.8.1	O	BCAST-CP_Form-C-023 AND BCAST-ClientID-C-027 AND BCAST-SAC-C-028
BCAST-CP_Recorded_SC-C-022	Content Protection of Recorded Material in Terminals with Smartcard	6.9.1, 8.5	O	BCAST-ClientID-C-027 AND BCAST-SC_Binding-C-012 AND BCAST-SCCommands-C-035 AND BCAST-SCCommands-C-040
BCAST-CP_Form-C-023	Content Protection Format	9.3.3	O	
BCAST-Domains-C-024	Broadcast Domains	5.3	O	
BCAST-MeteringDRM-C-025	Usage Metering for DRM Profile	5.9	O	
BCAST-MeteringSC-C-026	Usage Metering for Smartcard Profile	6.6.5	O	
BCAST-ClientID-C-027	BCAST Client ID	6.11	O	
BCAST-SAC-C-028	Use of SAC	6.8.1.3	O	
BCAST-ContentLayer-C-029	Processing of Content Layer authentication - ISMACRYP	9.3	O	BCAST-SRTPsignal-C-030
BCAST-SRTPsignal-C-030	SDP Signalling of SRTP	10.4	O	
BCAST-Recording_SC-C-031	Recording in Terminals with Smartcard for Smartcard profile	8.5	O	BCAST-SCCommands-C-040
BCAST-SC_ParentalControl-C-032	Support of Parental Control Message in Smartcard profile	6.6.5	O	BCAST-SCCommands-C-036
BCAST-SC_ParentalControl-C-033	Support of Smartcard-based parental control	6.7.3.11.1	O	BCAST-SCCommands-C-035 AND BCAST-SCCommands-C-042
BCAST-SC_LocationRestriction-C-034	Support of Smartcard-based location restriction	6.7.3.11.2	O	BCAST-SCCommands-C-035 AND BCAST-SCCommands-C-044 AND BCAST-SCCommands-C-045
BCAST-SCCommands-C-035	Support of MTK generation mode AUTHENTICATE Command for OMA BCAST operation	E.2.1	O	
BCAST-SCCommands-C-036	Support of MSK update mode AUTHENTICATE Command for OMA BCAST operation	E.2.2	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-SCCommands-C-037	Support of AUTHENTICATE Command for OMA BCAST operation: SPE Deletion Mode	E.2.3.1	O	
BCAST-SCCommands-C-038	Support of AUTHENTICATE Command for OMA BCAST operation: Recording Deletion Mode	E.2.3.2	O	
BCAST-SCCommands-C-039	Support of OMA BCAST Command: SPE Audit Mode	E.3.2	O	
BCAST-SCCommands-C-040	Support of OMA BCAST Command: Record Signalling Mode	E.3.3	O	
BCAST-SCCommands-C-041	Support of OMA BCAST Command: Recording Audit Mode	E.3.4	O	
BCAST-SCCommands-C-042	Support of VERIFY PIN Command	6.7.3.11.1	O	
BCAST-SCCommands-C-043	Support of UNBLOCK PIN Command	6.7.3.11.1	O	
BCAST-SCCommands-C-044	Support of DISPLAY TEXT Command	6.7.3.11.2	O	
BCAST-SCCommands-C-045	Support of PROVIDE LOCAL INFORMATION Command	6.7.3.11.2	O	

B.2 SCR for BSD/A

Item	Function	Reference	Status	Requirement
BCAST-BSDASPCP-S-001	Support Backend interface for Service Protection	13.1	O	BCAST-BSDASPCP-S-002 OR BCAST-BSDASPCP-S-003
BCAST-BSDASPCP-S-002	Support BCAST specific interface for SP-4	13.1.2	O	
BCAST-BSDASPCP-S-003	Support SP-4 by the adaptation of DVB Simulcrypt Head-end interfaces	13.1.1	O	
BCAST-BSDASPCP-S-004	Support Backend interface for Content Protection	13.2	O	BCAST-BSDASPCP-S-005
BCAST-BSDASPCP-S-005	Support CP-4 by BCAST specific interface	13.2	O	
BCAST-BSDASPCP-S-006	Support Service Protection	4	O	BCAST-BSDASPCP-S-007 OR BCAST-BSDASPCP-S-012
BCAST-BSDASPCP-S-007	Support DRM profile for Service Protection	5	O	BCAST-BSDASPCP-S-008 AND BCAST-BSDASPCP-S-009 AND BCAST-BSDASPCP-S-010 AND

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
				BCAST-BSDASPCP-S-011
BCAST-BSDASPCP-S-008	Support delivery of STKM for DRM profile	5.5	O	
BCAST-BSDASPCP-S-009	Support the encryption for Service Protection of Stream for DRM Profile	5.6.1.1	O	BCAST-BSDASPCP-S-028 OR (BCAST-BSDASPCP-S-029 AND BCAST-BSDASPCP-S-038) OR (BCAST-BSDASPCP-S-030 AND BCAST-BSDASPCP-S-037) OR BCAST-BSDASPCP-S-031
BCAST-BSDASPCP-S-010	Support the encryption for Service Protection of File for DRM Profile	5.6.2.1	O	BCAST-BSDASPCP-S-028 OR BCAST-BSDASPCP-S-033
BCAST-BSDASPCP-S-011	Support SDP signalling for Service Protection of DRM Profile	5.8	O	BCAST-BSDASPCP-S-034
BCAST-BSDASPCP-S-012	Support Smartcard Profile for Service protection	6	O	BCAST-BSDASPCP-S-013 AND BCAST-BSDASPCP-S-014 AND BCAST-BSDASPCP-S-015 AND BCAST-BSDASPCP-S-016
BCAST-BSDASPCP-S-013	Support delivery of STKM for Smartcard profile	6.7.3	O	
BCAST-BSDASPCP-S-014	Support the encryption for Service Protection of Stream for Smartcard Profile	6.7.4	O	BCAST-BSDASPCP-S-028 OR (BCAST-BSDASPCP-S-029 AND BCAST-BSDASPCP-S-038) OR (BCAST-BSDASPCP-S-030 AND BCAST-BSDASPCP-S-037)
BCAST-BSDASPCP-S-015	Support the encryption for Service Protection of File for Smartcard Profile	6.8.2.1	O	BCAST-BSDASPCP-S-028 OR BCAST-BSDASPCP-S-033
BCAST-BSDASPCP-S-016	Support SDP signalling for Service Protection of Smartcard Profile	6.10.1.2	O	BCAST-BSDASPCP-S-034
BCAST-BSDASPCP-S-017	Support Content Protection	4	O	BCAST-BSDASPCP-S-018 OR BCAST-BSDASPCP-S-023
BCAST-BSDASPCP-S-018	Support DRM profile for content protection	5	O	BCAST-BSDASPCP-S-019 AND BCAST-BSDASPCP-S-020 AND BCAST-BSDASPCP-S-021 AND BCAST-BSDASPCP-S-022
BCAST-BSDASPCP-S-019	Support delivery of STKM for DRM profile	5.5	O	
BCAST-BSDASPCP-S-020	Support the encryption for Content Protection of Stream for DRM Profile	5.6.1.2	O	BCAST-BSDASPCP-S-028 OR (BCAST-BSDASPCP-S-029 AND BCAST-BSDASPCP-S-038) OR (BCAST-BSDASPCP-S-030 AND BCAST-BSDASPCP-S-037)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-BSDASPCP-S-021	Support the encryption for Content Protection of File for DRM Profile	5.6.2.2	O	BCAST-BSDASPCP-S-028 OR BCAST-BSDASPCP-S-031 OR BCAST-BSDASPCP-S-033
BCAST-BSDASPCP-S-022	Support SDP signalling for Content Protection of DRM Profile	5.8	O	BCAST-BSDASPCP-S-034
BCAST-BSDASPCP-S-023	Support Smartcard Profile for content protection	6	O	BCAST-BSDASPCP-S-024 AND BCAST-BSDASPCP-S-025 AND BCAST-BSDASPCP-S-026 AND BCAST-BSDASPCP-S-027
BCAST-BSDASPCP-S-024	Support delivery of STKM for Smartcard profile	6.7.3	O	
BCAST-BSDASPCP-S-025	Support the encryption for Content Protection of Stream for Smartcard Profile	6.7.4	O	BCAST-BSDASPCP-S-028 OR (BCAST-BSDASPCP-S-029 AND BCAST-BSDASPCP-S-038) OR (BCAST-BSDASPCP-S-030 AND BCAST-BSDASPCP-S-037)
BCAST-BSDASPCP-S-026	Support the encryption for Content Protection of File for Smartcard Profile	6.8.2.2	O	BCAST-BSDASPCP-S-028 OR BCAST-BSDASPCP-S-033
BCAST-BSDASPCP-S-027	Support SDP signalling for Content Protection of Smartcard Profile	6.10.1.2	O	BCAST-BSDASPCP-S-034
BCAST-BSDASPCP-S-028	Support IPSEC	9.1	O	
BCAST-BSDASPCP-S-029	Support SRTP	9.2	O	
BCAST-BSDASPCP-S-030	Support ISMACryp	9.3	O	
BCAST-BSDASPCP-S-031	Support PDCF	9.4	O	
void-BCAST-BSDASPCP-S-032	Obsolete	N/A		
BCAST-BSDASPCP-S-033	Support DCF	9.4	O	
BCAST-BSDASPCP-S-034	Support SDP signalling for protection	10.1	O	BCAST-BSDASPCP-S-035 AND BCAST-BSDASPCP-S-036
BCAST-BSDASPCP-S-035	Support SDP signalling for STKM	10.1.3	O	
BCAST-BSDASPCP-S-036	Support SDP signalling for LTKM	10.1.4	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Item	Function	Reference	Status	Requirement
BCAST-BSDASPCP-S-037	Support SDP signalling for ISMACryp	0	O	
BCAST-BSDASPCP-S-038	Support SDP signalling for SRTP	10.4	O	
BCAST-BSDASPCP-S-039	Support for STKM generation	13.1.2.4	O	
BCAST-BSDASPCP-S-040	Support the common attribute of STKM	7	O	
BCAST-BSDASPCP-S-041	Support for the operation for recording	8	O	
BCAST-BSDASPCP-S-042	Support for sharing a protected data stream for the different operators using both DRM and Smartcard profile	11	O	BCAST-BSDASPCP-S-043 AND BCAST-BSDASPCP-S-044 AND (BCAST-BSDASPCP-S-045 OR BCAST-BSDASPCP-S-046)
BCAST-BSDASPCP-S-043	Support mapping for mapping of encryption and authentication keys	11.1	O	
BCAST-BSDASPCP-S-044	Support mapping for mapping between Key IDs for Smartcard profile and Key IDs for DRM profile	11.2	O	
BCAST-BSDASPCP-S-045	Support sharing SRTP Protected data Stream	11.3	O	
BCAST-BSDASPCP-S-046	Support sharing ISMACryp Protected data Stream	11.4	O	

B.3 SCR for BSM

Item	Function	Reference	Status	Requirement
BCAST-BSMSPCP-S-001	Support Backend interface for Service Protection	13.1	O	BCAST-BSMSPCP-S-002 OR BCAST-BSMSPCP-S-003
BCAST-BSMSPCP-S-002	Support BCAST specific interface for SP-4	13.1.2	O	
BCAST-BSMSPCP-S-003	Support SP-4 by the adaptation of DVB Simulcrypt Head-end interfaces	13.1.1	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-BSMSPCP-S-004	Support Backend interface for Content Protection	13.2	O	BCAST-BSMSPCP-S-005
BCAST-BSMSPCP-S-005	Support CP-4 by BCAST specific interface	13.2	O	
BCAST-BSMSPCP-S-006	Support Service Protection	4	O	BCAST-BSMSPCP-S-007 OR BCAST-BSMSPCP-S-008
BCAST-BSMSPCP-S-007	Support DRM profile for Service Protection	5	O	BCAST-BSMSPCP-S-009 AND BCAST-BSMSPCP-S-010 AND BCAST-BSMSPCP-S-011 AND BCAST-BSMSPCP-S-039
BCAST-BSMSPCP-S-008	Support Smartcard profile for Service Protection	6	O	BCAST-BSMSPCP-S-012 AND BCAST-BSMSPCP-S-013 AND BCAST-BSMSPCP-S-018 AND BCAST-BSMSPCP-S-019 AND BCAST-BSMSPCP-S-034 AND BCAST-BSMSPCP-S-039 AND BCAST-BSMSPCP-S-053
BCAST-BSMSPCP-S-009	Support registration for DRM Profile	5.3	O	
BCAST-BSMSPCP-S-010	Support LTKM generation for DRM Profile for Service Protection	5.4	O	
BCAST-BSMSPCP-S-011	Support STKM generation for DRM Profile for Service Protection	5.5	O	
BCAST-BSMSPCP-S-012	Support Subscriber Key Establishment for Smartcard Profile	6.5	O	
BCAST-BSMSPCP-S-013	Support LTKM generation for Smartcard Profile for Service Protection	6.6	O	
BCAST-BSMSPCP-S-014	Support of LTKM Push delivery over UDP	6.6	O	
BCAST-BSMSPCP-S-015	Support of LTKM Request	6.6	O	
BCAST-BSMSPCP-S-016	BSM Solicited Pull Procedure Initiation over SMS Bearer	6.6.2	O	BCAST-BSMSPCP-S-015
BCAST-BSMSPCP-S-017	BSM Solicited Pull Procedure to Initiate the Registration Procedure	6.6.3	O	BCAST-BSMSPCP-S-015 AND BCAST-BSMSPCP-S-053

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-BSMSPCP-S-018	Support the use of EXT BCAST for LTKM	6.6.4	O	
BCAST-BSMSPCP-S-019	Support of SPE=0x04	6.6.4	O	
BCAST-BSMSPCP-S-020	Support of SPE=0x00	6.6.4	O	
BCAST-BSMSPCP-S-021	Support of SPE=0x01	6.6.4	O	
BCAST-BSMSPCP-S-022	Support of SPE=0x02	6.6.4	O	
BCAST-BSMSPCP-S-023	Support of SPE=0x03	6.6.4	O	
BCAST-BSMSPCP-S-024	Support of SPE=0x05	6.6.4	O	
BCAST-BSMSPCP-S-025	Support of SPE=0x07	6.6.4	O	
BCAST-BSMSPCP-S-026	Support of SPE=0x08	6.6.4	O	
BCAST-BSMSPCP-S-027	Support of SPE=0x09	6.6.4	O	
BCAST-BSMSPCP-S-028	Support of SPE=0x0A	6.6.4	O	
BCAST-BSMSPCP-S-029	Support of SPE=0x0C	6.6.4	O	
BCAST-BSMSPCP-S-030	Support of SPE=0x0D	6.6.4	O	
BCAST-BSMSPCP-S-031	Support for Parental Control Message	6.6.5	O	
BCAST-BSMSPCP-S-032	Support for verification messages	6.6.6.1	O	
BCAST-BSMSPCP-S-033	Support for reporting messages	6.6.6.2 6.6.6.3	O	
BCAST-BSMSPCP-S-	Support STKM generation for Smartcard	6.7	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
034	Profile for Service Protection			
BCAST-BSMSPCP-S-035	Support the use of EXT BCAST for STKM	6.7.2	O	
BCAST-BSMSPCP-S-036	Parental control access criteria	6.7.3.11.1	O	
BCAST-BSMSPCP-S-037	Location access criteria	6.7.3.11.2	O	
BCAST-BSMSPCP-S-038	Support for BCAST Client ID for Smartcard Profile	6.11	O	
BCAST-BSMSPCP-S-039	Support the common attribute of STKM	7	O	
BCAST-BSMSPCP-S-040	Support Content Protection	4	O	BCAST-BSMSPCP-S-041 OR BCAST-BSMSPCP-S-042
BCAST-BSMSPCP-S-041	Support DRM Profile for Content Protection	5	O	BCAST-BSMSPCP-S-009 AND BCAST-BSMSPCP-S-044 AND BCAST-BSMSPCP-S-045
BCAST-BSMSPCP-S-042	Support Smartcard Profile for Content Protection	6	O	BCAST-BSMSPCP-S-053 AND BCAST-BSMSPCP-S-047 AND BCAST-BSMSPCP-S-048
void-BCAST-BSMSPCP-S-043	Obsolete	N/A		
BCAST-BSMSPCP-S-044	Support LTKM generation for DRM Profile for Content Protection	5.4	O	
BCAST-BSMSPCP-S-045	Support STKM generation for DRM Profile for Content Protection	5.5	O	
void-BCAST-BSMSPCP-S-046	Obsolete	N/A		
BCAST-BSMSPCP-S-047	Support LTKM generation for Smartcard Profile for Content Protection	6.6	O	
BCAST-BSMSPCP-S-048	Support STKM generation for Smartcard Profile for Content Protection	6.7	O	
BCAST-BSMSPCP-S-049	Support recording for DRM Profile for Content Protection	5.7 and 8	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Item	Function	Reference	Status	Requirement
BCAST-BSMSPCP-S-050	Support recording for Smartcard Profile for Content Protection	6.9 and 8	O	
BCAST-BSMSPCP-S-051	Usage Metering for DRM Profile	5.9	O	
BCAST-BSMSPCP-S-052	Support TBK for Smartcard Profile for Content Protection	12	O	
BCAST-BSMSPCP-S-053	MBMS registration and de-registration procedure	6.6	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

B.4 SCR for Smartcard

Item	Function	Reference	Status	Requirement
BCAST-SCSPCP-C-001	Smartcard is (U)SIM card	6.2	O	
BCAST-SCSPCP-C-002	Smartcard is (R)-UIM/CSIM Card	6.2	O	
BCAST-SCSPCP-C-003	Support of the GBA Subscriber Key establishment	6.2	O	BCAST-SCSPCP-C-001
BCAST-SCSPCP-C-004	Support of BCMCS Subscriber Key establishment	6.2	O	BCAST-SCSPCP-C-002
BCAST-SCSPCP-C-005	Support of all MBMS Key management features	6.2	O	BCAST-SCSPCP-C-003 OR BCAST-SCSPCP-C-004
BCAST-SCSPCP-C-006	Support of MBMS Key management features only related to the processing of MBMS MSK and MTK messages	6.2	O	BCAST-SCSPCP-C-004 OR BCAST-SCSPCP-C-003
BCAST-SCSPCP-C-007	Support of BCAST Key management	6.2	O	(BCAST-SCSPCP-C-005 OR BCAST-SCSPCP-C-006) AND BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-009
BCAST-SCSPCP-C-008	Support the use of EXT BCAST for LTKM	6.6.4	O	BCAST-SCSPCP-C-007 AND BCAST-SCSPCP-C-012 AND BCAST-SCSPCP-C-010 AND BCAST-SCSPCP-C-011
BCAST-SCSPCP-C-009	Support the use of EXT BCAST for STKM	6.7.2	O	BCAST-SCSPCP-C-007 AND BCAST-SCSPCP-C-028
BCAST-SCSPCP-C-010	Support of verification message	6.6.6.1	O	
BCAST-SCSPCP-C-011	Support of reporting message	6.6.6.2	O	
BCAST-SCSPCP-C-012	Support of SPE=0x04	6.6.4	O	BCAST-SCSPCP-C-028 AND BCAST-SCSPCP-C-029 AND BCAST-SCSPCP-C-030 AND BCAST-SCSPCP-C-032
BCAST-SCSPCP-C-013	Support of SPE=0x00	6.6.4	O	BCAST-SCSPCP-C-008
BCAST-SCSPCP-C-014	Support of SPE=0x01	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Item	Function	Reference	Status	Requirement
BCAST-SCSPCP-C-015	Support of SPE=0x02	6.6.4	O	BCAST-SCSPCP-C-008
BCAST-SCSPCP-C-016	Support of SPE=0x03	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034
BCAST-SCSPCP-C-017	Support of SPE=0x05	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034
BCAST-SCSPCP-C-018	Support of SPE=0x07	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034
BCAST-SCSPCP-C-019	Support of SPE=0x08	6.6.4	O	BCAST-SCSPCP-C-008
BCAST-SCSPCP-C-020	Support of SPE=0x09	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034
BCAST-SCSPCP-C-021	Support of SPE=0x0A	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-029
BCAST-SCSPCP-C-022	Support of SPE=0x0C	6.6.4	O	BCAST-SCSPCP-C-008
BCAST-SCSPCP-C-023	Support of SPE=0x0D	6.6.4	O	BCAST-SCSPCP-C-008 AND BCAST-SCSPCP-C-031 AND BCAST-SCSPCP-C-033 AND BCAST-SCSPCP-C-034
BCAST-SCSPCP-C-024	Support of Parental control messages	6.6.5	O	BCAST-SCSPCP-C-007 AND BCAST-SCSPCP-C-029
BCAST-SCSPCP-C-025	Support for Parental Control	6.6.5, 6.7.3.11.1	O	BCAST-SCSPCP-C-009
BCAST-SCSPCP-C-026	Support of PINCODE function for parental control	6.7.3.11.1	O	BCAST-SCSPCP-C-025
BCAST-SCSPCP-C-027	Support of location-based-restriction	6.7.3.11.2	O	BCAST-SCSPCP-C-009
BCAST-SCSPCP-C-028	Support of MTK generation mode AUTHENTICATE Command for OMA BCAST operation	E.2.1	O	
BCAST-SCSPCP-C-029	Support of MSK update mode AUTHENTICATE Command for OMA BCAST operation	E.2.2	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Item	Function	Reference	Status	Requirement
BCAST-SCSPCP-C-030	Support of AUTHENTICATE Command for OMA BCAST operation: SPE Deletion Mode	E.2.3.1	O	
BCAST-SCSPCP-C-031	Support of AUTHENTICATE Command for OMA BCAST operation: Recording Deletion Mode	E.2.3.2	O	
BCAST-SCSPCP-C-032	Support of OMA BCAST Command: SPE Audit Mode	E.3.2	O	
BCAST-SCSPCP-C-033	Support of OMA BCAST Command: Record Signalling Mode	E.3.3	O	
BCAST-SCSPCP-C-034	Support of OMA BCAST Command: Recording Audit Mode	E.3.4	O	

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Appendix C. Global Status Codes

Table 66: Global Status Codes

Code	Status
000	**Success** The request was processed successfully.
001	**Device Authentication Failed** This code indicates that the BSM was unable to authenticate the device, which may be due to the fact that the device is not registered with the BSM. In this case, the user may contact the BSM, and establish a contract, or get the credentials in place that are used for authentication.
002	**User Authentication Failed** This code indicates that the BSM was unable to authenticate the user, which may be due to the fact that the device is not registered with the BSM. In this case, the user may contact the BSM, and establish a contract, or get the credentials in place that are used for authentication.
003	**Purchase Item Unknown** This code indicates that the requested service item is unknown. This can happen e.g. if the device has a cached service guide with old information. In this case, the user may re-acquire the service guide.
004	**Device Authorization Failed** This code indicates that the device is not authorized to get Long-Term Key Messages from the RI, e.g. because the device certificate was revoked. In this case, the user may contact the BSM operator.
005	**User Authorization Failed** This code indicates that the user is not authorized to get Long-Term Key Messages from the RI, e.g. because the device certificate was revoked. In this case, the user may contact the BSM operator.
006	**Device Not Registered** This code indicates that the device is not registered with the RI that is used for the transaction. When this code is sent, the response message includes a registration trigger that allows the device to register. In this case, the device may automatically perform the registration, and, if the registration is successful, re-initiate the original transaction.
007	**Server Error** This code indicates that there was a server error, such as a problem connecting to a remote back-end system. In such a case, the transaction may succeed if it is re-initiated later.
008	**Mal-formed Message Error** This code indicates that there has been a device malfunction, such as a mal-formed XML request. In such a case, the transaction may or may not (e.g. if there is an interoperability problem) succeed if it is re-initiated later. Note: This code can also be used between network entities.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

009	**Charging Error** This code indicates that the charging step failed (e.g. agreed credit limit reached, account blocked). The user may in such a case contact the BSM operator. Note: This code can also be used between network entities.
010	**No Subscription** This code indicates that there has never been a subscription for this service item, or that the subscription for this item has terminated. The user may in such a case issue a service request for a new subscription.
011	**Operation not Permitted** This code indicates that the operation that the device attempted to perform is not permitted under the contract between BSM and user. The user may in this case contact BSM operator and change the contract. Note: This code can also be used between network entities.
012	**Unsupported version** This code indicates that the version number specified in the request message is not supported by the network. In this case, the user may contact the BSM operator. Note: This code can also be used between network entities.
013	**Illegal Device** This code indicates that the device requesting services is not acceptable to the BSM. E.g. Blacklisted. In this case, the user may contact the BSM operator.
014	**Service Area not Allowed** This code indicates that the device is not allowed services in the requested area due to subscription limits In this case, the user may contact the BSM operator or subscribe to the applicable service.
015	**Requested Service Unavailable** This code indicates that the requested service is unavailable due to transmission problems. In this case, the request may re-initiated at a later time. Note: This code can also be used between network entities.
016	**Request already Processed** This code indicates that an identical request has been previously processed. In this case, the user or the entity may check to see if the request had already been processed (i.e. received an LTK), if not retry the request.
017	**Information Element Non-existent** This code indicates that the message includes information elements not recognized because the information element identifier is not defined or it is defined but not implemented by the entity receiving the message. In this case related entities should contact each other.
018	**Unspecified** This code indicates that an error has occurred which cannot be identified. In this case related entities should contact each other.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

019	**Process Delayed** Due to heavy load, request is in the cue, waiting to be processed. In this case the user or entity should wait for the transaction to complete. Note: This code can also be used between network entities.
020	**Generation Failure** This code indicates that the request information (message) could not be generated. In this case the user or entity should retry later.
021	**Information Invalid** This code indicates that the information given is invalid and cannot be used by the system. In this case the request should be rechecked and sent again.
022	**Invalid Request** This code indicates that the requesting key materials and messages (e.g., LTKM) are not valid and can not be fulfilled. In this case the request should be rechecked and sent again.
023	**Wrong Destination** This code indicates that the destination of the message is not the intended one. In this case the request should be rechecked and sent again.
024	**Delivery of Wrong Key Information** This code indicates that the delivered key information and messages (e.g., LTKM) are invalid. In this case the request should be rechecked and sent again.
025	**Service Provider ID Unknown** This code indicates a confliction when the Visited Service or Home Provider requests a message to the Home Service or Visited Provider.
026	**Service Provider BSM_ID Unknown** This code indicates a confliction when the Visited Service or Home Provider BSM requests a message to the Home Service or Visited Provider BSM.
027 ~ 127	Reserved for future use
128 ~ 255	Reserved for proprietary use

The informative Table 67 below proposes example values from the Global Status Codes in the table above for the transaction messages that require the use of Global Status Codes. The values shown below are for guidance purposes and the full range of values of the Global Status Codes are applicable to all messages if deemed required.

Table 67: Cross Reference Table (Informative)

TS-BCAST_SvcCntProtection	13.1.2.2.1.2 Key Request Response	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	13.1.2.2.1.4 Key Delivery Confirmation	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	13.1.2.3.1.2 Key Request Response	000, 007, 008, 011, 012, 015, 016, 017, 018,

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

		019, 020, 021, 022, 023
	13.1.2.3.1.4 Key Delivery Confirmation	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	13.1.2.4.1.2 STKM Response	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	13.1.2.4.1.4 Partial STKM response message	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023
	13.1.2.4.1.6 STKM Delivery Confirmation	000, 007, 008, 011, 012, 015, 016, 017, 018, 019, 020, 021, 022, 023

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix D. Protected Outputs (Informative)

For some Protection_after_Reception values, rendering is allowed over appropriately protected output links. The definition of which protected link technologies are allowed is typically a deployment or trust authority issue and outside the scope of this specification.

Examples for such links could include, but are not restricted to, DTCP (DTCP-IP, DTCP-BT-Audio, DTCP-BT-Video, DTCP-USB, DTCP-1394), HDCP, BT-AD2P, BT-HFP.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix E. Terminal - BCAST Smartcard Interface in the Smartcard Profile (Normative)

E.1 Implementing BCAST Smartcard Functionality

BCAST Smartcard functionality can be implemented as follows:

- As a part of the USIM application, as defined in Appendix E.4;
- By implementing a BSIM application, as defined in Appendix E.5; or
- As part of CSIM application or (R-)UIM (to be defined).

If the BCAST Smartcard supports a BSIM application, as indicated by the presence of the BSIM AID in EF_DIR, and the Terminal supports the selection of the BSIM application, then the Terminal SHALL select the ADF_BSIM before issuing any of the commands defined in Appendix E.2 and Appendix E.3. This rule also applies to the BCAST related GBA commands and Local Key Establishment commands (see Appendix E.5.4.1).

If the BCAST Smartcard does not support a BSIM application, as indicated by the absence of the BSIM AID in EF_DIR, or the Terminal doesn't support the selection of the BSIM application, then the Terminal SHALL select the DF_BCAST under the network access application that is currently active (e.g. ADF_USIM) before issuing any command related to a BCAST operation.

The use of the BCAST functionality provided by the Smartcard SHALL NOT depend on the access method used by the Terminal, i.e. the BCAST commands sent and received by the Terminal SHALL be the same whether the Terminal has selected the ADF BSIM or the DF BCAST under the ADF_USIM or ADF_CSIM (to be defined), or in the (R-)UIM (to be defined). The commands defined in Appendix E.2 and Appendix E.3 are equally applicable to either approach.

As the selection of the ADF_BSIM or the DF_BCAST under the ADF_USIM or ADF_CSIM, or in the (R-)UIM is dependent on the capabilities of the Terminal and Smartcard, as described above, a Terminal will always select the same ADF_BSIM or DF_BCAST under ADF_USIM or ADF_CSIM, or in the (R-)UIM, when paired with the same Smartcard.

E.2 Extension of the MBMS Security Context

The parameters of the Authenticate command response for the MBMS security Context Mode, defined in 3GPP TS 31.102 [3GPP TS 31.102 v7], have been extended to enable BCAST information to be returned to the Terminal.

Additional Parameters and Data are defined for BCAST to the MBMS security context response in case of failure in the processing of MTK Generation Mode or MSK Update Mode for

- security_policy_extension
- parental control
- location based restriction

In addition, the Authenticate command for the MBMS Security Context, defined in 3GPP TS 31.102 [3GPP TS 31.102 v7], is extended with one additional OMA BCAST mode that comprises two sub-modes:

⇒ The SPE deletion sub mode to delete the SecurityPolicy Extension and associated data stored in the Smartcard

⇒ The Recording deletion sub mode to delete the Content Identifier and its association to flagged SPEs when the corresponding content is erased in the terminal

Other sub modes could be defined for future extension of the BCAST specification.

Table 68 contains the definition of Operation Status code used for the operations defined in the following sections.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 68: Operation Status Code Coding

Value	Description
0x00	Successful
0x01	Lack of credit in the live ppt_purse
0x02	Lack of credit in the playback_ppt_ purse
0x03	Lack of credit in the TEK counter
0x04	lack of credit in the user_purse
0x05	Play_back counter invalid or equal to zero
0x06	Tek counter invalid or equal to zero
0x07	User not authorized
0x08	PINCODE required
0x09	PINCODE not initialized
0x0A	PINCODE blocked
0x0B	Blackout
0x0C	Need specific permissions
0x0D	SPE used for Recording
0x0E	Parental control not supported
0x0F	PINCODE has been successfully changed
0x10	rating_type/level-granted pair has been successfully changed
0x11	PINCODE and rating_type/level-granted pair has been successfully changed
0x12	Security policy extension not supported
0x13	Security policy not satisfied; Secure Channel is required

E.2.1 MTK Generation Mode

In case of BCAST successful operation in the processing of MTK Generation Mode, the response parameters and data defined here after for the response of the Authenticate Command SHALL be used.

In case of failure in the BCAST processing of MTK Generation Mode, the additional parameters and data defined here after for the response of the Authenticate Command SHALL be used.

If the BCAST processing of MTK Generation Mode fails during operation on Security Policy Extension (i.e. during the update of purses, TEK counter and Playback counter), the OMA BCAST operation response for security policy extension operation described below, SHALL be returned to the Terminal.

If the BCAST processing of MTK Generation Mode fails during operation on parental control enforcement, the OMA BCAST operation response for parental control operation described below SHALL be returned to the Terminal.

If the BCAST processing of MTK Generation Mode fails during operation on location based restriction enforcement, the OMA BCAST operation response for location based restriction operation described below SHALL be returned to the Terminal.

E.2.1.1 OMA BCAST Operation Response: BCAST management_data Operation

Table 69: Coding of OMA BCAST Operation Response - BCAST management_data Operation (MTK Generation Mode)

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

2 to 1+A bytes (A ≤ 4)	MBMS operation response Data Object length (L)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2	OMA BCAST operation response TLV tag = 'AE'	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
A+3 to A+2+B	OMA BCAST operation response Data Object TLV Length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
A+2+B+1	BCAST management_data response Data Object tag ('80')		1
A+2+B+2	BCAST management_data response Data Object length (L=1)		1
A+2+B+3	BCAST management_data response Data Object	See Table 70	1
A+2+B+4	TEK Data Object tag '86' (Note1)		1
A+2+B+5	TEK Data Object Length (D)		1
A+2+B+6 to A+2+B+5+D	TEK Data Object		D
(A+2+B+5+D)+1	SALT Data Object tag '87'		1
(A+2+B+5+D)+2	SALT Data Object Length (E)		1
(A+2+B+5+D)+3 to (A+2+B+5+D)+2+E	SALT Data Object		E
Note1: TEK will only be returned if the BCAST management_data is success full (0x00) and if present in the incoming MIKEY message. Note2: SALT will only be returned if present in the incoming MIKEY message			

Table 70: Coding of BCAST management_data response Data Object tag'80'

Byte(s)	Description	Coding	Length
1	BCAST management_data Status code	See Table 68	1

Table 71: Coding of TEK Data Object tag'86'

Byte(s)	Description	Coding	Length
1-D	TEK	See TS 33.246 [3GPP TS 33.246 v7]	D

Table 72: Coding of SALT Data Object tag'87'

Byte(s)	Description	Coding	Length
1-E	SALT	See TS 33.246 [3GPP TS 33.246 v7]	E

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

E.2.1.2 OMA BCAST Operation Response: Parental Control Operation

Table 73: Coding of OMA BCAST Operation Response - Parental Control Operation

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A bytes (A ≤ 4)	MBMS operation response Data Object length (L)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2	OMA BCAST operation response tag ('AE')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
A+3 to A+2+B	OMA BCAST operation response Data Object TLV Length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
(A+2+B)+1	BCAST management_data response Data Object tag ('80')		1
(A+2+B)+2	BCAST management_data response Data Object length (L=1)		1
(A+2+B)+3	BCAST management_data response Data Object	See Table 70	1
(A+2+B+3) +1	Parental control operation Data Object tag ('88')		1
(A+2+B+3) +2	Parental control operation Data Object length L2 (Note1)		1
(A+2+B+3) + 3 to (A+2+B+3) + 2+L2	Parental control operation Data Object	See Table 74	L2
Note1: In BCAST_1.0 it is assumed that L2 is '04' Bytes.			

Table 74: Coding of Parental Control Operation Data Object tag'88'

Byte(s)	Description	Coding	Length
1	Key reference of the PIN defined for the parental control	See below	1
2	Current rating_type	See below	1
3	Current rating_value	See below	1
4	level_granted value for the current rating_type	See below	1

For the parental control operation, only the BCAST management_data response values 0x07, 0x08, 0x09 and 0x0A are allowed.

Key reference of the PIN:

The key references are defined in [ETSI TS 102.221].

The value of the key reference is in the range of values defined in table 9.3 in section 9.5.1 of [ETSI TS 102.221] and restricted to the values for level 2 (see Note below). The key reference chosen at the manufacture of the Smartcard SHALL be uniquely assigned in the UICC for the parental control function and SHALL NOT be used to access to files.

Where a Parental Control PIN is defined, the Parental Control PIN SHALL be assigned to the DF_BCAST under the ADF_USIM in the case that the BCAST functionality is implemented as part of the USIM or CSIM or (R-)UIM or to the ADF_BSIM in the case that the BCAST functionality is implemented as a BSIM.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

When PINCODE management is supported in the Smartcard, the FCP template in the response of selecting the ADF_BSIM or DF_BCAST SHALL contain the key reference of the Parental Control PIN.

The Terminal SHALL prompt the user for verification of the parental control PINCODE only after reception of the response of the AUTHENTICATE command in MTK generation mode where the operation status code is "PINCODE required", as specified in Section 6.7.3.11.1.

The Parental Control PIN assigned to USIM or CSIM or (R-)UIM MAY be connected with the Parental Control PIN used by the BSIM. This may be desirable to address the case in which a Smartcard supports both methods of implementing BCAST Smartcard functionality in order to be able to synchronise the values of the Parental Control PINs used by each Smartcard application. This is only necessary if the user switches between Terminals that have different levels of support for selecting the BSIM application. In the case that there are multiple USIM and/or BSIM applications on the Smartcard, the linking of Parental Control PINs used by the applications is determined by the Smartcard issuer.

Note: Key reference values for level 4 might be considered in the future.

Current rating_type:

This field indicates the rating_type of the content streams currently in decryption. This rating_type is indicated in the incoming STKM and returned to the terminal in case of error. The coding of this field is as specified in the OMA BCAST Parental Rating System Registry available at [OMNA].

Current rating_value:

This field indicates the current rating_value for to the current rating_type. The coding of this field is as specified in the OMA BCAST Parental Rating System Registry available at [OMNA].

level_granted value for the current rating_type:

This field indicates the level_granted value for the current rating_type. This value is stored in the Smartcard. The coding of this field is as specified in the rating_value column in the OMA BCAST Parental Rating System Registry available at [OMNA].

E.2.1.3 OMA BCAST Operation Response: Location-based Restriction Operation

Table 75: Coding of OMA BCAST Operation Response: Location-based Restriction Operation

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A bytes (A ≤ 4)	MBMS operation response Data Object length (L)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2	OMA BCAST operation response tag ('AE')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
A+3 to A+2+B	OMA BCAST operation response Data Object TLV Length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
(A+2+B) +1	BCAST management_data response Data Object tag ('80')		1
(A+2+B) +2	BCAST management_data response Data Object length (L=1)		1
(A+2+B) +3	BCAST management_data response Data Object	See below	1

For the BCAST management_data response Data Object tag '80' coding, see Table 70.

For the location based restrictions operation, only the BCAST management_data response values 0x0B and 0x0C are allowed.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

E.2.2 MSK Update Mode

1. In case of BCAST successful operation in the processing of MSK Update Mode, the response parameters and data defined here after for the response of the Authenticate Command SHALL be used.

The following cases may be encountered depending of the content of the MIKEY message:

- If the MIKEY message is a parental control message and contains only a PINCODE update that ends successfully, the Operation Status code 0x0F 'Parental control PINCODE has been successfully changed' is returned. If it contains a PINCODE and rating_type/level_granted pairs update, the Operation Status code 0x11 is returned. If it contains only the rating_type/level_granted pairs update, the Operation Status code 0x10 is returned.

- In all other cases, the Operation Status code '0x00' 'successful' is returned.

2. In case of failure in the BCAST operation during the processing of MSK Update Mode, the response parameters and data defined here after for the response of the Authenticate Command SHALL be used.

The following failure may be encountered:

- If the Security Policy Extension in the EXT BCAST extension of the MIKEY message is not supported by the Smartcard, the Operation Status code '0x12' 'security policy extension not supported' is returned.

- If the MIKEY message is a parental control message and the Smartcard does not support the parental control function, the Operation Status code 0x0E 'Parental control not supported' is returned.

E.2.2.1 OMA BCAST Operation Response: BCAST management_data Operation

Table 76: Coding of OMA BCAST Operation Response - BCAST management_data Operation (MSK Update Mode)

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A bytes (A ≤ 4)	MBMS operation response Data Object length (L)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2	OMA BCAST operation response tag = 'AE'	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
A+3 to A+2+B	OMA BCAST operation response Data Object Length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
A+3+B to A+2+B+L1	OMA BCAST operation response Data Object	See Table 77	L1

Table 77: Coding of OMA BCAST Operation response Data Object

Byte(s)	Description	Coding	Length
1	BCAST management_data response Data Object tag ('80')		1
2	BCAST management_data response Data Object length (L=1)		1
3	BCAST management_data response Data Object	See below	1
4	Parental rating Data Object Tag ('8A') (NOTE 2)		1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5	Parental rating Data Object Length		1
6 to 7	Parental rating Data Object	See below	2
3+4n+1	SPE Type not supported Data Object Tag ('8B') (NOTE 3)		1
3+4n+2	SPE Type not supported Data Object Length		1
3+4n+3	SPE Type not supported Data Object	See below	1
3+4n+4	MIKEY message Data Object Tag ('8C') (NOTE 1)		1
3+4n+5 to 3+4n+4+C	MIKEY message Data Object Length (L1)		C
3+4n+4+C+1 to 3+4n+4+C+L1	MIKEY Message Data		L1
NOTE 1: Data Object present if a LTKM Verification message or a LTKM Reporting message is returned. NOTE 2: The response data may contain none or multiple Parental Ratings Data Objects (n = number of Parental rating Data Objects). This number of data objects with tag '8A' indicates the number of rating_type/level_granted pairs received in the current parental control message. NOTE 3: This tag is present if the BCAST management_data response Data Object value is 0x0D. NOTE 4: Parental rating Data Object Tag present only if a MIKEY parental control message is included the AUTHENTICATE command. NOTE 5: SPE Type not supported Data Object Tag only present if a MIKEY LTKM is included the AUTHENTICATE command.			

For the BCAST management_data response Data Object tag '80' coding, see Table 70.

In this case, only the BCAST management_data response values 0x00, 0x0D, 0x0E, 0x0F, 0x10, 0x11, 0x12 are allowed.

Table 78: Coding of Parental Rating Data Object tag'8A'

Byte(s)	Description	Status	Value	Length
1	rating type	M	See coding in [OMNA]	1
2	level_granted	M	See coding in [OMNA]	1

Table 79: Coding of SPE Type not Supported Data Object tag'8B'

Byte(s)	Description	Status	Value	Length
1	Security Policy Extension	M	Coded as defined in 6.6.4.2	1

Security Policy Extension:

This is the value of security policy extension received in the LTKM that is not supported by the Smartcard.

rating_type:

This field indicates the rating_types stored in the Smartcard after the update caused by the received parental control message or the originally stored default value. The coding of this field is as specified in the OMA BCAST Parental Rating System Registry available at [OMNA].

level_granted:

This field indicates the level_granted value stored in the Smartcard after the update caused by the received parental control message or the originally stored default value. The level_granted value is associated to the rating_type above. The coding of this field is as specified in the rating_value column in the OMA BCAST Parental Rating System Registry available at [OMNA].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

E.2.3 MBMS Security Context – OMA BCAST Operation

The AUTHENTICATE command for the MBMS Security Context, defined in [3GPP TS 31.102 v7], is extended with one additional OMA BCAST mode of operation that comprises two sub modes:

- The SPE deletion sub mode to delete the Security Policy Extension and associated data stored in the Smartcard
- The Recording deletion sub mode to delete the Content Identifier and its association to flagged SPEs when the corresponding content is erased in the terminal

Other sub modes could be defined for future extension of the BCAST specification.

Command extension parameters and data:

The coding of the AUTHENTICATE command parameters and data is extended as follows:

Table 80: Coding of AUTHENTICATE Command Parameters and Data

Byte(s)	Description	Coding	Length
1	MBMS Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A (A ≤ 4)	MBMS Data Object length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2	OMA BCAST Operation TLV tag ('AE')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
A+3 to A+2+B	OMA BCAST Operation TLV Data Object Length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
A+3+B to A+2+B+L	OMA BCAST Operation TLV Data Object	See below	L

The coding of the OMA BCAST Operation TLV is as follows:

Table 81: Coding of OMA BCAST Operation TLV

Description	Value	M/O	Length (bytes)
OMA BCAST Operation TLV Tag	'AE'	M	1
OMA BCAST Operation TLV Data Object Length	L	M	B
OMA BCAST Operation Mode Tag	'90'	M	1
OMA BCAST Operation Mode Data Object Length	1	M	1
OMA BCAST Operation Mode Data Object	See below	M	1
Mode specific TLVs (Note1)		O	L2
Note1: For each BCAST operation mode, a list of TLV is defined in the following section			

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 82: Coding of OMA BCAST Operation Mode Data Object

Coding	Meaning
'00'	RFU
'01'	SPE Deletion Mode
'02'	Recording Deletion Mode
'03' to 'FF'	RFU

E.2.3.1 MBMS Security Context – OMA BCAST Operation - SPE Deletion Mode

E.2.3.1.1 SPE Deletion Mode: Command Description

BCAST Smartcards SHALL support the command described in this section.

The Smartcard receives from the terminal in the AUTHENTICATE command the Key Domain ID and SEK/PEK ID Key group part and optionally the SEK/PEK ID Key Number part, the Key Validity Data and the Security Policy Extension of the SEK/PEK that define the SPE to delete. The Smartcard shall identify all SPEs matching with all TLV received. If the Key Validity Data TLV is present, a SEK/PEK is matching only if its values for TS Low and TS High match the TS Low and TS High values in the command data.

If no match can be found on the Smartcard for the SPE defined in the command data, the Smartcard SHALL abandon the function and return the status word '6A88' (Referenced data not found).

If one or several matches can be found on the Smartcard for the SPE defined in the command data, the Smartcard SHALL delete all corresponding SPE and associated data if the UsedForRecording Flag is not set in the Smartcard for this SPE.

If one or several matches can be found on the Smartcard for the SPE defined in the command data, and the UsedForRecording Flag is set in the Smartcard, the Smartcard SHALL not delete the corresponding SPE that could still be usable for a recorded content. The Recording Deletion command SHALL be used to delete the UsedForRecording flag associated to SPEs.

If in the command, only mandatory fields (i.e. Key Domain ID TLV and SEK/PEK ID Key group part TLV) are present in the Key Identifier TLV, the Smartcard SHALL delete all SPE, SEK/PEKs associated to the Key Group and other information associated to the Key group as live_ppt_purse, playback_ppt_purse, kept_TEK_counter, etc.

Input:

- Key Domain ID and SEK/PEK ID Key Group part
- optionally the SEK/PEK ID Key Number part , the Key Validity Data and the Security Policy Extension

Output:

- Operation Status Code

E.2.3.1.2 SPE Deletion Mode: Parameters and Data

In case of SPE Deletion Mode (i.e. OMA BCAST Operation Mode Data Object value is ‘01’), the coding of OMA BCAST Operation TLV is as follows:

Table 83: Coding of OMA BCAST Operation TLV

Description	Value	M/O	Length (bytes)
OMA BCAST Operation TLV Tag	'AE'	M	1
OMA BCAST Operation TLV Data Object Length	L	M	1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

OMA BCAST Operation Mode Tag	'90'	M	1
OMA BCAST Operation Mode Data Object Length	1	M	1
OMA BCAST Operation Mode Data Object	'01'	M	1
Key Identifier TLV	See below		

Table 84: Coding of Key Identifier TLV

Description	Value	M/O	Length (bytes)
Key Domain ID Tag	'81'	M	1
Length	3	M	1
Key Domain ID	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	3
SEK/PEK ID Key Group part Tag	'82'	M	1
Length	2	M	1
SEK/PEK ID Key Group part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2
SEK/PEK ID Key Number part Tag	'83'	C (NOTE1)	1
Length	2	C	1
SEK/PEK ID Key Number part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	C	2
Key Validity Data Tag	'84'	C (NOTE1)	1
Length	8	C	1
TS Low || TS High	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	C	8
Security Policy Extension Tag	'85'	C (NOTE1)	1
Length	1	C	1
Security Policy Extension	Coded as defined in 6.6.4.2	C	1
NOTE1: If one of the conditional TLVs SEK/PEK ID Key Number Part, Key Validity Data, Security Policy Extension is present, these three TLVs SHALL be present.			

If present, the TLV shall be present in the order defined in the table.

If the command (SPE deletion sub mode) is successful the response parameters and data, SHALL be coded as follows:

Table 85: Coding of Response Parameters and Data if SPE Deletion Sub Mode Command Successful

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2	MBMS operation response Data Object length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
3	OMA BCAST operation response Tag = 'AE'		1
4	OMA BCAST operation response Data Object Length		1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

5	BCAST management_data response Data Object tag ('80')		1
6	BCAST management_data response Data Object length (L=1)		1
7	BCAST management_data response Data Object	See Table 70	1
Note: The BCAST management_data Data Object value 0x00 is used to indicate successful operation			

If the command fails (SPE concerned by the SPE deletion is Used For Recording) the response parameters/data, SHALL be coded as follows:

Table 86: Coding of Response Parameters and Data if SPE Deletion Sub Mode Command Fails because SPE concerned is Used For Recording

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2	MBMS operation response Data Object length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
3	OMA BCAST operation response Tag = 'AE'		1
4	OMA BCAST operation response Data Object Length		1
5	BCAST management_data response Data Object tag ('80')		1
6	BCAST management_data response Data Object length (L=1)		1
7	BCAST management_data response Data Object	See Table 70	1
Note: The BCAST management_data Data Object value 0x0D is used to indicate that SPE is used for recording			

If the command fails (no available SPE stored in the Smartcard corresponding to the SPE defined in the command), the status word ‘6A88’ (Referenced Data not found) is returned.

E.2.3.2 MBMS Security Context – OMA BCAST Operation – Recording Deletion Mode

E.2.3.2.1 Recording Deletion Mode: Command Description

BCAST Smartcards MAY support the command described in this section.

The Smartcard receives from the terminal in the AUTHENTICATE command the Terminal/content Identifier corresponding to the content that has been erased in the terminal. The Smartcard SHALL delete the content identifier specified in the command and stored in the smartcard and SHALL delete the association of this content identifier to all SPEs that have been flagged at the recording of this content, during the execution of Record Signalling command.

If the content identifier defined in the command data doesn’t exist in the Smartcard, the Smartcard SHALL abandon the function and return the status word '6A88' (Referenced data not found).

Input:

- Terminal Identifier and Content Identifier

Output:

- List of SPEs for which the association has been deleted.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

E.2.3.2.2 Recording Deletion Mode: Parameters and Data

In case of Recording Deletion Mode (i.e. OMA BCAST Operation Mode Data Object value is '02'), the coding of OMA BCAST Operation TLV is as follows:

Table 87: Coding of OMA BCAST Operation TLV

Description	Value	M/O	Length (bytes)
OMA BCAST Operation TLV Tag	'AE'	M	1
OMA BCAST Operation TLV Data Object Length	L	M	B
OMA BCAST Operation Mode Tag	'90'	M	1
OMA BCAST Operation Mode Data Object Length	1	M	1
OMA BCAST Operation Mode Data Object	'02'	M	1
Terminal/Content Identifier TLV	See below	M	L1

Table 88: Coding of Terminal/Content Identifier TLV

Description	Value	M/O	Length (bytes)
Terminal Identifier Tag	'96'	M	1
Length	17	M	1
Terminal Identifier type	Coded as defined in 6.11.1	M	1
Terminal Identifier	Coded as defined in 6.11.1	M	16
Content Identifier Tag	'97'	M	1
Content Identifier Length	L1-B-21	M	B
Content Identifier Data	Terminal specific coding	M	L1-B-21

If the command (Recording deletion sub mode) is successful the response parameters/data, SHALL be coded as follows:

Table 89: Coding of Response Parameters and Data if Recording Deletion Sub Mode Command Successful

Byte(s)	Description	Coding	Length
1	MBMS operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2	MBMS operation response Data Object length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
3	OMA BCAST operation response Tag = 'AE'		1
4	OMA BCAST operation response Data Object Length		1
5	BCAST management_data response Data Object tag ('80')		1
6	BCAST management_data response Data Object length (L=1)		1
7	BCAST management_data response Data Object (NOTE1)	See Table 70	1
8 to 35	OMA BCAST Flagged SPE TLV (NOTE2)		28

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

NOTE1: The BCAST management_data Data Object value 0x00 is used to indicate successful operation
NOTE2: There are as many OMA BCAST Flagged_SPE TLV as SPE concerned by this deletion.

The OMA BCAST Flagged SPE describes the SPE that had been flagged by the smartcard during the execution of the Record Signalling command sent when the content corresponding to the Terminal/content Identifiers pair had been recorded. During the execution of this current Recording deletion command, association of these SPEs to the content Identifier is erased. There are as many OMA BCAST Flagged_SPE TLV as SPE concerned by this deletion.

Table 90: Coding of OMA BCAST Flagged_SPE TLV

Byte(s)	Description	M/O	Length
1	OMA BCAST Flagged_SPE tag = 'A8'	M	1
2	OMA BCAST Flagged_SPE Records length = '0x1A'	M	1
3 to 7	Key Domain ID TLV	M	5
8 to 11	SEK/PEK ID Key Group part TLV	M	4
12 to 15	SEK/PEK ID Key Number part TLV	M	4
16 to 25	Key Validity Data TLV	M	10
26 to 28	Security Policy Extension TLV	M	3

Table 91: Coding of Key Domain ID TLV

Description	Value	M/O	Length (bytes)
Key Domain ID Tag	'81'	M	1
Length	3	M	1
Key Domain ID	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	3

Table 92: Coding of SEK/PEK ID Key Group part TLV

Description	Value	M/O	Length (bytes)
SEK/PEK ID Key Group part Tag	'82'	M	1
Length	2	M	1
SEK/PEK ID Key Group part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2

Table 93: Coding of SEK/PEK ID Key number part TLV

Description	Value	M/O	Length (bytes)
SEK/PEK ID Key Number part Tag	'83'	M	1
Length	2	M	1
SEK/PEK ID Key Number part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2

Table 94: Coding of Key Validity Data TLV

Description	Value	M/O	Length (bytes)
Key Validity Data Tag	'84'	M	1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Length	8	M	1
TS Low || TS High	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	8

Table 95: Coding of Security policy extension TLV

Description	Value	M/O	Length (bytes)
Security Policy Extension Tag	'85'	M	1
Length	1	M	1
Security Policy Extension	Coded as defined in 6.6.4.2	M	1

If the command fails (no available Content Identifier stored in the Smartcard corresponding to the Content Identifier defined in the command), the status word '6A88' (Referenced Data not found) is returned.

E.3 OMA BCAST COMMAND

E.3.1 Description of the Command

A new command has been defined in 3GPP/ETSI for OMA BCAST Smartcard Profile. BCAST Smartcards SHALL support the command described in this section.

The command can be used in several modes:

⇒ SPE Audit mode

⇒ SPE Record Signalling mode

⇒ Recording Audit Mode

⇒ Event Signalling Mode

Other modes may be defined in the future, for future release of OMA BCAST specification.

SPE Audit Mode is used by the terminal to retrieve in the Smartcard the SEK/PEK ID Key group or all SPE instances corresponding to a specific SEK/PEK ID Key group.

Record Signalling Mode is used by the terminal to signal to the Smartcard the recording of content and to allow the Smartcard to retrieve the SEK/PEKs used for the protection of this recorded content. This command results in flagging the corresponding SPE instance in the Smartcard as a SPE that should not be deleted by the Key management system, and could be used for further playback of the content.

Recording Audit Mode is used by the terminal to retrieve all the recordings that have been signalled in the Smartcard and their associated flagged SPEs. How the OMA BCAST command can chain successive blocks of OMA BCAST data and OMA BCAST response data is specified below.

Event Signalling Mode is used by the terminal to inform the Smartcard of an event that can e.g. impact the parental control decision process to specifically cater for deployment scenarios where the Smartcard does not have all the information necessary to perform such decision on its own. For BCAST1.0, one event is defined, zapping event. Other events can be specified in a future release of this specification in order to address local regulatory requirements for other event signalling (e.g. loss of signal, terminal OMA BCAST application switch-off/switch-on).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 96: Coding of OMA BCAST Command

Code	Value
CLA	'8X' or 'CX' or 'EX'
INS	'1B'
P1	See Table 97 below
P2	See Table 98 below
Lc	If present, length of subsequent data field
Data	See below
Le	Length of expected response data

P1 Parameter

Parameter P1 is used to control the data exchange between the terminal and the Smartcard. The P1 values for this OMA BCAST command are defined in Table 97 below. (The 'X' bits represent 'Reserved for Future Use' and '-' represents 'don't care'.)

Table 97: Coding of P1

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
-	-	-	x	x	x	x	x	RFU
1	0	0	-	-	-	-	-	First block of data
0	0	0	-	-	-	-	-	Next block of data
1	0	1	-	-	-	-	-	First block of response data
0	0	1	-	-	-	-	-	Next block of response data
1	1	1	1	1	1	1	1	No input data

As defined in [3GPP TS 31.101 v7] for the AUTHENTICATE Command with an ODD INS code, this OMA BCAST Command can chain successive blocks of data, with a maximum size of 255 command and 256 response bytes each, required for one OMA BCAST operation using P1 to indicate the first/next block.

The terminal performs the segmentation of the data, and the UICC the concatenation of the data. The first OMA BCAST Command APDU is sent with P1 indicating "First block of data". Following OMA BCAST Command APDUs are sent with P1 indicating "Next block of data". As long as the UICC has not received all segments of the data it SHALL answer with SW1 SW2 '63 F1'. When all segments of the data are received, the UICC answers with SW1 SW2 '62 F3'.

The response data is retrieved from the UICC using one or more separate OMA BCAST APDUs with the same chaining mechanism as for the input data. The UICC performs the segmentation of the data, and the terminal the concatenation of the response data. The first OMA BCAST Command APDU is sent with P1 indicating "First block of response data". When the UICC receives this first OMA BCAST Command APDU with P1 indicating "First block of response data", it shall perform the command and calculate the response. Following OMA BCAST Command APDUs are sent with P1 indicating "Next block of response data". As long as the UICC has not sent all segments of the response data it shall answer with SW1 SW2 '62 F1'. When all segments of the response data are sent, the UICC shall answer with SW1 SW2 '90 00' or '91XX' if a pro-active command is pending.

If no command data is sent from the terminal to the Smartcard in the command, then P1 is set to 'FF'. In this case, if the smartcard has data to send back to the terminal after performing the command, the smartcard answer SW1 SW2 '62 F3' to signal to the terminal that there is data available. Then the terminal sends the same command with P1 indicating 'first block of response data' using the same chaining mechanism as described in the previous paragraph.

P2 Parameter

Parameter P2 specifies the Mode of the OMA BCAST Command as follows:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 98: Coding of the Reference Control P2

Coding b8-b1	Meaning
0x01	SPE Audit Mode
0x02	SPE Record Signalling Mode
0x03	Recording Audit Mode
0x04	Event Signalling Mode
0x05 -0xFF	RFU

Coding of Data is specified in the following sections for each mode of the command.

OMA BCAST Command Status Words

Status of the card after processing of the OMA BCAST command is coded in the status bytes SW1 and SW2. The coding of the status bytes in the following table is specified for OMA BCAST command, in addition to the ones defined in [3GPP TS 31.101 v7]. The following table shows the possible status conditions returned (marked by an asterisk *).

Table 99: OMA BCAST Command and Expected Status Words

Status Words	OMA BCAST command	Description
90 00	*	Normal ending of the command
91 XX	*	Normal ending of the command, with extra information from the proactive UICC containing a command for the terminal. Length 'XX' of the response data
93 00		
98 50		
98 62		
98 64		
98 65		
98 66	*	Authentication error, no available memory space
98 67		
62 00		
62 81		
62 82		
62 83		
62 F1	*	More data available
62 F2	*	More data available and proactive command pending
62 F3	*	Authentication response data available
63 CX		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

63 F1	*	More data expected
63 F2	*	More data expected and proactive command pending
64 00		
65 00		
65 81	*	Memory problem
67 00	*	Wrong length
67 XX		
68 00	*	No information given
68 81	*	Logical channel not supported
68 82	*	Secure messaging not supported
69 81		
69 82	*	Security status not satisfied
69 83		
69 84	*	Referenced data invalidated
69 85	*	Conditions of use not satisfied
69 86		
6A 80	*	Incorrect parameters in the data field
6A 81	*	Function not supported
6A 82		
6A 83		
6A 86	*	Incorrect parameters P1 to P2
6A 87		
6A 88	*	Referenced data not found
6B 00	*	Wrong parameter(s) P1-P2
6D 00	*	Instruction code not supported or invalid
6E 00	*	Class not supported
6F 00	*	Technical problem, no precise diagnosis

E.3.2 SPE Audit Mode

E.3.2.1 Description of the Command

BCAST Smartcards SHALL support the command described in this section.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The response data of the OMA BCAST command in SPE Audit command is retrieved from the Smartcard using one or more separate SPE Audit APDUs with the chaining mechanism described in Section E.3.1.

In case the terminal wants to retrieve all Key Domain IDs and corresponding SEK/PEK ID Key group parts stored in the Smartcard, the terminal SHALL sent a command without command data and indicate this by setting the parameter P1 to 'FF'.

In case the terminal wants to retrieve all SEK/PEK IDs and their associated parameters corresponding to a Key Group stored in the Smartcard, the terminal SHALL send a command containing the respective Key Domain ID with the related SEK/PEK ID Key Group part in the command data.

If at least one SEK/PEK ID shall be returned in the response, the "Successful Key Availability Check operation" tag is returned.

If there is no SEK/PEK ID to be returned in the response, the command fails and the status word '6A88' (Referenced data not found) is returned.

Some examples of chaining for the SPE Audit mode are given in Appendix K.

If P1 indicates "no input data"

Input:

- None

Output:

- None

If P1 indicates "First block of data"

Input:

- Key Domain ID and SEK/PEK ID Key Group part; (see Table 100 for the detail command data)

Output:

- None

If P1 indicates "First block of response data"

Input:

- None

Output:

- List of SPE and their associated data stored in the Smartcard or list of Key DomainID and SEK/PEK ID Key group part

If P1 indicates "Next block of response data"

Input:

- None

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Output:

- Remaining part of the List of SPE and their associated data stored in the Smartcard or remaining part of the list of Key DomainID and SEK/PEK ID Key group part

E.3.2.2 Command Parameters and Data

If the P2 parameter in the OMA BCAST Command is '01': SPE Audit mode, the command parameters shall be coded as follows:

See Section E.3.1 for the coding of P1 parameter.

If P1 indicates "First block of response data" or "Next block of response data" or "No input data", then Input Data field is absent.

If P1 indicates "First block of data", then the following coding holds:

Table 100: Coding when P1 indicates "First block of data"

Byte(s)	Description	Coding	M/O	Length
1	SPE Audit Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	M	1
2 to 1+A bytes (A ≤ 4)	SPE Audit Data Object length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	M	A
A+2 to A+6	Key Domain ID TLV	See below	M	5
A+7 to A+10	SEK/PEK ID Key Group part TLV	See below	M	4

If one of the TLVs Key Domain ID TLV or SEK/PEK ID Key Group part TLV is present, then the following two TLVs SHALL be present.

Table 101: Coding of Key Domain ID TLV

Description	Value	M/O	Length (bytes)
Key Domain ID Tag	'81'	M	1
Length	3	M	1
Key Domain ID	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	3

Table 102: Coding of SEK/PEK ID Key Group Part TLV

Description	Value	M/O	Length (bytes)
SEK/PEK ID Key Group part Tag	'82'	M	1
Length	2	M	1
SEK/PEK ID Key Group part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2

If the OMA BCAST command (SEK/PEK Audit mode) is successful the response parameters and data, SHALL be coded as follows:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 103: Coding of Response Parameters and Data if SEK/PEK Audit Mode Command Successful

Byte(s)	Description	Coding	Length
1	SPE Audit operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A bytes (A ≤ 4)	SPE Audit operation response Data Object length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2 to (A+2+L1)	OMA BCAST Key Group description TLV or OMA BCAST SPE description TLV	See below	L1

In case Key Domain ID TLV and SEK/PEK ID Key Group part TLV are absent in the SPE Audit command, the OMA BCAST Key Group description TLV is returned. One or more OMA BCAST Key Group description TLV are returned.

The OMA BCAST Key Group description TLV coding is as follow:

Table 104: Coding of OMA BCAST Key Group Description TLV

Byte(s)	Description	M/O	Length
1	OMA BCAST Key group description tag = 'A5'	M	1
2	OMA BCAST Key group description length	M	1
3 to 7	Key Domain ID TLV	M	5
8 to 11	SEK/PEK ID Key Group part TLV	M	4
12 to 17	User_Purse TLV	O	6
18 to 23	Live_PPT_Purse TLV	O	6
24 to 29	Playback_PPT_Purse TLV	O	6
30 to 34	Kept_TEK_Counter TLV	O	5

For the coding of Key Domain ID TLV and SEK/PEK ID Key Group part TLV, see above the tables of the input parameters.

User_Purse TLV, Live_PPT_Purse TLV, Playback_PPT_Purse TLV, and Kept_TEK_Counter TLV are present if they are present in the smartcard for this Key Group.

Table 105: Coding of User_Purse TLV

Description	Value	M/O	Length (bytes)
User_Purse Tag	'8A'	M	1
Length	4	M	1
User_Purse Value		M	4

User_Purse Value is the number of tokens contained in the User_Purse.

Table 106: Coding of Live_PPT_Purse TLV

Description	Value	M/O	Length (bytes)
Live_PPT_Purse Tag	'8B'	M	1
Length	4	M	1
Live_PPT_Purse Value		M	4

Live_PPT_Purse Value is the number of tokens contained in the Live_PPT_Purse associated to this Key Group.

Table 107: Coding of Playback_PPT_Purse TLV

Description	Value	M/O	Length (bytes)
Playback_PPT_Purse Tag	'8C'	M	1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Length	4	M	1
Playback _PPT_Purse Value		M	4

Playback_PPT_Purse Value is the number of token contained in the Playback_PPT_Purse associated to this Key Group.

Table 108: Coding of Kept_TEK_Counter TLV

Description	Value	M/O	Length (bytes)
Kept_TEK_Counter Tag	'8D'	M	1
Length	3	M	1
Number_TEK Value		M	3

Number_TEK Value is the number of TEK contained in the Kept_TEK_Counter associated to this Key Group.

In case Key Domain ID TLV and SEK/PEK ID Key Group part TLV are present in the SPE Audit command, The OMA BCAST SPE description TLV is returned. One or more OMA BCAST SPE description TLV are returned.

The OMA BCAST SPE description TLV coding is as follow:

Table 109: Coding of OMA BCAST SPE Description TLV

Byte(s)	Description	M/O	Length
1	OMA BCAST SPE description tag = 'A6'	M	1
2	OMA BCAST SPE description length	M	1
3 to 7	Key Domain ID TLV	M	5
8 to 11	SEK/PEK ID Key Group part TLV	M	4
12 to 15	SEK/PEK ID Key Number part TLV	M	4
16 to 25	Key Validity DataTLV	M	10
26 to 28	Key properties TLV	M	3
29 to 31	Security_policy_extension TLV	M	3
32 to 35	Cost_value TLV	C(NOTE)	4
36 to 38	Play_back counter TLV	C(NOTE)	3
39 to 44	User_Purse TLV	C(NOTE)	6
45 to 50	Live_PPT_Purse TLV	C(NOTE)	6
51 to 56	Playback_PPT_Purse TLV	C(NOTE)	6
57 to 61	Kept_TEK_Counter TLV	C(NOTE)	5
62 to 66	TEK_Counter TLV	C(NOTE)	5

NOTE:

If Security_policy_extension is 0x00, 0x01, 0x02, 0x03, 0x08 or 0x09, the Cost_value TLV byte SHALL be present.

If Security_policy_extension is 0x07, the Play back counter TLV byte SHALL be present.

If Security_policy_extension is 0x00, the Live_PPT_Purse TLV SHALL be present, and the value is the content value of the live_ppt_purse associated to the SEK/PEK ID Key Group part and SPE.

If Security_policy_extension is 0x01 the Playback_PPT_Purse TLV SHALL be present, and the value is the content value of the playback_ppt_purse associated to the SEK/PEK ID Key Group part and SPE.

If security_policy_extension is 0x02, 0x03, 0x08, 0x09, the User_Purse TLV SHALL be present, and the value is the content value of the user_purse associated to the NAF_ID part of the SMK_ID.

If security_policy_extension is 0x0C, 0x0D, the TEK_Counter TLV SHALL be present, and the value is the content value of the TEK_counter associated to the SEK/PEK ID and KV and SPE.

If security_policy_extension is 0x0C, the Kept_TEK_Counter TLV SHALL be present, and the value is the

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

content value of the Kept_TEK_counter associated to the SEK/PEK ID Key group part and SPE.

For the coding of KeyDomainID TLV, SEK/PEK ID Key group part TLV, User_Purse TLV, Live_PPT_purse TLV, Playback_PPT_purse, and Kept_TEK_counter TLV see above in the coding of OMA BCAST Key Group description TLV.

Table 110: Coding of SEK/PEK ID Key Number Part TLV

Description	Value	M/O	Length (bytes)
SEK/PEK ID Key Number part Tag	'83'	M	1
Length	2	M	1
SEK/PEK ID Key Number part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2

Table 111: Coding of Key Validity Data TLV

Description	Value	M/O	Length (bytes)
Key Validity Data Tag	'84'	M	1
Length	8	M	1
TS Low || TS High	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	8

Table 112: Coding of Key Properties TLV

Description	Value	M/O	Length (bytes)
Key Properties Tag	'93'	M	1
Length	1	M	1
Key properties	See below	M	1

The Key properties byte shall be coded as follows:

Table 113: Coding of Key Properties Byte

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
X	X	X	X	X	X	X		RFU
-	-	-	-	-	-	-	0	Not used For Recording
-	-	-	-	-	-	-	1	Used For Recording

Table 114: Coding of Security Policy Extension TLV

Description	Value	M/O	Length (bytes)
Security Policy Extension Tag	'85'	M	1
Length	1	M	1
Security Policy Extension	Coded as defined in 6.6.4.2	M	1

Table 115: Coding of Cost Value TLV

Description	Value	M/O	Length (bytes)
Cost value Tag	'91'	M	1
Length	2	M	1

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Cost value		M	2

Table 116: Coding of Playback counter TLV

Description	Value	M/O	Length (bytes)
Playback counter Tag	'92'	M	1
Length	1	M	1
Playback counter Value		M	1

Table 117: Coding of TEK_counter TLV

Description	Value	M/O	Length (bytes)
TEK_counter Tag	'8E'	M	1
Length	3	M	1
TEK_counter Value		M	3

E.3.3 Record Signalling Mode

E.3.3.1 Description of the Command

BCAST Smartcards MAY support the command described in this section.

The terminal SHALL send this command to the Smartcard when it records or stores protected content using the mechanisms described in Section 8.1 and Section 8.3. The terminal SHALL send a command for each SEK/PEK Key number part covered by the recording.

At the reception of this command, the Smartcard SHALL flag the SPE instances corresponding to the Key Domain ID, SEK/PEK ID Key group part, SEK/PEK ID Key number part and TS interval found in the command and corresponding to the recorded/stored content. The SPEs flagged in the Smartcard SHALL be associated to a security policy extension allowing the play-back.

The Smartcard stores the content_ID received in the command and links it to the flagged SPEs.

The Smartcard SHALL return (in the response data) the number of SPE records available for SPEs instance required for the playback of recorded/stored content after the execution of this command and the description of the SPEs flagged internally in the Smartcard.

In the case that the Smartcard doesn't store a SPE instance corresponding to the parameters in the command or the security policy extension associated to the SEK/PEK doesn't allow the play-back recorded/stored content, the command fails and the status word '6A88' (Referenced data not found) is returned.

In case a part of the recorded content described in the input parameters of the command is not covered by an SPE in the Smartcard, the command fails, the status word '6A88' (Referenced data not found) is returned, and none of the SPE is flagged as a SPE Used For Recording.

In the case that there is no available SPE record for a SEK/PEK required for the playback of recorded/stored content, the command fails and the status word '9866' (Authentication error, no available memory space) is returned.

The chaining mechanism used for this command is described in Section E.3.1.

Input:

- Terminal and content identifier
- Key Domain ID, SEK/PEK ID Key Group part, SEK/PEK ID Key number part , TS Interval

Output:

- Key slot giving the free key slots available for recorded-content keys.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

- SPE description of the SPEs Flagged in the Smartcard corresponding to the input data.

Some examples of chaining for the Record Signalling Mode are given in Appendix K.

E.3.3.2 Command Parameters and Data

If the P2 parameter of the OMA BCASTCommand is '02': Record Signalling mode, the command parameters SHALL be coded as follows:

Input data SHALL be coded as follows:

Table 118: Input Data

Byte(s)	Description	Coding	Length
1	Record Signalling Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2 to 1+A bytes (A ≤ 4)	Record Signalling Data Object length (L1+L2)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	A
A+2 to (A+2+L1)	Terminal/Content Identifier TLV	See below	L1
(A+2+L1)+1 to (A+2+L1) + L2	Key Identifier of Recording TLV	See below	L2

Table 119: Coding of Terminal/Content Identifier TLV

Description	Value	M/O	Length (bytes)
Terminal Identifier Tag	'96'	M	1
Length	17	M	1
Terminal Identifier type	Coded as defined in 6.11.1	M	1
Terminal Identifier	Coded as defined in 6.11.1	M	16
Content Identifier Tag	'97'	M	1
Content Identifier Length	L1-B-21	M	B
Content Identifier Data	Terminal specific coding	M	L1-B-21

Table 120: Coding of Key Identifier of Recording TLV

Description	Value	M/O	Length (bytes)
Key Domain ID Tag	'81'	M	1
Length	3	M	1
Key Domain ID	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	3
SEK/PEK ID Key Group part Tag	'82'	M	1
Length	2	M	1
SEK/PEK ID Key Group part	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	2
SEK/PEK ID Key Number part Tag	'83'	M	1
Length	2	M	1
SEK/PEK ID Key Number part	Coded as defined in 3GPP TS 33.246 [3GPP	M	2

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

	TS 33.246 v7]		
TS Interval Tag	'94'	M	1
Length	8	M	1
TS start_recording	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	4
TS end_recording	Coded as defined in 3GPP TS 33.246 [3GPP TS 33.246 v7]	M	4

If the command (Record Signalling mode) is successful, the response parameters/data SHALL be coded as follows:

Table 121: Coding of Response Parameters and Data if Record Signalling Mode Command Successful

Byte(s)	Description	Coding	Length
1	Record Signalling operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2	Record Signalling operation response Data Object length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
3 to 6	OMA BCAST SPE Records TLV		4
7 to 34	OMA BCAST Flagged_SPE TLV		28
The Record signalling operation response SHALL contain as many OMA BCAST Flagged_SPE TLVs as SPEs that have been flagged internally by the Smartcard.			

Table 122: Coding of OMA BCAST SPE Records TLV

Byte(s)	Description	M/O	Length
1	OMA BCAST SPE Records tag = '88'	M	1
2	OMA BCAST SPE Records length = '2'	M	1
3 to 4	Available SPE records	M	2

The 'Available SPE records' field SHALL be binary coded and SHALL indicate the number of empty SPE records available for the storage of SEKs/PEKs required for the playback of recorded content, e.g. with their UsedForRecording Flag set, after the execution of this command

See Section E.2.3.2.2 for the coding of the OMA BCAST Flagged_SPE TLV.

The OMA BCAST Flagged SPE describes the SPE flagged by the smartcard corresponding to the input data of the Record Signalling command. There are as many OMA BCAST Flagged_SPE TLVs as SPEs flagged internally in the Smartcard.

E.3.4 Recording Audit Mode

E.3.4.1 Description of the Command

BCAST Smartcards MAY support the command described in this section.

The terminal SHALL send this command to the Smartcard to retrieve all Content Identifiers of recorded content and their associated flagged SPEs stored in the Smartcard.

The response data of the OMA BCAST command in Recording Audit command is retrieved from the Smartcard using one or more separate Recording Audit APDUs with the chaining mechanism described in Section E.3.1.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

At the reception of this command, the Smartcard SHALL return (in the response data) the list of content identifiers stored in the Smartcard with the description of all associated SPEs flagged at the time of the execution of the Record Signalling command sent during the recording of the corresponding content in the terminal.

In the case that the Smartcard doesn't store any content identifier the command fails and the status word '6A88' (Referenced data not found) is returned.

Input:

- None

Output:

- List of Content Identifiers with SPE description of the SPEs Flagged in the Smartcard and corresponding to the content identified by the content identifier.

Some examples of chaining for the Recording Audit Mode are given in Appendix K.

E.3.4.2 Command Parameters and Data

If the P2 parameter in the OMA BCAST Command is '03': Recording Audit mode, the command parameters shall be coded as follows:

Only P1 indicating "No input data", "First block of response data", and "Next block of response data" are applicable.

If the command (Recording Audit mode) is successful, the response parameters/data SHALL be coded as follows:

Table 123: Coding of Response Parameters and Data if Recording Audit Mode Command Successful

Byte(s)	Description	Coding	Length
1	Recording Audit operation response Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
2	Recording Audit operation response Data Object length	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	1
	Recording Audit operation response Data Object TLV	See below	L1
The Recording Audit signalling operation response SHALL contain as many Recording Audit operation response Data object TLV as content Identifiers stored internally in the Smartcard.			

Table 124: Coding of Recording Audit operation response Data Object

Description	Value	M/O	Length (bytes)
Recording Audit operation response Data object Tag	'A7'	M	1
Recording Audit operation response Data object Length	L	M	1
Terminal Identifier Tag	'96'	M	1
Terminal Identifier Length	17	M	1
Terminal Identifier type	Coded as defined in 6.11.1	M	1
Terminal Identifier	Coded as defined in 6.11.1	M	16
Content Identifier Tag	'97'	M	1
Content Identifier Length	L2	M	B
Content Identifier Data	Terminal specific coding	M	L2
OMA BCAST Flagged_SPE TLV	See below	M	28

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The Recording Audit operation response SHALL contain as many OMA BCAST Flagged_SPE TLVs as SPEs that have been flagged internally by the Smartcard for the content specified by the Terminal Identifier/Content Identifier pair.

See Section E.2.3.2.2 for the coding of the OMA BCAST Flagged_SPE TLV.

The OMA BCAST Flagged SPE describes the SPE flagged by the smartcard for the Content specified by its content identifier. There are as many OMA BCAST Flagged_SPE TLVs as SPEs flagged internally in the Smartcard for this content.

E.3.5 Event Signalling Mode

E.3.5.1 Description of the Command

BCAST Smartcards MAY support the command described in this section if the parental control is supported by the Smartcard.

The command "Event Signalling Mode" signals to the Smartcard that a specific event occurred.

Input:

- Event type

Output:

- none

An example of use of this Event Signalling Mode is given in Appendix K.

E.3.5.2 Command Parameters and Data

If the P2 parameter in the OMA BCAST Command is '04': "Event Signalling Mode", the command parameters shall be coded as follows:

See Section E.3.1 for the coding of P1 parameter.

As the input parameter is always present for this command, and no output is returned from the Smartcard, the P1 parameter will always take the value "first block of data" and the following coding holds:

Table 125: Coding input data

Byte(s)	Description	Coding	M/O	Length
1	Event Signalling Data Object tag ('73')	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	M	1
2 to 1+A bytes (A ≤ 4)	Event Signalling Data Object length (L1)	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	M	A
A+2 to A+6	Event Type TLV (Note1)	See below	M	3
A+7 to A+7+L2	Event Type Parameter TLVs (Note2)	See below	O	L2
Note1: Only one Event Type TLV (and associated Event Type Parameter TLVs, if any) is allowed in the command Note2: This TLV is present if the Event Type has parameters. There can be more than one Event Type Parameter TLV per Event Type TLV.				

Additional Event Type TLVs MAY be defined in the future. The Smartcard SHALL ignore data objects that it does not recognise.

Table 126: Coding of Event Type TLV

Description	Value	M/O	Length (bytes)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Event Type Tag	'8F'	M	1
Length	1	M	1
Event Type	See below	M	1

Table 127: Coding of Event Type Byte

Value	Description
0x00	Zapping
0x01 – 0x7F	Reserved for future use
0x80 – 0xFF	Reserved for specific event proprietary signalling (e.g. due to local regulations on parental control)

Table 128: Coding of Event Type Parameter TLV

Description	Value	M/O	coding	Length (bytes)
Event Type Parameter Tag	'95'	M		1
Length	L3	M	As defined in TS 31.101 [3GPP TS 31.101 v7] for BER-TLV data object	B
Event Type Parameter	Specific to each event type	M	Specific to each event type	L3

Note: For the Event type 0x00 (Zapping) no Event Type Parameter is defined in BCAST1.0. As a result, the Event Type Parameter TLV is absent for this event in BCAST1.0.

E.4 OMA BCAST DF – DF BCAST

A BCAST Dedicated File (DF) has been defined under the ADF_USIM with the identifier ‘5F80’.

E.5 OMA BCAST ADF - ADF_BSIM

This section defines the OMA BCAST application, hereafter referred to as the BSIM, that MAY reside on the Smartcard for the purpose of BCAST key management.

E.5.1 Structure of the BSIM Application IDentifier (AID)

In accordance with [ISO/IEC 7816-4], the BSIM AID has the following structure:

<---------- Application IDentifier (AID) ---------->	
Registered application provider IDentifier (RID)	Proprietary application Identifier eXtension (PIX)
<---------- 5 bytes ---------->	<---------- ≤ 11 bytes ---------->

The BSIM AID consists of a Registered application provider IDentifier (RID) of 5 bytes and a Proprietary application Identifier eXtension (PIX) of up to 11 bytes.

The RID assigned by ISO/IEC to Open Mobile Alliance is 'A000000412'.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

The PIX, defined by OMA has the following structure:

<---------- Proprietary application Identifier eXtension (PIX) ---------->	
Application Code	Application Specific PIX Data
<---------- 2 bytes ---------->	<---------- 5bytes ≤ length ≤ 9 bytes ----------→

Application Code (Digits 1 to 4 of PIX)

This field uniquely identifies the application and is maintained by OMNA. The Application code assigned by OMNA to BSIM is 0x0001.

Application Specific PIX Data

This field carries application specific additional PIX data and is defined for BSIM in this specification in the following section.

E.5.1.1 BSIM Specific PIX Data

<---------- BSIM Specific PIX Data ---------->			
Country Code	Application Provider Code	Specification Version	Application Provider Specific Data Field optional
<----- 2 bytes ---->	<---------- 3 bytes ---------->	<------ 2 bytes ------>	<----- 2bytes -------→

Country Code (Digits 5 to 8 of PIX)

Purpose: indicates the country of the application provider of the BSIM application.

Management: List of actual country codes is published by ITU.

Coding: According to [ITU-T Recommendation E.164]. The coding is right justified and padded with 'F' on the left.

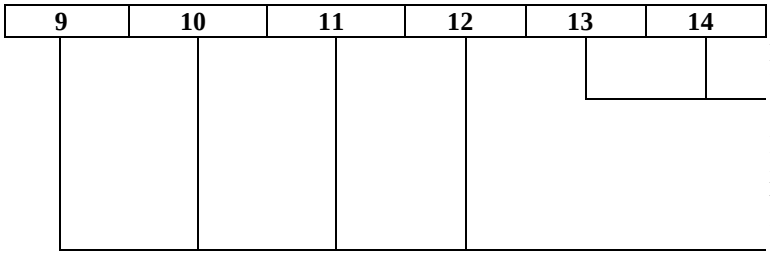
Application Provider Code (Digits 9 to 14 of PIX)

Purpose: Individual code for the application provider of the OMA standardized application.

Management: ITU

Coding: According to [ITU-T Recommendation E.118]. Hexadecimal. The coding is right justified and padded with 'F' on the left. As defined below.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.



Issuer Identifier Number (IIN) and Major Industry Identifier (MII) are coded in line with [ITU-T Recommendation E.118]. Application providers SHALL register at the ITU to get an Issuer Identifier Number.

Specification Version (Digits 15 to 18 of PIX)

Purpose: Indicates the version of the BCAST specification where the BSIM application is defined.

Management: OMA BCAST technical group

Coding: coded in BCD, refer to the specification version xx.yy The coding of xx and yy is right justified and padded with '0' on the left, i.e. for BCAST1.0: ' 01 00'

Application Provider Specific Data Field (Digits 19 to 22 of PIX)

Purpose: The use of this field is entirely up to the application provider and is optional. It may, for instance, be used to indicate "local" versions, revisions, etc. of the BSIM application. According to [ISO/IEC 7816-4], if this field is 2 bytes long, then the value 'FF' for the least significant byte is reserved for future use.

Management: Application provider.

Coding: Hexadecimal.

E.5.2 Contents of the EFs at the MF Level

There are four application independent files defined at the MF level: EF_{DIR}, EF_{ICCID}, EF_{PL} and EF_{ARR}. These EFs SHALL be as defined in [ETSI TS 102 221].

E.5.3 Contents of the ADF_BSIM

ADF_BSIM SHALL be selected using the AID and information in EF_DIR. The following figure depicts the file structure of the ADF_BSIM:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

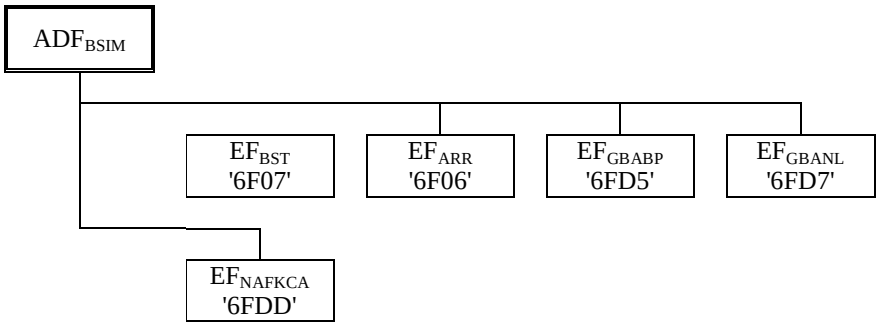


Figure 25 – File Identifiers and Directory Structures of BSIM

The files IDs '6F1X' (for EFs), '5F1X' and '5F2X' (for DFs) with X ranging from '0' to 'F' are reserved under the ADF_BSIM for administrative use by the card issuer.

E.5.3.1 EF_ARR (Access Rule Reference)

This EF contains the access rules for files located under the ADF_BSIM in the UICC. If the security attribute tag '8B' is indicated in the FCP it contains a reference to a record in this file.

Structure of EF_ARR at ADF-level

Identifier: '6F06'	Structure: Linear fixed	Mandatory
SFI: '06'		
Record Length: X bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	Access Rule TLV data objects	M	X bytes

This EF contains one or more records containing access rule information according to the reference to expanded format as defined in [ISO/IEC 7816-4]. Each record represents an access rule. Unused bytes in the record SHALL be set to 'FF'.

If the card cannot access EF_ARR, any attempt to access a file with access rules indicated in this EF_ARR SHALL NOT be granted.

E.5.3.2 EF_BST (BSIM Service Table)

This EF indicates which optional services are available. If a service is not indicated as available in the BSIM, the ME shall not select this service. The presence of this file is mandatory.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Identifier: '6F07'	Structure: transparent	Mandatory
SFI: '07'		
File size: X bytes, X >= 1	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Services n°1 to n°8	M	1 byte
2	Services n°9 to n°16	O	1 byte
3	Services n°17 to n°24	O	1 byte
4	Services n°25 to n°32	O	1 byte
etc.			
X	Services n°(8X-7) to n°(8X)	O	1 byte

Services Contents:

Service n°1	GBA-based Local Key Establishment Mechanism
Service n°2	Parental Control
Service n°3	Location based restriction
Service n°4	Support of SPE=0x00
Service n°5	Support of SPE=0x01
Service n°6	Support of SPE=0x02
Service n°7	Support of SPE=0x03
Service n°8	Support of SPE=0x05
Service n°9	Support of SPE=0x06
Service n°10	Support of SPE=0x07
Service n°11	Support of SPE=0x08
Service n°12	Support of SPE=0x09
Service n°13	Support of SPE=0x0A
Service n°14	Support of SPE=0x0B
Service n°15	Support of SPE=0x0C
Service n°16	Support of SPE=0x0D

Note: SPE 0x04 is not included in the table as it is mandatory to support for BCAST Smartcards and therefore the BSIM.

The EF shall contain at least one byte. Further bytes may be included, but if the EF includes an optional byte, then it is mandatory for the EF to also contain all bytes before that byte. Other services are possible in the future and will be coded on further bytes in the EF. The coding falls under the responsibility of OMA.

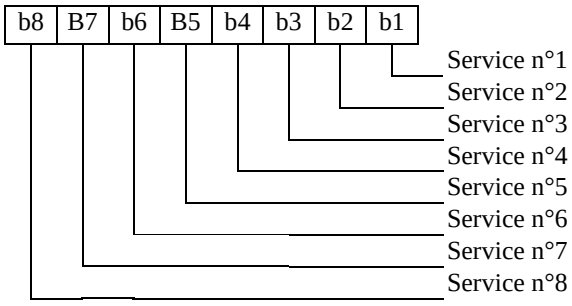
Coding:

1 bit is used to code each service:
bit = 1: service available;
bit = 0: service not available.

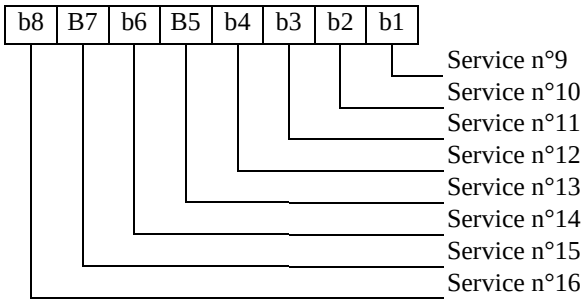
- Service available means that the BSIM has the capability to support the service and that the service is available for the user of the BSIM.
  Service not available means that the service shall not be used by the BSIM user, even if the BSIM has the capability to support the service.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

First byte:



Second byte:



etc.

E.5.3.3 EF_GBABP; ID = 6FD5

This EF contains the AKA Random challenge (RAND) and Bootstrapping Transaction Identifier (B-TID) associated with a GBA bootstrapping procedure.

The EF is a linked file which is mapped into the BSIM from the associated GBA serving application (e.g. USIM). Therefore file attributes and file content are identical to those of the EF with the same name of the associated application. For the definition of the EF see [3GPP TS 31.102 v7].

E.5.3.4 EF_GBANL; ID = 6FD7

This EF contains the list of NAF_ID and B-TID associated to a GBA NAF derivation procedure.

The EF is a linked file which is mapped into the BSIM from the associated GBA serving application (e.g. USIM). Therefore file attributes and file content are identical to those of the EF with the same name of the associated application. For the definition of the EF see [3GPP TS 31.102 v7].

E.5.3.5 EF_NAFKCA; ID = 6FDD

If service n°1 is "available", this file SHALL be present.

This EF contains one or more NAF Key Centre addresses. The first record in the EF shall be considered to be of the highest priority. The last record in the EF shall be considered to be the lowest priority.

The EF is a linked file which is mapped into the BSIM from the associated Local Key Establishment serving application (e.g. USIM). Therefore file attributes and file content are identical to those of the EF with the same name of the associated application. For the definition of the EF see [3GPP TS 31.102 v7].

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

E.5.4 BSIM Management Procedures

The requirements stated in [ETSI TS 102 221] SHALL apply to the BSIM application.

E.5.4.1 GBA and Local Key Establishment-related Procedures

The Terminal SHALL process all BCAST related GBA and Local Key Establishment commands, i.e. the AUTHENTICATE command with the GBA security context and the Local Key Establishment security context as defined in [3GPP TS 31.102 v7], and also the related EF READ and UPDATE commands as defined in [3GPP TS 31.101 v7] (referencing [ETSI TS 102 221]) within the BSIM ADF.

The BSIM SHALL be associated to a GBA and Local Key Establishment serving application on the UICC (e.g. a USIM in the case of a 3GPP smartcard) during application installation. When the BSIM receives a GBA or Local Key Establishment command from the terminal it SHALL forward the command to the associated application for processing and return the response received back to the Terminal.

In the case that more than one BSIM or USIM are present on the Smartcard it SHALL be possible to link multiple BSIMs with a single USIM application. It SHALL NOT be possible to link a single BSIM to more than one USIM.

E.5.4.2 BSIM Application Selection

If the Terminal wants to engage in BCAST operation, then after Smartcard activation (see [ETSI TS 102 221]), the Terminal SHALL select a BSIM application, if a BSIM application is listed in the EF_{DIR} file, using the SELECT by DF name as defined in [ETSI TS 102 221].

After a successful BSIM application selection, the selected BSIM (AID) SHALL be stored on the Smartcard. This application is referred to as the last selected BSIM application. The last selected BSIM application SHALL be available on the Smartcard after a deactivation followed by an activation of the Smartcard.

If a BSIM application is selected using partial DF name, the partial DF name supplied in the command SHALL uniquely identify a BSIM application. Furthermore, if a BSIM application is selected using a partial DF name as specified in [ETSI TS 102 221] indicating in the SELECT command the last occurrence, the Smartcard SHALL select the BSIM application stored as the last BSIM application. If, in the SELECT command, the options first, next/previous are indicated, they have no meaning if an application has not been previously selected in the same session and SHALL return an appropriate error code.

E.5.4.3 BSIM Application Initialisation

The BSIM SHALL NOT indicate any language preference. The terminal SHALL use the language indicated by any other application currently active on the Smartcard or choose a language from EF_{PL} at the MF level according the procedure defined in [ETSI TS 102 221].

If the terminal does not support the languages of EF_{PL}, then the Terminal SHALL use its own internal default selection.

Then the terminal runs the user verification procedure. If the procedure is not performed successfully, the BSIM initialisation stops.

After the previous procedures have been completed successfully, the terminal SHALL run the BSIM Service Table request procedure.

After the BSIM initialization has been completed successfully in the terminal, the terminal is ready for a BSIM session and SHALL indicate this to the BSIM by sending the corresponding STATUS command.

E.5.4.4 BSIM Session Termination

Note: This procedure is not to be confused with the deactivation procedure in [ETSI TS 102 221].

The BSIM session is terminated by the Terminal as follows:

- The Terminal SHALL indicate to the BSIM that the termination procedure is starting by sending the corresponding STATUS command (as defined in [ETSI TS 102 221]).

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- To actually terminate the session, the Terminal SHALL then use one of the mechanisms described in [ETSI TS 102 221].

E.5.4.5 BSIM Application Closure

After termination of the BSIM session, the BSIM application MAY be closed by closing the logical channels that are used to communicate with this particular BSIM application.

E.5.5 User Verification and File Access Conditions

This section gives information related to security features supported by the BSIM with respect to user verification and file access conditions.

The security architecture as defined in [ETSI TS 102 221] applies to the BSIM application with the following definitions and additions:

- The BSIM application SHALL use a global key reference as application PIN. This global PIN reference MAY be shared with the global Key reference used as application PIN for another application. In this case the BSIM and the application are seen as one application from the security context point of view.

  In order to avoid multiple entry of the PIN by the user, if the BSIM shares the application PIN with another application (signalled by the FCP in the response of the SELECT COMMAND or STATUS COMMAND), the terminal SHOULD NOT ask the user to verify the BSIM PIN. The terminal MAY check that the PIN is already verified by reading a file protected by the application PIN (e.g. EF_{BST}).If the access to the file terminates successfully, the application PIN is verified and the terminal MAY NOT proceed to the user verification procedure. If the access to the file returns an error 'security status not satisfied', then the user verification procedure MAY be executed.

- The only valid values for the usage qualifier is '08' (use the PIN for verification – Key reference data user knowledge based) as defined in [ISO/IEC 7816-4];

- The verification of the Parental Control PIN is allowed only after reception of the response of the AUTHENTICATE command in MTK generation mode where the operation status code is "PINCODE required" as specified in Section 6.7.3.11.1;

- A BSIM application may reside on either a single-verification capable UICC or a multi-verification capable UICC. If BSIM application resides on a single-verification capable UICC, the BSIM application SHALL share the application PIN with another application.

- Every file related to a BSIM application SHALL have a reference to an access rule stored in EF_{ARR} under BSIM.

The coding of the PIN and UNBLOCK PIN and security architecture as defined in [3GPP TS 31.101 v7] for 3GPP applications applies to the BSIM.

The security architecture as defined in [3GPP TS 31.101 v7] for terminals supporting 3GPP applications applies to terminals supporting the BSIM.

E.6 How to support a BSIM and USIM/CSIM Implementing BCAST Functionality on the same Smartcard

As described in Appendix E.1, BCAST Smartcard Functionality can be implemented as a dedicated application on the Smartcard – the BSIM (see Appendix E.5), or as part of a USIM application (see Appendix E.4).

BCAST 1.0 mandates Terminal and Smartcard support for implementations based on the USIM application. Support for the BSIM is optional for both Terminals and Smartcards. Terminals and Smartcards MAY support both implementations.

To address the case in which a Smartcard supports both implementations and is used with a new Terminal, resulting in the addition or removal of Terminal support for the selection of the BSIM, the Smartcard SHALL ensure that the BCAST data

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

used by one Smartcard application is synchronised with the data used by the other application, e.g. the data accessed by the BSIM and USIM is the same. How this is achieved is implementation specific.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix F. MIME Type Registrations

F.1 MIME Type Registration Request for application/vnd.oma.bcast.stkm

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name: application

Subtype name: vnd.oma.bcast.stkm

Required parameters:

streamid:

Unique positive integer identifying a particular key stream. Numbers are unique within a particular SDP session i.e. no global numbering is required. Used to indicate which media stream is protected by the actual STKM stream, by referencing to the value given by the “streamid” parameter by the “a=stkmstream” attribute in the according media section of the SDP file.

kmstype:

String identifying the Key Management System used. The list of allowed values can be found in the specification referenced under “Published specification”.

Optional parameters:

serviceproviders:

String identifying the service providers using the key stream, by referencing one or more BSMSelectors as declared in the Service Guide Delivery Descriptor in the BCAST Service Guide or one or more <X>/ServiceProvider values defined in the BCAST Management Object. This parameter carries a list of BSMSelector IDs or ServiceProvider values, separated by the character “|”. See alsoTable 53 in the Published Specification.

baseCID:

Part of the Service or Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object. This optional parameter is used in conjunction with the OMA BCAST DRM Profile for Service and Content protection.

srvCIDExt:

Part of the Service CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object; allowing to distinguish between multiple STKM streams for the same encrypted media stream. This optional parameter of type unsigned byte provides the value of the most significant byte of the service_CID_extension in the corresponding STKM stream. It is used in conjunction with the OMA BCAST DRM Profile for Service and Content protection.

prgCIDExt:

Part of the Program CID used to identify the corresponding asset within an OMA DRM 2.0 Rights Object; allowing to distinguish between multiple DRM Profile STKM streams for the same encrypted media stream. This optional parameter of type unsigned byte provides the value of the most significant byte of the program_CID_extension in the corresponding STKM stream. It is used in conjunction with the OMA BCAST DRM Profile for Service and Content protection.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

srvKEYList:

List of one or more so-called srvKEY values, separated by the "|" character and encoded using base64 encoding. Each of those "srvKEY" values corresponds to the concatenation of the Key Domain ID with the Key Group part of a SEK/PEK (ServiceEncryptionKey/ProgramEncryptionKey), which applies to the related STKM stream. It is used in conjunction with the OMA BCAST Smartcard Profile for Service and Content protection.

Encoding considerations: binary

Security considerations:

Short Term Key Messages carry encrypted key material to decrypt live streaming services. They are passive, meaning they do not contain executable or active content which may represent a security threat. The messages contain confidential and security critical fields. Encryption and authentication of such critical fields are provided by the type itself and does not have to be provided externally.

Interoperability considerations:

This content type carries BCAST Short Term Key Messages within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – Service and Content Protection for Mobile Broadcast Services, especially section 10.1.2, available from http://www.openmobilealliance.org

Applications, which use this media type:

OMA BCAST Services

Additional information:

Magic number(s): none

File extension(s): none

Macintosh File Type Code(s): none

Person & email address to contact for further information:

Uwe Rauschenbach

Uwe.Rauschenbach@nsn.com

Intended usage: Limited use.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Only for usage with OMA BCAST Services, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

F.2 MIME Type Registration Request for application/vnd.oma.bcast.ltkm

This section provides the registration request, as per [RFC 2048], to be submitted to IANA.

Type name: application

Subtype name: vnd.oma.bcast.ltkm

Required parameters:

kmstype:

String identifying the Key Management System used. The list of allowed values can be found in the specification referenced under “Published specification”.

Optional parameters:

serviceproviders:

String identifying the service providers using the key stream, by referencing one or more BSMSelectors as declared in the Service Guide Delivery Descriptor in the BCAST Service Guide or one or more <X>/ServiceProvider values defined in the BCAST Management Object. This parameter carries a list of BSMSelector IDs or ServiceProvider values, separated by the character “|”. See also Table 53 in the Published Specification.

Encoding considerations: binary

Security considerations:

Long Term Key Messages carry encrypted key material to decrypt live streaming services. They are passive, meaning they do not contain executable or active content which may represent a security threat. The messages contain confidential and security critical fields. Encryption and authentication of such critical fields are provided by the type itself and does not have to be provided externally.

Interoperability considerations:

This content type carries BCAST Long Term Key Messages within the scope of the OMA BCAST enabler. The OMA BCAST enabler specification includes static conformance requirements and interoperability test cases for this content.

Published specification:

OMA BCAST 1.0 Enabler Specification – Service and Content Protection for Mobile Broadcast Services, especially Section 10.1.4. Available from http://www.openmobilealliance.org

Applications, which use this media type:

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

OMA BCAST Services

Additional information:

Magic number(s):	none
File extension(s):	none
Macintosh File Type Code(s):	none

Person & email address to contact for further information:

Uwe Rauschenbach

Uwe.Rauschenbach@nsn.com

Intended usage: Limited use.

Only for usage with OMA BCAST Services, which meet the semantics given in the mentioned specification.

Author/Change controller: OMNA – Open Mobile Naming Authority, OMA-OMNA@mail.openmobilealliance.org

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix G. BCAST Compatibility with MBMS Smartcards (Informative)

The Smartcard Profile builds on the MBMS Security Specification [3GPP TS 33.246 v7], adding a number of features and capabilities, e.g. the ability to use IPsec and ISMACryp, support for multiple business models through the use of Security Policy Extensions, etc.

When specifying these additions the intention was that it should be possible for MBMS Smartcards to be used within a BCAST system. This section provides clarification of how BCAST 1.0 enables the use of MBMS only Smartcards.

G.1 Different LTKM Formats

BCAST 1.0 introduces two significant changes to the format of the MSK message defined by [3GPP TS 33.246 v7]:

- To carry BCAST specific information a new MIKEY General Extension Payload is defined - the EXT BCAST (see Section 6.6.4). The use of the EXT BCAST payload is dependent on the type of Smartcard being addressed (see below);
- When the EXT BCAST indicates the use of a Security Policy Extension, the Key Validity (KV) data will be formatted as an interval of 32 bit Timestamps rather than 16 bit TEK IDs.

As defined in [3GPP TS 33.246 v7], an MBMS only Smartcard will ignore the EXT BCAST payload. This allows for the use of the EXT BCAST payload in LTKMs sent to MBMS only Smartcards as the payload may include information that can be used by the Terminal, for example information relating to the Terminal Binding Key (TBK). However, MBMS only Smartcards should reject LTKMs with KV data formatted as an interval of Timestamps as the data will be twice as long as expected (note however that this error case is not addressed explicitly in [3GPP TS 31.102 v7]).

A BCAST Smartcard will support MBMS processing and can therefore process LTKMs with or without the EXT BCAST payload and with either type of KV data.

G.2 SEK/MSK Storage, Management and Use

An MBMS only Smartcard will store a SEK/MSK from a successfully processed LTKM in an internal Key Store and record the respective reference data in the file EF_{MSK}. When an MBMS only Smartcard processes a BCAST STKM or MBMS MTK message it looks for the relevant SEK/MSK reference in the file EF_{MSK} to retrieve the key.

A BCAST only Smartcard will store a SEK/MSK including the related reference data in the BCAST Key Store.

In the following the terms MBMS Key Store and BCAST Key Store refer to both, the storage of the keys and the related reference data, which in the case of MBMS is maintained in $EF_{MSK.}$

If an MBMS only Smartcard is used both as part of a Smartcard Profile implementation and as part of a parallel MBMS Security implementation, SEKs and MSKs will be stored in the same Key Store. The broadcast service provider must therefore ensure careful management of SEK/MSK IDs to avoid ID collisions, which could result in keys being incorrectly deleted or rejected.

A BCAST Smartcard cannot determine whether an LTKM has originated from a BCAST or MBMS system. This is due to re-use of MBMS commands by BCAST to forward the LTKM from the Terminal to the Smartcard (see Appendix E). This presents a problem: The BCAST Smartcard must process all LTKMs with the same format in the same way and therefore when it receives an LTKM without an EXT BCAST payload it will store the SEK/MSK in the MBMS Key Store and when it receives an LTKM with an EXT BCAST payload it will store the SEK in the BCAST Key Store. It follows that the MSKs for use within an MBMS system will always be stored in the MBMS Key Store (they are always sent in an LTKM without an EXT BCAST payload) but SEKs for use within a BCAST system could either be stored in the MBMS Key Store or BCAST Key Store. This was considered not to be acceptable as:

- The Smartcard would have to look in two places for the SEK required to process the STKM, adding complexity to an already complex process and possibly impacting performance;

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

- The Terminal would have to get information from two locations in order to know which SEKs where present on the Smartcard;
- In the case of parallel MBMS Security and Smartcard Profile implementations, there would be no easy way of differentiating which keys in the MBMS Key Store were for use within an MBMS system and which were for use within a BCAST system. Furthermore, the broadcast service provider would have to manage SEK/MSK IDs to avoid collisions (see below).

G.3 Terminal Filtering Based on UDP Port and Smartcard Type

LTKMs sent by a BCAST system are sent to UDP port 4359, while MSK messages sent by MBMS system are sent to UDP port 2269. The Terminal can therefore determine whether a LTKM originated from a BCAST or MBMS system. The Terminal can also determine whether it is paired with a BCAST or MBMS only Smartcard by examining the USIM Service Table.

Knowing both the origin of the message, the contents of the message and the type of Smartcard which it is paired with, the Terminal can decide whether or not it should forward the messages to the Smartcard.

Table 129: Example LTKM Filtering Based on Terminal Filtering Rules

LTKM format			UDP port	Smartcard type	Forwarded to Smartcard
EXT BCAST	FLAG set *	KV data format			
No	No	TEK ID	4359	MBMS	Yes
No	No	TEK ID	4359	BCAST	No
No	No	TEK ID	2269	MBMS	Yes
No	No	TEK ID	2269	BCAST	Yes
Yes	Yes	TimeStamp	4359	MBMS	No
Yes	Yes	TimeStamp	4359	BCAST	Yes
Yes	No	TEK ID	4359	MBMS	Yes
Yes	No	TEK ID	4359	BCAST	No

* Note: FLAG is set to “Yes” if one or more of the following LTKM flags is set to LTK_FLAG_TRUE: security_policy_ext_flag, consumption_reporting_flag and access_criteria flag. It also indicates that the KV data is formated as a TimeStamp.

G.4 Rules for LTKM Creation and Processing

When a BCAST EXT payload with one or more of the following LTKM flag

- security_policy_ext_flag,
- consumption_reporting
- flag and access_criteria flag.

is set to LTK_FLAG_TRUE in an LTKM, the KV data is formatted as a TimeStamp and as TEK_ID otherwise.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

G.5 Rules for the BSM

When addressing a BCAST Smartcard, the BSM SHALL send an LTKM that includes an EXT BCAST payload that either indicates the use of a security policy extension (in which case the KV data will be formatted as an interval of TimeStamps) or is being used for consumption reporting or is being used to set access criteria in the Smartcard.

All LTKMs sent by the BSM will be sent to UDP port 4359.

When addressing a MBMS only Smartcard, the BSM SHALL either:

- Send a LTKM that does not include an EXT BCAST payload; or
- Send an LTKM that includes an EXT BCAST payload in which the security_policy_ext_flag and the consumption_reporting_flag and the access_criteria flag are set to LTK_FLAG_FALSE. This case is allowed in order to support the use of the Terminal Binding Key (TBK) as described above.

In both cases the KV data is formatted as a TEK ID range.

G.6 Rules for the Terminal

The following text only covers the case in which the secure function is located on the Smartcard.

The Terminal applies the following filtering rules to LTKMs received over the UDP port 4359 to stop LTKMs that have been incorrectly formatted by the BSM being sent to the Smartcard. No filtering is applied to MSK messages received on UDP port 2269.

When the Terminal is paired with an MBMS only Smartcard and the received LTKM contains an EXT BCAST payload, if the security_policy_ext_flag or the consumption_reporting_flag or the access_criteria flag are set to LTK_FLAG_TRUE, the Terminal identifies that this message cannot be correctly processed by the MBMS only card and filters it out.

If the Terminal is paired with a BCAST Smartcard and the received LTKM does not contain an EXT BCAST payload, the Terminal identifies that this message would be incorrectly treated as an MBMS MSK message by the BCAST card and filters it out.

G.7 Rules for the Smartcard

Based on the rules for LTKM formatting defined for the BSM, a BCAST Smartcard should reject an LTKM if the EXT BCAST payload is present and the security_policy_extenstion flag, consumption_reporting_flag and access_criteria_flag are set to LTK_FLAG_FALSE (see section 6.6.7.2 for further details). Note that for this situation to occur the BSM must have generated the wrong type of LTKM for the Smartcard and the Terminal must have failed to filter out the incorrectly formatted LTKM.

G.8 Example Scenarios

To illustrate how the functionality described above supports BCAST/MBMS interoperability the following scenarios are used:

A broadcast service provider deploys a BCAST system and MBMS only Smartcards. The broadcast service provider does not use TBK functionality.

1. The broadcast service provider issues its subscribers with LTKMs that do not contain the EXT BCAST payload. The LTKMs are sent to UDP port 4359.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

2. The Terminal receives an LTKM. As the LTKM does not contain an EXT BCAST payload and the Terminal is paired with an MBMS only Smartcard, the Terminal forwards the message to the Smartcard.
3. The MBMS only Smartcard processes the LTKM as an MBMS MSK message, as defined by [3GPP TS 33.246 v7], storing the SEK in the MBMS Key Store.
4. The broadcast service provider issues STKMs than include the EXT BCAST payload.
5. The Terminal receives the STKMs and forwards them to the Smartcard.
6. The Smartcard processes the STKMs as it would for MBMS MTK messages, as defined by [3GPP TS 33.246 v7], ignoring the EXT BCAST payload and using the relevant SEK references stored in EF_{MSK} to process the message contents.

Note: In this scenario if the broadcast service provider wishes to deploy a parallel MBMS security solution, e.g. they use BCAST Smartcard Profile for content broadcast over a DVB-H bearer but MBMS Security for content distributed over an MBMS bearer, they must ensure that the range of SEK IDs used for each of the two systems do not overlap.

A broadcast service provider deploys a BCAST system and both MBMS only and BCAST Smartcards

1. The broadcast service provider issues its subscribers with MBMS only Smartcards with LTKMs that do not contain the EXT BCAST payload and its subscribers with BCAST Smartcards with LTKMs containing the EXT BCAST payload. Both types of LTKMs are sent to UDP port 4359. How the broadcast service provider determines which of its subscribers has which type of card is implementation specific.
2. A Terminal paired with a MBMS only Smartcard receives an LTKM that does not contain an EXT BCAST payload. As the Terminal is paired with an MBMS only Smartcard, it forwards the message to the Smartcard. A Terminal paired with a BCAST Smartcard receives an LTKM that contains an EXT BCAST payload. As the Terminal is paired with a BCAST Smartcard, it forwards the message to the Smartcard.
3. The MBMS only Smartcard processes the LTKM as an MBMS MSK message, as defined by [3GPP TS 33.246 v7], storing the SEK in the MBMS Key Store. The BCAST Smartcard processes the LTKM according to BCAST 1.0 and stores the SEK in the BCAST Key Store.
4. The broadcast service provider issues STKMs than include the EXT BCAST payload.
5. The Terminal receives the STKMs and forwards them to the Smartcard.
6. The MBMS only Smartcard processes the STKMs as it would for MBMS MTK messages, as defined by [3GPP TS 33.246 v7], ignoring the EXT BCAST payload and using the relevant SEK stored in EF_{MSK} to process the message contents. The BCAST Smartcard identifies that the STKM is a BCAST STKM and not an MBMS MTK message through the presence of the EXT BCAST payload and processes the message according to BCAST 1.0, using the relevant SEK from the BCAST Key Store.

Note: The TEK IDs used in the STKMs must be incremented according to the rules for MTK IDs defined in [3GPP TS 33.246 v7]. As the same STKMs are used by MBMS and BCAST Smartcards, the maximum lifetime of a SEK is 16^2 -1 TEKs, i.e. the range of TEK IDs that can be defined by the KV data of the LTKMs sent to the MBMS Smartcards. The KV data of the LTKMs sent to BCAST Smartcards should therefore correspond to the same period, e.g. number of TEKs multiplied by the TEK update period.

A broadcast service provider deploys both a BCAST system and a parallel MBMS system. Both systems use BCAST Smartcards

1. The broadcast service provider issues BCAST LTKMs containing the EXT BCAST payload and also MBMS LTKMs that do not contain the EXT BCAST payload, i.e. MSK messages. The BCAST LTKMs are sent to UDP port 4359 while the MBMS LTKMs are sent to UDP port 2269.
2. If the Terminal receives an LTKM on UDP port 4359 it determines that this is a BCAST LTKM. As the LTKM contains an EXT BCAST payload and the Terminal is paired with a BCAST Smartcard, the Terminal forwards the message to the Smartcard.
   If the Terminal receives an LTKM on UDP port 2269 it treats this message as an MBMS MSK message and processes it according to [3GPP TS 33.246 v7]. Assuming the Terminal processing is successful, the Terminal forwards the message to Smartcard.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

3. The BCAST Smartcard processes the MBMS LTKM as an MBMS MSK message, as defined by [3GPP TS 33.246 v7], storing the SEK in MBMS Key Store. The BCAST Smartcard processes the BCAST LTKM as defined by BCAST 1.0 storing the SEK in the BCAST Key Ktore.
4. The broadcast service provider issues BCAST STKMs that include the EXT BCAST payload. The broadcast service provider also issues MBMS STKMs (MTK Messages), which do not include the EXT BCAST payload.
5. The Terminal receives both sets of STKMs and forwards them to the Smartcard.
6. The BCAST Smartcard processes the BCAST STKMs as defined by BCAST 1.0, using the relevant SEK stored in BCAST Key Store to process the message contents. The BCAST Smartcard processes the MBMS STKMs as it would for MBMS MTK messages, as defined by [3GPP TS 33.246 v7], using the relevant MSK stored in the MBMS Key Store to process the message contents.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix H. Order of Restrictiveness of Parental Rating Values (Informative)

This table illustrates the concept of "order of restrictiveness" as defined in Section 6.7.3.11.1. The examples below are based on the first ten rating types in the OMA BCAST Parental Rating Registry at [OMNA]. For other schemes that will possibly be registered in the future, the order can be determined based on the semantics of the individual rating values as defined by the actual scheme in question.

Table 130: Examples of Order of Restrictiveness

rating_type	rating_value in order of increasing restrictiveness
0	0, 1, 2, 3, 4, 5,…,15
1	4, 0, 1, 2, 3
2	6, 5, 4, 3, 2, 1
3	6, 1, 2, 3, 4, 5
4	6, 5, 4, 3, 2, 1
5	2, 1
6	6, 5, 4, 3, 2, 1
7	7, 1, 2, 3, 4, 5, 6
8	6, 5, 4, 3, 2, 1
9	0, 1, 2, 3, 4, 5
10	0x00, ….0xFF

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix I. Registration of SDP Attributes (Informative)

The attributes

- bcastversion:<major>.<minor>
- stkmstream:<id of the stkm stream>
- SRTPAuthentication:<id for SRTP authentication algorithm value>
- SRTPROCTxRate:<ROC transmission rate>

are defined in this specification and have been registered with IANA. (See http://www.iana.org/assignments/sdp-parameters). [RFC5159] provides the descriptions of the SDP attributes used.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix J. Deriving the Zn/Zn' URL (Informative)

In some deployments, the same NAF may interact with the BSFs of multiple operators, e.g. where a DVB-H network is shared by multiple operators. Where Zn over web services is used, the NAF must therefore direct Zn requests to the correct BSF and to the correct URL on that BSF. If the Zn URLs are not preconfigured in the NAF, it is recommended that the method defined below is used to derive the Zn URL used in the GBA procedure. This applies to the case where the NAF is able to communicate directly with the BSF. In the case where a Zn-Proxy is used between the NAF and BSF (as specified in [3GPP TS 33.220 v6]), it is recommended that the Zn proxy use this method to derive the Zn' URL rather than the Zn URL.

OMA BCAST recommends that the default Zn URL be constructed using the B-TID received by the NAF from the UE over Ua as follows:

The fully qualified domain name is extracted from the B-TID and the path "/Zn" is added. The protocol is "https:". The port number is 5321 as defined by IANA for Zn over ssl. For example, for a B-TID of *base64encode(RAND)@ bsf.mnc015.mcc234.pub.3gppnetwork.org*, the Zn URL resolves to the following:

https:// bsf.mnc015.mcc234.pub.3gppnetwork.org:5321/Zn

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Appendix K. Examples of Command Chaining for OMA BCAST Instruction Code (Informative)

K.1 At Applicative Level

K.1.1 SPE Audit Mode

In the first example the SPE Audit Data Object length is 9 bytes ('09' hex) and the length of the response data is 100 bytes ('64' hex). In this case there is no output chaining, and the response data are sent in one SPE Audit APDU.

Table 131: Example of OMA BCAST Command with hort data in SPE Audit Mode

	Terminal		UICC
OMA BCAST Command (First Block of data)	80 1B 80 01 Lc=0B [73 09 81 03 xx .. 82 02 yy..]	→	
		←	62 F3
OMA BCAST Command (First Block of response data)	80 1B A0 01 Le=00	→	
		←	[73 64 zz ...zz] 90 00

In the second example SPE Audit Data Object length is 9 bytes ('09' hex) and the SPE Audit operation response Data Object length is 600 bytes ('0258' hex). In this case there is an output chaining, and the response data are sent in several SPE Audit APDU.

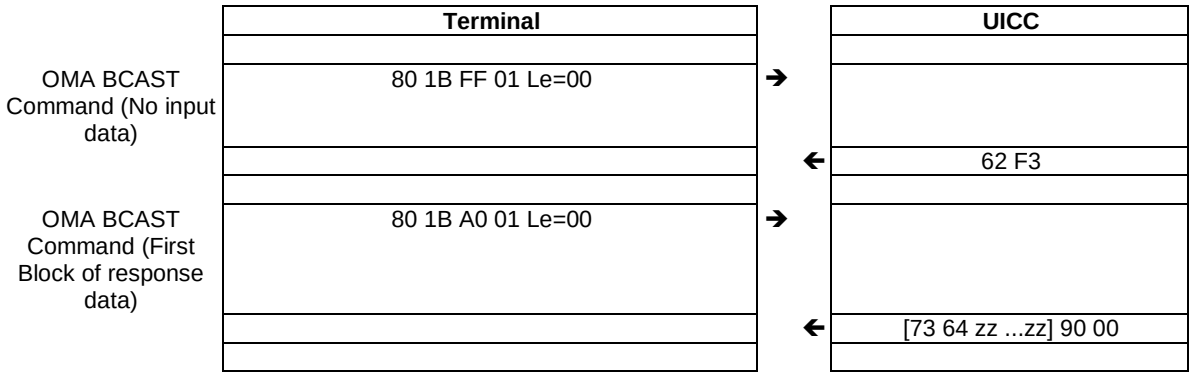
Table 132: Example of OMA BCAST Command with extended data in SPE Audit Mode

	Terminal		UICC
OMA BCAST Command (First Block of data)	80 1B 80 01 Lc=0B [73 09 81 03 xx .. 82 02 yy..]	→	
		←	62 F3
OMA BCAST Command (First Block of response data)	80 1B A0 01 Le=00	→	
		←	[73 82 02 58 zz ... zz] 62 F1
OMA BCAST Command (Next Block of response data)	80 1B 20 01 Le=00	→	
		←	[zz ... zz] 62 F1
OMA BCAST Command (Next Block of response data)	80 1B 20 01 Le=5C	→	
		←	[yy ... yy] 90 00

In the third example there is no input data and the SPE Audit operation response Data Object length is 100 bytes ('64' hex). In this case P1='FF' in the first APDU command, and the response data are sent in one SPE Audit APDU.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

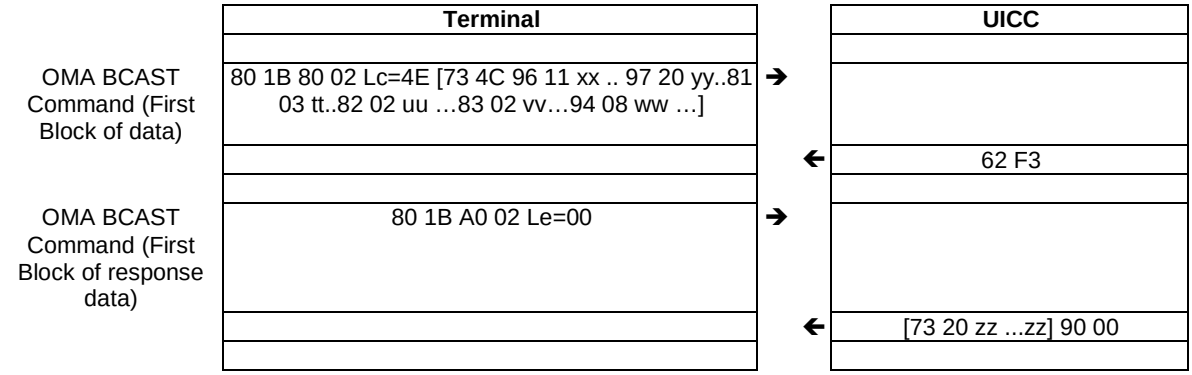
Table 133: Example of OMA BCAST Command without input data and with short data in SPE Audit Mode



K.1.2 Record Signalling Mode

In this example Record Signalling Data Object length is 76 bytes ('4C' hex) and Record Signalling operation response Data Object length is 32 bytes ('20' hex). In this case there is no input chaining and the data are transmitted in one Record Signalling APDU and there is no output chaining, and the response data are sent in one Record Signalling APDU.

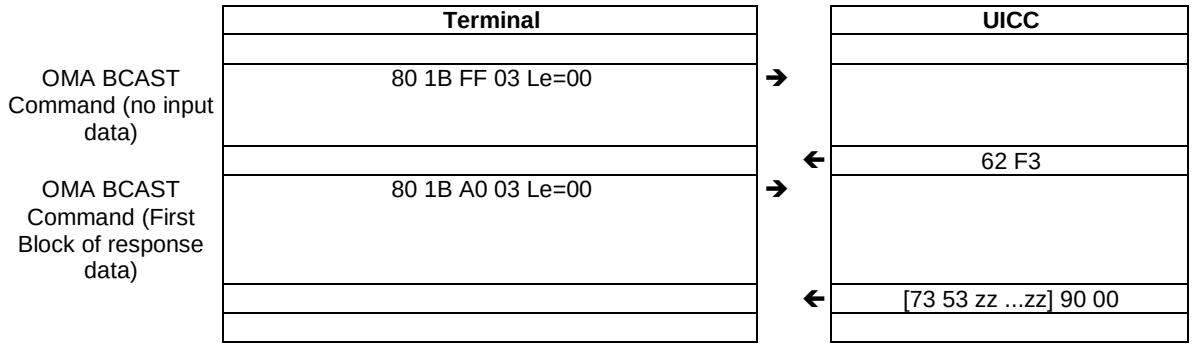
Table 134: Example of OMA BCAST Command with short data in Record Signalling Mode



K.1.3 Recording Audit Mode

In the first example Recording Audit operation response Data Object length is 83 bytes ('53' hex). In this case there is no output chaining, and the response data are sent in one Recording Audit APDU.

Table 135: Example of OMA BCAST Command with short data in Recording Audit Mode



© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

In the second example Recording Audit operation response Data Object length is 600 bytes ('0258' hex). In this case there is an output chaining, and the response data are sent in several Recording Audit APDU.

Table 136: Example of OMA BCAST Command with extended data in Recording Audit Mode

	Terminal		UICC
OMA BCAST Command (No input data)	80 1B FF 03 Le=00	→	
		←	62 F3
OMA BCAST Command (First Block of response data)	80 1B A0 03 Le=00	→	
		←	[73 82 02 58 zz ... zz] 62 F1
OMA BCAST Command (Next Block of response data)	80 1B 20 03 Le=00	→	
		←	[zz ... zz] 62 F1
OMA BCAST Command (Next Block of response data)	80 1B 20 03 Le=5C	→	
		←	[yy ... yy] 90 00

K.1.4 Event Signalling Mode

In this example a "zapping" event is signaled to the Smartcard.

Table 137: Example of OMA BCAST Command in Event Signalling Mode

	Terminal		UICC
OMA BCAST Command (First Block of data)	80 1B 80 04 Lc=05 [73 03 8F 01 00]	→	
		←	90 00

K.2 At applicative level and transport level with transport protocol T=0

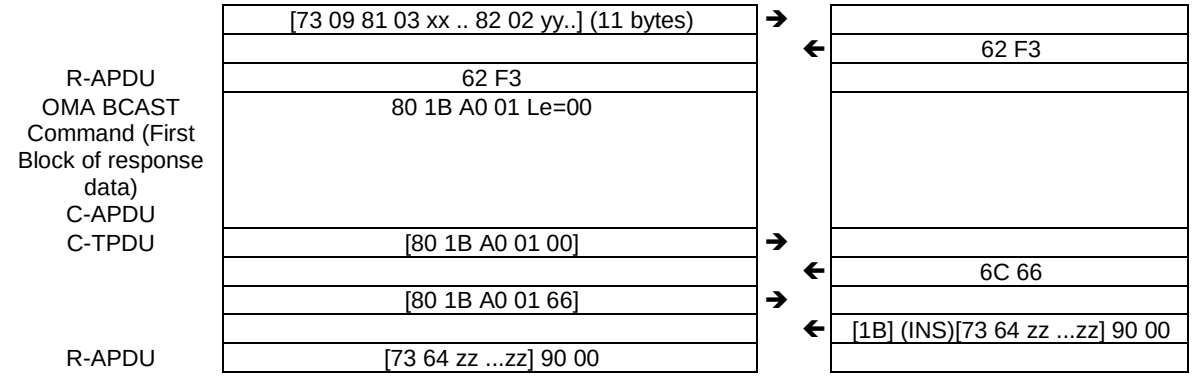
K.2.1 SPE Audit mode

In the first example the SPE Audit Data Object length is 9 bytes ('09' hex) and SPE Audit operation response Data Object length is 100 bytes ('64' hex). In this case there is no output chaining, and the response data are sent in one SPE Audit APDU.

Table 138: Example of OMA BCAST Command with short data in SPE Audit Mode

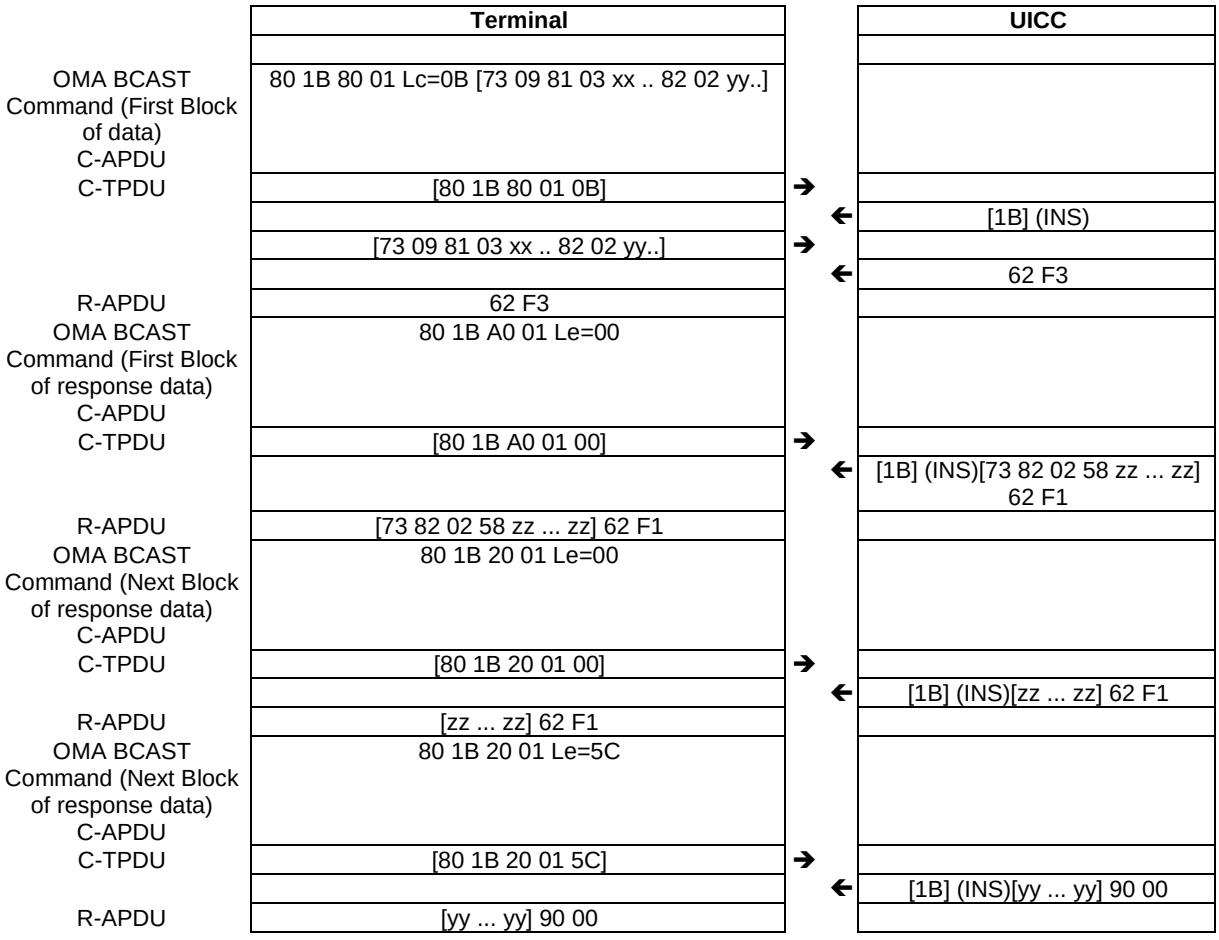
	Terminal		UICC
OMA BCAST Command (First Block of data) C-APDU	80 1B 80 01 Lc=0B [73 09 81 03 xx .. 82 02 yy..]		
C-TPDU	[80 1B 80 01 0B]	→	
		←	[1B] (INS)

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.



In the second example the SPE Audit Data Object length is 9 bytes ('09' hex) and the SPE Audit operation response Data Object length is 600 bytes ('0258' hex). In this case there is an output chaining, and the response data are sent in several SPE Audit APDU.

Table 139: Example of OMA BCAST Command with extended data in SPE Audit Mode



In the third example there is no input data and the SPE Audit operation response Data Object length is 100 bytes ('64' hex). In this case P1='FF' in the first APDU command, and the response data are sent in one SPE Audit APDU.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

Table 140: Example of OMA BCAST Command without input data and with short data in SPE Audit Mode

	Terminal		UICC
OMA BCAST Command (First Block of data) C-APDU	80 1B FF 01 Le=00		
C-TPDU	[80 1B FF 01 00]	→	
		←	62 F3
R-APDU	62 F3		
OMA BCAST Command (First Block of response data) C-APDU	80 1B A0 01 Le=00		
C-TPDU	[80 1B A0 01 00]	→	
		←	6C 66
	[80 1B A0 01 66]	→	
		←	[1B] (INS)[73 64 zz ...zz] 90 00
R-APDU	[73 64 zz ...zz] 90 00		

K.2.2 Record Signalling Mode

In this example the Record Signalling Data Object length is 76 bytes ('4C' hex) and Record Signalling operation response Data Object length is 32 bytes ('20' hex). In this case there is no input chaining and the data are transmitted in one Record Signalling APDU and there is no output chaining, and the response data are sent in one Record Signalling APDU.

Table 141: Example of OMA BCAST Command with short data in Record Signalling Mode

	Terminal		UICC
OMA BCAST Command (First Block of data) C-APDU	80 1B 80 02 Lc=4E [73 4C 96 11 xx .. 97 20 yy..81 03 tt..82 02 uu …83 02 vv…94 08 ww …]		
C-TPDU	[80 1B 80 02 4E]	→	
		←	[1B] (INS)
	[73 4C 96 11 xx .. 97 20 yy..81 03 tt..82 02 uu …83 02 vv…94 08 ww …] (76 Bytes)	→	
		←	62 F3
R-APDU	62 F3		
OMA BCAST Command (First Block of response data) C-APDU	80 1B A0 02 Le=00		
C-TPDU	[80 1B A0 02 00]	→	
		←	6C 22
	[80 1B A0 02 22]	→	
		←	[1B] (INS) [73 20 zz ...zz] 90 00
R-APDU	[73 20 zz ...zz] 90 00		

K.2.3 Recording Audit Mode

In the first example the Recording Audit operation response Data Object length is 83 bytes ('53' hex). In this case there is no output chaining, and the response data are sent in one Recording Audit APDU.

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document. [OMA-Template-EnablerTestSpec-201301011-I]

Table 142: Example of OMA BCAST Command with short data in Recording Audit Mode

	Terminal		UICC
OMA BCAST Command (No input data) C-APDU	80 1B FF 03 Le=00		
C-TPDU	[80 1B FF 03 00]	→	
		←	62 F3
R-APDU	62 F3		
OMA BCAST Command (First block of response data) C-APDU	80 1B A0 03 Le=00		
C-TPDU	[80 1B A0 03 00]	→	
		←	6C 55
	[80 1B A0 03 55]	→	
		←	[1B] (INS)[73 53 zz ...zz] 90 00
R-APDU	[73 53 zz ...zz] 90 00		

In the second example the Recording Audit operation response Data Object length is 600 bytes ('0258' hex). In this case there is an output chaining, and the response data are sent in several Recording Audit APDU.

Table 143: Example of OMA BCAST Command with extended data in Recording Audit Mode

	Terminal		UICC
OMA BCAST Command (No input data) C-APDU	80 1B FF 03 Le=00		
C-TPDU	[80 1B FF 03 00]	→	
		←	62 F3
R-APDU	62 F3		
OMA BCAST Command (First block of response data) C-APDU	80 1B A0 03 Le=00		
C-TPDU	[80 1B A0 03 00]	→	
		←	[1B] (INS)[73 82 02 58 zz ... zz] 62 F1
R-APDU	[73 82 02 58 zz ... zz] 62 F1		
OMA BCAST Command (Next Block of response data) C-APDU	80 1B 20 03 Le=00		
C-TPDU	[80 1B 20 03 00]	→	
		←	[1B] (INS) [zz ... zz] 62 F1
R-APDU	[zz ... zz] 62 F1		
OMA BCAST Command (Next Block of response data) C-APDU	80 1B 20 03 Le=5C		
C-TPDU	[80 1B 20 03 5C]	→	
		←	[1B] (INS) [yy ... yy] 90 00
R-APDU	[yy ... yy] 90 00		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

K.2.4 Event Signalling Mode

In this example a “zapping” event is signaled to the Smartcard.

Table 144: Example of OMA BCAST Command in Event Signalling Mode

	Terminal		UICC
OMA BCAST Command (First Block of data) C-APDU	80 1B 80 04 Lc=05 [73 03 8F 01 00]		
C-TPDU	[80 1B 80 04 05]	→	
		←	[1B] (INS)
	[73 03 8F 01 00] (5 bytes)	→	
		←	90 00
R-APDU	90 00		

© 2013 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.