ISMA™
Internet Streaming Media Alliance

Presidio of San Francisco
P.O. Box 29920
572 B Ruger Street
San Francisco, CA 94129-0920

T 415.561.6276
F 415.561.6120

www.isma.tv

INTERNET STREAMING MEDIA ALLIANCE

Implementation Specification

ISMA Encryption and Authentication, Version 1.1

AREA / Task Force: DRM

September 15, 2006

Table of Contents

1.0 Introduction

This document specifies payload encryption and message authentication (integrity) services for ISMA 1.0 [ISMASPEC] and ISMA 2.0 [ISMASPEC2]. The official name for this service specification is "ISMA 1.1 Encryption and Authentication" but it is unofficially called "ISMACryp" throughout much of this document. Thus "ISMACryp" serves as a nickname for ISMA 1.1 Encryption and Authentication, which is a cryptographic framework for ISMA 1.0 and ISMA 2.0 media streams. Note that this specification can also be applied to other context/codecs than those specified in ISMA 1.0 or ISMA 2.0.

The framework is extensible to new media encodings, can be upgraded to new cryptographic transforms, and is applicable to a variety of key management, security, or digital rights management (DRM) systems. ISMACryp also defines a default encryption of media streams and authentication of media messages for ISMA 1.0 and ISMA 2.0. The ISMA 1.1 Encryption and Authentication transforms (i.e. enc-mpeg4-generic and RTP/SAVP) conform to the ISMA DRM Recommendations [ISMADRM].

Sections 4, 5 and 6 repeat the requirements and architecture sections of the DRM Recommendations for the convenience of the reader. Sections 7, 8, 9 and 10 define ISMACryp cryptography, RTP payload type, SDP signaling, and MP4 file format [14496-12, 14496-14].

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 RFC 2119 [RFC2119]. Section 2.0 gives definitions for important terms that are used throughout this document.

2.0 Glossary of Terms

AU	An ISMA transport packet contains at least one AU header and MPEG media AU. See "media frame" regarding the definition of "media AU."
Authentication	See "Entity authentication" and "Message authentication"
Authorization	The process of allocating access to resources, such as key-management keying material, to an authenticated entity. Authorization is outside of the the scope of ISMA 1.1 Encryption and Authentication.
Byte Stream Offset (BSO)	The position of a stream byte relative to the first stream byte, which has a BSO of zero. The default encryption transform defines and uses the BSO for the IV (see "Initialization Vector").
Confidentiality	Access control that is applied to media using ISMA 1.1 Encryption and Authentication.
Entity authentication	Entity authentication confirms that an entity (e.g. a rights holder) has possession of a secret associated with its identity. Key management procedures often perform entity authentication during key establishment such as in an authenticated key exchange [Krawczyk].
Initialization Vector (IV)	The cryptographic metadata needed by a payload-encryption and/or message-authentication transform in the ISMA 1.1 Encryption and Authentication framework. The contents of the IV depend on the particular transform. The default ISMACryp encryption transform uses the BSO as the IV (See Section 10.0). ISMACryp default message-authentication does not use the IV.
Integrity	See "Message integrity" and "Message Authentication"
IPMP	Intellectual Property Management and Protection - the part in MPEG-4 dealing with authentication and protection of presentations.
IPMP-X	IPMP extensions – an amendment to MPEG-4 aimed to enable interoperation between different IPMP tools and/or systems.
Media	Continuous-time data that share a common timebase.
Media authentication	Media authentication validates the integrity of the media data, independently of the message, and authenticates the rights holder who mastered and

	authenticated the data.
Media frame	The smallest clocked unit of media.
Message authentication	Message authentication validates that the received message is identical to what was sent by the sender. A transport security protocol performs message authentication using an integrity check of a message authentication code or through the verification of a digital signature.
Message integrity	See "Message authentication."
Packetization	Process of assigning media frames or fragments of media frames to MPEG transport packets or to RTP packets before, during, or after encryption of the media data. Packetization is performed by a "packetizer."
Privacy	Access controls applied to the user's identity and activity on a network.
Sample	See "media frame"

3.0 References

3.1 Normative

[14496-1] ISO/IED 14496-1
Information technology -- Coding of audio-visual objects --
Part 1: Systems,
Third Edition, November 2004.

[14496-8] ISO/IEC 14496-8
Information technology -- Coding of audio-visual objects --
Part 8: Carriage of ISO/IEC 14496 contents over IP networks,
First Edition, May 2004.

[14496-10] ISO/IEC 14496-10
Information technology -- Coding of audio-visual objects --
Part 10:, Advanced Video Coding
2003
ITU-T Recommendation H.264 (2003): "Advanced video coding for generic audiovisual services"

[14496-12] ISO/IEC 14496-12|15444-12
Information technology -- Coding of audio-visual objects --
Part 12: ISO base media file format,
Information technology -- JPEG 2000 image coding system --
Part 12: ISO base media file format,
Second Edition, April 2005.

[14496-13] ISO/IEC 14496-13
Information technology -- Coding of audio-visual objects --
Part 13: Intellectual Property Management and Protection (IPMP),
First Edition, September 2004.

[14496-14] ISO/IEC 14496-14
Information technology -- Coding of audio-visual objects --
Part 14: MP4 file format,
First Edition, November 2003.

[AES] NIST FIPS 197
Advanced Encryption Standard (AES),
(http://csrc.nist.gov/encryption/aes/index.html),

(http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf).

[AES-C] NIST SP 800-38
Recommendation for Block Cipher Modes of Operation Methods and Techniques,
M. Dworkin, December 2001,
(http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf).

[AES-CTR] NIST Workshop1
CTR-Mode Encryption,
H.Lipmaa, P.Rogaway, D. Wagner,
(http://csrc.nist.gov/encryption/modes/workshop1/papers/lipmaa-ctr.pdf).

[ISMADRM] ISMA DRM
Recommendations Version 1.0, Internet Streaming Media Alliance, 2570 W. El Camino Real,
Mountain View, California, 94040, January 2002.

[ISMASPEC] ISMA Implementation Specification
Version 1.0, August 2001.

[ISMASPEC2] ISMA Implementation Specification
Version 2.0, April 2005.

[RFC1630] IETF RFC 1630
Uniform Resource Identifiers in WWW,
T. Berners-Lee, June 1994,
(ftp://ftp.rfc-editor.org/in-notes/rfc1630.txt).

[RFC2119] IETF RFC 2119
Key words for use in RFCs to Indicate Requirements Levels,
S. Bradner, March 1997,
(http://www.ietf.org/rfc/rfc2119.txt?number=2119).

[RFC2327] IETF RFC 2327
SDP: Session Description Protocol,
M. Handley, V. Jacobson, April 1998,
(ftp://ftp.rfc-editor.org/in-notes/rfc2327.txt).

[RFC3016] IETF RFC 3016
,
RTP Payload Format for MPEG-4 Audio/Visual Streams,
Y. Kikuchi, et. Al, November 2000,
(ftp://ftp.rfc-editor.org/in-notes/rfc3016.txt).

[RFC3640] IETF RFC 3640
RTP Payload Format for Transport of MPEG-4 Elementary Streams,
J. van der Meer, D. Mackie, V. Swaminathan, D. Singer, P. Gentric, November 2003,
(ftp://ftp.rfc-editor.org/in-notes/rfc3016.txt).

[RFC3711] IETF RFC 3984
SRTP: The Secure Real Time Transport Protocol,
M. Baugher, D. McGrew, D. Oran, R. Blom, E. Carrera, M. Näslund, K. Normann, March 2004,
(ftp://ftp.rfc-editor.org/in-notes/rfc3711.txt).

[RFC4568] IETF RFC 4568

SDP Security Descriptions for Media Streams,
F.Andreasen, M.Baugher, D.Wing, July 2006
(ftp://ftp.rfc-editor.org/in-notes/rfc4568.txt).

3.2 Informative

[21000-5] ISO/IEC 21000-5
Part 5: Rights Expression Language,
First Edition, April 2004.

[21000-6] ISO/IEC 21000-6
Information Technology — Multimedia Framework —
Part 6: Rights Data Dictionary,
First Edition, May 2004.

[AMPBH] Proc. Security Protocols Workshop '97
Secure Books: Protecting the Distribution of Knowledge,
R.J.Anderson, V. Matyás Jr., F. Petitcolas, I.E. Buchan, R. Hanka, 1997,
(http://www.cl.cam.ac.uk/users/rja14/).

[CPRM] 4C Entity Content Protection for Recordable Media
(http://www.4centity.com/docs/versions.html).

[EPIC] Pretty Poor Privacy: An Assessment of P3P and Internet Privacy
EPIC, June 2000,
(http://www.epic.org/Reports/prettypoorprivacy.html).

[FFSS] Privacy Engineering in Digital Rights Management Systems
Revised papers from the ACM CCS-8 Workshop on Security and Privacy in DRM,
J. Fegenbaum, M.Freedman, T. Sander, A. Shostack, 2001,
(http://www.homeport.org/~adam/privacyeng-wspdrm01.pdf).

[IPMPH] ISO/IEC JTC1/SC29/WG11/N2614
MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Applications,
J. Lacy, N. Rump, P. Kudumakis, December 1998,
(http://www.chiariglione.org/mpeg/working_documents.htm#MPEG-4).

[IPSEC] IETF RFC 2411
IP Security Document Roadmap,
R. Thayer, N. Doraswamy, R. Glenn, November 1998,
(http://www.ietf.org/rfc/rfc2411.txt).

[Krawczyk] SKEME: A Versatile Secure Key Exchange Mechanism for Internet,
1996 Symposium on Network and Distributed System Security,
H. Krawczyk, February 1996.

[Marks & Turnbull] WIPO WCT-WCT-WPPT/IPMP/3
Technical protection measures: The intersection of technology, law, and commercial licenses,
D.S. Marks, B.H. Turnbull, December 1999,
(http://www.wipo.int/documents/en/meetings/1999/wct_wppt/doc/imp99_3.doc).

[MF00] Attacks on Encryption of Redundant Plaintext and Implications on Internet Security
Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC),

D. McGrew, S. Fluhrer,2000,
(http://www.mindspring.com/~dmcgrew/dam-srf-sac00.pdf).

[NNL] Revocation and Tracing Schemes for Stateless Receivers,
Lecture Notes in Computer Science,
D. Naor M. Naor, J. Lotspiech, March 2002
(http://www.wisdom.weizmann.ac.il/%7Enaor/PAPERS/2nl.pdf),
(http://www.securemulticast.org/GSEC/gsec1_Naor.pdf).

[OFT] Key Establishment in Large Dynamic Groups Using One-Way Function Trees
D.McGrew, A.Sherman, 1998,

[OMADCFV2] OMA-TS-DRM-DCF-V2_0-20060303-A
DRM-DCF Specification V2.0 (Approved version),
March 2006
(http://member.openmobilealliance.org/ftp/Public_documents/BAC/DLDRM/Permanent_documents/OMA-TS-DRM-DCF-V2_0-20060303-A.zip)

[OMADRMV2] OMA-TS-DRM-DRM-V2_0-20060303-A
DRM-DRM Specification V2.0 (Approved version),
March 2006
(http://member.openmobilealliance.org/ftp/Public_documents/BAC/DLDRM/Permanent_documents/OMA-TS-DRM-DRM-V2_0-20060303-A.zip)

[OMARELV2] OMA-TS-DRM-REL-V2_0-20060303-A
DRM-REL Specification V2.0 (Approved version),
March 2006
(http://member.openmobilealliance.org/ftp/Public_documents/BAC/DLDRM/Permanent_documents/OMA-TS-DRM-REL-V2_0-20060303-A.zip)

[RFC2104] IETF RFC 2104
HMAC: Keyed-Hashing for Message Authentication,
H.Krawczyk, M.Bellare, R.Canetti, February 1997,
(http://www.ietf.org/rfc/rfc2104.txt).

[RFC2627] IETF RFC 2627
Key Management for Multicast: Issues and Architectures,
D.Wallner, E.Harder, R.Agee, June 1999,
(ftp://ftp.rfc-editor.org/in-notes/rfc2627.txt).

[RFC3547] IETF RFC 3547
The Group Domain of Interpretation, IETF,
M.Baugher, T.Hardjono, H.Harney, B.Weis, July 2003,
(http://www.ietf.org/rfc/rfc3547.txt).

[SECURITY] A concrete security treatment of symmetric encryption
Proceedings of the 38th Symposium on Foundations of Computer Science, IEEE,
M.Bellare, A.Desai, E.Jokipii, P.Rogaway, 1997,
(http://www-cse.ucsd.edu/users/mihir/papers/sym-enc.html).

[SMPEG] Cryptographic Message Syntax for MP3/MPEG I, II, and III Secure MPEG
J. Kahn, M. Gaur, August 2001, Work in Progress,
(http://search.ietf.org/internet-drafts/draft-khan-gaur-secure-mpeg-syntax-00.txt).

4.0 ISMA Architecture

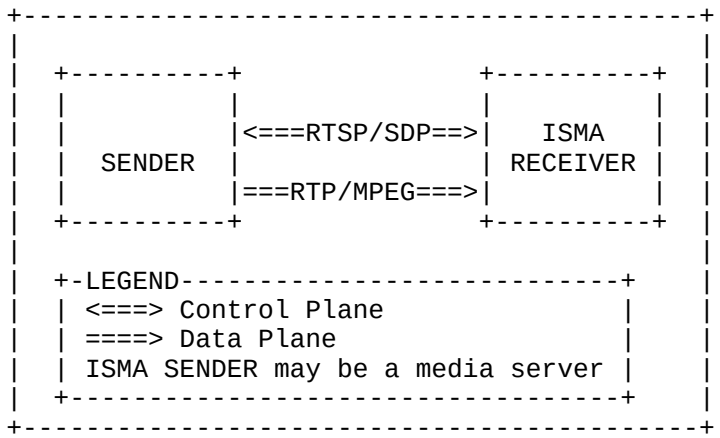


Figure 4.0-1: ISMA Architecture

The ISMA 1.0 and ISMA 2.0 streaming specifications define an RTP/MPEG protocol for the "data plane." ISMA 1.0 and ISMA 2.0 also include Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP) definitions for the "control plane." Figure 4.0-1 illustrates the ISMA data protocols on the arc labeled "RTP/MPEG" and control protocols on the arc labeled "RTSP/SDP," where RTSP is one means to provide VCR-like controls for an ISMA session.

5.0 ISMA DRM Overview

This section briefly reviews the ISMA DRM Recommendations [ISMADRM] document.

5.1 ISMA DRM Goals

There are three general goals for the first release of the ISMA DRM specification according to the ISMA DRM Recommendations.

1. Ensure the "support and consistency" of ISMA 1.0 specifications [ISMASpec] when DRM is added. Interoperability is a goal of ISMA, preserving interoperability when DRM is added is the major goal of ISMA DRM: ISMA 1.0 "interoperability" is data plane, e.g. MPEG over RTP[14496-8], and control plane (RTSP) interoperability at the ISMA receiver with or without cryptographic services.

2. Remove technical barriers to the dissemination of rights-managed content on platforms that run the ISMA protocols [Marks & Turnbull]. If ISMA media and protocols are to serve as open, standard interfaces to "content protection" devices, ISMA needs to accommodate the technical protection measures (TPM) in those devices and it needs to implement the change control needed for platform licensing.

3. Identify what needs to be standardized, which bodies are developing needed standards and what needs to be invented. For example, MPEG is developing the decoder interfaces and services for DRM; the IETF, SMPTE, and OMA are developing cryptographic and key management protocols. The IETF defines cryptographic protocols for IP-network applications, particularly for the high-security needs of governments, enterprises, and individuals [IPSEC, RFC3711, RFCSDPSD].

The ISMA DRM Recommendations document surveys DRM technology and considers all that technology that ISMA media and protocols might need in the future. Following this survey, it concludes that ISMA immediately needs encryption and authentication of ISMA 1.0 media and protocols. Thus, the scope of ISMA 1.0 is much more narrowly focused than the more general problems of digital rights management and content protection. The general model is given to better understand the role that the present work, ISMA 1.1 Encryption and Authentication, plays in a more general environment of DRM, content protection, and security. It can also be applied to ISMA 2.0.

5.2 ISMA DRM Model

The technical requirements of ISMA DRM can be understood in the context of the archetypal end-to-end DRM architecture shown in Figure 5.2-1.

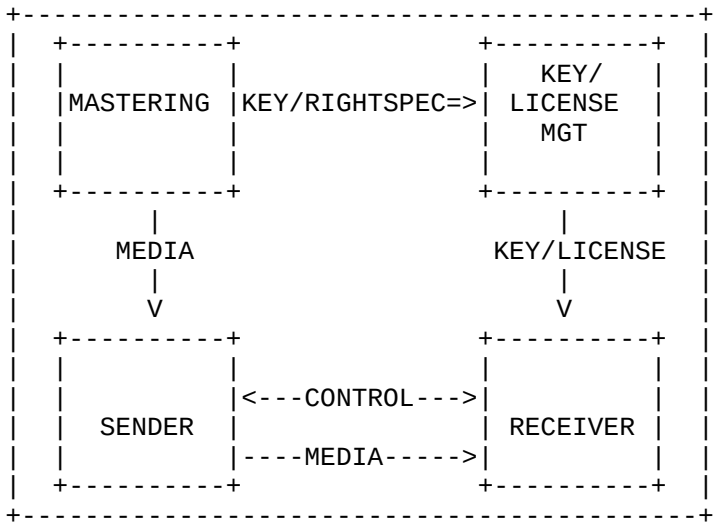


Figure 5.2-1: END-TO-END DRM ARCHITECTURE

Figure 5.2-1 is a minimal description that folds multiple functions in a single box in some cases (rights clearing is not shown in Figure 5.2-1, for example). In the Figure, "MASTERING" is where a content work is prepared for dissemination. It may be encrypted and associated with a rights specification that is formatted according to a "rights expression language" [21000-5, 21000-6, OMARELv2]. Mastering is important to interoperability when cryptographic transforms are applied: The standard receiver needs standard cryptographic transforms for the decryption, authentication, and integrity of content works. The Mastering step may apply encryption to the mastered media work.

The KEY/LICENSE MGT entity in Figure 5.2-1 associates a rights specification and cryptographic keys with an ISMA content work. KEY/LICENSE MGT translates the rights specification into a license. The license authorizes particular types of access to the work, possibly according to a set of "business rules." The access may be at a highly-granular level of access such as to view/hear the content, write to a DVD, or send to a friend. Key/License Management is referenced but not specified; ISMA needs to accommodate a variety of Key/License management systems, both standard and proprietary.

Although Figure 5.2-1 shows KEY/LICENSE MGT providing only keys to the RECEIVER, the SENDER may obtain the key using the same elements of procedure as the RECEIVER. The process may be completely different on the SENDER if the content is pre-encrypted at the MASTERING step leaving the SENDER with no need to hold the key to the content work. This latter scenario is depicted in Figure 5.2-1, which shows a key/license flow to the RECEIVER but not to the SENDER. Thus, implementations may vary from Figure 5.2-1. In another possible variation, the MASTERING entity that encrypts a content work could obtain the key from KEY/LICENSE MGT instead of providing the key to latter. Figure 5.2-1 collapses these alternatives to a minimal set. In any case, the SENDER entity manages the requests from receivers (labeled "CONTROL") and disseminates content works ("MEDIA") to receivers; both CONTROL and MEDIA flows may use encryption, authentication and integrity services.

The RECEIVER of Figure 5.2-1 decrypts and authenticates content works contained in the MEDIA flow and may decrypt and authenticate CONTROL flows. Depending on the nature of the key management protocol in use, the RECEIVER may perform mutual authentication with the KEY/LICENSE MGT entity to prove that the receiver is an authorized platform. This process is controlled by the license, which specifies the terms and conditions under which a key is provided to an ISMACryp device [21000-5, 21000-6, OMARELv2]. The license determines what authenticating information is exchanged, such as information about the RECEIVER's hardware, software or human user. This information needs to be governed by a specification as to what can be collected and how it can be used. Exchanges with

KEY/LICENSE MGT need to be secured in practically all circumstances to protect the user identity and the user's content-work transactions as well as the content-work keys. The media decryption keys must also be secured, and the receiver may be a licensed content-protection platform [CPRM]. The first release of ISMA DRM supports but does not specify the interfaces, messaging, or processing of content-protection platforms.

Thus, the information assets to be protected go beyond content works; they include information related to the user's privacy [FFSS, EPIC] and authenticating information. These assets also include resources such as the CPU, storage, service and bandwidth of the provider and the user.

According to the ISMA DRM Recommendations, ISMA must specify cryptographic transforms for the media flows of Figure 5.2.1 and define the needed interfaces for a variety of Key and License Management systems and for a variety of receiver content-protection platforms. Thus, Figure 5.2.1 provides the "big picture" and ISMA 1.1 Encryption and Authentication is only one part of the complete solution.

5.3 Technical Requirements for ISMA 1.1 Encryption and Authentication

1. It MUST be possible to encrypt ISMA media and protocols; media encryption may take place either during mastering or sending. Ideally, the solution should cater to either scenario without modification and MUST support MPEG IPMP [14496-1] minimal signaling to make the encrypted presentation MPEG-4 compliant (ISO/IEC 14496 compliant, as specified in ISO/IEC 14496-4 and its amendments).

2. It MUST be possible to authenticate the origin of ISMA media and protocols.

The cryptographic transforms used to authenticate messages or encrypt media need to be replaceable, such as when particular algorithms are compromised or superior algorithms are invented. The following two requirements mandate that ISMA provide a cryptographic framework to promote replacement and upgrade of its cryptographic services.

3. There MUST be a set of default transforms for encryption, authentication, and integrity with the default encryption transform having the following properties.

The same encryption scheme MUST be usable for video, audio and any associated data. The cipher MUST be symmetric and MUST have a relatively small memory "footprint" for a given cryptographic strength, which MUST be at least 128 bits and SHOULD support key sizes larger than 128 bits. The cipher MUST have known cryptographic strength against an appropriate set of attack methods. The work factor of the best-known attack MUST be known and deemed to be acceptable. The cipher MUST have undergone extensive analysis by the world cryptographic community, and should be widely adopted.

4. There MUST be a framework that allows for the replacement of the default encryption, authentication and integrity transforms.

A cryptographic framework enables ISMA cryptographic services to be robust in a variety of environments, such as licensed content-protection devices that mandate specific ciphers. Another measure of robustness is the ability of the default transforms to satisfy requirements for a variety of media and applications. The following two requirements mandate that ISMA provide robust cryptographic services in its default transforms.

5. Content MAY be encrypted by the default cipher using a single master key, or a sequence of time-varying session keys. Means for the accurate and deterministic synchronization between media packets and their associated key messages MUST be provided.

6. The default cryptosystem MUST be self-synchronizing, providing random access or seek capabilities, as well as recovery from data loss. Although these are different scenarios, in practice they depend on the same

criteria: The lack of all or part of any preceding data blocks MUST NOT affect the decryption of the current data block. The availability of reliable (explicit or implicit) continuity information for the data to be decrypted may be assumed.

7. The default transforms MUST offer good performance in both hardware and software across a wide range of computing environments. Key setup time, key update and parallelism are all seen as important. Our choice of algorithm should reflect a "security to a point" policy in which acceptable security concessions could be made to increase efficiency and reduce complexity.

8. The cryptosystem MUST offer low data expansion. The size of the resulting cipher text should be the same as the size of the plaintext, and the size of any additional "security headers" should be kept to a minimum.

9. Encryption or authentication of more than one message with the same "master" key MUST be possible, without compromising security in any way. The data key MAY be derived using IVs in stream or hybrid stream/block ciphers so as to prevent re-use of a keystream. Thus, it is acceptable to achieve all the goals listed above by means of combining ciphers and/or modes, provided that the resulting solution offers benefits that are not achievable using a single cipher.

10. ISMA transforms MUST use open, standard technology with the additional goal of having no known intellectual property rights (IPR) encumbrances.

Thus, ISMACryp focuses only on the encryption and authentication of ISMA 1.0 and ISMA 2.0 media and protocols. The following section explains why ISMACryp only partially meets the interoperability goals for ISMA 1.0 and ISMA 2.0 receivers as described in Section 5.1.

5.4 ISMA DRM Interoperability

ISMA 1.0 targets two types of receivers, namely, ISMA-only receivers and MPEG-4 systems-capable receivers. "ISMA-only receivers" are defined here as receivers that are not MPEG-4 systems capable, i.e. cannot process the MPEG-4 signaling and control (elementary) streams that can accompany any MPEG-4 (elementary) media stream. "MPEG-4 systems-capable receivers" can process the MPEG-4 systems layer information as well as ISMA 1.0. ISMA 2.0 makes no such distinction; in ISMA 2.0 MPEG-4 systems elements are optional.

Interoperability with MPEG systems-capable receivers is achieved through an MPEG Initial Object Description (IOD) [14496-1] that conveys at least a minimal level of MPEG-4 systems signaling. The IOD is Base-64 encoded and included as an SDP attribute (i.e. the SDP mpeg4-iod or "SDP IOD").

ISMACryp is applicable to both types of receivers: ISMACryp extends SDP fmtp signaling, and extends the binary ISMA IOD inside the SDP message. The IOD signaling can be either “minimal” or “IPMP-X”. “Minimal” signaling serves to alert receivers that the presentation, or part of it, is protected. Receivers then refer to the SDP signaling for details. “IPMP-X” signaling conveys to terminals (that are capable of processing IPMP-X data) all the authentication/protection information carried in the SDP, and additionally can be used to carry extra data for increasing interoperability between different Key Management Systems. Terminals that are not capable of processing IPMP-X data will handle “IPMP-X” signaling as “minimal.” See Section 8 for details. When used with ISMA 2.0 the signaling will, as usual, be in the SDP; if MPEG-4 systems elements are present, they MUST contain appropriate signaling, as above.

The current level of interoperability that is provided by ISMACryp is limited. The presence of encryption and authentication implies the use of access controls, authentication and authorization. These mechanisms are undefined in ISMACryp. Thus, there is a requirement for interoperability at the level of authorization and key management. These mechanisms, moreover, will likely differ between client/server, broadcast, and conferencing applications as well as between streaming and file download. A streaming server, for

example, may use SSL authorization, authentication and access-control mechanisms, but SSL might be unsuitable for certain applications such as media broadcast. Thus, ISMACryp interoperability is KMS-specific, which means that interoperability between ISMA senders and receiver is guaranteed only if they use a common KMS with common mechanisms of authorization, authentication, and key management. The use of IPMP-X signaling may improve the situation, but this, as well as other aspects of KMS interoperability, is left for further study.

6.0 ISMA DRM Architecture

This section briefly reviews the architecture of the ISMA DRM Recommendations [ISMADRM] document to identify what parts are covered by ISMA 1.1 Encryption and Authentication.

Figure 6.0-1 summarizes the flow within the ISMA DRM architecture. The ISMA DRM scope in Figure 6.0-1 is the encryption of ISMA media and authentication of ISMA messages, which is labeled "ISMACryp" in the figure, and the signaling of ISMACryp, which is labeled "RTSP/SDP+."

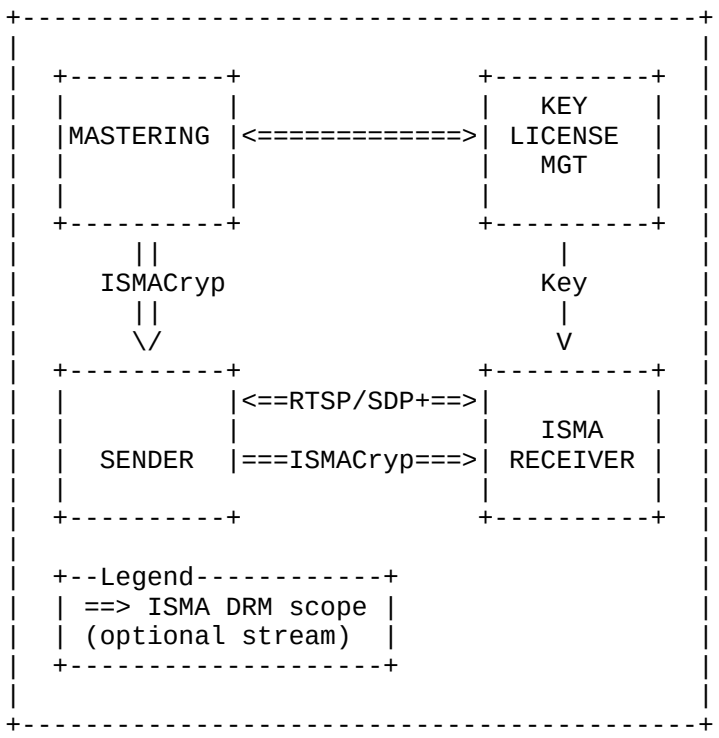


Figure 6.0-1: ISMA DRM Architecture

The Mastering entity is responsible for the preparation and publication of the content. The protocols for the key/license management interfaces are out of scope for this document (see Section 5.4). Annex A, however, suggests an Informative key management interface between mastering and key management. Annex E proposes an informative adaptation to OMA DRM v2 rights and key management [OMADRMv2].

In Figure 6.0-1, the delivery of the Key (and License) from the Key/License Management entity to the ISMA Receiver is out of scope for this document (see Section 5.4). It is assumed for the purposes of the discussion here that there are secure methods for the delivery of such information to the end devices. Section 8, however, gives a Normative interface for describing ISMACryp streams and encoding keys.

The Sender entity is responsible for the delivery of the content to the ISMA Receiver via an open-standard protocol called "ISMACryp," which MAY be signaled by the sender using RTSP/SDP+ (ISMA 1.0 and 2.0 SDP definitions plus ISMACryp signaling, see Section 8) or it MAY be signaled by a third device.

In the ISMA DRM architecture, the ISMA Receiver can process the ISMACryp encrypted streams, authenticated messages, and signaling. The following sections define "ISMACryp," the technology that

provides ISMA 1.0 and 2.0 media and protocols with encryption and message authentication/integrity services.

7.0 ISMA 1.1 Encryption and Authentication Framework

This section specifies the ISMA 1.1 Encryption and Authentication cryptographic framework for ISMA media and protocols. Nicknamed "ISMACryp" throughout this document, ISMA 1.1 Encryption and Authentication is a family of cryptographic media encodings and protocols. Section 10 specifies the default encryption and authentication transforms for ISMACryp. Sections 7, 8, 9, 10 form a complete specification for ISMACryp.

7.1 Receiver Architecture and End-to-End Flows

Figure 7.1-1 shows the ISMA Receiver architecture in more detail with interfaces to Key/License Management (KEY MGT), an RTSP control interface, and ISMACryp, the cryptographic services for ISMA data. The ISMACryp Receiver can decrypt, authenticate, and check the integrity of ISMA data.

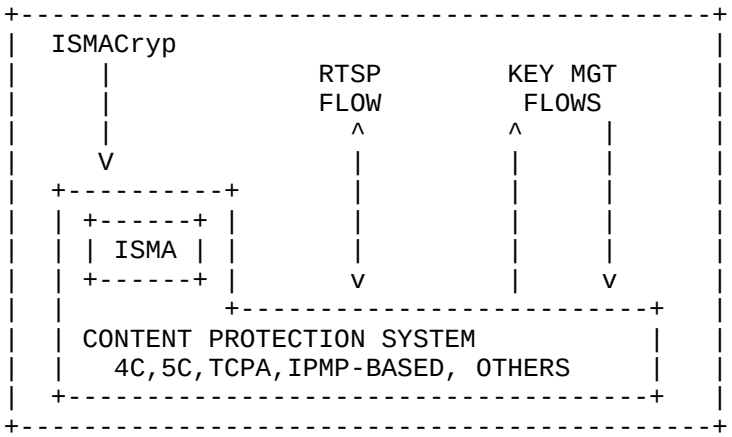


Figure 7.1-1: ISMACryp Receiver Architecture

Figure 7.1-1 also indicates the scope of the ISMACryp specification, which ends upon the decryption of the ISMA stream. The decoding and presentation of the decrypted media is according to ISMA 1.0 [ISMASPEC] or to ISMA 2.0 [ISMASPEC2]. Receiver content protection system and key management are out of scope.

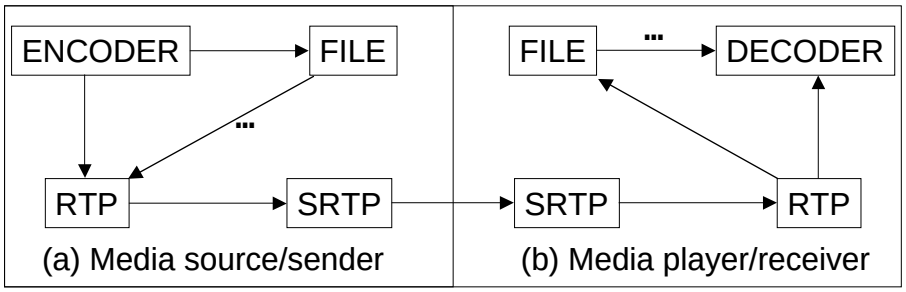


Figure 7.1-2: ISMACryp End-to-end Flows

Figure 7.1-2a shows the ISMACryp environment where a stream MAY be mastered into a file or streamed directly from an encoding application to a network. In both cases, encryption occurs prior to transport although message authentication SHOULD be performed at the transport layer. At the receiver (media player/decoder) of Figure 7.1-2b, a stream may be received into a file, such as a personal recorder at a player or cache server, or directly to a decoder. ISMACryp's transforms are applied at the arcs that

emanate from the Figure 7.1-2 ENCODER; decryption occurs at the arcs that terminate at the DECODER; message authentication transforms are applied at the SRTP sender and receiver shown in Figure 7.1-2.

The following sections describe the cipher and bit-stream specifications for the ISMACryp specifications.

7.2 Transforms

ISMACryp supports the replacement of its encryption and message-authentication (integrity) transforms. This section identifies the default transforms and references their specification, which is in Section 10. It's possible to replace the ISMACryp default encryption or message authentication transform with different ones: The present (framework) specification will not need revision but Section 10 would be replaced with a new transforms specification (i.e. in an ISMACryp version later than version 1.1, which is revision-controlled by ISMA). In some cases, existing transforms MAY be augmented. For example, the default ISMACryp transforms include only message authentication, but media authentication can be added to ISMACryp without necessarily replacing message authentication.

The ISMACryp modular design relies on external standards where appropriate and is therefore suitable for the greatest variety of network environments. Thus ISMACryp uses SRTP for message authentication of real-time media packets [RFC3711]. A modular design also permits use of a variety of key management and key establishment systems.

In addition to message authentication, there is also media authentication (see Section 2.0). Authentication of the rights holder that creates or publishes a content work - and validation of the integrity of that work - can be an important function for many applications. Typically, digests from hash functions will suffice for the integrity check, and digital signatures serve to authenticate the creator of the digest or catalog of published works [AMPBH]. This solution is efficient for files that are reliably delivered and for which a single hash of the contents is feasible. Such a hash cannot be computed when parts of the file are streamed over a lossy channel.

Whereas public-key cryptography (PKC) is suitable to authenticate media data in files [SMPEG], this is inefficient for packet data: Practical security systems generally do not use asymmetric cryptography for packets owing to the excessive per-packet overhead of digital signatures or public-key encryption. The packet-size and computation overhead are worse for media frames since there are often multiple frames to a packet. There are, however, two methods more efficient than PKC to authenticate stored and streamed media data independently of the message.

The first method is to authenticate each media frame using symmetric keys. Use of a message authentication code (MAC) is arguably feasible for large media frames, but these frames may be fragmented across transport packets thus causing additional complications. In fact, even symmetric authentication techniques are generally infeasible for small media frames since a message authentication code (MAC) can add 10 or more bytes to the length of each media frame [RFC2104]. This technique has orders of magnitude less overhead than PKC and is not excessive for large (~500 byte) video frames. But it is excessive for small frames and low bit-rate audio and video data.

The second method avoids the problem of authenticating fragments of media frames by pre-assigning media data into packet payloads, which are authenticated independently of the transport packet. This pre-assignment assumes that the size of the transport-packet payload is fixed prior to the time of transport.

Even the second choice, which has much less overhead than the first, usually doubles the amount of MAC data in each transport packet. This doubling is unavoidable if both message authentication and media authentication are desired. Of the two, message authentication is chosen as more important for several reasons. First, the receiver must trust the sender to have rights to disseminate the media and this trust relationship is realized in message authentication. Second, the authentication of the file creator can be no better than that of the sender whom the creator has authorized to disseminate the work. Third, it is not

necessarily the receiver's responsibility to ensure that what the sender sends is exactly what the file creator authorized it to send. Finally, the sender may have rights to alter the media in various ways. Thus, media authentication is redundant to message authentication for many practical applications. The preferred packet design of section 7.3, therefore, uses SRTP message authentication only and does not support media authentication independent of the SRTP message.

7.2.1 Default message authentication (integrity) transform

SRTP message authentication [RFC3711] SHOULD be used for ISMACryp messages. This transform is described in Section 10.

7.2.2 Default cipher and mode

The AES [AES] in Counter mode [AES-C, AES-CTR] SHALL be used as the encryption cipher. This cipher is described in Section 10.

7.3 Transport Packet Structure

This section defines the RTP payload formats for content encrypted according to this specification. The reader needs to be familiar with the mpeg4-generic packet format [MPEGIPTS] in order to understand the ISMACryp packet design.

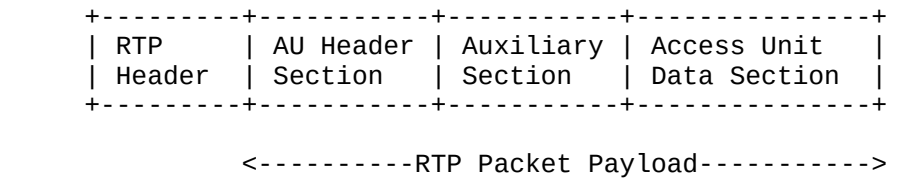


Figure 7.3-1: Data sections within an mpeg4-generic RTP packet

7.3.1 General principles

In RTP, a payload format is generally designed for the specific media being carried. ISMACryp 1.1 tries here to use a generic approach which is as much as possible codec agnostic.

For ISMA 1.0 payloads (MPEG-4 Video, AAC and CELP MPEG-4 Audio) and for ISMA 2.0 payloads (MPEG-4 AVC, AAC and HE-AAC MPEG-4 Audio), this specification defines RTP payload formats derived from the mpeg4-generic specification for MPEG-4 content [RFC3640]. It may also be adaptable to other codecs.

The ISMACryp payload type specified here tries to preserve optimal packetization and loss recovery while acknowledging that the packetization or the de-packetization processes might not have access to media data, which are encrypted at these stages. Hence media data are invisible to the packetization process except for access unit boundaries, which are available to the packetization process. Therefore, the ISMACryp payload type imposes specific restrictions at the access-unit level, such as allowing or disallowing fragmentation, or interleaving of access units. No restrictions that require access to unencrypted media are imposed here, e.g., requiring that video frames be split at video packet boundaries.

Thus, the ISMACryp RTP payload format is based on the RTP payload format defined in mpeg4-generic [RFC3640]. Support for encryption is added by defining a new payload type (see Section 8.3) that is upwardly compatible with mpeg4-generic and that supports different media codecs including MPEG-4 Part 2 Video, MPEG-4 Part 10 AVC, MPEG-4 AAC Audio, MPEG-4 HE-AAC and MPEG-4 CELP Audio.

7.3.2 Decryption process

Each packet is sent according to the ISMACryp payload format to the receiver, who must have all the needed cryptographic data to decrypt the packet. Upon decryption, the receiver must have all the needed information to process the mpeg4-generic packet. ISMACryp packetization accomplishes this by inserting an initialization vector (IV) in each packet, and the IV contains all the needed information to decrypt each access unit (AU) contained in the packet. How this is done is specific to the cipher, see Section 10 for operational details for the default cipher (where the IV is instantiated as a BSO). It is REQUIRED that ISMACryp ciphers accommodate the loss, delay, and reordering of a packet stream The IV contains all cryptographic data that are needed to make the decryption of a packet independent of previous or successive packets in a stream.

7.3.3 The enc-mpeg4-generic payload type

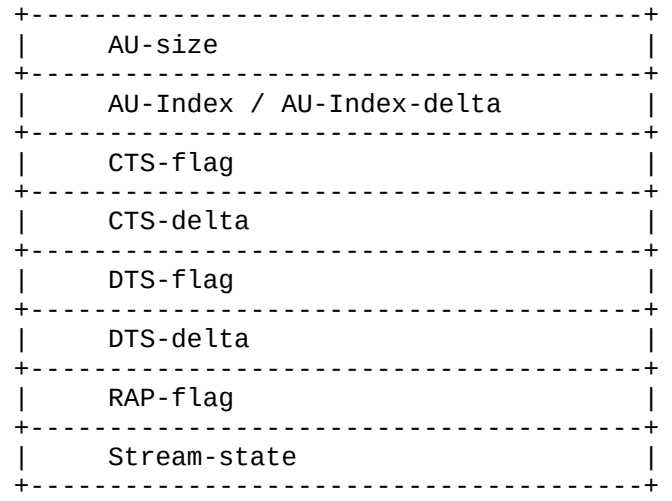
The RTP payload structure defined in this specification is based upon the RTP payload format defined in mpeg4-generic [RFC3640]. Support for encrypted media is added by adding new fields to the access unit (AU) header section.

Each ISMACryp RTP packet SHALL contain either:

1. Exactly one access unit,
2. Two or more complete access units, or
3. One fragment of an access unit.

7.3.4 The enc-mpeg4-generic header

The AU Header for each AU is defined in mpeg4-generic as follows:

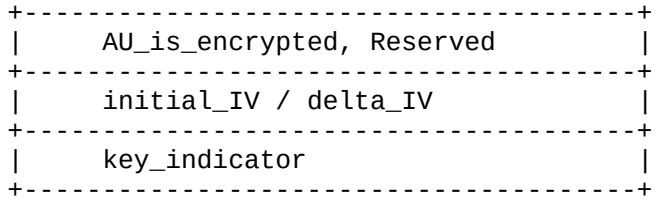


ISMACryp inserts cryptographic metadata at the beginning of each AU header. The format of the first AU header is different from the second and subsequent AU headers (similar to the treatment of AU-Index and AU-Index-Delta in mpeg4-generic). This block supplies the Cryptographic context for each AU or AU fragment in an RTP packet and is defined as follows:

```
class ISMACrypContextAU(int auNum).
{
        if (ISMACrypSelectiveEncryption) {
                bit(1)    AU_is_encrypted;
                bit(7)    Reserved;
```

```
        }
        else AU_is_encrypted = 1;
        if (auNum==0) // First AU in packet?
        {
                unsigned int(ISMACrypIVLength*8)                initial_IV;
                unsigned int(ISMACrypKeyIndicatorLength*8)      key_indicator;
        }
        else
        {
                int(ISMACrypDeltaIVLength*8)                            delta_IV;
                if (ISMACrypKeyIndicatorPerAU)
                        unsigned int(ISMACrypKeyIndicatorLength*8)      key_indicator;
        }
}
```

Reserved fields MUST have all bits set to zero though a compliant receiver MAY choose to ignore this field. Note that in bit(n), unsigned int(n) and int(n), n is always a bit count. Note also that delta_IV is the only signed int in the above definition. See Section 8.1 for the signaling of the parameter constants (ISMACrypSelectiveEncryption, ISMACrypIVLength, ISMACrypKeyIndicatorLength, ISMACrypDeltaIVlength, and ISMACrypKeyIndicatorPerAU). Diagrammatically, this means that these fields are inserted just before the AU-size field in the diagram above:



All "Reserved" fields MUST be zero and MAY be ignored by the receiver. Note that it is possible to compute the access unit count by using the configuration parameters, and the signaled length of the access unit headers. This is because the total bit-length of the AU-headers is given in each packet, and the length of the first AU Header as well as the second and subsequent AU-headers can be computed from the signaled parameters. This is still true with this extended AU header. So we have:

```
AU-count = (AU-header-length – first-header-size)/
                              subsequent-header-size + 1;
```

Note that this equation does not hold if either CTS or DTS timestamps are present; however, this normally applies only to video, and in that case, the payload format restricts the packet to containing only one AU or a fragment of an AU.

The fields in the ISMACrypContextAU structure have the following meaning.

AU_is_encrypted.

An optional single bit field to signal selective encryption. A 1 value signals that the corresponding access unit is encrypted, a value of 0 means it is not. The presence of this field is configured with the ISMACrypSelectiveEncryption parameter. All fragments of a single access unit SHALL have the same value for AU_is_encrypted.

key_indicator

Contains the key indicator for an access unit when ISMACrypKeyIndicatorLength is non-zero (Section 8.1). If the Section 8.1 ISMACrypKeyIndicatorPerAU is 0, then only the first access unit in a packet has an explicit key indicator value included in the cryptographic context; all subsequent access units SHALL have the same value for key_indicator as the first access unit. If ISMACrypKeyIndicatorPerAU is 1, then a value of key_indicator is included in the cryptographic

header for each access unit or fragment in the packet. If AU_is_encrypted is 0 for an access unit, then the value of this field is ignored.

initial_IV
: Contains the initial IV value for the first access unit or fragment contained in the packet. In most cases, this is the only IV in the packet. In some cases such as interleaved media, however, there MAY be an IV per AU. See Section 10 for initial_IV definitions for the default transform

delta_IV
: This field contains IV data on a per-AU basis when ISMACrypDeltaIVLength (see Section 8) is non-zero and the data are interleaved in packet payloads.

The actual IV to be used for each access unit in a packet is computed as follows, with the first access unit in a packet indexed as zero:

```
IV[0]   := AUHeader[0].Initial_IV;          // First AU in packet
IV[N+1] := IV[N] + AUSize[N]
                 + (ISMACrypDeltaIVLength == 0 ? 0 :
                        AUHeader[N+1].delta_IV)   // Subsequent
```

Note: The number of access units in a packet is not signaled in this payload format. The number of access units in the packet can be deduced from the access unit header as for unencrypted modes. A packet that has the M-bit cleared contains a fragment that is not the last of an AU. If the M-bit is set, then the packet has one or more access units or the last fragment of an access unit. The access unit header indicates whether there are two or more access units in the packet. To distinguish between one whole AU and the last fragment, compare the payload data size and the access unit size conveyed in the access unit header. The access unit size will be the size of the whole AU and not the fragment.

Note: In the simple case where there is one AU per packet, or the AUs are contiguous, this structure reduces to signaling a key indicator and an initial IV per packet.

Note: In the case of selective encryption, if the AU is not encrypted, the initial_IV/delta_IV and the key_indicator are still present but these values are not needed to the clients as AU is not encrypted.

7.3.5 All mpeg4-generic modes are inherited by enc-mpeg4-generic

The mpeg4-generic specification defines 5 "modes" of the RTP payload format. Four of these modes are used in ISMACryp as they are defined in ISMA 1.0 or in ISMA 2.0. Thus, the following modes are inherited from the ISMA specifications.

- **MPEG-4 AAC Low and High Bit Rates**. These two modes support encrypted MPEG-4 AAC (including HE-AAC).
- **MPEG-4 CELP CBR and VBR.** These two modes support encrypted constant and variable bit rate MPEG-4 CELP Audio.

7.3.6 The enc-mpeg4-generic encrypted MPEG-4 video mode

Unlike AAC and CELP, ISMA 1.0 MPEG-4 video uses the payload format defined in RFC 3016 [RFC3016]. To enable the carriage of encrypted video, a new mode is defined here for its carriage solely in the context of this encrypted payload format. This mode MUST not be used with the ISMA 1.0 payload format, mpeg4-generic, which is unencrypted video.

Encrypted MPEG-4 video Mode. This mode supports encrypted MPEG-4 Video (Part 1).

This mode is signaled by mode=mpeg4-video. In this mode, the configuration of the payload is as follows:

- SizeLength = 0. Each packet contains only one AU or AU fragment.
- IndexLength = 0 and IndexDeltaLength = 0. No interleaving.
- CTSDeltaLength = 0. No signaling of CTS needed as it is obtained from the RTP timestamp.
- DTSDeltaLength = 22. Video access units may have a DTS that differs from the CTS when B-frames are present, therefore the DTS can be signaled in this mode.
- RandomAccessIndication = 1. The RAP-flag is set to 1 to indicate video I-frames.
- Config. Must be present.

At most one access unit or fragment appear in an RTP packet. Multiple access units in a packet are not allowed. The rules for packetization specified in RFC 3016 are not required in this mode. An access unit may be split anywhere at all, without regard to video packet, video header, or any other boundaries.

For this mode, the AU header takes one of two sizes (not counting the cryptographic metadata):

1. If the DTS is equal to the CTS, the DTS-flag is set to zero. In this case, the AU header consists of only two bits (the DTS-flag followed by the RAP-flag), which is then followed by six padding bits.

2. If the DTS is not equal to the CTS, the DTS-flag MUST be set to 1. In this case, the AU header consists of 24 bits: the DTS-flag, the 22 bit signed integer DTS-delta, and the RAP-flag.

7.3.7 The enc-mpeg4-generic encrypted AVC video mode

Unlike AAC and CELP, ISMA 2.0 AVC video uses the payload format defined in RFC 3984 [RFC3984]. To enable the carriage of encrypted video, a new mode of RFC 3640 is defined here for its carriage solely in the context of this encrypted payload format. This mode MUST not be used with the ISMA payload format, mpeg4-generic, which is unencrypted video.

The media that is encrypted is transformed before encryption, in order to enable recovery of NAL units even if RTP packets are lost. The length-codes which are stored before each NAL Unit in the standard AVC file format are replaced by start-codes, so that each NAL Unit is now structured in a legal byte-stream format according to Annex B of the AVC specification [14496-10]. This process is safe since AVC specification [14496-10] required start-code emulation avoidance (even in length-field oriented applications).

Note: Byte-stream format [14496-10 annex B] allows two types of start-codes: "four bytes 00 00 00 01" and "three bytes 00 00 01". Since AVC File format [14496-15] does not permit 3-byte length fields, the byte-stream MUST use 4-byte start-codes (to enable easy transformation back into length fields after decryption). In other words, the fields "leading_zero_8bits" and "trailing_zero_8bits" MUST NOT be present and the field "zero_byte" MUST be present in each NAL unit. After decrypting the stream, two consecutive start-codes are needed to discover the length of each NAL unit.

Encrypted AVC video Mode. This mode supports encrypted AVC Video.

This mode is signaled by mode=avc-video. In this mode, the configuration of the payload is as follows:

- SizeLength = 0. Each packet contains only one AU or AU fragment.
- IndexLength = 0 and IndexDeltaLength = 0. No interleaving.
- CTSDeltaLength = 0. No signaling of CTS needed as it is obtained from the RTP timestamp.
- DTSDeltaLength = 22. Video access units may have a DTS that differs from the CTS when B-frames are present, therefore the DTS can be signaled in this mode.
- RandomAccessIndication = 1. The RAP-flag is set to 1 to indicate video I-frames.

- config : Must be present and is the hexadecimal value of `AVCDecoderConfigurationRecord` [14496-15]. In this structure, `LengthSizeMinusOne` indicates the length of start-codes and MUST be set to 3 (corresponding to 4-byte start-codes).
- At most one access unit or fragment appears in an RTP packet. Multiple access units in a packet are not allowed.

The rules for packetization specified in RFC 3984 are not required in this mode. An access unit may be split anywhere at all, without regard to video packet, video header, or any other boundaries.

For this mode, the AU header takes one of two sizes (not counting the cryptographic metadata):

1. If the DTS is equal to the CTS, the DTS-flag is set to zero. In this case, the AU header consists of only two bits (the DTS-flag followed by the RAP-flag), which is then followed by six padding bits.
2. If the DTS is not equal to the CTS, the DTS-flag MUST be set to 1. In this case, the AU header consists of 24 bits: the DTS-flag, the 22 bit signed integer DTS-delta, and the RAP-flag.

8.0 ISMA 1.1 Encryption and Authentication Signaling

All ISMA 1.1 Encryption and Authentication (ISMACryp) receivers recognize a special SDP attribute for "isma-compliance." ISMA 1.1 Encryption and Authentication does not change the isma-compliance "lowest-spec-version" from 1.0 nor from 2.0. ISMACryp is signaled, therefore, by means other than a new version number, and these means are the SDP fmtp and iod. The fmtp has new ISMACryp parameters and a new ISMACryp mode for video. Both are described in this Section.

8.1 Overview

ISMACryp signaling has session and stream signaling parameters. The stream signaling parameters describe the encryption of the stream.

1. Crypto suite: Identifies the cipher, mode, keylength, authentication algorithm, etc…
2. IV length: Describes the size of the initialization vector in bytes.
3. Key indicator length: Describes the size of the key indicator in bytes.
4. Selective encryption: Indicate whether selective encryption is used for the session or not.
5. Salt key: Initiates the salt key value used, given an additional IV, to generate the AES-CTR counter.
6. Key: A structure that describes the key management system or conveys the key for the stream.

Key indicator length and selective encryption are optional since ISMACryp streams are not required to rotate keys or have unencrypted media frames. Salt key is also optional; if non present, the salt key is directly managed with the same system as the main key.

In addition to the stream-signaling parameters, there are two optional transport parameters.

7. delta IV length: Describes the maximum size of the optional media-frames initialization vector.
8. Key indicator per AU: Indicates key rotation on a media frame basis.

The delta IV length parameter is needed when media frames are interleaved in packets and unneeded otherwise. Key indicator per AU is needed when the stream has multiple keys and the packetizer might rotate a key between two media frames that are in the same packet. Key rotation and other techniques can be used by a Key Management System (KMS) for both MPEG and non-MPEG (i.e. ISMA-only in Section 5.4) categories of receivers, which are defined next, in Section 8.1.1.

8.2 Receiver categories and signaling schemes

There are three categories of receivers that are considered in this context:

1. ISMA-only receiver: A receiver that plays back streamed presentations of MPEG audio/video or other media and ignores any MPEG4 Systems [14496-1] related information
2. MPEG-receiver: A receiver that plays back streamed or locally stored presentations using MPEG4 Systems information in both cases. This receiver, however, cannot process IPMP-X data, although it can identify IPMP-X descriptors and skip the parts in them that it does not understand.
3. IPMP-X receiver: An MPEG-receiver that is also capable of fully parsing and processing IPMP-X descriptors.

Three signaling schemes are defined to accommodate all three categories. The first two categories require the SDP fmtp scheme. The second category requires also the “minimal” MPEG scheme in order to signal the receiver that the presentation is protected. The third category uses the IPMP-X scheme All senders MUST implement the SDP fmtp scheme **and** either the (minimal) MPEG scheme or the IPMP-X scheme.

8.3 Session Description Protocol Signaling

This section defines SDP [RFC2327] fmtp signaling for ISMA-only and MPEG receivers. Some of this signaling is redundant for IPMP-X receivers but nevertheless MUST be present in all cases.

The SDP fmtp signaling SHALL use enc-mpeg4-generic as its format. All ISMACryp SDP signaling parameters and names are case-insensitive.

Generic SDP signaling:

```
m=<media> <port>/<number of ports> <transport> <fmt list>
a=rtpmap:<payload type> <encoding name>/<clock rate>[/<encoding parameters>]
a=fmtp:<payload type> mode=<mode>; <MPEG4-GENERIC-PARMS> <ENC-MPEG4-GENERIC-
PARAMS>


<media> = "audio"|"video"
<transport> = "RTP/AVP"|"RTP/SAVP"
<encoding name> = "enc-mpeg4-generic"
<mode> = "aac-hbr"|"aac-lbr"|"celp-cbr"|"celp-vbr"|"mpeg4-video"|"avc-video"
```

MODE is MANDATORY and is defined in mpeg4-generic [RFC3640] or it is "mpeg4-video" as defined in Section 7.3.6 for MPEG-4 video streams, or it is "avc-video" as defined in Section 7.3.7 for AVC video streams. For avc-video the NAL Unit structure MUST be transformed into byte-streams prior to encrypting and back into length fields after decrypting.
MPEG4-GENERIC-PARMS are OPTIONAL parameters that are defined in mpeg4-generic [RFC3640].
ENC-MPEG4-GENERIC-PARMS are OPTIONAL and defined below.

Note: the parameters in the fmtp line (mode=<mode>; <MPEG4-GENERIC-PARMS> <ENC-MPEG4-GENERIC-PARAMS>) may appear in any order.

Table 8.3.2: enc-mpeg4-generic fmtp parameters

DESCRIPTOR	DEFINED VALUES	DEFAULT
ISMACrypCryptoSuite	AES_CTR_128[1]	AES_CTR_128[1]
ISMACrypIVLength	0..8	4[1]
ISMACrypDeltaIVLength	0..2	0
ISMACrypSelectiveEncryption	0 (False) or 1 (True)	0
ISMACrypKeyIndicatorLength	0..255	0
ISMACrypKeyIndicatorPerAU	0 (False) or 1 (True)	0
ISMACrypSalt	Base64 encoded 64-bit number	0
ISMACrypKey	(type) string	""
ISMACrypKMSID	4CC	
ISMACrypKMSVersion	unsigned int 32	
ISMACrypKMSSpecificData	quoted-string	""

ISMACrypCryptoSuite identifies the default cipher, mode, key length and other descriptors used to describe the encryption of ISMA media (see Section 10). AES-CTR is the default and mandatory-to-implement cipher and mode, see Section 10 of this document.

ISMACrypIVLength describes the byte length of the initialization vector that is conveyed initially in the ISMACryp packet. For the default cipher and mode, this is the BSO value (see Sections 2 and 10).

ISMACrypDeltaIVLength describes the byte length of the initialization vector, if any, that is conveyed with an individual AU. See Section 10 for the encoding used for the default AES counter value.

[1] See Section 10

ISMACrypSelectiveEncryption declares that the media stream uses selective encryption when it is set to 1, which indicates that the selective encryption bit will appear in the ISMACryp header.

When ISMACrypKeyIndicatorLength is non-zero, a key indicator will appear in the ISMACryp header. ISMACrypKeyIndicatorLength can signal a key indicator field that is 0 to 255 bytes in length.

When ISMACrypKeyIndicatorPerAU is non-zero, a key indicator number appears on a per-AU basis.

ISMACrypSalt initializes the salt key with a randomly generated and non-null value that will be used for counter generation in the entire media stream (see Section 10). Unless specifically supplied by the KMS, the default salt value is 0.

ISMACrypKey identifies the key to the receiver. The parameter "type" is either "URI," "IPMP," or "KEY." "URI" is a URI for the key management system, which identifies the key or a location from which to obtain the key [RFC1630]. Thus, a uniform resource identifier follows the type "uri" as in *ISMACrypKey=(uri)https://example.isma.tv*. "IPMP" indicates that the key management signaling is done via IPMP-X, and nothing follows the type as in *ISMACrypKey=(ipmp)*. "KEY" is an encoding of the key used to decrypt the stream. This encoding is specific to the encryption transform. Section 10 defines the ISMACryp default encoding for ISMACrypKey. When ISMACrypKey is not explicitly signaled, it is an empty string, meaning that there is no key in the SDP, the key will be delivered from the key management entity (e.g. Conditional Access System or OMA DRM v2).

ISMACrypKMSID is the identification of the KMS used.

ISMACrypKMSVersion is the version of the KMS used.

ISMACrypKMSSpecificData contains information necessary to the kms such (e.g. a KMS Content ID).
Note :

- quoted-string = (<"> *(qdtext) <">)
- qdtext = <any TEXT except <">>
- TEXT = <any OCTET except CTLs>
- OCTET = <any 8-bit sequence of data>
- CTL = <any US-ASCII control character (octets 0 - 31) and DEL (127)>

The signaling message SHOULD be authenticated when carrying these parameters and SHOULD be encrypted when the ISMACrypKey parameter appears with an encoded key. In addition to signaling the encryption of the ISMA stream, ISMACryp defines signaling for the authentication of ISMA messages. See Section 10 for the default message authentication (integrity) transform for ISMACryp and its signaling method.

For examples of fmtp statements, see Annex F.

The ISMACrypKey=(uri) in ISMACrypKey serves as a string identifier for the key. The ISMACrypkey=(key) parameter, however, is the key itself, base-64 encoded, and with an optional lifetime parameter. Since the key is crypto-suite dependent, it is defined by the particular crypto-suite (see Section 10). Despite the above example, there is no reason to specify defaults in the fmtp parameters and this practice is NOT RECOMMENDED (e.g., there is no reason for ISMACrypCryptoSuite=AES_CTR_128 since it does not change the configuration and takes up space in the SDP message).

It is also possible to convey several keys in parallel to allow key renewal using an extended key signaling: ISMACrypKey = (key) BASE64(aes-key1||salt1)|lifetime1|KI1,BASE64(aes-key2||salt2)|lifetime2|KI2,BASE64(aes-key3||salt3)|lifetime3|KI3;

Therein, the KI1 is the corresponding key indicator of aes-key1 of a lifetime1 and a salt1, KI2 is the corresponding key indicator of aes-key2 of a lifetime2 and a salt2, KI3 is the corresponding key indicator of aes-key3 of a lifetime3 and a salt3…

8.4 IPMP Signaling

This section defines SDP/IOD minimal MPEG signaling and IPMP-X signaling. ISMA-only receivers accept only SDP fmtp signaling parameters but the SDP IOD MUST signal any MPEG receiver that the stream has ISMACryp protections (signaling is done by either Minimal or IPMP-X signaling). The KMS MAY signal ISMACryp parameters and other data by using IPMP-X signaling in the SDP IOD. Minimal signaling is specified in 8.4.1 and IPMP-X signaling is specified in 8.4.2.

8.4.1 Minimal MPEG

When Minimal signaling is used the following IPMP information MUST be included in the ISMA IOD for MPEG-receivers. The ES Descriptor of each encrypted ES MUST contain the following IPMP_DescriptorPointer [14496-1]:

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
		IPMP_DescriptorPointer	
1	8	IPMP_DescriptorPointer tag	10
2	8	descriptor size	1
3	8	IPMP_DescriptorID	1

The OD stream must contain the following IPMP_DescriptorUpdate in the first access unit:

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
		IPMP_DescriptorUpdate	
1	8	IPMP_DescriptorUpdate tag	5
2	8	descriptor size	5
		IPMP_Descriptor	
3	8	IPMP_Descriptor tag	11
4	8	descriptor size	3
5	8	IPMP_DescriptorID	1
6	16	IPMPS_Type	0x4953

See Section 11 for name bindings.

8.4.2 IPMP-X Signaling

For the IPMP-X systems-capable receiver, the following IPMP-X signaling SHOULD be used as a basis for a more capable MPEG IPMP signaling.

IPMP-X defines an IPMP Tool List Descriptor in IOD, which identifies a list of required IPMP Tools to process the protected content. IPMP-X terminal uses this information to make sure all required tools are retrieved and are ready to use before attempting to play the content.

In the scope of ISMACryp, the ISMA decryption tool MUST always be listed in the Tool List with the tool ID that is 0x4953 (see Section 12), as shown below. If an IPMP-X KMS tool is used, it may be included in the Tool List as well.

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
1	8	IPMP_ToolListDescTag	0x60
2	16	Descriptor size	0x13
		IPMP_Tool	
3	8	IPMP_ToolTag	0x61
4	8	Descriptor size	0x11
5	128	IPMP_ToolID	0x4953
6	1	isAltGroup	0
7	1	isParametric	0
8	6	reserved	0b0000.00

The ES Descriptors of each encrypted elementary streams MUST contain the following IPMP_DescriptorPointer [14496-13]:

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
		IPMP_DescriptorPointer	
1	8	IPMP_DescriptorPointer tag	10
2	8	descriptor size	5
3	8	IPMP_DescriptorID	0xFF
4	16	IPMPX_DescriptorID	0x0001
5	16	IPMP_ES_ID	0x0000

The OD stream MUST contain at least one IPMP_DescriptorUpdate at time 0. This update MUST contain an IPMP Descriptor, to which the descriptor pointer above points. The following IPMP Descriptor is used to signal the protection of ISMACryp decryption tool, inside of which the ISMACryp_Data is extended from IPMP-X's IPMP_Data_BaseClass with DataTag of 0xD0 to carry ISMACryp decryption parameters.

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
		IPMP_DescriptorUpdate	
1	8	IPMP_DescriptorUpdate tag	5
2	8	descriptor size	36
		IPMP_Descriptor	
1	8	IPMP_Descriptor tag	11
2	8	descriptor size	34
3	8	IPMP_DescriptorID	0xFF
4	16	IPMPS_Type	0xFFFF
5	16	IPMP_DescriptorIDEx	0x0001
6	128	IPMP_ToolID	0x4953

7	8	ControlPointCode	0x01 (between the decode buffer and the decoder)
8	8	SequenceCode	0x80
		ISMACryp_Data	
9	8	ISMACryp_DataTag	0xD0
10	8	data size	9
11	8	Version	0x01
12	32	dataID	0x4953
13	8	Crypto-suite	Identifies encryption and authentication transforms
14	8	IV-length	Byte length of the initialization vector
15	1	Selective-encryption	1 if selective encryption is used
16	7	Reserved	MUST be zero
17	8	Key-indicator-length	Byte length of the key indicator

The possible values for each parameter, and its default value.

Table 8.4.2: IPMP parameters

DESCRIPTOR	DEFINED VALUES	DEFAULT
CryptoSuite	1..255	1 (AES_CTR_128)
IVLength	1..8	4
SelectiveEncryption	0 (False) or 1 (True)	0
KeyIndicatorLength	0..255	0

Note: The salt key value (see section 8.1) MAY be initiated thanks to an OPTIONAL salt-key parameter. The salt-key parameter can be added in the data class ISMACryp_Data if necessary.

9.0 ISMA 1.1 Encryption and Authentication MPEG-4 File Structure

The file format transformation supports the encryption of files either for local playback (including file download) or prior to hinting for streaming. The transformation is self-contained; all the information needed to either play the file, or hint it for streaming (including generating SDP information) is in the file. This does not mean, of course, that the file contains, for example, all the keys; but it does mean that enough information MAY be included to identify the KMS used and to enable a client to contact it and acquire the correct set of keys. The file transformation involves:

a) transforming the media samples themselves (encrypting them);
b) transforming the description of the media samples, both to document the transformation of the media samples, and to avoid the encrypted samples being read as if they were in the clear.

The file may be optionally hinted, and the extra signaling MUST be generated in the hint tracks.

9.1 General principles

This section documents the general format of encrypted media and the principles applied. The specific cases that apply to the ISMA 1.0 & 2.0 media formats and initial encryption scheme are then described later.

9.1.1 Sample transformation

In encryption, the samples are transformed – encrypted – so that the underlying media cannot be accessed by readers without the appropriate information (e.g. keys). The format of the encrypted samples is "owned" and documented by the encryption system.

9.1.2 Sample description transformation

The purpose of the sample description transformation is twofold: The sample description prevents accidental treatment of encrypted data as if it were un-encrypted and documents the transforms applied.

The documentation of the encryption scheme and its parameters is supplied in a uniform way. Note that in the following definitions that "n" in bit(n), unsigned int(n) and int(n) is always a bit count.

9.1.2.1 Transformation procedure

The transformation of the sample description is described entirely by the following procedure:

1. The 4CC of the sample description is replaced with a 4CC indicating the encryption: ‘mp4v’ or '264b' are replaced with ‘encv’ and ‘mp4a’ is replaced with ‘enca’.

In the case of 'avc1' stream, the ‘original format’ indicator for the stream MUST be changed from ‘avc1’ to '264b’. Unencrypted streams MUST NOT be stored in files using the byte-stream format as this is not the standard format for AVC. The ‘avc1’ sample-entry name MUST NOT be used for byte-stream structured AVC, either as a sample-entry name or as an original-format name.

Note: Byte-stream format [14496-10 annex B] allows two types of start-codes: "four bytes 00 00 00 01" and "three bytes 00 00 01". Since the AVC File format [14496-15] does not permit 3-byte length fields, the byte-stream MUST use 4-byte start-codes (to enable easy transformation back into length fields after decryption). In other words, the fields "leading_zero_8bits" and "trailing_zero_8bits" MUST NOT be present and the field "zero_byte" MUST be present in each NAL unit. After decrypting the stream, two consecutive start-codes are needed to discover the length of each NAL unit.

2. In the AVCDecoderConfigurationRecord, the LengthSizeMinusOne field indicates the length of start-codes and MUST be set to 3 (corresponding to 4-byte start-codes)

```
aligned(8) class AVCDecoderConfigurationRecord {
      unsigned int(8) configurationVersion = 1;
      unsigned int(8) AVCProfileIndication;
      unsigned int(8) profile_compatibility;
      unsigned int(8) AVCLevelIndication;
      bit(6) reserved = '111111'b;
      unsigned int(2) lengthSizeMinusOne = 3;
      bit(3) reserved = '111'b;
      unsigned int(5) numOfSequenceParameterSets;
      for (i=0; i< numOfSequenceParameterSets; i++) {
            unsigned int(16) sequenceParameterSetLength ;
            bit(8*sequenceParameterSetLength)
      sequenceParameterSetNALUnit;
      }
      unsigned int(8) numOfPictureParameterSets;
      for (i=0; i< numOfPictureParameterSets; i++) {
            unsigned int(16) pictureParameterSetLength;
            bit(8*pictureParameterSetLength)
      pictureParameterSetNALUnit;
      }
}
```

3. A ProtectionInfoBox (defined below) is appended to the sample description, leaving all other boxes unmodified.

The Protection Info Box contains all the information required both to understand the encryption transform applied and its parameters, and also to find other information such as the kind and location of the key management system. It also documents the original (unencrypted) format of the media. The Protection Info Box is a container Box.

```
aligned(8) class ProtectionInfoBox(fmt) extends Box('sinf') {
   OriginalFormatBox(fmt)  original_format;
   SchemeTypeBox           scheme_type;
   SchemeInformationBox    info;
}
```

9.1.2.2 Original format

The Original Format Box 'frma' contains the 4CC of the unencrypted sample description:

```
aligned(8) class OriginalFormatBox(codingname) extends Box ('frma') {
   unsigned int(32)  data_format = codingname;
               // format of decrypted, encoded data
}
```

9.1.2.3 Protection scheme

The Scheme Type Box ('schm') specifies the encryption scheme.

```
aligned(8) class SchemeTypeBox extends FullBox('schm', 0, flags) {
   unsigned int(32)  scheme_type;         // 4CC identifying the scheme
   unsigned int(32)  scheme_version;      // scheme version
   if (flags & 0x000001) {
      unsigned int(8)   scheme_uri[];     // browser uri
   }
}
```

The scheme_type is the 4CC defining the protection scheme (i.e. "iAEC" in ISMACryp, see Section 9.2). The scheme_version is the version of the scheme used to create the content. The scheme_uri optionally directs the user to a web-page if they do not have the scheme installed on their system. It is an absolute URI formed as a null-terminated (C Programming Language) string in UTF-8 characters.

9.1.2.4 Scheme Information

The Scheme Information Box is a container Box that is only interpreted by the scheme being used. Any information the protection system needs is stored here. The content of this box is a series of boxes whose type and format are defined by the scheme declared in the SchemeTypeBox.

```
aligned(8) class SchemeInformationBox extends Box('schi') {
   Box   scheme_specific_data[];
}
```

9.1.2.5 MPEG-4 streams

For MPEG-4 streams, appropriate IPMP descriptors are also added to the ESDescriptor, as defined in the signaling section of this specification. Note that all ISMA 1.0 streams are indeed MPEG-4 media streams.

9.2 ISMA Encryption

In the definitions which follow, the value n in bit(n), unsigned int(n) and int(n) is always a bit count.

9.2.1 Encryption scheme

The AES-CTR-128 mode (Section 10) used by ISMA uses the scheme-type "iAEC" in the scheme-type box with a version of 1 for ISMACryp, which is revision-controlled by ISMA.

9.2.2 Encryption information

This section describes for the ISMA scheme how to convey similar parameters to some of those in Table 8.3.2. For example, following information is required in the scheme information box:

a) the identification of the Key Management System (KMS) used, its URI and KMS version
b) the description of the format of the samples when in the file format; this includes the presence of a selective-encryption indicator, the size of the key_indicator, and the size of the initial-offset.

The KMS supplies the keying material. The "string" used for KMS_URI below is a null-terminated string.

```
aligned(8) class ISMAKMSBox extends FullBox('iKMS', version, 0) {
   if (version==0) {
      string   KMS_URI;       // the KMS URI which the hinter or server
                              // MAY add to the ISMACryp SDP
information
   } else { // version ==1
      unsigned int(32)  KMS_ID;           // 4CC identifying the KMS
      unsigned int(32)  KMS_version;      // KMS version
      string   kms_URI;       // the KMS URI which the hinter or server
                              // MAY add to the ISMACryp SDP
information
}
```

Note : Writers should use version 0 to be compatible with ISMACryp 1.0 readers and should use version 1 if extended KMS information is needed.

```
aligned(8) class ISMASampleFormatBox extends FullBox('iSFM', 0, 0) {
   bit(1)   selective_encryption;         // see Section 8.1
```

```
   bit(7)   reserved;                    // MUST be zero
   unsinged int(8)   key_indicator_length;// see Section 8.1
   unsigned int(8)   IV_length;           // see Section 8.1
}
```

Other additional information about the ISMA scheme may be required. It is possible to add an ISMACrypSaltBox to convey the salt key used in the media encryption. This parameter is similar to the fmtp parameter defined in Table 8.3.2. This is an OPTIONAL sub-box of the scheme-information box.

```
aligned(8) class ISMACrypSaltBox extends Box('iSLT') {
   unsigned int(64)  salt;       // see Section 8.1, MUST be non null
}
```

9.2.3 Sample transformation

ISMACryp adopts the approach of embedding ISMACryp signaling information (Section 8.1) in the media data. While this scheme has redundant data, which is bad, it does not require redefinition of the MP4 file format. In this scheme, the samples are encrypted using a key from the key-set and using the ISMACryp default encryption in ISMACryp or some other encryption transform (in a future ISMACryp version). In order to permit random access, editing, and hinting, without scanning the file, we add to each sample the following parameters as defined in Section 8.1:

a) selective encryption indicator;
b) key indicator;
c) the initialization vector (for the ISMACryp default encryption, the initial counter value).

```
aligned(8) class ISMASample {
   if (selective_encryption == 1) {// from the sample description
      bit(1)   sample_is_encrypted;
      bit(7)   reserved;      // must be zero
   }
   else sample_is_encrypted = 1;
   if (sample_is_encrypted==1) {
      unsigned int(8 * IV_length)               IV;
      unsigned int(8 * key_indicator_length)    key_indicator;

   }
   unsigned int(8)   data[];  // encrypted media data, to end of sample
}
```

When selective-encryption (from the ISMASampleFormatBox) is zero, there is no storage allocated for the fields sample-is-encrypted and Reserved. When no storage is allocated for the sample-is-encrypted field, the "else sample-is-encrypted = 1" refers to a local variable named "sample-is-encrypted" and not the field named "sample-is-encrypted". If key_indicator_length is zero (0), then the key_indicator is also always zero (0). This means a single key is being used for the stream.

10.0 ISMA 1.0 & 2.0 Encryption (Default) Cryptography Specification

This section describes the default encryption and authentication transform for ISMA 1.1 Encryption and Authentication (ISMACryp). Future specifications MAY define new encryption and/or authentication transforms to supercede or augment these definitions. Section 10.1 describes the ISMACryp encryption transform and 10.2 describes the ISMACryp message authentication (integrity) transform.

10.1 ISMA 1.0 AES-CTR Encryption Transform

Section 10.1.1 defines the cipher, mode, and key length. 10.1.2 defines needed transport packet fields. Section 10.1.3 defines signaling parameters.

10.1.1 Default cipher, mode, and configuration

The AES blocksize is 128 bits and so is the counter, which is defined below. The key length SHALL be 128 bits. The 128-bit key, blocksize, and counter describe the AES-CTR cipher exactly. The AES block cipher encrypts the counter to form a pseudorandom block of 128 bits; successive AES blocks form a stream of AES-encrypted blocks called the "keystream." AES-CTR generates a stream of pseudo-random blocks based on AES encryption; encryption is done by exoring those bytes with plaintext to encipher and exoring those bytes with ciphertext to decipher.

Figure 10.1-1 shows AES in Counter Mode with a 128-bit counter, key, and a 128-bit keystream block. The figure assumes "big-endian" order where the left-most bit or byte is the most significant. The "div" denotes truncated integer division that is effectively a logical right shift of n bits for "div 2^n" with zero bits moved into the shifted positions. The exclusive-or operation, designated by "(*)" in Figure 10.1-1, is bit-wise exclusive-or, which is applied to the keystream and a 128-bit block of plaintext to produce a block of ciphertext, or it is applied to a 128-bit block of ciphertext to produce the original block of plaintext.

Each plaintext byte corresponds to one and only one BSO (see Section 2, Glossary) for the stream. The BSO information is included in the packet and called the "IV" or initialization vector. The encryption operation is performed using the BSO whereas decryption is performed using the IV, which is the BSO value of the first byte of payload data. This section generally uses only one term, "IV," and it should be understood to mean the BSO when the operation is encryption. The IV starts at the first byte in the packet payload when the packet does not contain interleaved AUs. When the packet contains interleaved AUs, an IV is associated with each AU.

Given the IV, salt and key, the counter is formed using the IV and a 64-bit salting key, k_s, which is left shifted 64 bits and right-padded with zeros. As shown in Figure 10.1-1, the IV is exored to salting key k_s. The multiply by 2^64 effectively shifts k_s left by 64 bits and could be written as "<< 64" using C programming language notation; the div operation is truncated integer division and could be written as ">> 4." Since the BSO and corresponding IV are stream byte-counters, the div is by 16 since there are 16 bytes (octets) in a 128-bit number, i.e. the div operation yields that keystream block that contains the IV byte. The result of the exor operation is the counter as shown in Figure 10.1-1.

AES-CTR [AES-CTR] defines the value shown as "counter" in Figure 10.1-1 as a "nonce" meaning that it is used once and only once in the counter space for a given key. A value of the counter MUST NOT be repeated for a given key or else the security of AES-CTR is compromised. The IV carries the unique value that forms the counter. For ISMACryp default encryption, the BSO SHALL be incremented for every byte in the stream regardless of whether that byte is encrypted or selectively unencrypted. The BSO is conveyed as an IV value in an ISMACryp packet (Section 7). The length of the IV is application dependent and MAY be signaled at the start of the session and SHALL remain fixed for the duration of the session (see 10.1.2 and 10.1.3). AES-CTR places no restriction on the nonce/counter to be randomized in any way. Indeed, some scholarly work proves superior security for AES-CTR without assuming that the nonce is randomized (e.g. starts from a random offset and incremented from there modulo the 128-bit blocksize, see

[SECURITY]). Some experts assert, however, that the counter needs to be initialized from a random offset within the counter space to prevent a "key-collision attack" [MF00]. This is the reasons that the default mode and cipher adds a salting key (k_s) random offset to the AES counter.

In addition to the counter there is the AES key, which is 128 bits. This key MUST be secret and difficult for an adversary to guess (there MUST be 128 bits of randomness in a 128-bit key). The key is denoted as "k" in Figure 10.1-1 and is used to encrypt the counter for the encryption and decryption operations, which are symmetric. The AES key SHOULD NOT be used to encrypt more than one ISMACryp stream owing to the danger of counter reuse [AES-CTR].

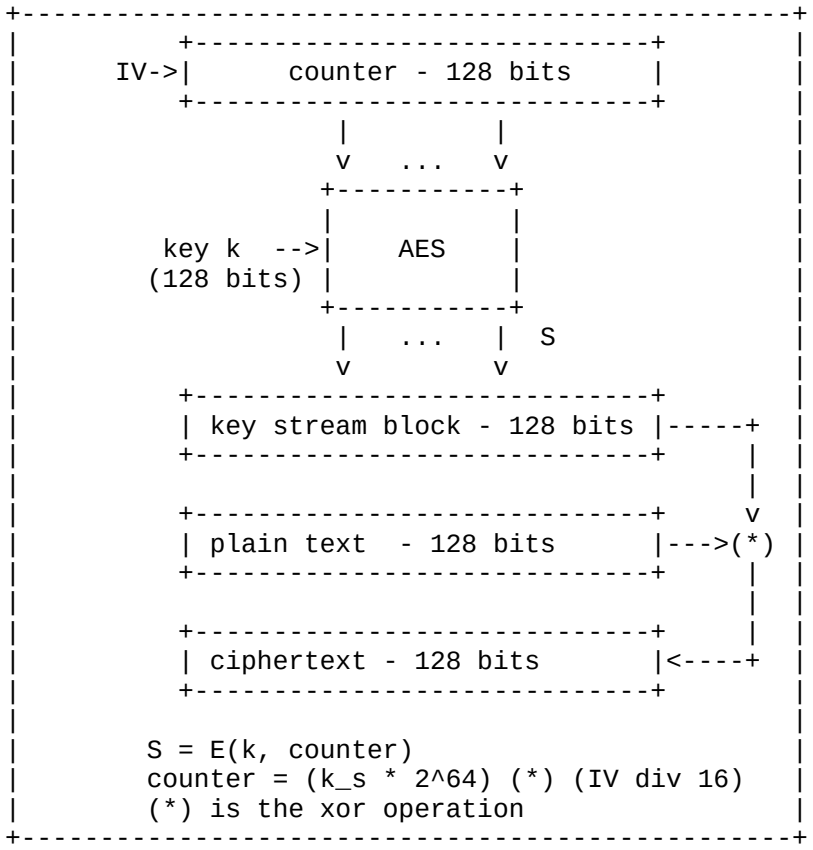


Figure 10.1-1: AES-CTR Mode Encryption

As shown in 10.1-1, the key stream is a sequence of 128-bit blocks (the figure shows just one of them) that are exored to the plaintext to encrypt and the ciphertext to decrypt. The encrypted byte stream is therefore dependent on the cipher key, the salting key (the non-secret random offset in the counter space) and the counter. With this, we have the following useful properties:

- The keystream may be pre-computed prior to the processing of the cleartext (on the encryption side) or the encrypted text (at the decryption side).
- The data are encrypted in 128-bit blocks, which are cryptographically independent of each other; thus it is not necessary to have a previous or subsequent block of cleartext for encryption or ciphertext for decryption of a given block of data.
- There is zero expansion of the data: Each ciphertext byte is in 1:1 correspondence with the plaintext byte.

- From the preceding bullets, the keystream is "seekable" to any given byte (BSO) of clear or cipher text, where the initial block and byte of the keystream for the packet-payload AU can be computed from the IV as

  keystream-block = IV div 16
  keystream-byte = IV mod 16

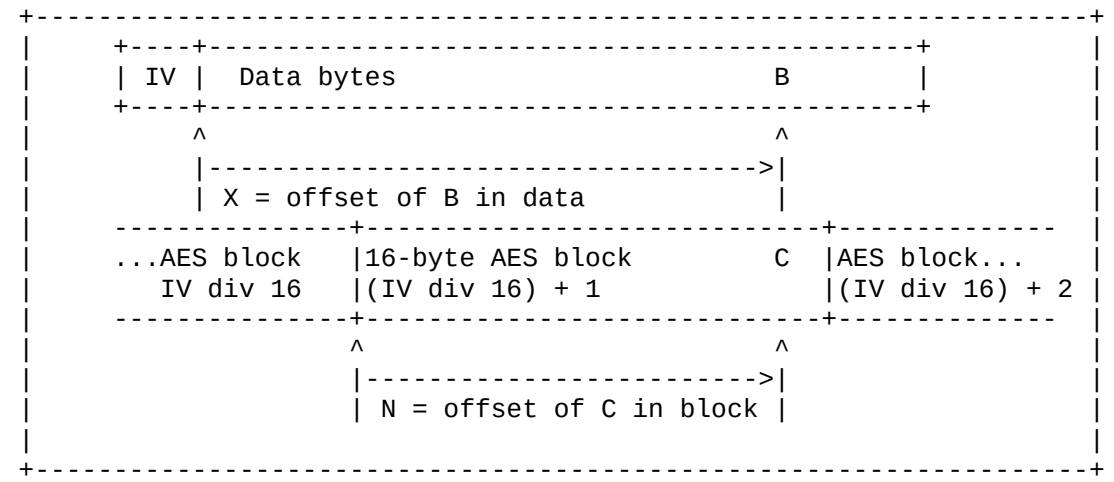


Figure 10-1.2: Locating a keystream byte within the packet payload

Whereas "IV div 16" and "IV mod 16" locate the first byte of keystream in a packet payload AU from the IV (or delta IV), Figure 10-1.2 shows how any byte within the payload can be matched to the corresponding byte of keystream that was used to encrypt it. Given the counter, the keystream is "seekable" to any given byte of cleartext or ciphertext in the ISMACryp packet payload. The counter is formed from the Initialization vector (IV) or vectors in the packet (one per non-interleaved packet or one per AU in an interleaved packet). In the diagram above, consider byte B which is at offset X after the periodically-supplied initialization vector IV. It is encrypted by exoring it with byte C, which is at offset N in a 128-bit AES keystream block. The counter value for that AES keystream block, and the offset N within that AES keystream block are given by:

```
counter =  k_s * 2^64 XOR ((IV + X) div 16)
N       = (IV + X) mod 16
```

The AES-CTR mode decryption has the same structure as its encryption counterpart except that the key stream block is exored with the cipher text block in order to obtain the plain text.

Note: The IV SHOULD start from zero and MUST be reset before it overflows. Keys SHOULD be changed as well at least once between every IV reset to avoid AES counter reuse. Specifically, for every access unit, the value of the access unit's IV plus the length of the access unit in bytes MUST NOT exceed (1 << (8 * IV_Length)). This restriction avoids an ambiguity as to whether the AES counter would continue to increment or would wrap around to zero after the largest possible IV value is reached within the access unit.

10.1.2 Fixed parameters and signaling values

Parameters describing packet fields that are an integral number of bytes are stated in bytes (octets) rather than bits. There are several fixed parameters for the ISMACryp default encryption transform.

1. AES_CTR_128 is the AES-CTR encryption cipher and mode defined in this section with a 128-bit key, a 128-bit blocksize and a 64-bit salting key.
2. The Salt-Key length is 8 bytes.

3. The AES block size is 16 bytes.
4. The AES key length is 16 bytes.

There are a few parameters that MAY be set through signaling.

5. ISMACrypCryptoSuite MAY be set to AES_CTR_128, or it MAY default to this value.
6. ISMACrypIVLength defaults to 4 bytes but MAY be set to any value between 1 byte and 8 bytes.

ISMACrypKey MAY carry a URI that identifies a key server when its type is "URI," or it MAY indicate that the MPEG IOD carries all key-management information when its type is "IPMP." When its type is "KEY," ISMACrypKey encodes one or several inline ISMACryp decryption key(s) as follows :

BASE64(aes-key||salt)|lifetime|KI{,BASE64(aes-key||salt)|lifetime|KI}*

Thus, when its type is "KEY," ISMACrypKey is a UTF-8 string with three components. The first component is the concatenated aes-key and salt; "||" is the concatenate operator. The aes-key is 16 bytes and the salt is 8 bytes so "aes-key || salt" is 24 bytes before Base 64 encoding, which results in 32 bytes since Base 64 is a "three in four" encoding scheme. The value MUST be unique and is followed by a "|" and the key lifetime, which is the number of bytes that may be encrypted and decrypted using this master key. When empty, the lifetime defaults to 2^64 bytes, which is the maximum width of the ISMACrypIVLength, parameter #6 above (the AES-CTR limitation is 2^64 blocks [SECURITY]). The lifetime is followed by a "|" and the optional key indicator (KI) for this master key. The "|" characters are needed only when the lifetime and/or key indicator is present in the ISMACrypKey parameter.

The following is an example of an ISMACrypKey that is provided with no explicit lifetime (defaults to 2^64 bytes) and no KI:

ISMACrypKey=(key)MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0

The lifetime may be specified as follows:

ISMACrypKey=(key)MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0|2^24

In this case, after decrypting 2^24 bytes of data, the receiver MUST NOT use the key. Either the stream is at an end or the new key is obtained from the KMS. Note that the ISMACrypKey parser MUST be able to evaluate expressions of the form 2^x.

When the ISMACryp stream uses a key indicator, this value MUST be specified as follows.

ISMACrypKey=(key)MTIzNDU2Nzg5MDEyMzQ1Njc4OTAxMjM0|4000000|1492

In the example above, the lifetime is 4 million bytes and a KI of 1492 is associated with the key.

10.1.3 Transport packetization values

Parameters describing packet fields that are an integral number of bytes are stated in bytes (octets) rather than bits. The only transport packetization value needed by the cipher is an IV input for decryption and a BSO input for encryption, where the IV is a BSO sample that the packetizer adds to an ISMACryp packet. Section 2, Glossary, defines the BSO.

An enc-mpeg4-generic packet contains the "initial IV" value for the first access unit or fragment contained in the packet; in many cases when packet data are not interleaved, the Initial_IV is the only IV in the packet. Its length is determined by the signaling parameter ISMACrypIVLength, and has a default length of 4 bytes. If the media stream is longer than 2^32 bytes, then there are two options. The first option is to use a longer IV length that is as large as the stream. The second option is to rotate the key before the IV wraps.

When interleaving is used, there SHALL be an IV assigned to each enc-mpeg4-generic AU. See Section 7 for specifics of how to encode multiple IVs in an enc-mpeg4-generic packet.

10.2 ISMA 1.0 & 2.0 Message Authentication (Integrity) Transform

The default ISMACryp message authentication (Integrity) transform is SRTP with an HMAC-SHA1 with an 80-bit output tag and a 128-bit master key [RFC3711]. The master key is used with SRTP key derivation function to compute a 160-bit authentication key. This service is signaled using SDP security descriptions [RFC4568]. ISMACryp applications SHOULD NOT signal or use SRTP encryption but MAY signal SRTCP encryption. Here is an example of such SDP signaling:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
   inline:azerRTazad1223dsdsfEhtgdjj12ZSzerefigtyt|2^20|1:32
   UNENCRYPTED_SRTP
```

Where

- crypto:1 = we are using master key 1 (
- AES_CM_128_HMAC_SHA1_80 specify the mode (mandatory)
- inline = key (optional)
- 2^20 = lifetime (optional)
- 1:32 = master key indicator (MKI) of 32-bits wide and associates the number 1 (optional)
- UNENCRYPTED_SRTP means that ISMACryp doesn't use SRTP encryption (mandatory)

10.3 The Security of ISMA 1.0 & 2.0 Cryptography

Attacks on the encryption or authentication of ISMACryp media and messages encounter too high a workload to be perceived as a threat to the media confidentiality of an ISMA stream [SECURITY, RFC2104]. ISMACryp signaling is secure against forgery, replay attack, and unauthorized disclosure of an ISMACryp parameter when the SDP signaling uses a data-security protocol such as TLS or IPsec. When an SDP message conveys an enc-mpeg4-generic parameter, it SHOULD be authenticated and integrity-protected using TLS or IPsec. When an SDP message conveys the enc-mpeg4-generic ISMACrypKey key parameter, it SHOULD be encrypted. When properly used, ISMACryp is appropriate for government, enterprise, and individual security applications.

There are three key-management risks, however, to the proper use of ISMACryp encryption and message authentication.

1. Counter reuse: Additive stream ciphers share the security properties of the One-Time-Pad encryption system and can disclose information about the plaintext segment when two different plantexts are encrypted using the same keystream segment (under certain circumstances, the plaintext can be disclosed). It is RECOMMENDED that an ISMACryp encryption key be used for one an only one unidirectional stream: Two or more streams SHOULD NOT use the same ISMACryp key. It is NOT RECOMMENDED that the salting_key be used to ensure that the keystream is unique among streams since it is there to protect against key collision attacks [MF00] and not to make the keystream unique.
2. Key collision: McGrew and Fluhrer describe an attack that weakens the ISMACryp key (i.e. reduces the effective bit length) through an attack where the attacker precomputes a large number of keys starting from beginning of the counter space, and looks for key matches based on known plaintext in the stream. This attack is prevented by randomizing the start of the counter space to an unpredictable starting value [MF00]. This random offset is the salting key, k_s, of Section 10, which MUST be unpredictable to the attacker.
3. Key disclosure: When used properly (i.e. avoids counter reuse) ISMACryp would not be the target of an attacker seeking to get unauthorized access to media-stream plaintext. The easier and oftentimes feasible approach is to attack the key management system. Thus, the protections of ISMACryp for government, enterprise, or individual security are no stronger than the security of the particular key management system and the protection of keys by devices and their users.

There is an additional risk to message confidentiality when there is no authentication: If a cryptoanalyst knows the plaintext in a particular position in the stream, and if the attacker can benefit from transforming the bytes of know plaintext into different values, then the attacker can ensure that those byte decrypt to the different value rather than the original (known) plaintext. To prevent such an attack from succeeding, an

ISMACryp implementation SHOULD use message authentication when the ISMACryp stream traverses an insecure network. When the ISMACryp stream is stored on an insecure host computer, the ISMACryp implementation SHOULD use file authentication techniques [SMPEG].

In addition to organizational or individual security applications, ISMACryp MAY be incorporated into content protection applications to serve as a scrambling mechanisms. Consumer devices that process encrypted content generally violate the security assumptions of computers in a government or enterprise security environment since consumer devices routinely suffer key disclosure [DeCSS]. Studios, record labels, and other distributors of copyright content that use ISMACryp SHOULD evaluate the robustness and compliance of the key management system that handles ISMACryp keys. This system, however, is outside the scope of ISMACryp.

11.0 Name Assignment and Registration

The following table lists each name assignment that MUST be reserved for ISMACryp, its description, and the relevant naming authority.

VALUE	TYPE	DESCRIPTION	AUTHORITY
enc-mpeg4-generic	MIME	ISMA encrypted payload type	IETF/IANA
AES_CTR_128	UTF-8	enc-mpeg4-generic parameter	IETF/IANA
all of Table 8.3.2	UTF-8	enc-mpeg4-generic parameters	IETF/IANA
0x4953	IPMPS_Type	ISMACryp stype	www.ipmp-ra.org
enca or encv	4CC	MP4 FF 4CC sample description	MPEG
iAEC	scheme-type box	MP4 FF ISMACryp default encryption	MPEG
264b	4CC	MP4 FF 4CC original format	MPEG
OMA2	4CC	MP4 FF 4CC OMA DRM v2 KMS	MPEG

12.0 Open Issues

ISMA has not yet obtained a value for IPMPS_Type. This value is needed for IPMPS_Type, IPMP_ToolID, and dataID.

Annex A: Key Management Interfaces (Informative)

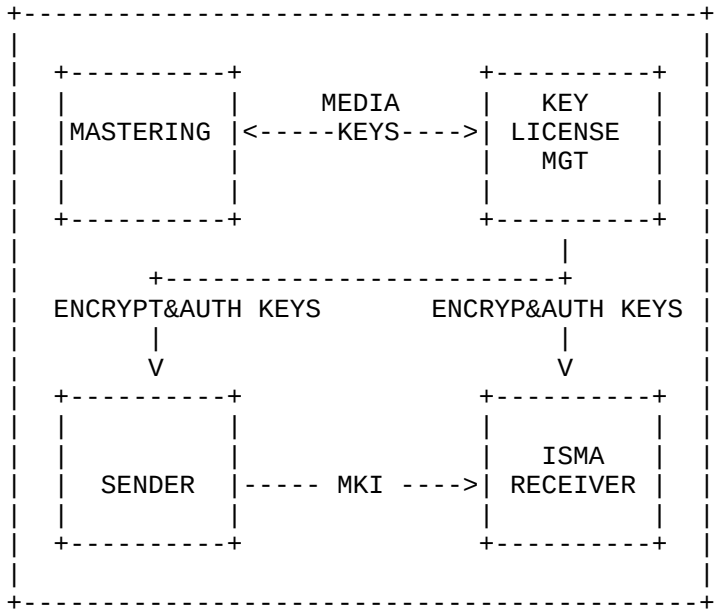


Figure A-1: Key/License Interfaces

Figure A-1 shows three key/license interfaces. The box labeled "KEY/LICENSE MGT" is the key store in this functional block diagram. The functional nature of this diagram does not assume that there is only one possible key distribution center on a single computer; Figure A-1 shows the KEY/LICENSE MGT function and not an implementation. Some means are needed to provide decryption keys to KEY/LICENSE MGT that match the keys used to encrypt a media stream in the MASTERING step. Alternatively, KEY/LICENSE MGT could provide all needed keys to MASTERING. Thus, whether or not MASTERING creates the keys is outside the scope of ISMACryp.

Nonetheless, any operational system needs some means to associate one or more keys with a media stream, as shown by the directed arcs between MASTERING, KEY/LICENSE MANAGEMENT, the SENDER, and the ISMA RECEIVER in Figure A-1. These associations are depicted in Figure A-1.

This association of keys and stream between MASTERING and KEY/LICENSE MANAGEMENT can be through a secure Remote Procedure Call interface, through a secure file transfer, a Web language interface specification such as XML, or possibly by other means - all of which are outside the scope of ISMACryp. To illustrate one possible approach, this section presents a method to convey keys in an MPEG-4 file. This MP4-file approach clearly defines how keys are to be synchronized with media; this is particularly needed if key changes occur at specific points in the media stream. Section A.1 describes some possible key-management extensions to the MPEG-4 file format for keystreams. This description is informative and does not preclude future definition of an RPC interface, API, or other file format such as XML.

A pair interfaces exists between KEY/LICENSE MANAGEMENT and the functional boxes labeled "SENDER" and "ISMA RECEIVER." These interfaces carry encryption and/or authentication keys. ISMA uses SRTP message authentication and thus uses SRTP key management interfaces for this function. At present, work is being done in the IETF MMUSIC working group to define an SDP key management interface for SRTP. This work [RFCSDPSD] is referenced by ISMA for ISMACryp. ISMA encryption-key signaling is discussed in Section 8, above. Section A.2 considers alternatives to the SDP interface.

Section A.3 discusses the key rotation feature of ISMACryp that is signaled out of band at the time that keys are established at the receiver and is signaled in band in the OPTIONAL MKI field in the enc-mpeg4-generic header. Figure A-1, therefore, shows an interface, labeled "MKI," between the SENDER and RECEIVER. The MKI flow is with the media as depicted in Figure A-1.

A.1 MPEG-4 Key Management Interface: ISMACryp Key Track

In some circumstances, it might be desirable for functional entities to be able to exchange an mp4 file with ISMACryp key information. One example is when the Master Encryptor wishes to transfer the key information to key management. In this scenario, it is necessary that both the key-descriptors and key-values be communicated to key management. We assume that the ISMACryp mp4 file transfer is done via a secure channel outside the scope of this specification (e.g. S/MIME, SSH, SSL, IPsec).

A second example is when the content controlling entity wishes to distribute a locked copy of the ISMACryp mp4 file to Receivers via a non-streaming mechanism. Examples of this are HTTP download or physical media distribution. In this case, the ISMACryp key track should contain only key indicators and not the actual key-values. The Receiver will need to contact key management to retrieve the key values to decrypt the media. This exchange is assumed to occur via a secure channel that is outside the scope of this specification.

This proposal illustrates how ISMACryp key information can be stored in an ISMA compliant mp4 file, but does not attempt to cast the ISMACryp key track into the MPEG-4 Systems IPMP framework. Note that in bit(n), unsigned int(n) and int(n), n is always is a bit count [14496-1].

A.1.1 Assumptions

A media AU is either unencrypted or encrypted with a single key, i.e. there is no AU-fragment encryption. There is a null encryption key-indicator. This will be used to signal selective encryption of media AUs.

A.1.2 Problems and proposed solutions

A basic problem is the identification of encrypted media tracks and association with a key track: We desire to have a simple mechanism by which an application that understands the mp4 file format but not necessarily the media encodings within, can determine if a media track has been encrypted via ISMACryp. We also need a mechanism whereby the encrypted media track can be associated with a key track that will allow decryption of the media (once licensing requirements have been met).

As background, we note that the standard mp4 box, "moov.trak.tref", is defined to contain an array of track reference type boxes. Five standard reference box types are currently defined (hint, dpnd, ipir, mpod, sync). Each box type follows the following syntax template:

```
aligned(8) class TrackReferenceTypeBox (unsigned int(32) reference-
type)
      extends Box(reference-type) {
            unsigned int(32) track-IDS[];
}
```

We propose an ISMA defined track reference type "IKEY" following the above template. We would impose a restriction that only one track-ID be present for "IKEY". The referenced track contains the ISMACryp key information necessary to decrypt the media frames of the referencing track. Identification of an ISMACryp media track then simply is a matter of checking if the box "moov.trak.tref.IKEY" exists for a given track. Note that we anticipate that there will typically be a one to one correspondence between encrypted media tracks and key tracks, however our proposal allows for multiple media tracks to reference the same key track.

A.1.3 Definition of an ISMACryp key track

Currently in an mp4 file, the type of a track is specified in the value "moov.trak.mdia.hdlr.handler-type". Existing values used in an ISMA-compliant mp4 file are "vide", "soun", "hint", "odsm", and "sdsm" corresponding to video, audio, hint, object descriptor, and scene description tracks. We propose to define a new value "IKEY" for the key track.

Existing mp4 track types have a media handler atom under "moov.trak.media.minf" (i.e. vmhd, smhd, hmhd, and nmhd). As the media handler box for an ISMACryp key track should be an empty box, we propose to (re)use the existing "moov.trak.mdia.minf.nmhd" box used for MPEG-4 Systems tracks which is currently defined to be empty.

ISMACryp requires some initialization information, which we propose to store in the sample description table of the track, i.e. "moov.trak.mdia.minf.stbl.stsd". We impose a restriction that only one sample description be present for an ISMACryp key track. The syntax of the sample description is as follows:

```
class ISMACrypSampleDescriptionEntry
   extends SampleEntry("ICSD") {
      string   cipher_mode_keylength;
      unsigned int(8)   sample-version;
      string            key-manager-uri;
}
```

cipher_mode_keylength - the ISMACryp encryption transform and mode having fixed key size (key-size) and block size. See Annex E for the value that MUST be supplied when the ISMACryp default cipher is used.
sample-version - indicator of the structure of the information stored in the key track samples. This document defines version 1 of this data.
key-manager-uri - A URI by which the key manager can be contacted to retrieve the key-values associated with key indicators. If the key track samples contain the keys, then this URI is not necessary and may be empty.

A.1.4 Definition of a ISMACryp key track sample

To decrypt an encrypted media sample, the key track sample that corresponds in time to the sample time of the media sample is consulted. Note this requires that the key track lifetime equal (or exceed) the lifetime of the media track. It is anticipated that typically there will be one key track per encrypted media track, and that their time scales, start times, and durations will be identical as is typically the case for hint tracks. The key track sample will contain the key-indicator for the given media AU, and optionally the key-value. If the key-value is not present, the key-manager-uri should be used to contact the Key Manager and request the key-value associated with the key-indicator. The details of this exchange are beyond the scope of this specification.

Each sample (AU) in the ISMACryp key track will have the following structure:

```
aligned(8) class ISMACrypKeySample {
   unsigned int(8)   key-indicator;
   if (sample-size == 1 + key-size) {
      unsigned int(8)   key-value[key-size];
   }
}
```

The key-size value is determined by the cipher_mode_keylength in the ISMACryp sample description. (see section 4.2.2). The sample-size should be either be one byte, i.e only the key-indicator is present, or should be equal to the key-size specified by the cipher-mode plus 1 byte. If the sample-size is not one of these two values it should be considered an error.

Some media AUs may be unencrypted within the encrypted media track, this is indicated by having a corresponding key track sample with the key-indicator value of 0.

A.2 Receiver Key Management Interfaces Considerations

The Receiver Key Management Interface is standardized with an ISMACryp description for SDP in this specification. The ISMACryp description for SDP, however, SHOULD be encrypted if it conveys an ISMACryp key in an SDP k= statement. S/MIME, SSL, IPsec or some other means could serve to encrypt the SDP message having a k= statement, but this solution is not likely to suit all applications. If the SDP description cannot be used, it can serve an heuristic purpose for a receiver key management application programming API, a key management stream that gets delivered to the receiver, or by other means that are not defined in this specification.

A.3 Example use of the key-indicator

This annex provides an informative example of how the key-indicator may be used by a KMS using "loose" synchronization.

The client gets the following information at start-up: The encryption info includes the following information (see section 8).

a) Optional information about a key management service (i.e. a URI).
b) Information describing the encryption parameters, such as cipher, mode, and key length.
c) An optional encryption key

In addition to this information, the client has some kind of program or key-set identifier (which is probably not the URI the client used, as that may have been altered by the content delivery network etc.) that identifies to the KMS what key-set it needs (i.e. what content needs to be decrypted). The program or key-set identifier is specific to the service and is outside the scope of ISMACryp.

By communicating with the key-management service, using the program identifier, key-set identifier, or other service-specific means, the client gets a key set. This key-set may be (a) "static" and sufficient for the length of this content or (b) "dynamic" and dependent on a particular point in the stream. In the case of dynamic key-sets, the KMS will supply keys roughly for the "current" point in the stream, looking forward. The key-set contains key-indicator/key pairs.

The key-indicator, when used, is delivered with the data. This key-indicator matches the key-set with a key. That key is used to decrypt the content (along with the algorithm in use, e.g. AES counter mode). Key-indicators are used in sequence.

In one possible usage, the key-indicator runs in a space of B bits with a value-range of V (e.g. 8 bits, 256 values). The KMS might supply to the client fewer than V/2 keys at any one time. While those keys are being processed, and before they all expire, the client might return to the KMS and "pull" a new set (which may well overlap the set it already has, but which will extend further into the future). Alternatively, the KMS might "push" the keys as done in some broadcast networks. In the "push" scenario, the KMS sends the keys to the client in an unsolicited fashion.

In the "pull" scenario, the KMS provides to the client a "suggested" time to get a new key-set, such as by identifying a key indicator value, Q, that serves as a trigger for the client to pull a new key set.. If Q is never used in the stream (for "short" content) then the client has all the keys it needs. The KMS can spread its clients out, to smooth the traffic flow, by supplying different Qs, and can also choose how many keys to supply in any one transaction, thus controlling the traffic for each client. When the client does in fact, get back to the KMS, is its own decision (if it does it "early" it may gather few if any new keys, and if it does it "late" it may run out of keys before getting a reply).

When the client sees a value key-indicator X in the data stream, it discards all key-indicator/value pairs that it has that contain a key-indicator outside the range X through X+V/2 (in modulo V arithmetic). This

ensures that keys that have been used once are discarded. It allows the link to the KMS to be "loose" – the KMS might supply a few keys that are in the past with respect to the client, which will promptly discard them as they lie outside the valid range (X, X+V/2). In particular, the key-set can be static for short programs needing fewer than V/2 keys.

In some KMS designs, the client needs to tell the Key Server its current position in the stream in addition to the Key Indicator. This is achieved by proprietary means but recommended values are: the sample number (when playing files), or the Normal Play Time (streams or files). A KMS can also use a larger Key Indicator and private encoding in this KI to achieve this goal. The current key-indicator being used by the client is always supplied to the KMS; this may be sufficient to determine stream position, if the key-indicator does not wrap within the program.

Annex B: Local Playback (Informative)

This annex provides a walkthrough of how local playback, random access, and editing, can be performed, when the ISMA AES-CTR encryption is used.

1. First, check that the ISMA AES-CTR mode encryption is being used.

2. Then, get the keys needed. Retrieve the KMS indicator and parameters from the sample description, and interface to the KMS to acquire a set of key-indicators/key mappings. This interaction may use the time-offset in the stream.

3. From the sample description, extract the salt, selective-encryption-used, key-indicator-size, and initial-counter-size.

4. Read the sample from disk using standard MP4 means, and place it into a buffer.

```
encrypted := ((selective-encryption-used == 1) ? read-bytes(1) & 0x80 : 0);
if (encrypted == 0x80) then {
        key-indicator := read-bytes( key-indicator-size );
        initial-counter := read-bytes( initial-counter-size );
        decrypt-buffer-bytes( key-map[key-indicator], salt, initial-counter );
}
```

Annex C: Encryption Process Example (Informative)

1) Basic encryption procedure

In this example an (AES) encryptor block size of 3 is used to illustrate the process. The counter at starts with a value of 0. By incrementing the counter a byte stream containing "key stream bytes" is generated as indicated in Figure A-1:

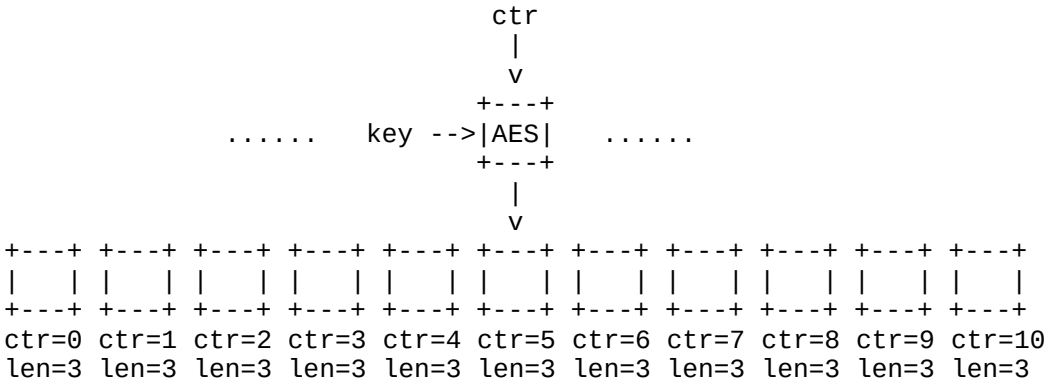
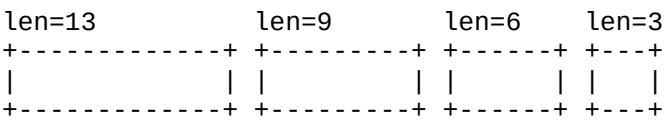


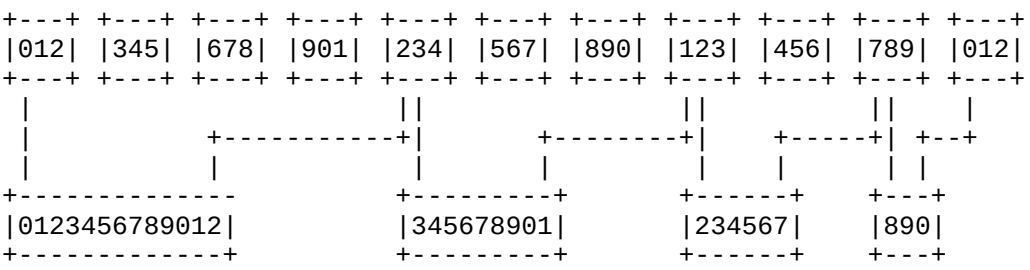
Figure C.3-1: Indicating a high level overview of the key stream generation

To encrypt a plain text byte stream with AES in Counter Mode, the plaintext byte stream is exored with the key stream to obtain the corresponding cipher-text byte stream. The decryption process is the reverse of the encryption process, i.e., the cipher text byte stream is exored with the same key byte stream to receive the original plaintext byte stream.

In the following, the plain text is media stream access Units, where, to illustrate the encryption process, 4 access units of different lengths are used:



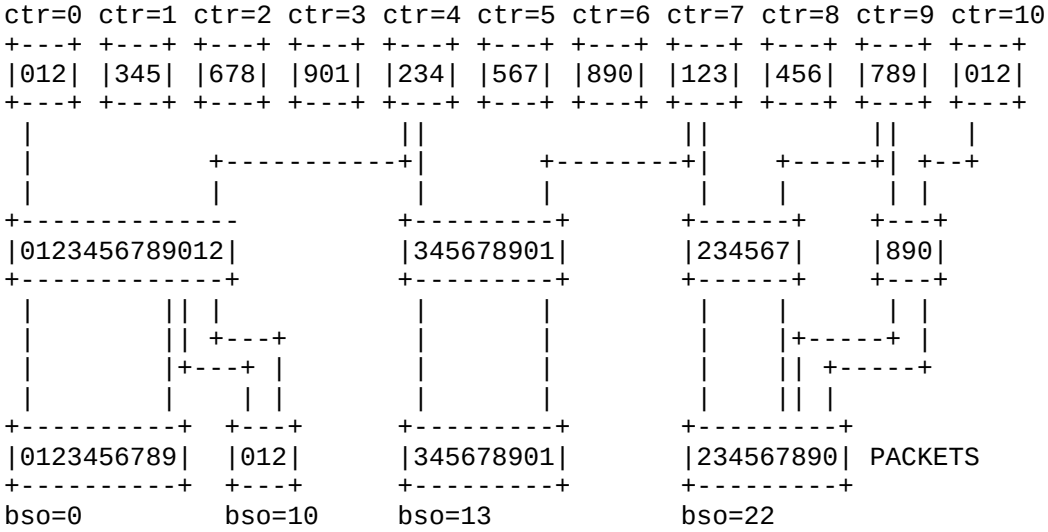
For encryption these access units packets are treated as a single byte stream and are lined up in sequence against the key stream bytes as indicated in the following figure:



Thus the media access units are encrypted by performing the byte-wise XOR of the their bytes with the corresponding key stream bytes. Note that if after encrypting an access unit, if the bytes from the previous encryptor block have not been exhausted, the remaining bytes of that block will carry-over to the beginning

of the next access unit. Thus the encryptor block boundaries are NOT aligned with the access unit boundaries.

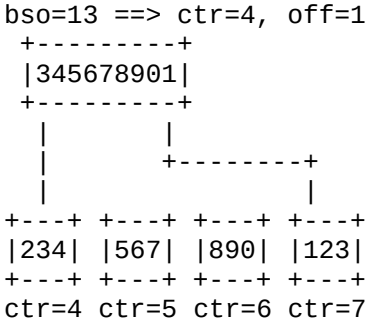
The encrypted access units are now packetized by inserting the "AU byte stream offset" (BSO) into the ISMACryp header (see Section 7). In the example below, the first AU gets fragmented over 2 packets, the second AU goes in as a whole into one packet, and the third and fourth AU are small enough that they may be combined into a single packet:



Having the byte stream offset (bso) in each packet allows the decryption of each packet independent of the other packets. To illustrate this, suppose the third packet, which happens to contain one complete access unit, is received. The byte stream offset (bso) received in the packet is used to derive the counter (ctr) and encryptor block offset (off) as follows:

```
ctr = bso / 3
off = bso % 3
```

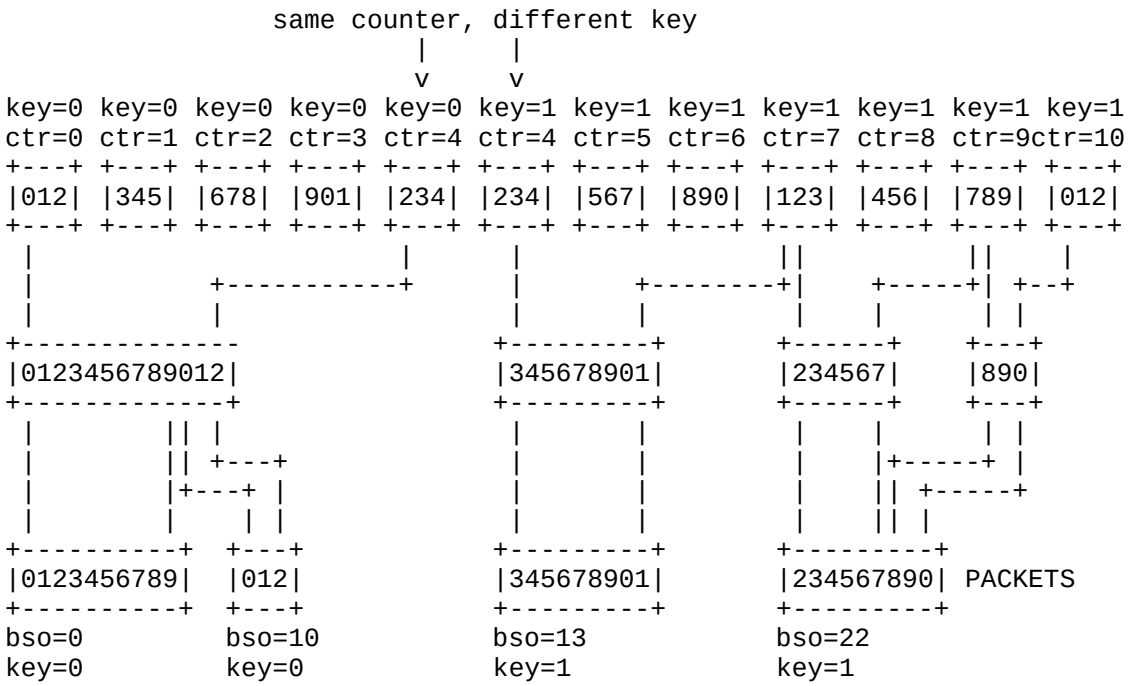
where 3 is the encryptor block size (3 in our example; it's 16 bytes for AES). And that's all is needed to regain synchronization of the data with the key stream. The packet data may be decrypted as follows:



2) Key change

The notion of a key change is illustrated taking the same example from above. Without any loss of generality, assume that the key changes to the next key at the beginning of the second access unit. Note that, in this example, this happens to fall right in the middle of an encryptor block. In the case of an

encryption wherein the key remained the same, the last couple of unused key stream bytes from the last encryptor block would be used for the encryption or decryption of the bytes in the next access unit block. But since the key has changed, the key stream block must be regenerated with the same counter, but with the new key, as indicated in the figure below:

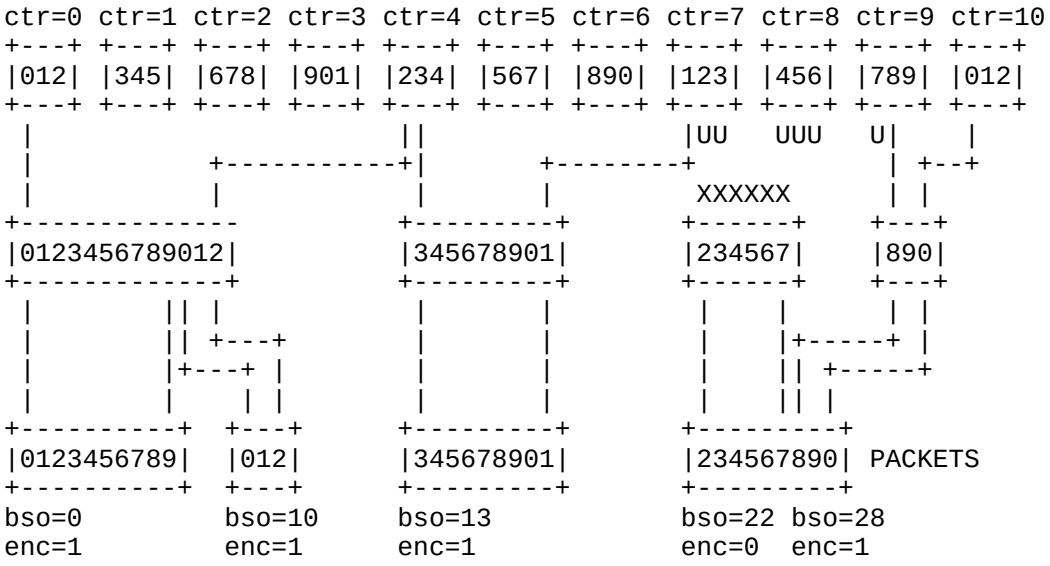


For the decryption process, the "bso" and key indicator are sufficient to re-generate the correct key stream and thus, each packet corresponding to an encrypted access is independently decrypted, even with key changes.

Note that the Key Manager can select the key based upon the key indicator, but it is also possible for the Key Manager to determine which key to use based upon the bso (in the example above, key=1 takes effect at bso=13).

3) Selective encryption

Suppose all access units except the 3rd are to be encrypted. It is desired that the 3rd access unit be sent in the clear. During encryption process this is easily accomplished by not performing the XOR for those bytes that have to be left in the clear (marked with XXX below), and, in addition, skipping the unused key stream bytes (marked as UUU below).



The fourth RTP packet now must have an ISMACryp header with two segments, one indicating that the first AU in the payload is in the clear, while the second AU in the paylaod is encrypted. The decryptor for this packet thus has all the information to independently decrypt the data, be it selectively.

Annex D: Sample IOD (Informative)

Table 8.2-0.1 shows the IOD of a sample session in which both video and audio streams are protected. The rows that were added to a non-protected IOD appear in blue. The table is followed by the binary and Base64 representations of the table data. This sample implements the Minimal MPEG signaling.

Table 8.2-0.1 – A sample IOD for an audio-visual presentation – binary syntax

Descriptor Name			
Field No.	**Size in Bits**	**Field Name**	**Value**
InitialObjectDescriptor			
1	8	InitialObjectDescriptor tag	2
2	16	descriptor size	0x823F
3	10	ObjectDescriptorID	0
4	1	URI_Flag	0
5	1	includeInlineProfilesFlag	0
6	4	Reserved	15
7	8	ODProfileLevelIndication	255
8	8	sceneProfileLevelIndication	255
9	8	audioProfileLevelIndication	15
10	8	visualProfileLevelIndication	1
11	8	graphicsProfileLevelIndication	1
	ES_Descriptor (of OD stream)		
12	8	ES_Descriptor tag	3
13	16	Descriptor size	0x814a
14	16	ES_ID	1
15	1	streamDependenceFlag	0
16	1	URI_Flag	1
17	1	OCRstreamFlag	0
18	5	streamPriority	0
19	8	string-size	172
20	36*8	URIstring	"data:application/mpeg4-od-au;base64,"
		ObjectDescriptorUpdate (Fields 21 through 107 are shown before the encoding in Base64)	
21	8	ObjectDescriptorUpdate tag	1
22	8	Descriptor size	92
		ObjectDescriptor (of video)	
23	8	ObjectDescriptor tag	1
24	8	descriptor size	45
25	10	ObjectDescriptorID	20
26	1	URI_Flag	0
27	5	Reserved	31
		ES_Descriptor (of video)	
28	8	ES_Descriptor tag	3
29	8	descriptor size	41
30	16	ES_ID	201
31	1	streamDependenceFlag	0
32	1	URI_Flag	0

33	1	OCRstreamFlag	1
34	5	streamPriority	4
35	16	OCR_ES_Id	101
		DecoderConfigDescriptor (of video)	
36	8	DecoderConfigDescriptor tag	4
37	8	descriptor size	13
38	8	objectTypeIndication	32
39	6	streamType	4
40	1	upStream	0
41	1	Reserved	1
42	24	bufferSizeDB	8000
43	32	maxBitrate	64000
44	32	avgBitrate	64000
		SLConfigDescriptor (for the video stream)	
45	8	SLConfigDescriptor tag	6
46	8	descriptor size	16
47	8	predefined	0
48	1	useAccessUnitStartFlag	0
49	1	useAccessUnitEndFlag	1
50	1	useRandomAccessPointFlag	0
51	1	hasRandomAccessUnitsOnlyFlag	0
52	1	usePaddingFlag	0
53	1	useTimeStampsFlag	1
54	1	useIdleFlag	0
55	1	durationFlag	0
56	32	timeStampResolution	1000
57	32	OCRResolution	0
58	8	timeStampLength	32
59	8	OCRLength	0
60	8	AU_Length	0
61	8	instantBitrateLength	0
62	4	degradationPriorityLength	0
63	5	AU_seqNumLength	0
64	5	packetSeqNumLength	0
65	2	reserved	3
		IPMP_DescriptorPointer (of video)	
66	8	IPMP_DescriptorPointer tag	10
67	8	descriptor size	1
68	8	IPMP_DescriptorID	1
		ObjectDescriptor (of audio)	
69	8	ObjectDescriptor tag	1
70	8	descriptor size	40
71	10	ObjectDescriptorID	10
72	1	URI_Flag	0
73	5	Reserved	31
		ES_Descriptor (of audio)	
74	8	ES_Descriptor tag	3
75	8	descriptor size	36

76	16	ES_ID	101
77	1	streamDependenceFlag	0
78	1	URI_Flag	0
79	1	OCRstreamFlag	0
80	5	streamPriority	5
		DecoderConfigDescriptor (of audio)	
81	8	DecoderConfigDescriptor tag	4
82	8	descriptor size	13
83	8	objectTypeIndication	64
84	6	streamType	5
85	1	upStream	0
86	1	Reserved	1
87	24	bufferSizeDB	8000
88	32	maxBitrate	64000
89	32	avgBitrate	64000
		SLConfigDescriptor (of audio)	
90	8	SLConfigDescriptor tag	6
91	8	descriptor size	16
92	8	predefined	0
93	1	useAccessUnitStartFlag	0
94	1	useAccessUnitEndFlag	1
95	1	useRandomAccessPointFlag	0
96	1	hasRandomAccessUnitsOnlyFlag	0
97	1	usePaddingFlag	0
98	1	useTimeStampsFlag	1
99	1	useIdleFlag	0
100	1	durationFlag	0
101	32	timeStampResolution	1000
102	32	OCRResolution	1000
103	8	timeStampLength	32
104	8	OCRLength	32
105	8	AU_Length	0
106	8	instantBitrateLength	0
107	4	degradationPriorityLength	0
108	5	AU_seqNumLength	0
109	5	packetSeqNumLength	0
110	2	reserved	3
		IPMP_DescriptorPointer (of audio)	
111	8	IPMP_DescriptorPointer tag	10
112	8	descriptor size	1
113	8	IPMP_DescriptorID	1
		IPMP_DescriptorUpdate	
114	8	IPMP_DescriptorUpdate tag	5
115	8	descriptor size	5
		IPMP_Descriptor	
116	8	IPMP_Descriptor tag	11
117	8	descriptor size	3
118	8	IPMP_DescriptorID	1
119	16	IPMPS_Type	0x4953

		DecoderConfigDescriptor (of OD stream)	
120	8	DecoderConfigDescriptor tag	4
121	8	descriptor size	13
122	8	objectTypeIndication	1
123	6	streamType	1
124	1	upStream	0
125	1	Reserved	1
126	24	bufferSizeDB	200
127	32	maxBitrate	0
128	32	avgBitrate	0
		SLConfigDescriptor (of OD stream)	
129	8	SLConfigDescriptor tag	6
130	8	descriptor size	9
131	8	predefined	1
132	32	startDecodingTimeStamp	0
133	32	startCompositionTimeStamp	0
	ES_Descriptor (of BIFS stream)		
134	8	ES_Descriptor tag	3
135	8	descriptor size	105
136	16	ES_ID	2
137	1	StreamDependenceFlag	0
138	1	URI_Flag	1
139	1	OCRstreamFlag	0
140	5	StreamPriority	0
141	8	string-size	70
142	38*8	URIstring	"data:application/mpeg4-bifs-au;base64,"
143	32*8	URIstring	Binary data before encoding in Base64: 0x" C0 10 12 81 30 2A 05 6D 26 10 41 FC 00 00 01 FC 00 00 04 42 82 28 29 F8"
		DecoderConfigDescriptor (of BIFS)	
144	8	DecoderConfigDescriptor tag	4
145	8	Descriptor size	18
146	8	ObjectTypeIndication	2
147	6	StreamType	3
148	1	Upstream	0
149	1	Reserved	1
150	24	BufferSizeDB	20
151	32	MaxBitrate	0
152	32	AvgBitrate	0
		BIFSv2Config	
153	8	BIFSv2Config tag	5
154	8	Descriptor size	3
155	1	use3DmeshCoding	0
156	1	UsePredictiveMFField	0
157	5	NodeIDbits	0
158	5	RouteIDbits	0
159	5	ProtoIDbits	0
160	1	IsCommandStream	1
161	1	PixelMetric	1

162	1	HasSize	0
163	4	byte align	0
		SLConfigDescriptor (of BIFS)	
164	8	SLConfigDescriptor tag	6
165	8	descriptor size	9
166	8	Predefined	1
167	32	startDecodingTimeStamp	0
168	32	startCompositionTimeStamp	0

This table yields the following hexadecimal string:

```
02 82 3F 00 4F FF FF 0F 02 01 03 81 4A 00 01 40
AC 64 61 74 61 3A 61 70 70 6C 69 63 61 74 69 6F
6E 2F 6D 70 65 67 34 2D 6F 64 2D 61 75 3B 62 61
73 65 36 34 2C 41 56 77 42 4C 51 55 66 41 79 6B
41 79 53 51 41 5A 51 51 4E 49 42 45 41 48 30 41
41 41 50 6F 41 41 41 44 36 41 41 59 51 41 45 51
41 41 41 50 6F 41 41 41 41 41 43 41 41 41 41 41
41 41 77 6F 42 41 51 45 72 41 70 38 44 4A 77 42
6C 42 51 51 4E 51 42 55 41 48 30 41 41 41 50 6F
41 41 41 44 36 41 41 59 51 41 45 51 41 41 41 50
6F 41 41 41 44 36 43 41 67 41 41 41 41 41 77 6F
42 41 51 55 46 43 77 4D 42 53 56 4D 3D 04 0D 01
05 00 00 C8 00 00 00 00 00 00 00 00 06 09 01 00
00 00 00 00 00 00 00 03 69 00 02 40 46 64 61 74
61 3A 61 70 70 6C 69 63 61 74 69 6F 6E 2F 6D 70
65 67 34 2D 62 69 66 73 2D 61 75 3B 62 61 73 65
36 34 2C 77 42 41 53 67 54 41 71 42 57 30 6D 45
45 48 38 41 41 41 42 2F 41 41 41 42 45 4B 43 4B
43 6E 34 04 12 02 0D 00 00 14 00 00 00 00 00 00
00 00 05 03 00 00 60 06 09 01 00 00 00 00 00 00
00 00
```

```
a=mpeg4-iod: "data:application/mpeg4-iod;base64,
AoI/AE///w8CAQOBSgABQKxkYXRhOmFwcGxpY2F0aW9uL21wZWc0LW9kLWF1O2Jhc2U2NCx
B
VndCTFFVZkF5a0F5U1FBWlFRTklCRUFIMEFBQVBvQUFBRDZBQVlRQUVRQUFBUG9BQUFBQUN
B
QUFBQUFBd29CQVFFckFwOERKd0JsQlFRTlFCVUFIMEFBQVBvQUFBRDZBQVlRQUVRQUFBUG9
B
QUFENkNBZ0FBQUFBd29CQVFVRkN3TUJTVk09BA0BBQAAyAAAAAAAAAAABgkBAAAAAAAAAAA
D
aQACQEZkYXRhOmFwcGxpY2F0aW9uL21wZWc0LWJpZnMtYXU7YmFzZTY0LHdCQVNnVEFxQlc
w
bUVFSDhBQUFCL0FBQUJFS0NLQ240BBICDQAAFAAAAAAAAAAABQMAAGAGCQEAAAAAAAAAAA=
=
"
```

Annex E: Interoperability with OMA DRM Version 2.0 (Informative)

This annex provides guidelines on how ISMACryp can be used together with the key and rights management system of OMA DRM v2. It specifies those hooks in the file format and SDP which allow to obtain keys from the OMA DRM KMS for streaming as well as downloading ISMA content.

E.1 Overview

OMA DRMv2.0 is split into four parts: Architecture, Rights Expression Language [OMARELv2], DRM protocols (e.g. Rights Object Acquisition and Management) [OMADRMv2] and DRM Content Format [OMADCFv2]. For more information, the specifications can be consulted from the OMA website (http://www.openmobilealliance.org).

Familiar readers may have noticed that in OMA DRM DCF v2.0 candidate specification [OMADCFv2], Packetized DRM Content Format is almost based on ISMACryp format.

E.2 MPEG-4 File Structure

The file format provides a mean to identify when the OMA DRM key management system is used.

E.2.1 Sample description transformation

The Scheme Information Box is only a container box used to carry DRM KMS specific information. The series of contained boxes can be of any type and format, thus it can include OMA specific boxes. To use OMA DRM KMS, this box MUST include exactly:

a) In first position, one ISMA KMS Box "iKMS" with
   - version = 1
   - KMS ID = OMA2
   - KMS version = 0x00000200
   - KMS URI = OMA DRM v2 right issuer URI

b) Then one OMA DRM Common Headers Box "ohdr" [OMADCFv2] that specifies the encryption scheme and its parameters and provides information about the Rights Issuer as well.

c) In third position, one ISMASampleFormat Box "iSFM".

d) In last position, one ISMACrypSaltBox.

These hooks should be enough to launch the OMA DRM v2.0 Right Object Acquisition Protocol (ROAP) [OMADRMV2] to acquire the license and then to access the associated content. If there is only one salt-key for the stream, the ISMACrypSaltBox defines it.

E.3 Transport Signaling

When using both ISMACryp and OMA DRM v2.0 combined, session and stream signaling parameters must identify these standards. These parameters identify the crypto suite and the structure of the OMA DRM KMS.

E.3.1 Session Description Protocol Signaling

Regarding ISMACryp current signaling, this solution SHOULD use the enc-mpeg4-generic format and its associated generic parameters as a base. In addition, three MANDATORY parameters are required: a salt key, a content identifier parameter and a rights issuer URL parameter:
The SDP fmtp signaling SHALL use enc-mpeg4-generic as its format, which is case sensitive.

Generic SDP signaling:

```
m=<media> <port>/<number of ports> <transport> <fmt list>
a=rtpmap:<payload type> <encoding name>/<clock rate>[/<encoding parameters>]
```

```
a=fmtp:<payload type> mode=<mode>; <MPEG4-GENERIC-PARMS> <ENC-MPEG4-GENERIC-
PARAMS>
```

TableE.3.1: fmtp parameters

DESCRIPTOR	**Known in** [OMADRMV2] **as**	**Defined values for OMA DRM v2**	**Default value for OMA DRM v2**
ISMACrypSalt		*Base64 encoded 64-bit number*	0
ISMACrypKey	Right Issuer URL	(uri) *string*	
ISMACrypKMSID		OMA2	OMA2
ISMACrypKMSVersion		0x0000200	0x0000200
ISMACrypKMSSpecificData	ContentID	*URI, quoted using <">*	""

For examples of fmtp statements, see Annex F.

E.3.2 IPMP Signaling

ContentIdentificationDescriptor descriptor [14496-1] SHOULD BE used to identify the content in the IOD.
Concerning the protection scheme signaling, this solution SHOULD use a specific OMA DRM v2.0 IPMPX tool to manage OMA DRM v2 rights object acquisition protocol. IPMPX tool specific information and a rights issuer URL SHOULD BE inserted in the IOD. It is important that the IPMP_Descriptor and the IPMP_ToolListDescriptor both refer to the ISMA and OMA Tool IDs.

Annex F: SDP Examples (Informative)

Note about transport type: In media announcement, "SRTP/AVP" MUST be used when either SRTP authentification or SRTP encryption is used. In all other cases, "RTP/AVP" MUST be used (even though the stream is encrypted).

F.1 The enc-mpeg4-generic encrypted and authenticated MPEG-4 audio mode

Notes:

- ISMACrypSelectiveEncryption: this example uses selective encryption, so this field is explicitly set to 1 in the fmtp line.

Example: (for AAC-hbr mode)

```
m=audio 0 RTP/AVP 96
a=rtpmap:96 enc-mpeg4-generic/22050
a=fmtp:96 streamtype=5; profile-level-id=15; mode=AAC-hbr; config=1388; SizeLength=13; IndexLength=3;
IndexDeltaLength=3; ISMACrypSelectiveEncryption=1; ISMACrypKey=(uri)shttp://talkingHeads.isma.tv
a=mpeg4-esid:1
a=crypto:1 AES_CM_128_HMAC_SHA1_80
   inline:azerRTazad1223dsdsfEhtgdjj12ZSzerefigtyt|2^20|1:32
   UNENCRYPTED_SRTP
```

F.2 The enc-mpeg4-generic encrypted MPEG-4 video mode

Notes:

- ISMACrypDeltaIVLength: this parameter is not used since each packet contains only one AU or AU fragment.
- ISMACrypKeyIndicatorPerAU: this parameter is not used since each packet contains only one AU or AU fragment.

Example:

```
m=video 0 RTP/AVP 96
a=rtpmap:96 enc-mpeg4-generic/600
a=fmtp:96 streamtype=4; profile-level-id=1; mode=mpeg4-video;
config=000001b0f3000001b50ee040c0cf000001000000012000844 0fa28202056a21f; DTSDeltaLength=22;
RandomAccessIndication=1; ISMACrypKey=(uri)shttp://talkingHeads.isma.tv
a=mpeg4-esid:2
```

F.3 The enc-mpeg4-generic encrypted AVC video mode

Notes:

- ISMACrypDeltaIVLength: this parameter is not used since each packet contains only one AU or AU fragment.
- ISMACrypKeyIndicatorPerAU: this parameter is not used since each packet contains only one AU or AU fragment.

Example :

```
m=video 0 RTP/AVP 96
a=rtpmap:96 enc-mpeg4-generic/90000
a=fmtp:96 streamtype=4; mode=avc-video; config=0142E00DFFE1000A6742E00D965202C12C8001000468CE3C80;
DTSDeltaLength=22; RandomAccessIndication=1; ISMACrypKey=(uri)shttp://talkingHeads.isma.tv
a=mpeg4-esid:3
```

F.4 Interoperability with OMA DRM Version 2.0

Example: CELP-cbr mode

m=audio 0 RTP/AVP 96
a=rtpmap:96 enc-mpeg4-generic/16000/1
a=fmtp:96 streamtype=5; profile-level-id=14; mode=CELP-cbr; config=440E00; constantSize=27; constantDuration=240;
ISMACrypSelectiveEncryption =1;
ISMACrypSalt=aXNtYUITTUE; ISMACrypKey=(uri)http://www.rightsserver.org/; ISMACrypKMSID=OMA2;
ISMACrypKMSVersion=512; ISMACrypKMSSpecificData ="content145678@ContentIssuer.com"
a=mpeg4-esid:4

Annex G: Use of ISMACryp prior to OMA DRM 2.0 super-distribution\ (informative)

G.1 Introduction

Assume a mobile operator introduces OMA DRM 2.0 based services over a 3G mobile network, complemented with delivery over a broadcast network, such as DVB-H. Much content is real-time streamed, using RTP, in particular over the broadcast network. The user can store the streamed content on his device and has the option to share the recorded content that he particularly likes with others by sending it to his friends so that they can consume the content on their OMA DRM 2.0 compliant devices.

This concept is called super-distribution, whereby the content is first distributed to the primary users, but where these primary users can distribute it further to their friends, while these friends can further distribute it again to their friends, etc. Of course the friends typically will need to pay for the rights to consume the super-distributed content, but such payment is at the operator's discretion.

For such applications, the use of ISMACryp is very interesting, as ISMACryp supports storage of the content and optional further distribution without any re-encryption. However, for consumption at OMA DRM 2.0 compliant devices, the stored content has to comply with the OMA DRM 2.0 specification. This can be achieved easily by using ISMACryp in a specific manner. This Annex describes the required constraints for the use of ISMACryp so that subsequent OMA DRM 2.0 super-distribution of the content to OMA DRM 2.0 clients can take place without any re-encryption of the content.

G.2 ISMACryp streaming followed by OMA DRM 2.0 super-distribution

In figure G-1, OMA DRM 2.0 super-distribution subsequent to ISMA streaming is depicted. The content to be streamed over RTP is either encrypted in real time or is stored in encrypted form in a file. When streaming, the encrypted content is packetized in RTP packets as specified by ISMACryp and the RTP packets are broadcast to the users. When recording, the user stores the encrypted content in a file; so as to be OMA DRM 2.0 compliant, the file complies with the (P)DCF format specified in OMA DRM 2.0. For random access purposes and various other reasons, it is strongly recommended to store the streamed content in a PDCF file instead of a DCF file. Once the content is stored in the PDCF file, the user can send the file to the OMA DRM 2.0 clients of his friends, and, if so desired, these friends can distribute the files further to their friends, etc., as depicted in figure G-1.

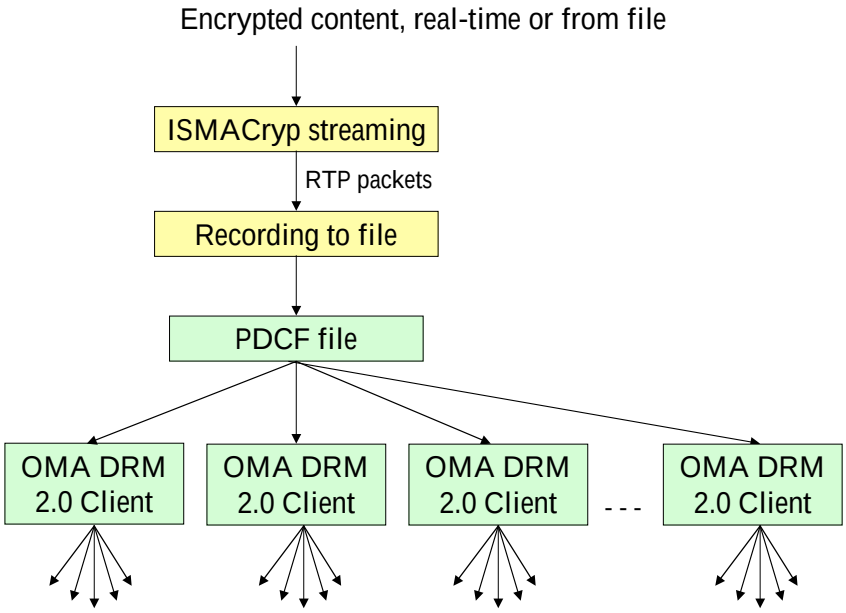


Figure G-1 OMA DRM 2.0 super-distribution subsequent to ISMA streaming

Users receiving super-distributed content can only playback the content if they obtain playback rights from a Rights Issuer. To allow users to contact the Rights Issuer for this purpose, the Common Header Box in the super-distributed PDCF file must contain a Rights Issuer URL. To allow the Rights Issuer to provide the Rights for the content, the super-distributed content in the PDCF file must be identified by a unique content-id that must be provided in the Common Header Box. The content-id may contain timing information (such as date and start / end times) of the recorded content and may be included in the Rights Issuer URL. Next to the content-id also the group-id concept as defined in OMA DRM 2.0 may be used. In case the Rights Issuer URL contains sensitive information, integrity protection on the Rights Issuer URL may be needed, e.g. by using a key derived from the content encryption key. Note also that the Rights Issuer can only provide Rights for the super-distributed content if the Rights Issuer has knowledge of the key that is used to encrypt the content. The format of the Rights Issuer URL, the format of the content-id and the associated issues are application specific and beyond the scope of this Specification, but applications must ensure their proper definition.

G.3 Requirements for ISMACryp streaming

In Annex E, "Interoperability with OMA DRM Version 2.0" it is described how to apply OMA DRM 2.0 on top of the file format specified by ISMACryp, but not how to produce an OMA DRM 2.0 compliant PDCF file from one or more ISMACryp streams. The following should be taken into account to ensure that the encrypted content received after ISMACryp streaming can be stored in OMA DRM 2.0 compliant PDCF files.

1. OMA DRM 2.0 and ISMACryp use AES_CTR_128 as encryption algorithm, which allows the construction of OMA DRM 2.0 compliant streams without any re-encryption, but in DCF the salt key is not supported. When encrypting an ISMACryp stream, then a salt key of zero MUST be used. Note: in DCF, the initial value of the 128 bit counter is prefixed to the ciphertext, and the counter is incremented for each AES cipherblock by 1 (modulo 2^128).

2. OMA DRM 2.0 DCF supports selective encryption, but requires that the selective encryption flag must be transported for each Access Unit; therefore the ISMACrypSelectiveEncryption in SDP must be set to 1.
3. OMA DRM 2.0 PDCF does not support key cycling, hence this feature in ISMACryp must not be used. Consequently, for ISMACryp in SDP the ISMACrypKeyIndicatorLength must be set to 0.
4. The structure of the AU header as used in ISMACryp differs from the AU header in PDCF; therefore the AU header in the ISMACryp RTP packets must be transformed in PDCF compliant AU headers; in this context it should be noted that in PDCF allows an AU to consist of a group of samples.
5. In ISMAcryp, AUs are encrypted as a sequence of bytes without any requirement for alignment between AUs and AES blocks. In PDCF however a new cipherblock starts at the beginning of each "PDCF Access Unit" (corresponding to a group of one or more samples), and hence, unlike in ISMACryp, "PDCF Access Units" are always AES Block aligned. Thus, ISMACryp uses a byte counter and the concept of a byte stream offset, while PDCF simply uses an AES block counter, whereby each AES block consists of 16 bytes. As a consequence, an ISMACryp stream can only be packetized in an OMA DRM 2.0 compliant PDCF without re-encryption if AES Block alignment is applied within the ISMACryp stream. Therefore ISMACryp must provide the IV information for each "PDCF Access Unit" carried in the RTP payload, and for each "PDCF Access Unit" the IV data must indicate the start of a new AES block, thereby typically introducing a IV discontinuity at the beginning of each "PDCF Access Unit".
6. In ISMACryp an AU is equivalent with a sample, but in OMA DRM 2.0 it is specified that one "PDCF Access Unit" may contain multiple samples, which allows storage of small samples (such as AMR speech samples) in an efficient manner in a PDCF file. Though this feature is not very useful for the samples that can be transported by this ISMACryp specification, its usage is described here for the purpose of future ISMACryp specifications that may be capable of carrying small samples. When multiple samples are to be contained in one "PDCF Access Unit", then it is sufficient to apply in ISMACryp AES block alignment at the level of the group of samples that will form a "PDCF Access Unit". However, in ISMACryp an IV or IV delta is provided for each AU and hence each sample. For storage of multiple samples in one "PDCF Access Unit", (a) no IV discontinuity must be applied at non-first samples that form a "PDCF Access Unit" (consequently, non-first samples are typically not AES block aligned), and (b) ISMACryp receivers that construct a PDCF file must verify for each AU whether it is AES block aligned and whether an IV discontinuity occurs. Each AU that is not AES block aligned is a non-first sample of a "PDCF Access Unit"; for such AU there must be no IV discontinuity. If an AU is AES block aligned, and there is no IV discontinuity, it may be either a first or a non-first sample of a "PDCF Access Unit" and the choice is at the receiver's discretion. If an AU is AES block aligned, and there is an IV discontinuity, it is the first sample of a "PDCF Access Unit". See also figure G-2.

AU is AES block aligned	IV discontinuity at AU	
yes	Yes	First sample in "PDCF Access Unit"
yes	No	First or non-first sample in "PDCF Access Unit"
no	Yes	Forbidden
no	No	Non-first sample in "PDCF Access Unit"

Figure G-2 AUs and samples in a "PDCF Access Unit".

G.4 Conclusion

Content that is streamed to devices by means of ISMACryp can be super-distributed to OMA DRM 2.0 compliant devices if the requirements described in this Annex are taken into account. Section G.3 provides the requirements, such as ISMACryp parameter initialisation, IV constraints and transformation of headers, that are within the scope of this Specification. However, some other requirements need further definition at application level, and are therefore beyond the scope of this Specification. For this purpose, in summary, applications must ensure that the following is specified:

- A method for unique identification of super-distributed content, so that the Rights Issuer, when so requested by recipients of the super-distributed content, knows for which content to grant rights.
- A method to ensure that the Rights Issuer has knowledge of the key that is used to encrypt the super-distributed content, so as to allow the Right Issuer to provide the correct keys for decryption of the content.
- The format of the Rights Issuer URL in the Common Header Box, so as to allow proper communication between the recipient of the super-distributed content and the Rights Issuer.
- A method to apply integrity protection on the Rights Issuer URL, if needed, to protect sensitive information contained in this URL.

Users can distribute OMA DRM 2.0 protected content without explicit permission, but of course this only makes sense if the content provider / rights issuer is willing and prepared to grant rights to the recipient users to consume the content. Therefore it is desirable that applications provide means to indicate whether OMA DRM 2.0 super-distribution is a meaningful option.