Marlin Broadband Transport Stream Specification

Version 2.0.2
Final

Source Marlin Developer Community
Date September 20, 2018

Notice

THIS DOCUMENT IS PROVIDED "AS IS" WITH NO REPRESENTATION OR WARRANTY, EXPRESS OR IMPLIED, CONCERNING THE COMPLETENESS, ACCURACY, OR APPLICABILITY OF ANY INFORMATION CONTAINED IN THIS DOCUMENT. THE MARLIN DEVELOPER COMMUNITY (“MDC”) ON BEHALF OF ITSELF AND ITS PARTICIPANTS (COLLECTIVELY, THE "PARTIES") DISCLAIM ALL LIABILITY OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED, ARISING OR RESULTING FROM THE RELIANCE OR USE BY ANY PARTY OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. THE PARTIES COLLECTIVELY AND INDIVIDUALLY MAKE NO REPRESENTATIONS CONCERNING THE APPLICABILITY OF ANY PATENT, COPYRIGHT (OTHER THAN THE COPYRIGHT TO THE DOCUMENT DESCRIBED BELOW) OR OTHER PROPRIETARY RIGHT OF THIS DOCUMENT OR ITS USE, AND THE RECEIPT OR ANY USE OF THIS DOCUMENT OR ITS CONTENTS DOES NOT IN ANY WAY CREATE BY IMPLICATION, ESTOPPEL OR OTHERWISE, ANY LICENSE OR RIGHT TO OR UNDER ANY PATENT, COPYRIGHT, TRADEMARK OR TRADE SECRET RIGHTS WHICH ARE OR MAY BE ASSOCIATED WITH THE IDEAS, TECHNIQUES, CONCEPTS OR EXPRESSIONS CONTAINED HEREIN.

Use of this document is subject to the agreement executed between you and the Parties, if any.

Any copyright notices shall not be removed, varied, or denigrated in any manner.

Intellectual Property

A commercial implementation of this specification requires a license from the Marlin Trust Management Organization.

Contact Information

Feedback on this specification should be addressed to: editor@marlin-community.com

Contact information for the Marlin Trust Management Organization can be found at: http://www.marlin-trust.com/

Contents

Deleted: 21

1 Introduction

This document describes how to use Marlin with content packaged as an MPEG-2 Transport Stream (MPEG-2 TS) [MPEG2-TS]. This enables for example Marlin protection of content that is delivered over an IP Multicast Channel (IPMC).

1.1 Document Organization

This document is organized as follows:

- (This) introduction, including abbreviations, definitions and references.
- BBTS
    - Scope
    - The relation between the MPEG2TS deployment environment and Marlin.
    - The content protection format
- Single Key Layer BBTS
    - Scope
    - BBTS Compendium

1.2 Conformance Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119].

These capitalized key words are used to unambiguously specify requirements and behavior that affect the interoperability and security of implementations. When these key words are not capitalized they are meant in their natural-language sense.

All Elements of this specification are considered **Normative** unless specifically marked **Informative**. All Normative Elements are **Mandatory** to implement, except where such an element is specifically marked **OPTIONAL**. Finally, where **Normative** elements are described as **OPTIONAL**, they MAY be omitted from an implementation, but when implemented, they MUST be implemented as described.

1.3 References

1.3.1 Normative references

[8pus]	Octopus DRM Technology Platform Specifications, Version 1.0
[AES]	Advanced Encryption Standard (AES), FIPS PUB 197, November 26, 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[IEC62455]	Internet protocol (IP) and transport stream (TS), based service access IEC 62455 © IEC:2010(E) Edition 2, 2010-12
[MPEG2-TS]	Information technology – Generic coding of moving pictures and associated audio information: Systems, ISO/IEC 13818-1:2000(E), Second edition, 200-12-01
[MRL CORE]	Marlin Core System Specification Version 1.3
[MS3]	Marlin – Simple Secure Streaming Specification, Version 1.1

[MOC] Marlin Output Control Specification, Version 1.0

[OMARLIN] OMArlin specification, version 1.0, final.

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels. Internet Engineering Task Force, 1997

[RFC2396] Uniform Resource Identifiers (URI): Generic Syntax

[SHA1] FIPS PUB 180-1. *Secure Hash Standard.* U.S. Department of Commerce/National Institute of Standards and Technology. http://www.itl.nist.gov/fipspubs/fip180-1.htm

[TTS] ARIB STD-B24 Version 5.1, Data Coding and Transmission Specification for Digital Broadcasting, Association of Radio Industries and Businesses, March 14, 2007.

[DVBCA] ETSI ETR 289, Digital Video Broadcasting (DVB); Support for the use of scrambling and Conditional Access (CA) within digital broadcasting systems, October 1996

[MURIT10] URI Templates for Marlin, Version 1.0, Sept 10, 2007.

[DVBID] DVB Identifiers
http://www.dvbservices.com/identifiers/

1.3.2 Informative references

[Marlin] Marlin
http://www.marlin-community.com/

1.4 Acronyms & Abbreviations

AES	Advanced Encryption Standard
BBTS	Broadband Transport Stream
CA	Conditional Access
CA-ID	Conditional Access Identity
CAS	Conditional Access System
CAT	Content Access Time
CBC	Cipher Block Chaining
CK	Content Key
DRM	Digital Rights Management
ECM	Entitlement Control Message
EMM	Entitlement Management Message
EPG	Electronic Program Guide
ES	Elementary Stream
ETSI	European Telecommunications Standards Institute
ID	Identity

IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IP	Internet Protocol
IV	Initialization Vector
KSM	Key Stream Message
License	Marlin License or a MS3 Stream Access Statement (SAS)
MPEG	Moving Pictures Expert Group
MPEG-2 TS	MPEG-2 Transport Stream
PAT	Program Association Table
PEK	Program Encryption Key
PID	Packet Identifier
PMT	Program Map Table
RFC	Request For Comments
SAS	Stream Access Statement
SKL	Single Key Layer
socID	service operation centre ID
TK	Traffic Key
TS	Transport Stream
TTS	Timed Transport Stream
URL	Uniform Resource Locator

2 BBTS

2.1 Scope (informational)

This section describes the relation between the technology defined in this document and its environment.

2.1.1 Deployment

Figure 1 indicates the typical deployment of an MPEG2 TS system. A Marlin Client receives information on what content is available from the web. This information can be provided by a web-page or using an Electronic Program Guide (EPG).

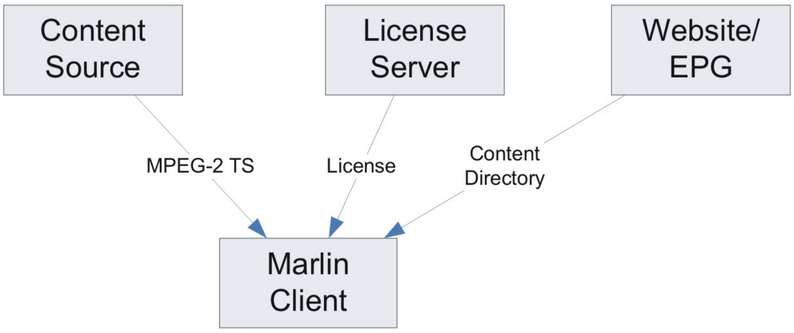


Figure 1: Marlin Clients and MPEG2 TS systems

In the case of an EPG, EPG information is provided to the client. Based on this information, a local application constructs the EPG to be shown on screen, In the case of a web site. The EPG information is provided as a web page. When a content item is selected, the Client is instructed to connect to a certain Content Source to retrieve the content and, in some cases to retrieve a License for the content from a License Server. The content can be sent over IP Multicast.

In this document, the content is an MPEG-2 TS, as defined in section 15 “Protection of MPEG2 TS-based IP systems” of the IEC 62455 standard [IEC62455], formatted as defined in section 2.2.

2.1.2 Marlin and streaming content

For Marlin, content access to a stream should be seen as accessing a file that is downloaded from a content service; the same mechanisms are used. When a client requires access to the content, it acquires and evaluates the License. When content access is granted, the client is allowed to access the content; typically for a certain period of time. During the indicated time, the client is allowed to decrypt all Entitlement Control Messages (ECMs) and the content corresponding to the content item referred to by the License.
The License must be re-evaluated when the content id of the content changes.

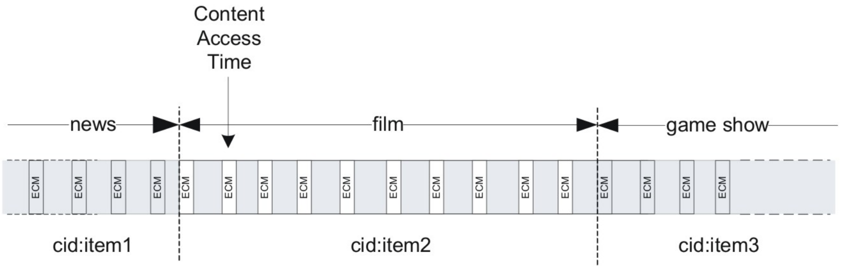


Figure 2: Content access and License evaluation in MPEG2 TS

In the stream indicated in Figure 2, first the news is broadcasted (content item 1), then a film (content item 2), followed by a game show (content item 3). Each content item has its own content id. The content id of each content item is indicated in the ECM. Thus, if the client has evaluated the License and has the permission to access the content of item 2 (film), it may decrypt the content. At the moment the first ECM of content item 3 has been received, a License for content item 3 is required. If no valid License is available, the device will not be allowed to further decrypt the content.

2.1.3 Content access

An important aspect of MPEG2 TS is the time it takes to switch between streams. There are two possible procedures: the standard and the optimized one.

2.1.3.1 Standard procedure

Figure 3 indicates the standard content access sequence.

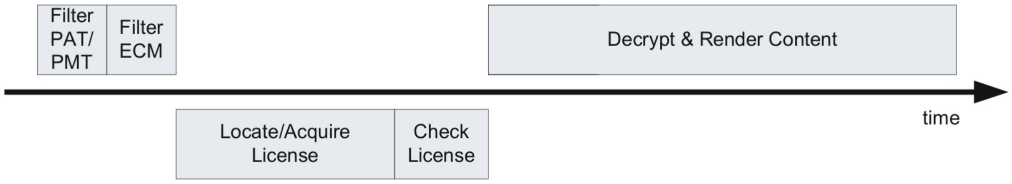


Figure 3: Standard Content Access Procedure

In the standard content access procedure the client accesses the content stream, and filters the PAT and PMT to determine where the audio and video elementary streams and the ECMs are located.

After the first ECM has been filtered, the corresponding License should be located. If the License is not available, the device has to update its registration information and/or acquire a (new) Licence. For this purpose, the service provider can include information in the EPG or download descriptor.

When the License is located, its integrity is checked and the License is evaluated. If content rendering is allowed, the content can be decrypted and decoded.

2.1.3.2 Optimized procedure

In the case of a service-based licensing scheme, the content id of the stream and the License are unlikely to change often. In this case, the standard procedure can be improved if the client remembers which License corresponds to which MPEG2 TS (Figure 4) This will allow it to locate and check the License while or before the PAT/PMT and ECM are being filtered.

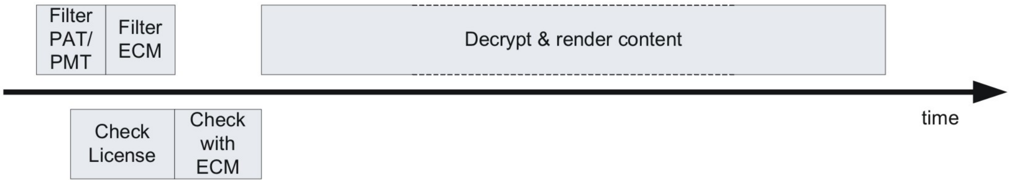


Figure 4: Optimized content Access Procedure

If the License allows content rendering and the ECM has been filtered, the content ID from the ECM is compared with the Content ID(s) in the License. If they match, the content can be decrypted and decoded. In this case, rendering can start almost immediately.

If the License does not allow content rendering, the registration information has to be updated and/or a License has to be acquired as has been described in section 2.1.3.1.

2.2 Content protection and formats

Streaming content protected by Marlin uses an MPEG-2 Transport Stream [MPEG2-TS] or an MPEG-2 Timed Transport Stream (TTS) [TTS].

2.2.1 Stream format

The stream SHALL include Entitlement Control Messages (ECMs) and MAY include Entitlement Management Messages (EMMs). As such, only PAT and PMT are REQUIRED and, the CAT is OPTIONAL.

As defined in [MPEG2-TS], a PMT refers to one or more elementary streams (ES). Transport stream packets protected by Marlin SHALL be encrypted as specified in section 2.2.2. Transport streams protected using the mechanism defined in this document SHALL include CA-descriptors and ECMs as defined in section 2.2.3.

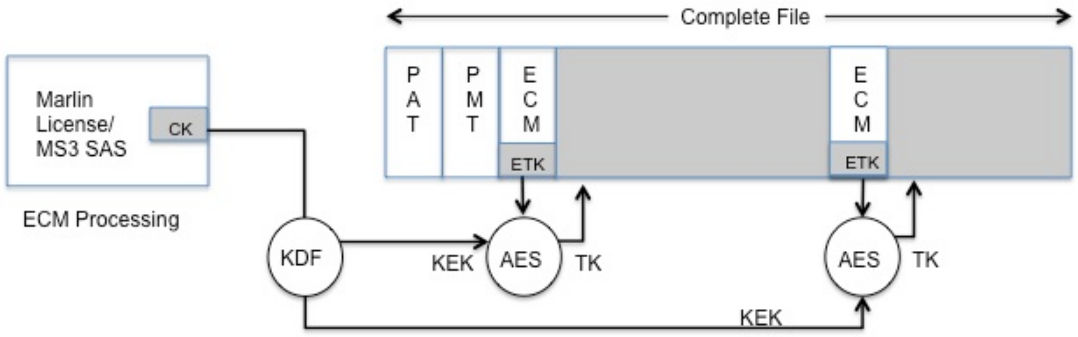


Figure 5: Content Access for ECM Processing implementations

2.2.2 Content encryption

The MPEG-2 TS/TTS is partially encrypted as is the case in the Conditional Access System (CAS) defined in [MPEG2-TS]. Whether a TS/TTS packet is encrypted SHALL be signaled with transport_scrambling_control bits in the TS packet header, according to Table 1.

transport_scrambling_control bits	Description
10	The TS/TTS packet is encrypted with the “even” Scramble Key.
11	The TS/TTS packet is encrypted with the “odd” Scramble Key.
00	The TS/TTS packet is not encrypted.
01	Reserved.

Table 1: transport_scrambling_control bits

The following applies:

- The encryption algorithm SHALL be either AES or DVB/CSA [IEC62455 section 6.4.5]. When AES is used, then it SHALL use a 128-bit key using the Cipher Block Chaining (CBC) encryption mode with the residual termination block process as specified in [IEC62455 section 6.4.6]. When DVB/CSA is used it SHALL follow the guidance given in [IEC62455 section 6.4.5].
- The encryption SHALL be performed per TS/TTS packet. PES level scrambling SHALL NOT be used.
- The Traffic Keys encrypting TS/TTS packets are encrypted and enclosed in ECMs defined in section 2.2.4.1.
- The ECM includes information to calculate the Marlin Content ID with which the content is associated with License.

The use of odd and even keys in transport stream encryption is illustrated in Appendix A.

2.2.3 BBTS CA_descriptor

Programs/Services protected using the mechanism defined in this document SHALL include in the PMT the BBTS CA_descriptor defined below and the ECMs as defined in section 2.2.4.1.

Transport streams protected using the mechanism defined in this document MAY include in the CAT the BBTS CA_descriptor defined below and the EMMs as defined in section 2.2.5

DRM Clients SHALL support presence of the BBTS CA_descriptor in the PMT and the CAT and SHALL support receipt of ECM’s in section 2.2.4.1 and EMM’s specified in desction 2.2.5.

The BBTS CA descriptors SHALL be formatted as specified in Table 2 - BBTS CA_descriptor.

Syntax	No. of bits	Mnemonic	Value
BBTS CA_descriptor() {			
descriptor_tag	8	Uimsbf	9
descriptor_length	8	Uimsbf	
CA_system_ID	16	Uimsbf	
MPEG2_Reserved	3	Bslbf	
CA_PID	13	Uimsbf	
for (i = 0; i < N; i++) {			

private_data_byte	8	uimsbf	
}			
}			

Table 2 - BBTS CA_descriptor

descriptor_tag, MPEG has defined the value of 9 to the CA-descriptor.
descriptor_length, the number of bytes of the descriptor immediately following descriptor_length field.
CA_system_ID, Marlin system ID, assigned by [DVBID].
CA_PID, the PID on which the ECMs or EMMs as defined in sections 0 and 2.2.5 can be found
MPEG2_reserved, bits reserved by [ISO/IEC 13818-1].

private_data_byte*, the private information specific to this specification. This field consists of descriptors defined in Table 3 - Descriptors in private_data_byte*

and following clauses.

Deleted: *Tab*

Descriptor_tag	Identification	Mandatory
0x00	Reserved	n/a
0x01	Service Base CID descriptor	No
0x02	Reserved for Single_key_layer_descriptor	No
0x03	KeyDerivationFunction_descriptor	Yes
0x04 – 0xff	Reserved	n/a

Table 3 - Descriptors in private_data_byte

2.2.3.1 ServiceBaseCID descriptor

The definition of the Service Base CID descriptor is specified below. Presence of this descriptor is OPTIONAL.

Syntax	No. of bits	Mnemonic	Value
ServiceBased_descriptor() {			
descriptor_tag	8	uimsbf	0x01
descriptor_length	8	uimsbf	
for (i = 0; i < N; i++) {			
baseCID_byte	8	uimsbf	
}			
}			

Table 4 - ServiceBaseCID_descriptor

descriptor_tag, MUST have the value of 0x01 to signal the ServiceBaseCID_descriptor
descriptor_length, the number of bytes of the descriptor immediately following descriptor_length field.
baseCID_byte, the Service Base Content ID for this content.

As specified in section 2.2.4.2, the Service Base Content ID is the base part of the Content IDs of the programs and/or services contained in the BBTS. These Content IDs MUST be globally unique URIs. It is therefore RECOMMENDED that the Service Base Content ID is composed as:

serviceBaseCID=organization||"-"||service

Where:

- organization equals an organization identifier for the organization providing the service that is registered with Marlin as described in [MRL CORE] section 1.3.2.

- service is an identifier for the service chosen by the organization providing the service.

The Service Base Content ID MAY be embedded into the BBTS_CA_Descriptor, as specified in Table 2 - BBTS CA_descriptor.
DRM Clients MUST support receipt of the Service Base Content ID from the BBTS_CA_descriptor.

Alternatively, the Service Base Content ID MAY be delivered to the Marlin Client and associated to the BBTS via an unspecified out-of-band mechanism. It could for example be derived from an EPG, specified outside of scope for Marlin.

One or more BBTS_CA_descriptors may be present in a PMT or CAT. Each BBTS_CA_descriptor MAY via the CA_PID field reference a different stream of ECMs or EMMs, all of which enable access to the same content. This mechanism allows multiple Marlin service providers, with different Service Base Content ID's, different ECM or EMM streams and different license management, to independently provide access to the same BBTS.

When providing access to a BBTS with multiple parallel BBTS_CA_descriptors, the DRM Client SHOULD check if it has a License associated with any possible ContentID that can be composed (as specified in section 2.2.4.1) from the Service Base Content IDs in a BBTS_CA_descriptor and the programme_CID_extensions and service_CID_extensions, that are present in the referenced IEC62455 ECMs. It is outside of scope of this specification to specify a more efficient mechanism to determine which ECM stream and ContentID a License is available.

2.2.3.2 Key Derivation Function Descriptor

The definition of the KeyDerivationFunction_descriptor is specified below. For this version of the specification, this descriptor is REQUIRED to be present in the BBTS CA_descriptor.

This descriptor signals to an ECM Processing implementation whether or not a key derivation function has to be applied to derive the key that is subsequently used to encrypt or decrypt Programme Encryption Keys or Traffic Keys carried in the ECMs.

The KDF defined in §2.2.4.1.1 SHALL be applied to the Content Key when AES-128-CBC is used to encrypt PEKs or Traffic Keys carried in the IEC 62455 ECM. Thus kdf_type 0x01 defined in Table 6 - Key Derivation Function type values SHALL be signaled in this descriptor.

N.B.: This descriptor will not be known to ECM Processing implementations that only conform to versions 1.0 and 1.1 of this specification and therefore the key protection defined here is not backward compatible with versions 1.0 and 1.1. Such implementations can detect this incompatibility by inspecting the content_key_index in the key stream messages.

Syntax	No. of bits	Mnemonic	Value
KeyDerivationFunction_descriptor() {			
descriptor_tag	8	uimsbf	0x03
descriptor_length	8	uimsbf	0x01
kdf_type	8	uimsbf	See Table 6 - Key Derivation Function type values
}			

Table 5 - KeyDerivationFunction_descriptor

descriptor_tag, MUST have the value of 0x03 to signal the KeyDerivationFunction_descriptor

descriptor_length, MUST be 0x01 as the private data for this descriptor must signal the type of the key derivation function signaled in the kdf_type byte

kdf_type MUST be one of the values defined in Table 6 - Key Derivation Function type values

kdf_type value	**Description**
0x00	None. No key derivation function used.
0x01	Truncated SHA1 as defined in §2.2.4.1.1.
0x02 – 0xff	Reserved.

Table 6 - Key Derivation Function type values

2.2.4 Entitlement Control

2.2.4.1 IEC 62455 ECM

As specified in section 14.2 of [IEC62455], the table from ETSI ETR 289 [DVBCA] shall be used to carry the KSM defined in section 7.2 of [IEC62455] as payload. The value of the traffic protection protocol and syntax of the KSM for this protocol have been revised in the referenced version of [IEC62455]. They are repeated below as a convenience to the reader.

The following usage restrictions SHALL apply to the KSM:

- traffic protection protocol is set to KSM_ALGO_MPEG2_TS_CRYPT.

Name	**Value**
KSM_ALGO_MPEG2_TS_CRYPT	7

Table 7 - Value from [IEC62455] Table 21

Syntax	No. of bits	Mnemonic
if (traffic_protection_protocol == KSM_ALGO_MPEG2_TS_CRYPT) {		
content_key_index	4	uimsbf
odd_even_flag	1	bslbf
cipher_mode	3	uimsbf
reserved_for_future_use	8	bslbf
if (cipher_mode == 0x1) {		
initial_vector_length	8	uimsbf
initial_vector	8 * initial_vector_length	bslbf
If (next_traffic_flag == KSM_FLAG_TRUE) {		
next_initial_vector	8 * initial_vector_length	Bslbf
}		
}		
}		

Table 8 - Syntax from KSM_ALGO_MPEG2_TS_CRYPT specific fields of the key_stream_message (per [IEC62455] Table 7)

- Traffic_authentication_flag is set to KSM_FLAG_FALSE (traffic authentication is not used).

  In case the encryption method is chosen to be AES:

  - content_key_index is set to 0xF (Traffic Key is AES key with 128 bit length. A key derivation function has to be applied to derive the key that is subsequently used to encrypt or decrypt Programme Encryption Keys or Traffic Keys carried in the ECMs, see 2.2.3.2).
  - cipher_mode is set to 0x1 (CBC mode).

    - initial_vector_length is set to 16 (128 bits).
    - encrypted_traffic_key_material_length is set to 16 (128 bits).

  In case the encryption method is chosen to be CSA:

    - content_key_index is set to 0x0 (Traffic Key is DVB-CSA key with 64 bit length).
    - cipher_mode is set to 0x2 (CSA mode).
    - encrypted_traffic_key_material_length is set to 16 (128 bits).

- permissions_flag is set to 0.
- the access criteria descriptor loop MAY contain any access criteria descriptor. The DRM Client MUST support the copy_control_information access criteria descriptor as specified in [IEC62455] and MUST handle it according to section 2.2.4.3 and 2.2.4.4. The DRM Client MAY ignore any other access criteria descriptor. Note that non-Marlin access criteria descriptors may be defined which may need to be enforced by the client device for compliancy to other specifications.

2.2.4.1.1 Truncated SHA1 Derived AES key

When the KeyDerivationFunction descriptor defined in §2.2.3.2 signals the use of the TruncatedSHA1 KDF (kdf_type=0x01) in the BBTS CA_descriptor, then the traffic key material or the programme encryption key (PEK) carried in the IEC 62455 ECM SHALL be encrypted with a key encryption key (KEK) derived from the Content Key (CK) (obtained from the License) using the algorithm described below.

This key encryption key SHALL be derived from CK using the following key derivation function (KDF):

KEK = TRUNCATE(SHA1(CK))

Where,

- CK is the content key (IEC 62455 service/programme key)
- SHA1 is the one-way hash function defined in [SHA1]
- TRUNCATE takes the 128 most significant bits of the 160-bit output of SHA1

2.2.4.2 Relation between Marlin Broadband License and BBTS

As specified in [IEC62455], an IEC 62455 ECM SHALL either contain key material that provides access to the current program or to the whole service or both. Programs and Services are accessed with a different key. This requires a program and a service content identifier. The key management is defined in the IEC 62455 ECM and illustrated in Appendix B.

In order to relate a License to content (program or service) in a BBTS stream, it is needed to construct the Content ID of the content as it is referred to from the License defined in [MRL CORE] or [MS3].

The Content ID for a program SHALL be constructed using the syntax defined below:
CID="cid:"||socID ||"#P"|| serviceBaseCID||"@"||hex(programme_CID_extension)

The Content ID for a service SHALL be constructed using the syntax defined below:
CID ="cid:"||socID||"#S"||serviceBaseCID||"@"||hex(service_CID_extension)

Where,

- socID equals "marlin" or is retrieved via an unspecified out-of-band mechanism.
- serviceBaseCID is the Service Base Content ID retrieved out-of-band or from the BBTS_CA_descriptor as described in section 2.2.3.

- programme_CID_extension is part of the IEC 62455 ECM described in section 2.2.4.1. .
- service_CID_extension is part of the IEC 62455 ECM described in section 2.2.4.1. .
- The hex() function is a hexadecimal presentation of the parameter containing hexadecimal characters 0-9 and a-f (in lowercase) with possible preceding zeros. See [IEC62455]

The ContentIDs used for the programs and services contained in the BBTS MUST be globally unique URIs.

From the License for a BBTS, the DRM Client SHALL compute an authentication_key via:

authentication_key = HMAC-SHA1(CK, SEED)

Where,

- CK is the content key
- SEED is the fixed string ‘MRL-BBTS-AUTH-KEY-10’ as ASCII characters.

When a DRM Client encounters an IEC62455 ECM in the process of providing access to Content in a BBTS based on a License associated with a ContentID for a program, the DRM Client SHALL compute the programme MAC using the authentication key and HMAC-SHA-1-96 according to [IETF RFC 2104] and [IETF RFC 2404] calculated over all fields of the IEC62455 KSM preceding the programme_MAC field. In case the computed MAC is unequal to the programme_MAC field in the encountered IEC62455 ECM, the DRM Client SHALL disregard the encountered IEC62455 ECM.

When a DRM Client encounters an IEC62455 ECM in the process of providing access to Content in a BBTS based on a License associated with a ContentID for a service, the DRM Client SHALL compute the service MAC using the authentication key and HMAC-SHA-1-96 according to [IETF RFC 2104] and [IETF RFC 2404] calculated over all fields of the IEC62455 KSM preceding the service_MAC field. In case the computed MAC is unequal to the service_MAC field in the encountered IEC62455 ECM, the DRM Client SHALL disregard the encountered IEC62455 ECM.

2.2.4.3 Output Control

Marlin defines a default set of output control information and the output control mechanism defined in [MOC] enables a deviation from the default set by including permission(s)/obligation(s) in a License.
This section describes the processing rule to compute the effective output control information from the specified default set, permission(s)/obligation(s) in License, and the IEC 62455 ECM.

For each output control parameter, the DRM Client SHALL apply the value from the first available source in the following list:

1. Output control information included in a License for the program contentID as Obligations.
2. Output control information included in the copy_control_information access criteria descriptor of ECM
3. Output control information included in a License for the service contentID as Obligations.
4. The default set of output control information.

The relationship between parameters of copy_control_information access criteria descriptor defined in [IEC62455] and output control information defined in [MOC] is shown in Table 9. Parameters which are defined in [MOC] but don’t have corresponding parameters in [IEC62455] are not overridden.

Deleted: [IEC

Deleted: [MC

Formatted: (

Formatted: E

Parameter defined in [IEC62455]	Parameter of Basic CCI defined in [MOC] expressed as an Obligation
EMI	CCI
APS	APS
CIT	ImageConstraintToken
RCT	EPN

Table 9 - Relationship among parameters of [IEC62455] and [MOC]

Deleted: [IEC

Deleted: [MC

Formatted: F

Formatted: F

If a License also contains output control information expressed as a permission, the DRM Client MAY alter the effective output control according to the permission.

2.2.4.4 Copy Control

Export Action defined in [8pus] §3 SHALL be used to export content which means that copy_control_information in ECM is not used for this purpose.

2.2.5 Entitlement Management

2.2.5.1 Rights URL section

The Rights URL section is an EMM that contains all the information potentially needed by a Marlin Client to request rights for the MPEG2-TS stream.

Syntax	No. of bits
Rights_URL_section() {	
Table_id = 0x82	8 uimsbf
section_syntax_indicator	1 uimsbf
DVB_reserved	1 uimsbf
ISO_reserved	2 uimsbf
section_length	12 uimsbf
For (i=0; i<N; i++){	
URL_parameters_byte	8 bslbf
}	
}	

Table 10 - Rights URLs section syntax

table_id, this specification has defined the value of 0x82 for the Rights URLs section.
Section_syntax_indicator, set to 0
DVB_reserved, bit reserved for future use by DVB.
ISO_reserved, bits reserved by [ISO/IEC 13818-1].
Section_length, the number of bytes that follow the section_length field up to the end of the section.
URL_parameters_byte, This field contains at least one or several optional parameters encoded in TLV format. To ensure evolution of the specification, parameters with unknown tag SHALL be ignored.

The parameters that can be found in the URL_parameters_byte of the Rights URLs section are the following:

- Rights Issuer URL parameter
- Silent Rights URL parameter
- Preview Rights URL parameter

The encoding of the text and the character set used for URL data bytes SHALL conform to Annex A of [ETSI EN300 468] unless it contains a URI template as specified in [MURIT10].

The URL's MAY contain a URI template as specified in [MURIT10]. The DRM Client MUST support [MURIT10]. Note that the minimal mandatory processing required by [MURIT10] is to remove the template placeholders (i.e., delimited by a “{“ and “}” character) from the URL or replace it with a “~”.

Rights Issuer URL parameter

The coding of this parameter in the TLV format is the following:

Syntax	No. of bits
Rights_Issuer_URL () {	
Rights_Issuer_URL_tag = 0x02	8 uimsbf
Rights_ Issuer_URL_length	8 uimsbf
For (i=0; i<N; i++){	
Rights_ Issuer_URL_data_byte	8 bslbf
}	
}	

Table 11 - Rights Issuer URL parameter syntax

Rights_ Issuer_URL_tag, this specification has defined the value of 0x02 for the Rights Issuer URL parameter.
Rights_ Issuer_URL_length, specifies the length of the Rights_Issuer_URL_data_bytes in bytes (N).
Rights_ Issuer_URL_data_byte, the Rights Issuer URL for this content.

2.2.5.1.1 Silent_Rights URL parameter

The coding of this parameter in the TLV format is the following:

Syntax	No. of bits
Silent_Rights_URL () {	
Silent_Rights_URL_tag = 0x03	8 uimsbf
Silent_Rights_URL_length	8 uimsbf
Silent_method	8 uimsbf
For (i=0; i<N; i++){	
Silent_Rights_URL_data_byte	8 bslbf
}	
}	

Table 12 - Silent Rights URL parameter syntax

Silent_Rights_URL_tag, this specification has defined the value of 0x03 for the Silent Rights URL parameter.
Silent_Rights_URL_length, adds Silent_method (1 byte) and the length of Silent_Rights_URL_data_byte in bytes (N), so is equal to (N+1).
Silent-method, the silent-method as described in section 4.1.1 in [Omarlin] specification. The method is one byte encoded in this specification as follows:

silent method	Byte encoding
“on-demand”	0x00
“in-advance”	0x01

Silent_Rights_URL_data_byte, the Silent Rights URL for this content.

2.2.5.1.2 Preview_Rights URL parameter

The coding of this parameter in the TLV format is the following:

Syntax	No. of bits
Preview_Rights_URL () {	
Preview_Rights_URL_tag = 0x04	8 uimsbf
Preview_Rights_URL_length	8 uimsbf
For (i=0; i<N; i++){	
Preview_Rights_URL_data_byte	8 bslbf
}	
}	

Table 13 - Preview Rights URL parameter syntax

Preview_Rights_URL_tag, this specification has defined the value of 0x04 for the Preview Rights URL parameter.
Preview_Rights_URL_length specifies the length of the Preview_Rights_URL_data_byte in bytes (N)
Preview_Rights_URL_data_byte, the Preview Rights URL for this content.

2.2.5.1.3 Rights URL section processing

The following processing rules for handling the Rights URLs defined above SHALL be complied with:

- If the MPEG2-TS contains either a Silent rights URL or a Preview rights URL parameter and there is no local available rights, then an attempt to request a Marlin Action Token or a License SHALL be made automatically silently (without further user interaction).
- If the MPEG2-TS contains both a Silent Rights URL and a Preview Rights URL parameter, the parameter appearing first in the Rights URLs section MUST be prioritized and used to attempt to request a Marlin Action Token or a License first.
- If the MPEG2-TS contains a Rights Issuer URL parameter and there is no local available rights and the context is a user-initiated session, requesting user consent and then getting a Marlin Action Token or a License SHALL be attempted by sending a HTTP GET request to the Rights Issuer URL. If the context is not a user-initiated session, then it is RECOMMENDED to abandon the rights acquisition effort.
- When requesting rights to the Rights Issuer URL, either of a Marlin Action Token, a License or a HTML that is defined by a Marlin-adopting system SHALL be returned and appropriately handled.

2.2.6 License for ECM

The usage rule specified in License can vary between real-time stream and non real-time stream. To distinguish the real-time stream from non real-time stream, a timestamp included in ECM which expresses the time and date of the ECM delivery is used during the evaluation of License.

2.2.6.1 Object Path for timestamp in ECM

This section defines the object path for timestamp in ECM to enable System.Host.GetObject Plankton system call to access such a timestamp in ECM. When evaluating a License associated with ECM, the timestamp included in ECM MAY be visible to the Plankton Virtual Machine for the corresponding License under the following container:

Object Path
/Marlin/Stream/timestamp

The value of this object has type Integer as defined in [8pus] §4 and is the number of minutes elapsed since January 1, 1970 00:00:00. The value is a UTC date. This date is converted from the timestamp in the ECM as specified in [IEC62455].

The usage rule specified in the License may require this container to be visible. If the container is not made visible, the usage rule may disallow access to the content.

If a DRM Client supports this timestamp, then it SHALL also support the obligation specified in §2.2.6.2

2.2.6.2 Obligation for ECM

Even when a usage for real-time stream is granted by using the mechanism defined in §2.2.6.1, License can indicate a requirement of monitoring for subsequent ECMs by using the obligation mechanism defined in [8pus] §3.4.

The following obligation parameters are defined for ECM monitoring:

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>ECMMonitor</td><td>ValueList</td><td>
<table>
<tr><th>Type</th><th>Description</th></tr>
<tr><td>Integer</td><td>Monitor interval expressed in seconds. The host application MUST monitor subsequent ECMs regularly at this interval.</td></tr>
<tr><td>Integer</td><td>Time-lag expressed in minutes. The time-lag is the absolute time of the delta between timestamp in ECM and current time.
The host application MUST stop the action if the absolute value of the delta between the two timestamps is bigger than this time-lag.</td></tr>
</table>
</td></tr>
</table>

2.2.6.3 Constraint for ECM Freshness

The following Temporal Constraint ([8pus] §3.3.4.2.2) MAY be included in an ESB to indicate the freshness of the timestamp in ECM is required by the usage rule.

Name	Type	Description
NotLaterThan	Integer	Time-lag expressed in minutes. The time-lag is the absolute time of the delta between the timestamp in the ECM and the current time permitted by the License.

3 Single Key Layer BBTS

3.1 Scope (Informative)

This section describes the rationale for the definition of a Single Key Layer BBTS, which is a special form of a BBTS (§2). The BBTS is designed for live broadcast streaming, allowing clients to tune in to a program at any time Therefore, in order to allow for the client to obtain the traffic keys to decrypt the TS packets, the Entitlement Control Message (ECM) that contains the traffic keys is repeated frequently to facilitate this operation. In addition, BBTS is designed to enable Simulcrypt with (existing) other protection systems. In the past, the CSA encryption algorithm was frequently used to encrypt the TS, This algorithm is vulnerable to attacks, and hence frequent changes of traffic keys were necessary. But for non-broadcast use cases (i.e. on-demand download or streaming) and when using AES, it is not necessary to repeat the ECM through the TS. Essentially, the content key obtained from the License could serve as the traffic key and the need for additional key layer (ECM) is not needed.

Nevertheless, support for the BBTS requires a client to expect key changes and filter ECMs. A client implementation that unwraps the Traffic Key from an ECM to decrypt the BBTS is referred to as an ECM Processing implementation. A manufacturer of ECM Processing implementations must therefore anticipate the related high processing overhead. The rationale for the Single Key Layer BBTS is to allow simpler clients that do not have ECM filtering capability. A Single Key Layer BBTS is a special form of a BBTS in which the Traffic Key is constant. A client implementation that retrieves the Traffic Key from the License without ECM processing is referred to as an SKL Processing implementation.

The packaging and protection format of a Single Key Layer BBTS is interoperable with both ECM Processing implementations and SKL Processing implementations.

3.1.1 Marlin and Downloaded/Streaming Content

For Marlin, content downloaded and streamed to the client would require the client to acquire and evaluates the corresponding License before access can be granted. The content key delivered in the License can be used to decrypt the ECM in order to obtain the (same) traffic keys or the content key itself may serve as the traffic key for decrypting the content.

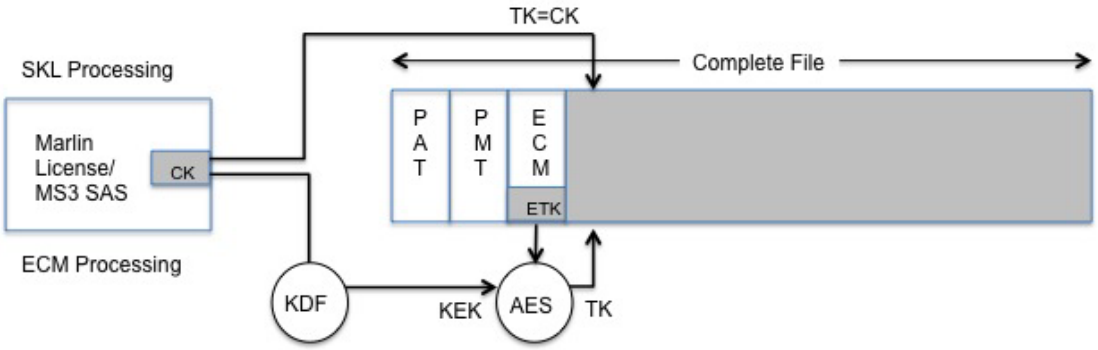


Figure 6: Content Access for ECM and Single Key Layer Processing implementations

The figure shows a complete content file in MPEG2-TS that is protected according to [IEC62455]. In this example, an ECM is present at the beginning of the content file, just after the PAT/PMT.

A client that only supports Single Key Layer BBTS could first acquire the License for accessing the content, calculate the IV and setup the MPEG2-TS decrypting hardware, It may then download or stream the content file for rendering.

Content IDs in BBTS are carried in ECMs. Clients that implement Single Key Layer BBTS and cannot parse ECMs will need to rely on other methods to associate the key(s) delivered in a License with the key(s) used to encrypt the content.

A client that supports BBTS, as defined in §2.2, may filter the ECM and use the information contained therein to associate key(s) in a License to derive a key and then use the derived key to decrypt the traffic key in the ECM.

3.1.2 Content Access

The content access sequence for the Single Key Layer BBTS is similar to the standard procedure as described in §2.1.3.1 except that it is now optional for the clients that implement the Single Key Layer BBTS to filter the ECM when the key materials can be obtained directly from the License. Also, the Acquisition of the License may be performed prior to downloading and accessing the MPEG2-TS (e.g. using MS3 compoundURI)

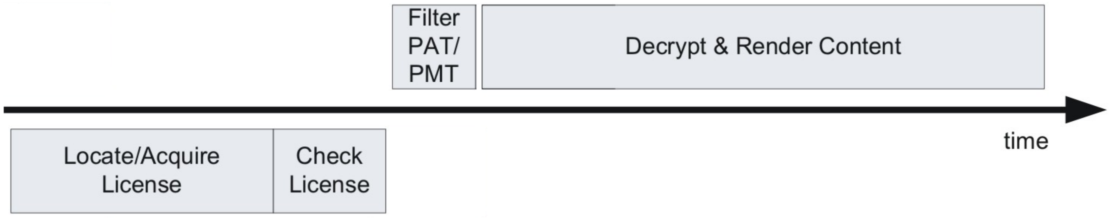


Figure 7: Alternate Content Access Procedure

3.2 BBTS Compendium

The intention of this compendium is to aid the implementation of BBTS for content download, so that the ECM processing overhead can be removed. Unless stated otherwise in this section, the normative descriptions defined in §2 apply.

3.2.1 Stream Format per §2.2.1

The stream MUST have one Entitlement Control Message (ECM) located at the beginning in between TS packets that contain the PAT/PMT and TS packets that contain the media. The ECM MAY be repeated in the stream.

3.2.2 Content Encryption per §2.2.2

In a Single Key Layer BBTS, the Content Key from a License SHALL be used as the (even and odd) traffic key(s) that are used to encrypt the traffic.

The Content Key in a License SHALL be used to as the input to the key derivation function defined in §2.2.4.1.1 to compute a key encryption key which is then used to encrypt the traffic key in the ECM(s).

The traffic key for encrypting the TS media packets in a BBTS SHALL be the same, changing traffic keys MUST NOT be used.

There MUST be only one program in the BBTS, all elementary streams MUST use the same traffic key as the odd and even key.

A Single Key Layer BBTS SHALL use AES as the encryption algorithm with an Initialization Vector (IV) of zero for the encryption of TS media packets.

3.2.3 BBTS CA_descriptor per §2.2.3

To signal a Single Key Layer BBTS in the stream, a “Single_key_layer_descriptor” SHALL be embedded into the BBTS CA_descriptor. The BBTS CA_descriptor including the “Single_key_layer_descriptor” SHALL be included in the PMT as program information.

Descriptor_tag	Identification	Mandatory
0x00	Reserved	n/a
0x01	Service Base CID descriptor	No
0x02	Single_key_layer_descriptor	Yes
0x03	Key Derivation Function descriptor	Yes
0x04 – 0xff	Reserved	n/a

Table 14 - Descriptors in BBTS CA_descriptor private_data_byte, per Table 3

3.2.3.1.1 Single_key_layer_descriptor

The definition of the Single_key_layer_descriptor is specified below.

Syntax	No. of bits	Mnemonic	Value
Single_key_layer_descriptor() {			
descriptor_tag	8	uimsbf	0x02
descriptor_length	8	uimsbf	0x00
}			

Table 15 - Single_key_layer_descriptor

descriptor_tag, MUST have the value of 0x02 to signal the Single_key_layer_descriptor
descriptor_length, MUST be 0x00 as no private data is defined for this descriptor

3.2.4 Entitlement Control per §2.2.4

It is OPTIONAL for the client to process the ECM and EMM in a single-key-layer BBTS.

The content ID SHALL be constructed using the syntax as defined for a program in §2.2.4.2.

The key layer to enable service access SHALL NOT be used. (i.e. the [IEC62455] service_flag = 0.)

The ECM SHALL NOT carry the copy_control_information access criteria descriptor defined in [IEC62455].

3.2.5 License for ECM per §2.2.6

It is not necessary to distinguish between a real-time stream from a non real-time stream in a Single Key Layer BBTS. Therefore, timestamp defined in §2.2.6.1 and obligation for ECM as specified in §2.2.6.2 SHALL NOT be used the License.

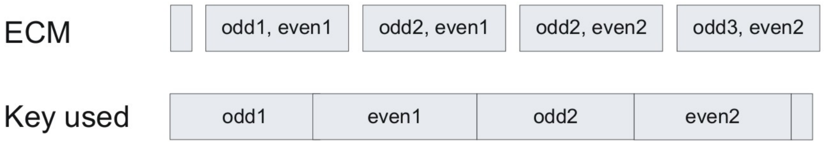
3.2.6 Mime-type

The following MIME-type MAY be used to signal that the content is a Single Key Layer BBTS:
application/vnd.marlin.drm.bbts-single-key-layer

For example in HTML5 context:
if (canPlayType('application/vnd.marlin.drm.StreamAccessStatement;
container="application/vnd.marlin.drm.bbts-single-key-layer";
codecs="avc1.42E01E, mp4a.40.2"') == "probably")

Appendix A Odd/even key management (§2)

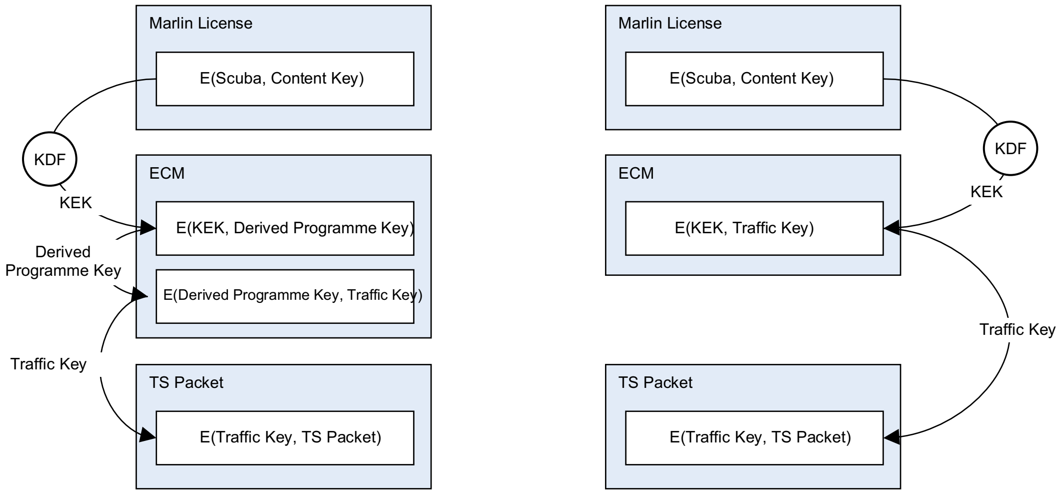
The picture below indicates both the ECM stream and the key used to encrypt the content.



The ECM contains two keys, an odd and an even key. While using the odd key, the even key is updated and vice versa. This prevents that the insertion of the ECMs (and processing of the ECM in the client) has to correspond exactly with a change in the key used to encrypt the content.
When the odd_even_flag in the ECM is KSM_FLAG_ODD, it signals that the encrypted_traffic_key_material in the ECM contains the odd key. As such, to prepare the descrambler, the encrypted_traffic_key_material and the intial_vector is used to initialize the odd descrambler register. In addtion, to prime the descrambler for the next key change, the next_encrypted_traffic_key_material and the next_initial_vector is used to intialize the even descrambler register. When the odd_even_flag is KSM_FLAG_EVEN the converse of the above description applies.
Alternate key management schemes SHALL NOT be used.

Appendix B IEC 62455 Key Management (§2)

The figure below indicates the key management as defined in [IEC62455]. The left side of the picture indicates the situation where part of the content has both a service and a program-based license. The right site indicates the situation where either a service or program license is used.



A service-based license is a license that gives access to all programs in a channel. A program-based license provides access to one of the programs in a channel.

The right side indicates the situation where either a service or program license is used. The Content Key that is stored in the License is decrypted using the information from SCUBA [MRL CORE]. The KEK (Key Encryption Key) derived from the Content Key will be used to decrypt of the Traffic Key stored in the ECM. The Traffic Key provides access to the content in this crypto period.

When both programme and service keys are used, an extra layer is introduced in the key hierarchy. The KEK (Key Encryption Key) derived from the Content Key (which is the IEC 62455 service key) will be used to decrypt the derived programme key, which is subsequently used to decrypt the Traffic Key. This prevents a client having access to the Traffic Key from accessing the service key.