

1
2
3
4
5
6
7
8
9
10
11
12

Marlin IPTV End-point Service Specification

13
14
15 Version 2.0.0
16 Final

17
18
19
20
21
22
23
24
25
26
27
28
29

Source	Marlin Developer Community
Date	January 16, 2026

30

31 **Notice**
32

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

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

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

56 Copyright © 2003 - 2026 by MDC, 415-112 North Mary Avenue #383 Sunnyvale, CA
57 94085, USA. All rights reserved. Third-party brands and names are the property
58 of their respective owners.

59 **Intellectual Property**

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

62 **Contact Information**

63 Feedback on this specification should be addressed to: [editor@marlin-
community.com](mailto:editor@marlin-
64 community.com)

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

67

68	Contents	
69		
70	1 Introduction	4
71	1.1 Document Organization	4
72	1.2 Terminology and Conventions	4
73	1.3 Abbreviations	4
74	1.4 Terms and Definitions	5
75	1.5 References	6
76	1.5.1 Normative References	6
77	1.6 Bit/Byte ordering	7
78	2 Marlin IPTV-ES System entities (Informative)	8
79	2.1 Marlin IPTV-ES Device	8
80	2.2 Marlin IPTV-ES Server	8
81	3 Architecture of Marlin IPTV End-point Service	9
82	3.1 Architecture (Informative)	9
83	3.2 Marlin IPTV-ES Device	9
84	3.2.1 Functions	9
85	3.2.2 Credentials and Device Information	9
86	3.3 Marlin IPTV-ES Server	10
87	3.3.1 Functions	10
88	3.3.2 Credentials	11
89	4 Marlin IPTV-ES SAC and Marlin IPTV-ES Service Protocols	12
90	4.1 Secure Authenticated Channel (SAC) Protocol	12
91	4.1.1 Protocol overview	12
92	4.1.2 Crypto Algorithm	15
93	4.1.3 Protocol	16
94	4.1.4 Processing Rules	25
95	4.2 Marlin IPTV-ES Service Protocols over SAC	38
96	4.2.1 Get Permission Protocol	38
97	4.2.2 Get Trusted Time Protocol	46
98	4.2.3 Packed Message Protocol	47
99	4.2.4 Processing Rules	49
100	5 Marlin IPTV-ES Trust Management	56
101	5.1 Certificates	56
102	5.1.1 Certificate Contents	56
103	5.1.2 Certificate Extensions	57
104	5.2 Certificate Revocation List	58
105	5.2.1 CRL Contents	58
106	5.3 DRL	59
107	5.3.1 Node and Device IDs	59
108	5.3.2 DRL Fields	60
109	5.3.3 DRL Format	60
110	6 File Format for Marlin IPTV-ES Content	63
111	6.1 Standalone Format	63
112	6.1.1 Stream encryption	63
113	6.1.2 ECM format	63
114	6.1.3 Processing Rules of ECM	64
115	6.2 Interoperable Format	65
116	Appendix A Profiles (Normative)	67
117	A.1 SAC Protocol	67
118	A.2 Service Protocol	69
119	A.3 File Format	72
120		

1 Introduction

1.1 Document Organization

This document covers the Marlin IPTV End-point Service Specification. It is organized as follows:

- (This) introduction, including abbreviations, definitions and references.
- Marlin IPTV End-point Service System entities.
- Architecture of Marlin IPTV End-point Service Specification.
- Communication Protocol.
- Trust Management.
- File Format.

1.2 Terminology and 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 [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 Abbreviations

ACK	ACKnowledgement
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CPRM	Content Protection for Recordable Media
CRL	Certificate Revocation List
DH	Diffie-Hellman
DRL	Device Revocation List
EC-DH	Elliptic Curve Diffie-Hellman
EC-DSA	Elliptic Curve Digital Signature Algorithm
ECM	Entitlement Control Message
ECSP-DSA	Elliptic Curve Signature Primitive, Digital Signature Algorithm version
ECSVDP-DH	Elliptic Curve Secret Value Derivation Primitive, Diffie-Hellman version
ECVP-DSA	Elliptic Curve Verification Primitive, Digital Signature Algorithm version
IV	Initialization Vector
MG-R	MagicGate Type-R
MPEG-2 TS	MPEG 2 Transport Stream
OFB	Output FeedBack
PI	Protection Information

PMT	Program Map Table
RFC	Request For Comments
SAC	Secure Authenticated Channel
TS	Transport Stream
TTS	Timed Transport Stream
VCPS	Video Content Protection System

149

150

1.4 Terms and Definitions

Authentication	The process of validating the identity of an individual, device, entity or system.
Channel	A TS/TTS that consists of one or more contents.
Channel Tier Bits	A bit string that is transferred within an ECM and specifies the subscription to which the Channel carrying the ECM belongs.
Content Key	The symmetric key that encrypts the content.
Direct Key Delivery	A key delivery scheme to deliver a key directly necessary for content consumption with a validity period to a Marlin IPTV-ES Device via SAC. The key delivered by this scheme is called a Content Key.
ECM	Information used on descrambling contents as a sub-permission of consuming the contents.
EXPORT	Output to other protection systems.
EXTRACT	Output contents for rendering.
Indirect Key Delivery	A key delivery scheme to deliver a key indirectly necessary for content consumption to a Marlin IPTV-ES Device via SAC. The key delivered by this scheme is called a Work Key.
IPTV	TV system with broadband connection.
IPTV-ES Network	A network between the Marlin IPTV-ES Server and the Marlin IPTV-ES Device.
Marlin IPTV-ES Device	Client device of a Marlin IPTV-ES Server.
Marlin IPTV-ES Server	Server on the Marlin IPTV-ES Network.
Persistent Storage	Storage areas of the Marlin IPTV-ES Device or the Marlin IPTV-ES Server that can retain the stored data in absence of power.
Protected Storage	A local storage that incorporates a mechanism by which to secure the data it persists. Protected Storage may be physically or logically bound to the Marlin IPTV-ES Device.
RECORD	Output to Protected Storage.
Scramble Key	The symmetric key that scrambles the content.
Service Provider ID	An identifier of a service provider.
Simple Key Delivery	A key delivery scheme to deliver a key directly necessary for content consumption without a validity period to a Marlin IPTV-ES Device via SAC. The key delivered by this scheme is called a Content Key but is only allowed to be cached during the rendering of the content.
Subscription Tier Bits	A bit string that is transferred with a Work Key and specifies the subscription of a Marlin IPTV-ES Device.

Tier Bits	A bit string used to control the ability of consuming Channels. Subscription to one or more Channels of a single service provider is to be assigned to each bit.
TransactionFlag Management	A set of procedures performed by both Marlin IPTV-ES Device and Marlin IPTV-ES Server in conjunction with storing the TransactionFlag on their Persistent Storage and sending the stored TransactionFlag from the Marlin IPTV-ES Device in order to make the Marlin IPTV-ES Server possible to check the reception status of a communication message on the Marlin IPTV-ES Device after a communication cut-off.
Trusted Time	A secure and internal time source of a Marlin IPTV-ES Server or a Marlin IPTV-ES Device defined in the Marlin compliance rules.
Work Key	The symmetric key that encrypts ECMs.
Work Key ID	An identifier of a Work Key.
Work Key Management ID	An identifier of a unit for managing Work Keys of a single service provider.
Work Key Version	A value that specifies the version of Work Key for a single Work Key Management ID.

152

153

1.5 References

154

1.5.1 Normative References

[AES]	NIST FIPS 197: Advanced Encryption Standard (AES). November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
[AES-MODES]	Recommendation of Block Cipher Modes of Operation. NIST. NIST Special Publication 800-38A. http://csrc.nist.gov/CryptoToolkit/modes/800-38_Series_Publications/SP800-38A.pdf
[DTCP]	Digital Transmission Content Protection Specification Revision 1.4 Volume 1
[FIPS PUB 186-5]	Digital Signature Standard (DSS)
[IEEE1363-2000]	IEEE Standard Specifications for Public-Key Cryptography
[IEEE1363a-2004]	IEEE Standard Specifications for Public-Key Cryptography — Amendment 1: Additional Techniques
[RFC2119]	S. Bradner, <i>Key words for use in RFCs to Indicate Requirement Levels</i> , IETF RFC 2119, March 1997 http://www.ietf.org/rfc/rfc2119.txt
[MEXP]	Marlin - Export Parameter Specification
[MFF]	Marlin – File Formats Specification, Version 1.0
[MP2S]	ISO/IEC 13818-1 “Information technology – Generic coding of moving pictures and associated audio information: Systems” Second edition 2000-12-01
[ISOMFF]	“Information technology – Coding of audio-visual objects – Part 12: ISO base media file format”, second edition, ISO/IEC 14496-12:2005(E), 2005-04-01
[PKIX]	R. Housley, W. Ford, W. Polk, D. Solo. <i>Internet X.509 Public Key Infrastructure Certificate and CRL Profile</i> . IETF RFC 3280. April 2002 http://www.ietf.org/rfc/rfc3280.txt

[SCTE52]	ANSI/STCE 52: "Data Encryption Standard – Cipher Block Chaining Packet Encryption Specification"
[Starfish]	Starfish – Marlin Broadcast Encryption Scheme, Version 1.1
[X509]	<i>ITU-T Recommendation X.509 (1997 E): Information Technology - Open Systems Interconnection - The Directory: Authentication Framework, June 1997.</i>
[X509Cor1]	<i>ITU-T Recommendation X.509 (2000) Corrigendum 1: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks Technical Corrigendum 1</i>

155

156

1.6 Bit/Byte ordering

157

All data in this specification are presented with the most significant bit (or byte) on the left hand side and the least significant bit (or byte) on the right hand side.

158

159

Also, all data in this specification are encoded using the big-endian byte order (also known as network byte order) and all bit vectors are multiples of 8 bit bytes in big-endian byte order.

160

161

162

163 **2 Marlin IPTV-ES System entities (Informative)**

164 **2.1 Marlin IPTV-ES Device**

165 Marlin IPTV-ES Devices are devices such as TV sets, Set top Boxes, etc. that are
166 continuously connected to services located in the IPTV-ES Network. These devices
167 are IPTV service clients and can have caches of contents that can be played
168 autonomously, implementing a number of subscription and rental business models.
169 Marlin IPTV-ES Devices are able to play streamed and downloaded contents.
170 Moreover, the devices are able to export streamed and downloaded contents to a
171 certain media, and to record streamed contents to a local storage. To play content, a
172 Marlin IPTV-ES Device acquires a Content Key or a Work Key, respectively with a
173 simple validity expression, from the Marlin IPTV-ES Servers located in the IPTV-ES
174 Network, and renders by decrypting the contents with the Content Key or a Scramble
175 Key, which is obtained by decrypting an ECM with the Work Key. The Marlin IPTV-
176 ES Devices can export contents to a certain media system (e.g. DTCP, CPRM, MG-
177 R, VCPS, etc) by acquiring a Content Key or a Work Key, respectively with export
178 information from the Marlin IPTV-ES Servers located in the IPTV-ES Network and
179 transforms the contents for the Media System. When the technology defined in this
180 specification is used for key delivery for conditional access services, the simple
181 validity expression corresponding to a Work Key indicates the period in which a
182 Marlin IPTV-ES Device can access to a streamed content. In such case, the Marlin
183 IPTV-ES Device is able to play or export the streamed content even after the
184 expiration of the Work Key by recording the content to the local storage.

185
186 Note that a Marlin IPTV-ES Device is not capable of sharing the received content
187 with other devices in the network.
188

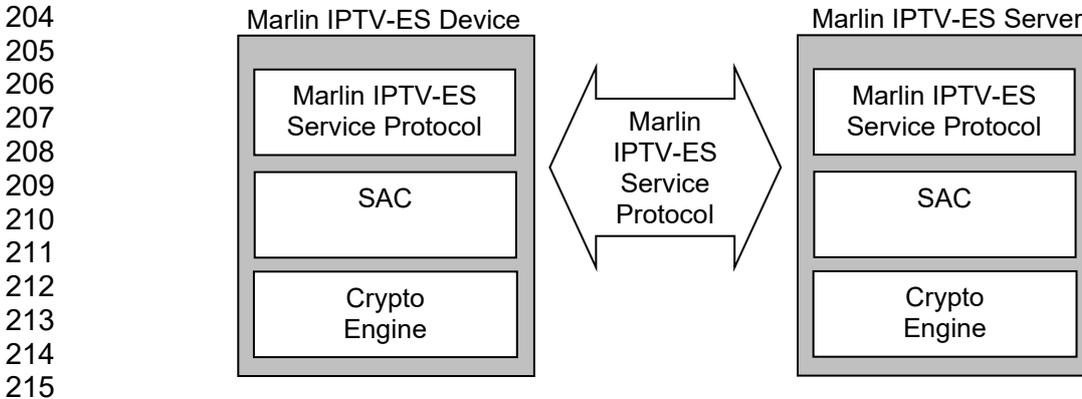
189 **2.2 Marlin IPTV-ES Server**

190 The Marlin IPTV-ES Server is located in the IPTV-ES Network and operated by
191 service providers. When Marlin IPTV-ES Server receives a request of a certain action
192 (play, export, or record) for content from the Marlin IPTV-ES Device, it checks the
193 availability of the action for the content. When the request is granted, the Content
194 Key, or the Work Key and its related Work Key ID, and other related information such
195 as a validity expression for the key, export information, or record information are sent
196 to the Marlin IPTV-ES Device.
197

198 3 Architecture of Marlin IPTV End-point Service

199 3.1 Architecture (Informative)

200 The Figure 1 shows a high level architecture of a Marlin IPTV-ES Device and a
201 Marlin IPTV-ES Server. The entities which are included in the Marlin IPTV End-point
202 Service are depicted in the figure.



215
216
217 *Figure 1: Architecture of Marlin IPTV-ES Device and Marlin IPTV-ES Server*

217 3.2 Marlin IPTV-ES Device

218 3.2.1 Functions

219 The Marlin IPTV-ES Device at least SHALL implement the following functionality.

- 220
- 221 • Marlin IPTV-ES Service Protocol: Generates and analyzes messages for
- 222 Marlin IPTV-ES Service Protocols defined in section 4.2.
- 223 • SAC: Communicates messages for Authentication and Encryption defined in
- 224 section 4.1.
- 225 • Crypto Engine: Manages Crypto operation for SAC.
- 226

227 The detail of which function that the Marlin IPTV-ES Device SHALL implement
228 depends on its supporting profile defined in Appendix A.

230 3.2.2 Credentials and Device Information

231 The Marlin IPTV-ES Device SHALL have the following information which is used in
232 Marlin IPTV-ES Service Protocols over SAC:

- 233
- 234 • *Credentials*: X.509 Key Pair used for Authentication and SAC establishment
- 235 with a Marlin IPTV-ES Server
- 236 • *DeviceInformation*: DeviceInformation indicates the characteristic of the
- 237 Marlin IPTV-ES Device. In Marlin IPTV-ES Service Protocols defined in
- 238 section 4.2, DeviceInformation is encoded as in Table 3-1.
- 239 ➤ *SpecificationVersion*: SpecificationVersion represents the major and
- 240 minor versions of the Marlin IPTV-ES specifications the client supports.
- 241 ➤ *Capabilities*: Capabilities indicates a certain functionality the client
- 242 supports. The following capabilities are defined in this specification.

- 243 ✧ bit0: This bit indicates that the client implements and has the secure
- 244 clock function using the Trusted Time.
- 245 ✧ bit1-bit7: Reserved.
- 246 ➤ *Manufacturer*: Manufacturer indicates a unique identity of each of
- 247 manufacturers. The manufacturer-specific value obtained from MTMO
- 248 SHALL be set.
- 249 ➤ *ManufacturerModel*: ManufacturerModel indicates an identity of a model
- 250 in the specified Manufacturer.
- 251 ➤ *ManufacturerModelVersion*: ManufacturerModelVersion represents the
- 252 major and minor versions of the specified ManufacturerModel.
- 253

Byte index ¹	Description
0	Marlin IPTV-ES SpecificationVersionMajor. For the client that implements this specification, "01h" SHALL be set for ECC 224-bit keys. "02h" SHALL be set for ECC 384-bit keys.
1	Marlin IPTV-ES SpecificationVersionMinor. For the client that implements this specification, the value for minor is set to "00h".
2	Capabilities. "00h" SHALL be set for this specification.
3-4	Manufacturer.
5-6	ManufacturerModel. "0000h" SHALL be set for this specification.
7	ManufacturerModelVersionMajor. "00h" SHALL be set for this specification.
8	ManufacturerModelVersionMinor. "00h" SHALL be set for this specification.
9-11	Reserved. "000000h" SHALL be set for this specification.

Table 3-1: Device Information Encoding

254

255 **3.3 Marlin IPTV-ES Server**

256 **3.3.1 Functions**

257 The Marlin IPTV-ES Server at least SHALL implement the following functionality.

258

- 259 • Marlin IPTV-ES Service Protocol: Generates and analyzes messages for
- 260 Marlin IPTV-ES Service Protocols defined in section 4.2.
- 261 • SAC: Communicates messages for Authentication and Encryption defined in
- 262 section 4.1.
- 263 • Crypto Engine: Manages Crypto operation for SAC.

264

265 The detail of which function that the Marlin IPTV-ES Server SHALL implement

266 depends on its supporting profile defined in Appendix A.

267

¹ The Byte index value is a relative value from the data format defined here. Hereinafter, the Byte index value is always a relative value if not specified.

268 **3.3.2 Credentials**

269 The Marlin IPTV-ES Server SHALL have the following information which is used in
270 Marlin IPTV-ES Service Protocols over SAC.

271

- 272 • *Credentials*: X.509 Key Pair used for Authentication and SAC establishment
273 with Marlin IPTV-ES Device.

274

275 **4 Marlin IPTV-ES SAC and Marlin IPTV-ES Service**
 276 **Protocols**

277 This section defines communication protocols between a Marlin IPTV-ES Device and
 278 Marlin IPTV-ES Server.
 279

280 **4.1 Secure Authenticated Channel (SAC) Protocol**

281 **4.1.1 Protocol overview**

282 The outline of the message exchange sequence is shown in Figure 2. Marlin IPTV-
 283 ES Devices and Marlin IPTV-ES Servers SHALL use the Authentication and Key
 284 Exchange (AKE) protocol defined in [DTCP].
 285

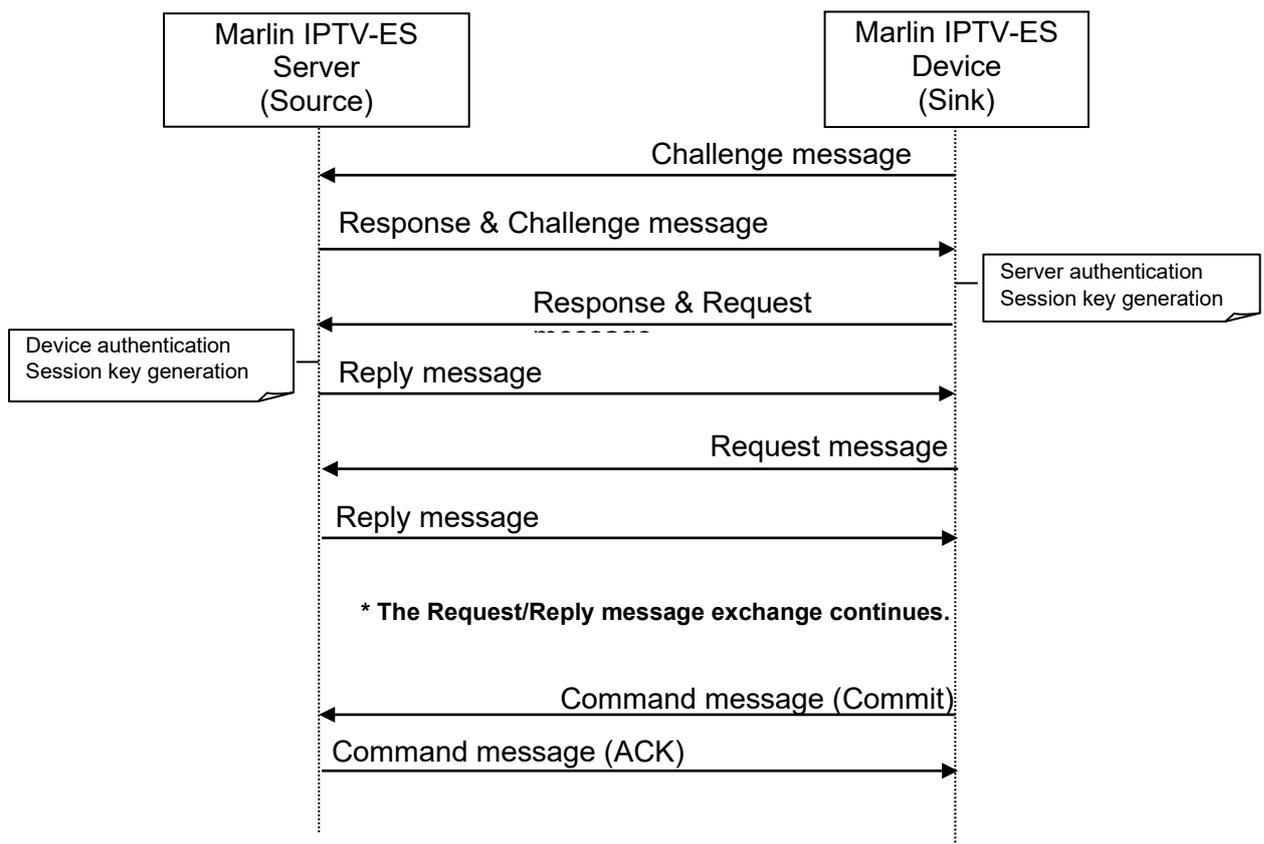


Figure 2: Outline of secure authenticated channel protocol

305
 306 The Marlin IPTV-ES Device sends a Challenge message which includes
 307 Authentication information (random number and certificate) of the Marlin IPTV-ES
 308 Device to the Marlin IPTV-ES Server. Then the Marlin IPTV-ES Server sends, in
 309 response to the Challenge message, a Response & Challenge message which
 310 includes information for Authentication and key sharing of the Marlin IPTV-ES
 311 Server. Receiving this message, the Marlin IPTV-ES Device authenticates the Marlin IPTV-
 312 ES Server and generates a session key.
 313

314 Message communications between the Marlin IPTV-ES Device and the Marlin IPTV-
 315 ES Server are performed by exchanging Request messages and Reply messages.

316 A Response & Request message is a Request message to which response data for
317 Authentication and key sharing are affixed. The Marlin IPTV-ES Device sends this
318 type of messages to the Marlin IPTV-ES Server in response to the received
319 Response & Challenge message. Receiving the Response & Request message, the
320 Marlin IPTV-ES Server authenticates the Marlin IPTV-ES Device and generates a
321 session key.

322
323 ACK, ERROR, and Commit for the secure authenticated communications are
324 performed with a Command message.

325
326 The Command message, after session key generation, is sent after being encrypted
327 by a session key as an Encrypted command message, while a reply to a Challenge
328 message and a Response & Request message from the Marlin IPTV-ES Server to
329 the Marlin IPTV-ES Device is respectively sent as a Plain command message.

330
331 In addition to these messages, although not shown in Figure 2, the Marlin IPTV-ES
332 Device can send a Response & Commit message to the Marlin IPTV-ES Server at
333 the same timing as a Response & Request message to commit a SAC session.

334
335 Communication messages between a single pair of Marlin IPTV-ES Server and
336 Marlin IPTV-ES Device SHALL be exchanged sequentially, not concurrently. Also a
337 single pair of Marlin IPTV-ES Device and Marlin IPTV-ES Server SHALL use one
338 Marlin IPTV-ES SAC session at the same time.

339
340 The Marlin IPTV-ES Device and the Marlin IPTV-ES Server SHALL terminate SAC
341 connection after a string of continuous communications in accordance with the
342 criteria described in section 4.1.4.10.

343

344 **4.1.1.1 Transaction Flag Management**

345 In the secure authenticated communications between a Marlin IPTV-ES Server and a
346 Marlin IPTV-ES Device, both the Marlin IPTV-ES Device and Marlin IPTV-ES Server
347 respectively manage a transaction flag so that the Marlin IPTV-ES Server can check
348 the reception status on the Marlin IPTV-ES Device of a communication message
349 which it sends as a response to the request sent by the Marlin IPTV-ES Device.

350

351 Timings of the status transition are shown in Figure 3.

352

353

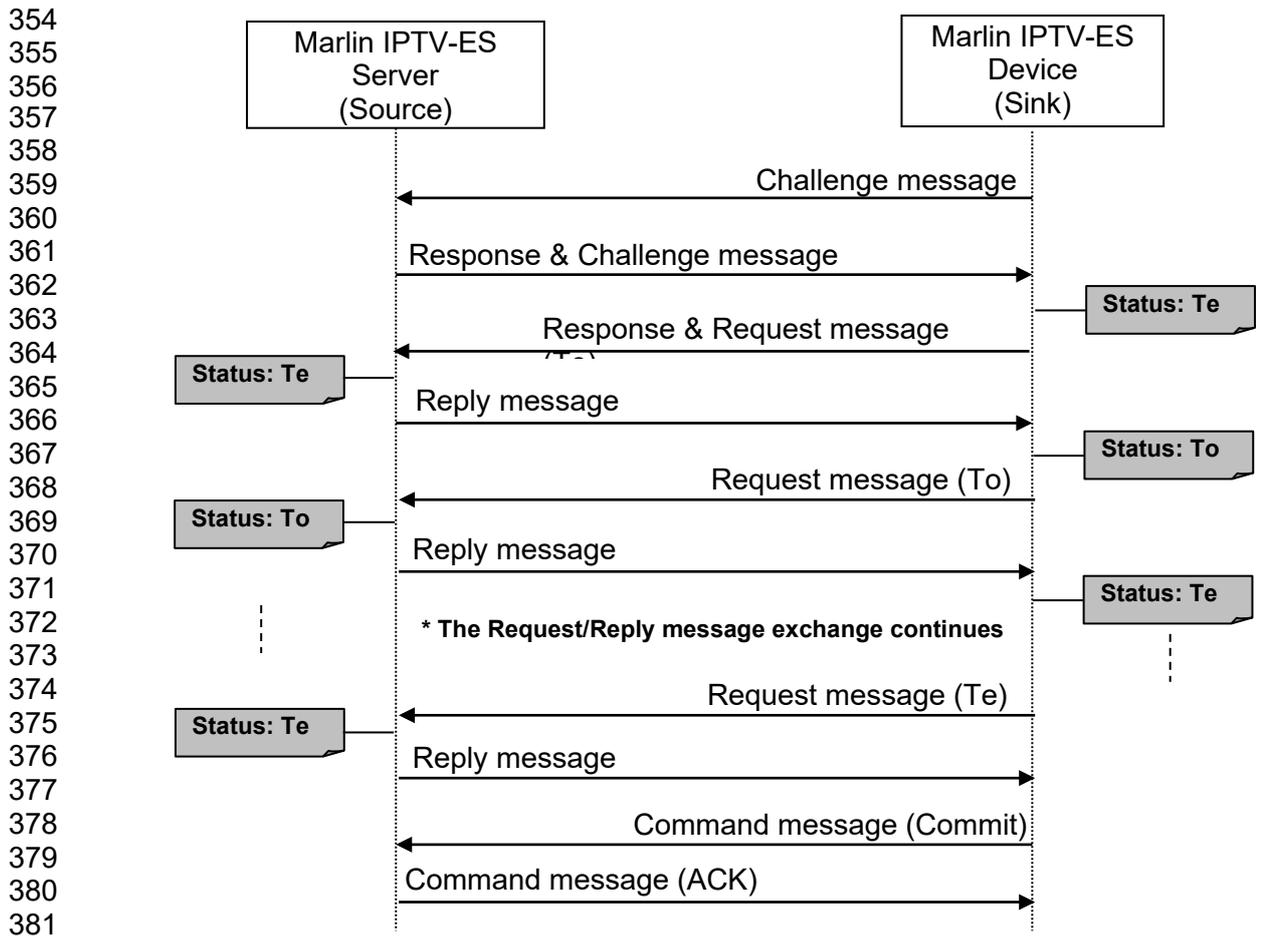


Figure 3: Status transition of secure authenticated communications

382
383 The Marlin IPTV-ES Server and the Marlin IPTV-ES Device manage the transaction
384 flag after generating session keys, and it is inverted by the Marlin IPTV-ES Device.
385 The transaction flag indicates even transaction “Te” and odd transaction “To”. The
386 Marlin IPTV-ES Device alternately inverts in the order of “Te” and “To” of the
387 transaction flag every time it receives a Reply message from the Marlin IPTV-ES
388 Server. The Marlin IPTV-ES Server acknowledges that the Marlin IPTV-ES Device
389 has received a communication message with a change in status of the transaction
390 flag which it receives.

391
392 In the event of communication error or accidental power down, at the subsequent
393 connection the Marlin IPTV-ES Device notifies the reception status of the
394 communication message at a communication cut-off by sending the stored
395 transaction flag.

396
397 For example, in cases where count constraint is applied to the usage of a
398 ContentKey, the IPTV-ES Server needs to know the precise number of times of
399 which an IPTV-ES Device acquired the ContentKey necessary for that action. In such
400 cases, the IPTV-ES Server can know whether or not an IPTV-ES Device acquired a
401 ContentKey by performing as described below.

- 402
403 • If a communication cut-off occurs and an IPTV-ES Device fails to receive a
404 Reply message which carries a ContentKey, the IPTV-ES Device sends its
405 storing TransactionFlag to the IPTV-ES Server after re-establishing SAC
406 connection.

407 • After re-establishing SAC connection, the IPTV-ES Server acknowledges that
408 an IPTV-ES Device failed to receive the lastly sent Reply message, i.e. failed
409 to acquire the ContentKey, by comparing the TransactionFlag sent by the
410 IPTV-ES Device and the TransactionFlag that the IPTV-ES Sever stores
411 because, in such case, those TransactionFlags have different values.
412

413 Therefore, in order to prepare for events such as communication error or accidental
414 power down, the Marlin IPTV-ES Device, which has the capability of handling the Get
415 Permission Request message that requires TransactionFlag Management,
416 processes the following for TransactionFlag Management.

- 417
- 418 • Storage of TransactionFlag and ContentKey.
- 419 • Deletion of stored TransactionFlag and status change of stored ContentKey.
- 420 • Sending of stored TransactionFlag as a Response & Commit message.
- 421

422 Accordingly, in order to prepare for events such as communication error or accidental
423 power down, the Marlin IPTV-ES Server, which has the capability of handling the Get
424 Permission Request message that requires TransactionFlag Management,
425 processes the following for TransactionFlag Management.

- 426
- 427 • Storage of TransactionFlag.
- 428 • Deletion of stored TransactionFlag.
- 429 • Judgement of Reply message reception.
- 430

431 Conditions and details of each process for the Marlin IPTV-ES Device and the Marlin
432 IPTV-ES Server are specified in section 4.1.4.11, while conditions for Get Permission
433 Request messages of whether the TransactionFlag Management is required or not is
434 specified in section 4.2.1.1.
435

436 **4.1.1.2 URI signature verification**

437 Before establishment of connection, the Marlin IPTV-ES Device SHALL verify the
438 signature of the URI that indicates the network location of the Marlin IPTV-ES Server
439 to whom the Marlin IPTV-ES Device is attempting to connect in accordance with
440 section 4.1.4.12.
441

442 **4.1.2 Crypto Algorithm**

443 **4.1.2.1 Crypto Algorithm for ECC 224-bit keys**

444 The following algorithms SHALL be used for authentication, key exchange, and
445 message encryption.

- 446 • Authentication: EC-DSA (224 bits) with SHA-256.
- 447 • Key exchange: EC-DH (224 bits).
- 448 • Message encryption: AES (128 bits).
- 449

450 The EC-DSA signature generation algorithm for the SAC SHALL use ECSP-DSA and
451 EMSA1 to which SHA-256 are applied, defined in the reference [IEEE1363-2000].
452

453 The EC-DSA signature verification algorithm for the SAC SHALL use ECVP-DSA and
454 EMSA1 to which SHA-256 are applied, defined in the reference [IEEE1363-2000].
455

456 The EC-DH key sharing algorithm SHALL use the ECSVDP-DH primitive defined in
457 the reference [IEEE1363-2000]. The session key used SHALL be made up of low

458 order 128 bits on the x coordinate of the shared secret value generated by the EC-
459 DH.

460
461 The messages encryption SHALL use AES [AES] in CBC [AES-MODES] mode. The
462 IV for the CBC mode SHALL be a value with all bits equal to zero. If a fraction is
463 produced, OFB mode SHALL be used as described in [SCTE52]

464 **4.1.2.2 Crypto Algorithm for ECC 384-bit keys**

465 The following algorithms SHALL be used for authentication, key exchange, and
466 message encryption.

- 467 • Authentication: EC-DSA (NIST P-384(secp384r1)) with SHA-384.
- 468 • Key exchange: EC-DH (NIST P-384(secp384r1)).
- 469 • Message encryption: AES (256bit)

470
471 The EC-DSA signature generation algorithm for the SAC SHALL use ECSP-DSA and
472 EMSA1 to which SHA-384 are applied, defined in the reference [IEEE1363-2000]
473 [IEEE1363a-2004].

474
475 The EC-DSA signature verification algorithm for the SAC SHALL use ECVP-DSA and
476 EMSA1 to which SHA-384 are applied, defined in the reference [IEEE1363-2000].
477 [IEEE1363a-2004]

478
479 The EC-DH key sharing algorithm SHALL use the ECSVDP-DH primitive defined in
480 the reference [IEEE1363-2000] [FIPS PUB 186-5]. The session key used SHALL be
481 made up of low order 256 bits on the x coordinate of the shared secret value
482 generated by the EC-DH.

483
484 The messages encryption SHALL use AES [AES] in CBC [AES-MODES] mode. The
485 IV for the CBC mode SHALL be a value with all bits equal to zero. If a fraction is
486 produced, OFB mode SHALL be used as described in [SCTE52].

487 **4.1.3 Protocol**

488 **4.1.3.1 Message header and payload**

489 Each message SHALL include the following header.

- 490
- 491 • *ProtocolID*: ProtocolID is a constant byte value and identifies the head of a
492 message.
- 493 • *ProtocolVersion*: ProtocolVersion determines types of payload that can be
494 handled, data formats and cryptographic parameter to be used. The version
495 is a two-byte value.
- 496 • *SenderID*: SenderID determines message sender. Device ID defined in
497 [Starfish] §3.2.2 is used in the messages sent by the sink, and NULL value
498 (00h) is used in the messages sent by the source.
- 499 • *PayloadType*: PayloadType identifies a message type.
- 500 • *PayloadSize*: PayloadSize defines the number of bytes of the payload data
501 subsequent to the header.

502 **4.1.3.1.1 Message header and payload for ECC 224-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.

Byte index	Description
4-5	ProtocolVersion (major and minor version number). "0100h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType.
16-19	PayloadSize.

503
504
505

506 **4.1.3.1.2 Message header and payload for ECC 384-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion (major and minor version number). "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType.
16-19	PayloadSize.

507

508 **4.1.3.2 Challenge message parameters**

509 Challenge message SHALL include the header defined in section 4.1.3.1 followed by
510 the parameters defined below.

511
512
513
514
515

- *SinkRandomNumber*: Random value generated by the sink.
- *SinkCertificateSize*: Size of certificate of the sink.
- *SinkCertificate*: The certificate of the sink. PKIPath defined in [X509Cor1] is used in the case that SinkCertificate contains certificate chain.

516 **4.1.3.2.1 Response & Challenge message parameters for ECC 224-bit keys**

517

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0100h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0001h" SHALL be set for this specification.
16-19	PayloadSize.
20-35	SinkRandomNumber.
36-37	SinkCertificateSize.
38- (38+SinkCertificate Size-1)	SinkCertificate.

518 **4.1.3.2.2 Response & Challenge message parameters for ECC 384-bit keys**

519

Byte index	Description
------------	-------------

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0001h" SHALL be set for this specification.
16-19	PayloadSize.
20-51	SinkRandomNumber.
52-53	SinkCertificateSize.
54- (54+SinkCertificate Size-1)	SinkCertificate.

520

521 4.1.3.3 Response & Challenge message parameters

522 Response & Challenge message SHALL include the header defined in section
523 4.1.3.1 followed by the parameters defined below.

524

- 525 • *SourceRandomNumber*: Random value generated by the source.
- 526 • *SourceEC-DHPhase1Value*: The phase1 value of DH generated by the
527 source. The value contains the x coordinate followed by the y coordinate.
- 528 • *Signature*: The signature with the source private key corresponding to the
529 SourceCertificate which covers the concatenation of SinkRandomNumber
530 and SourceEC-DHPhase1Value. The value contains c value followed by d
531 value defined in [IEEE1363-2000].
- 532 • *SourceCertificateSize*: Size of certificate of the source.
- 533 • *SourceCertificate*: The certificate of the source. PKIPath defined in
534 [X509Cor1] is used in the case that SourceCertificate contains certificate
535 chain.

536 4.1.3.3.1 Response & Challenge message parameters for ECC 224-bit keys

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	<i>ProtocolVersion</i> . <i>"0100h" SHALL be set for this specification.</i>
6-13	SenderID.
14-15	PayloadType. "0002h" SHALL be set for this specification.
16-19	PayloadSize.
20-35	SourceRandomNumber.
36-91	SourceEC-DHPhase1Value.
92-147	Signature.
148-149	SourceCertificateSize.
150- (150+SourceCertifi cateSize-1)	SourceCertificate.

537 4.1.3.3.2 Response & Challenge message parameters for ECC 384-bit keys

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	<i>ProtocolVersion.</i> "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0002h" SHALL be set for this specification.
16-19	PayloadSize.
20-51	SourceRandomNumber.
52-147	SourceEC-DHPhase1Value.
148-243	Signature.
244-245	SourceCertificateSize.
246- (246+SourceCertifi cateSize-1)	SourceCertificate.

540 4.1.3.4 Response & Request message parameters

541 Response & Request message SHALL include the header defined in section 4.1.3.1
542 followed by the parameters defined below.

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

- *SinkEC-DHPhase1Value*: The phase1 value of DH generated by the sink. The value contains the x coordinate followed by the y coordinate.
- *Signature*: The signature with the sink private key corresponding to the SinkCertificate which covers the concatenation of SourceRandomNumber and SinkEC-DHPhase1Value. The value contains c value followed by d value defined in [IEEE1363-2000].
- *EncryptedDataSize*: Size of encrypted data.
- *SequenceNumber*: 3-byte sequence number. The usage rules of the sequence number are as follows.
 - The initial value of a sequence number SHALL be set to "1" each time a SAC session is started. Therefore, SequenceNumber of a Response & Request message or of a Response & Commit message SHALL be set to "1".
 - A sequence number is incremented by 1 when a message that contains cipher text is sent, and can be incremented until the maximum value of $(2^{24}-2)$.
 - On receiving a message that contains cipher text, the sequence number of the cipher text is checked to match the sequence number of the receiver. If the two values match, the sequence number is incremented by 1.
- *TransactionFlag*: TransactionFlag is associated with the transaction currently being processed. The flag corresponds to even transaction and odd transaction respectively. "00h" for even transaction and "01h" for odd transaction SHALL be set.
 - As specified in Figure 3, TransactionFlag of a Response & Request message SHALL be set to "00h".
- *Request*: The request message from the sink.
- *MessageDigest*: MessageDigest SHALL be calculated as described in section 4.1.2 over the clear-text message parameters (with the exclusion of the MessageDigest) in their respective order.

574

575 SequenceNumber, TransactionFlag, Request, and MessageDigest SHALL be
576 encrypted with the session key as described in section 4.1.2.

577 **4.1.3.4.1 Response & Request message parameters for ECC 224-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0100h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0003h" SHALL be set for this specification.
16-19	PayloadSize.
20-75	SinkEC-DHPhase1Value.
76-131	Signature.
132-135	EncryptedDataSize.
136-138	SequenceNumber.
139	TransactionFlag.
140- (140+RequestSize -1)	Request.
(140+RequestSize)- (171+RequestSize)	MessageDigest.

578

579 **4.1.3.4.2 Response & Request message parameters for ECC 384-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0003h" SHALL be set for this specification.
16-19	PayloadSize.
20-115	SinkEC-DHPhase1Value.
116-211	Signature.
212-215	EncryptedDataSize.
216-218	SequenceNumber.
219	TransactionFlag.
220- (220+RequestSize -1)	Request.
(220+RequestSize)- (267+RequestSize)	MessageDigest.

580

581

582 **4.1.3.5 Request message parameters**

583 Request message SHALL include the header defined in section 4.1.3.1 followed by
584 the parameters defined in section 4.1.3.4.

585 SequenceNumber, TransactionFlag, Request, and MessageDigest SHALL be
586 encrypted with the session key as described in section 4.1.2.

587 **4.1.3.5.1 Request message parameters for ECC 224-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0100h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0004h" SHALL be set for this specification.
16-19	PayloadSize.
20-23	EncryptedDataSize.
24-26	SequenceNumber.
27	TransactionFlag.
28- (28+RequestSize- 1)	Request.
(28+RequestSize)- (59+RequestSize)	MessageDigest.

588 **4.1.3.5.2 Request message parameters for ECC 384-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0004h" SHALL be set for this specification.
16-19	PayloadSize.
20-23	EncryptedDataSize.
24-26	SequenceNumber.
27	TransactionFlag.
28- (28+RequestSize- 1)	Request.
(28+RequestSize)- (75+RequestSize)	MessageDigest.

589

590 **4.1.3.6 Reply message parameters**

591 Reply message SHALL include the header defined in section 4.1.3.1 followed by the
592 parameters defined below. Definitions of all parameters are the same as section
593 4.1.3.4 with the exception of parameters defined below.

594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615

- *TransactionFlagRecordFlag*: TransactionFlagRecordFlag indicates whether or not it is required to record the transaction identification flag in Non-volatile Memory Area on the sink. “00h” for indication of not to record and “01h” for indication of record SHALL be set, provided that “01h” SHALL be set if and only if both of the following two conditions are met, and otherwise “00h” SHALL be set.
 - Either of the following two Service Protocol message is set to Reply, defined below.
 - ✧ Get Permission Reply message which corresponds to a Get Permission Request message that requires TransactionFlag Management.
 - ✧ Packed Message Reply message which corresponds to a Packed Message Request message which packs one or more Get Permission Request messages that requires TransactionFlag Management.
 - A Content Key which requires TransactionFlag Management, e.g. count constraint is applied to the consumption of a Content Key, is set to Get Permission Reply message.
- *Reply*: The reply message from the source.

SequenceNumber, TransactionFlagRecordFlag, Reply, and MessageDigest SHALL be encrypted with the session key as described in section 4.1.2.

616 **4.1.3.6.1 Reply message parameters for ECC 224-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of “IPTV” SHALL be set for this specification.
4-5	ProtocolVersion. “0100h” SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. “0005h” SHALL be set for this specification.
16-19	PayloadSize.
20-23	EncryptedDataSize.
24-26	SequenceNumber.
27	TransactionFlagRecordFlag.
28-(28+ReplySize-1)	Reply.
(28+ReplySize)-(59+ReplySize)	MessageDigest.

617 **4.1.3.6.2 Reply message parameters for ECC 384-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of “IPTV” SHALL be set for this specification.
4-5	ProtocolVersion. “0200h” SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. “0005h” SHALL be set for this specification.
16-19	PayloadSize.
20-23	EncryptedDataSize.

Byte index	Description
24-26	SequenceNumber.
27	TransactionFlagRecordFlag.
28-(28+ReplySize-1)	Reply.
(28+ReplySize)-(75+ReplySize)	MessageDigest.

618

619 4.1.3.7 Plain command message parameters

620 This message SHALL be sent from the source to the sink before session key
621 generation to notify an ERROR command in response to a Challenge message,
622 Response & Request message or Response & Commit message. Plain command
623 message SHALL include the header defined in section 4.1.3.1 followed by the
624 parameters defined below.

625

- 626 • *Command*: ERROR (0002h) SHALL be set.
- 627 • *Status*: Error status defined in Table 4-1 SHALL be indicated.

628

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0100h" SHALL be set for ECC 224-bit keys. "0200h" SHALL be set for ECC 384-bit keys.
6-13	SenderID.
14-15	PayloadType. "0006h" SHALL be set for this specification.
16-19	PayloadSize.
20-21	Command.
22-23	Status.

629

630 Status values are below.

631

Values	Details
8001h	Error other than the below.
8002h	Message error.
8003h	Authentication error.
8004h	Revoked.
8005h	Certificate issuer mismatch.

Table 4-1: Status value of Plain command message

632

633 4.1.3.8 Encrypted command message parameters

634 Encrypted command message SHALL be used to send a command message after
635 session key generation and is encrypted with the session key. Encrypted command
636 message SHALL include the header defined in section 4.1.3.1 followed by the
637 parameters defined below. Definitions of all parameters are the same as sections
638 4.1.3.4 and 4.1.3.7 with the exception of parameters defined below.

639

- 640 • *EncryptedDataSize*: Size of encrypted data.

- 641 • *TransactionFlag*: This TransactionFlag MAY be set to either of “00h” or “01h”,
- 642 regardless of even or odd transaction.
- 643 • *Command*: “ACK” (0001h), “ERROR” (0002h), or “Commit” (0003h) SHALL
- 644 be set.
- 645 • *Status*: Status defined in Table 4-2 SHALL be returned.

646 SequenceNumber, TransactionFlag, Command, Status and MessageDigest SHALL

647 be encrypted with the session key as described in section 4.1.2.

648

649

Byte index	Description
0-3	ProtocolID. ASCII value of “IPTV” SHALL be set for this specification.
4-5	ProtocolVersion. “0100h” SHALL be set for ECC 224-bit keys. “0200h” SHALL be set for ECC 384-bit keys.
6-13	SenderID.
14-15	PayloadType. “0007h” SHALL be set for this specification.
16-19	PayloadSize.
20-23	EncryptedDataSize.
24-26	SequenceNumber.
27	TransactionFlag.
28-29	Command.
30-31	Status.
32- (32+MessageDigestSize-1)	MessageDigest. MessageDigestSize = 32 for ECC 224-bit keys. MessageDigestSize = 48 for ECC 384-bit keys.

650

651 Status values are below.

652

Values	Details
0000h	Success.
8001h	Error other than the below.
8002h	Message error.
8003h	Authentication error.

653 *Table 4-2: Status value of Encrypted command message*

654

654 **4.1.3.9 Response & Commit message parameters**

655 This message is used to send Commit command following SAC establishment.

656 Response & Commit message SHALL include the header defined in section 4.1.3.1

657 followed by the parameters defined below. Definitions of all parameters are the same

658 as section 4.1.3.4 with the exception of parameters defined below.

659

- 660 • *TransactionFlag*: This TransactionFlag MAY be set to either of “00h” or “01h”,
 - 661 regardless of even or odd transaction, in accordance with the “Sending of
 - 662 stored TransactionFlag” process specified in section 4.1.4.11.1.
- 663

664 SequenceNumber, TransactionFlag and MessageDigest SHALL be encrypted with

665 the session key as described in section 4.1.2.

666 **4.1.3.9.1 Response & Commit message parameters for ECC 224-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0100h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0008h" SHALL be set for this specification.
16-19	PayloadSize.
20-75	SinkEC-DHPhase1Value.
76-131	Signature.
132-135	EncryptedDataSize.
136-138	SequenceNumber.
139	TransactionFlag.
140-171	MessageDigest.

667 **4.1.3.9.2 Response & Commit message parameters for ECC 384-bit keys**

Byte index	Description
0-3	ProtocolID. ASCII value of "IPTV" SHALL be set for this specification.
4-5	ProtocolVersion. "0200h" SHALL be set for this specification.
6-13	SenderID.
14-15	PayloadType. "0008h" SHALL be set for this specification.
16-19	PayloadSize.
20-115	SinkEC-DHPhase1Value.
116-211	Signature.
212-215	EncryptedDataSize.
216-218	SequenceNumber.
219	TransactionFlag.
220-267	MessageDigest.

668

669 **4.1.4 Processing Rules**

670 A Marlin IPTV-ES Server and a Marlin IPTV-ES Device SHALL verify SAC messages
671 in accordance with the processing rules defined in the following subsections. If the
672 verification defined in sections 4.1.4.1 through 4.1.4.9 resulted in a "verification
673 failure", the Marlin IPTV-ES Server or the Marlin IPTV-ES Device SHALL terminate
674 the SAC connection as defined in section 4.1.4.10.

675 Note that verification is considered successful unless it is defined otherwise in the
676 processing rules to follow.

677

678 **4.1.4.1 Message header and payload**

679 Whenever receiving a SAC message, the Marlin IPTV-ES Server and the Marlin
680 IPTV-ES Device SHALL verify its header, which is defined in section 4.1.3.1. If one or
681 more parameters in the header of the SAC message received by the Marlin IPTV-ES
682 Device are invalid values as shown in Table 4-3, the verification SHALL be deemed
683 as "verification failure". On the other hand, if one or more parameters in the header of
684 the SAC message received by the Marlin IPTV-ES Server are invalid values as

685 shown in Table 4-3, the verification SHALL be deemed as “verification failure” and
 686 the Status of the Plain command message or the Encrypted command message sent
 687 to the Marlin IPTV-ES Device SHALL be set to “Message error” (8002h).
 688

Parameters	Invalid Values
ProtocolID	Other than the ASCII value of “IPTV” (49505456h).
ProtocolVersion	Other than “0100h” or “0200h”.
PayloadType	Other than values corresponding to the Currently Received Message specified in Table 4-4 for the Marlin IPTV-ES Server.
	Other than values corresponding to the Currently Received Message specified in Table 4-5 for the Marlin IPTV-ES Device.
PayloadSize	For fixed size messages (Plain command message, Encrypted command message and Response & Commit message), other than each of PayloadSizes defined in section 4.1.3.
	For variable size messages, other than ((received message size)-(message header size)).

Table 4-3: Invalid parameter values of message header

689
 690 A valid message to be received depends on the message previously sent by the
 691 Marlin IPTV-ES Server or the Marlin IPTV-ES Device, respectively. Table 4-4 and
 692 Table 4-5 show the combinations of valid message sequences of the Marlin IPTV-ES
 693 Server and the Marlin IPTV-ES Device, respectively.

694
 695 Note that a Response & Commit message is to be sent from a Marlin IPTV-ES
 696 Device to a Marlin IPTV-ES Server if and only if the Marlin IPTV-ES Device has the
 697 capability of handling the Get Permission Request message that requires
 698 TransactionFlag Management. Therefore, in other words, although the Response &
 699 Commit message is shown in Table 4-4 and Table 4-5, it is an invalid message to be
 700 sent and received, respectively by a Marlin IPTV-ES Device and by a Marlin IPTV-ES
 701 Server, when the Marlin IPTV-ES Device and the Marlin IPTV-ES Server have no
 702 capability of handling the Get Permission Request message that requires
 703 TransactionFlag Management.
 704

Combina tion	Previously Sent Message	Currently Received Message
S-1	None (i.e. before starting SAC establishment.)	Challenge message
S-2	Encrypted command message (i.e. after termination of another SAC by sending ACK or ERROR.)	
S-3	Plain command message (i.e. after termination of another SAC.)	
S-4	Response & Challenge message	Response & Request message
S-5		Challenge message
S-6		Response & Commit message
S-7	Reply message	Request message
S-8		Encrypted command message
S-9		Challenge message

Table 4-4: Valid message sequence of Marlin IPTV-ES Server

705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735

Figure 4 shows a brief diagram of the Marlin IPTV-ES Server about combinations of valid messages that are previously sent to and currently received from the Marlin IPTV-ES Device. The numbers shown in Figure 4 corresponds to the message combination of Table 4-4, e.g. "S-4" means that Response & Challenge message is previously sent and Response & Request message is currently received.

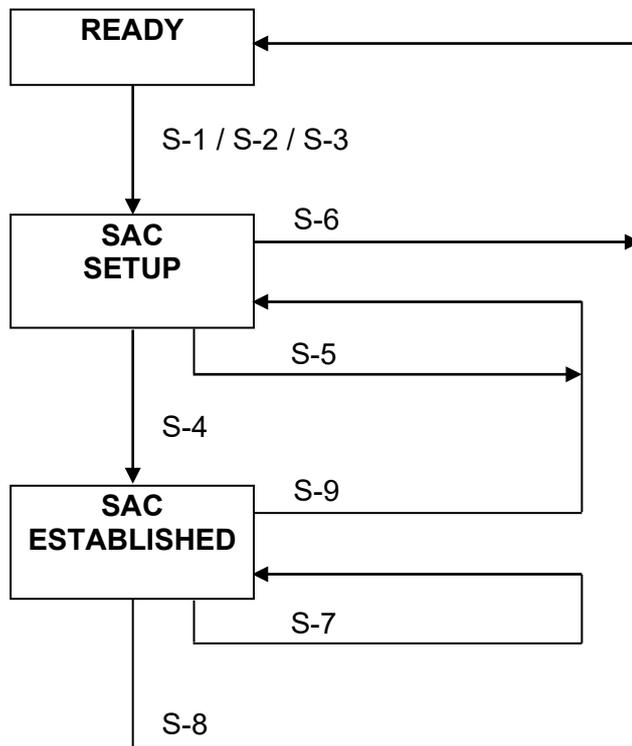


Figure 4: Valid message combinations of Marlin IPTV-ES Server (Informative)

736

Combina tion	Previously Sent Message	Currently Received Message
D-1	Challenge message	Response & Challenge message
D-2		Plain command message
D-3	Response & Request message	Reply message
D-4		Plain command message
D-5	Response & Commit message	Encrypted command message
D-6		Plain command message
D-7	Request message	Reply message
D-8		Encrypted command message
D-9	Encrypted command message	Encrypted command message

Table 4-5: Valid message sequence of Marlin IPTV-ES Device

737
738
739
740
741
742
743

Figure 5 shows a brief diagram of the Marlin IPTV-ES Device about combinations of valid messages that are previously sent to and currently received from the Marlin IPTV-ES Server. The numbers shown in Figure 5 corresponds to the message combination of Table 4-5, e.g. "D-1" means that Challenge message is previously sent and Response & Challenge message is currently received.

744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761

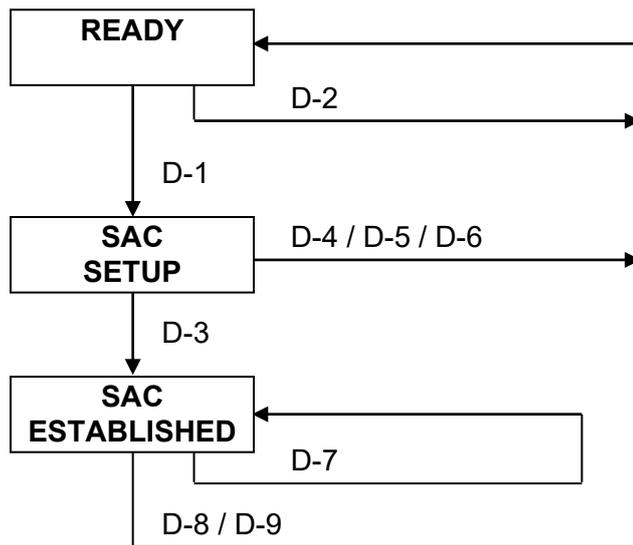


Figure 5: Valid message combinations of Marlin IPTV-ES Device (Informative)

762

763 4.1.4.2 Challenge message parameters

764 Whenever receiving this Challenge message, the Marlin IPTV-ES Server SHALL
765 verify its parameters as shown below.

766

- 767 • *SinkCertificateSize*
 - 768 ➤ for ECC 224-bit keys
 - 769 ✧ If Sink CertificateSize is other than (PayloadSize – 18 bytes), the
770 verification SHALL be deemed as “verification failure” and the Status of
771 the message sent to the Marlin IPTV-ES Device SHALL be set to
772 “Message error” (8002h).
 - 773 ➤ for ECC 384-bit keys
 - 774 ✧ If Sink CertificateSize is other than (PayloadSize – 34 bytes), the
775 verification SHALL be deemed as “verification failure” and the Status of
776 the message sent to the Marlin IPTV-ES Device SHALL be set to
777 “Message error” (8002h).
- 778 • *SinkCertificate*
 - 779 ➤ Using the public keys of the Issuers of the certificates, Marlin IPTV-ES
780 Server SHALL verify the signature of SinkCertificate. If failed, the
781 verification SHALL be deemed as “verification failure” and the Status of
782 the message sent to the Marlin IPTV-ES Device SHALL be set to
783 “Authentication error” (8003h).
 - 784 ➤ If path length validation is failed, the verification SHALL be deemed as
785 “verification failure” and the Status of the message sent to the Marlin
786 IPTV-ES Device SHALL be set to “Authentication error” (8003h).
 - 787 ➤ Using a DRL, Marlin IPTV-ES Server SHALL check whether the Marlin
788 IPTV-ES Device has been revoked or not. If the Marlin IPTV-ES Device is
789 determined to have been revoked, the verification SHALL be deemed as
790 “verification failure” and the Status of the message sent to the Marlin
791 IPTV-ES Device SHALL be set to “Revoked” (8004h). The Marlin IPTV-
792 ES Server SHALL use a DRL whose Signature is successfully verified in
793 accordance with section 4.1.4.13.

794

795 **4.1.4.3 Response & Challenge message parameters**

796 Whenever receiving this Response & Challenge message, the Marlin IPTV-ES
797 Device SHALL verify its parameters as shown below.

798
799

- 800 • *Signature*
 - 801 ➤ Marlin IPTV-ES Device SHALL verify the Signature using the
 - 802 SourceCertificate. If failed, the verification SHALL be deemed as
 - 803 “verification failure”.
- 804 • *SourceCertificateSize*
 - 805 ➤ for ECC 224-bit keys
 - 806 ✧ If SourceCertificateSize is other than (PayloadSize – 130 bytes), the
 - 807 verification SHALL be deemed as “verification failure”.
 - 808 ➤ for ECC 384-bit keys
 - 809 ✧ If SourceCertificateSize is other than (PayloadSize – 226 bytes), the
 - 810 verification SHALL be deemed as “verification failure”.
- 811 • *SourceCertificate*
 - 812 ➤ Using the public keys of the Issuers of the certificates, Marlin IPTV-ES
 - 813 Device SHALL verify the signature of SourceCertificate. If failed, the
 - 814 verification SHALL be deemed as “verification failure”.
 - 815 ➤ Marlin IPTV-ES Device SHALL verify the validity of SourceCertificate. If it
 - 816 is expired, the verification SHALL be deemed as “verification failure”.
 - 817 ➤ If path length validation is failed, the verification SHALL be deemed as
 - 818 “verification failure”.
 - 819 ➤ Using a CRL, Marlin IPTV-ES Device SHALL check whether the Marlin
 - 820 IPTV-ES Server has been revoked or not. If the Issuer and the serial
 - 821 number of the Source Certificate matches to those listed in the CRL, the
 - 822 Marlin IPTV-ES Server is determined to be revoked, and, therefore, the
 - 823 verification SHALL be deemed as “verification failure”. The Marlin IPTV-
 - 824 ES Device SHALL use a CRL whose Signature is successfully verified in
 - 825 accordance with section 4.1.4.14.
 - 826

827 **4.1.4.4 Response & Request message parameters**

828 Whenever receiving this Response & Request message, the Marlin IPTV-ES Server
829 SHALL verify its parameters as shown below.

830
831

- 831 • *Signature*
 - 832 ➤ Marlin IPTV-ES Server SHALL verify the Signature using the
 - 833 SinkCertificate. If failed, the verification SHALL be deemed as “verification
 - 834 failure” and the Status of the message sent to the Marlin IPTV-ES Device
 - 835 SHALL be set to “Authentication error” (8003h).
- 836 • *EncryptedDataSize*
 - 837 ➤ for ECC 224-bit keys
 - 838 ✧ If EncryptedDataSize is other than (PayloadSize – 116 bytes), the
 - 839 verification SHALL be deemed as “verification failure” and the Status of
 - 840 the message sent to the Marlin IPTV-ES Device SHALL be set to
 - 841 “Message error” (8002h).
 - 842 ➤ for ECC 384-bit keys
 - 843 ✧ If EncryptedDataSize is other than (PayloadSize – 196 bytes), the
 - 844 verification SHALL be deemed as “verification failure” and the Status of
 - 845 the message sent to the Marlin IPTV-ES Device SHALL be set to
 - 846 “Message error” (8002h).
- 847 • *SequenceNumber*

- 848 ➤ If SequenceNumber is other than 1, the verification SHALL be deemed as
849 "verification failure" and the Status of the message sent to the Marlin
850 IPTV-ES Device SHALL be set to "Message error" (8002h).
- 851 • *TransactionFlag*
 - 852 ➤ If TransactionFlag is not even (other than "00h"), the verification SHALL
853 be deemed as "verification failure" and the Status of the message sent to
854 the Marlin IPTV-ES Device SHALL be set to "Message error" (8002h).
 - 855 • *MessageDigest*
 - 856 ➤ If the hash value of the decrypted Response & Request message
857 excluding this MessageDigest is other than the value of this
858 MessageDigest, the verification SHALL be deemed as "verification failure"
859 and the Status of the message sent to the Marlin IPTV-ES Device SHALL
860 be set to "Authentication error" (8003h).
- 861

862 **4.1.4.5 Request message parameters**

863 Whenever receiving this Request message, the Marlin IPTV-ES Server SHALL verify
864 its parameters as shown below.

- 865
- 866 • *EncryptedDataSize*
 - 867 ➤ If EncryptedDataSize is other than (PayloadSize – 4 bytes), the
868 verification SHALL be deemed as "verification failure" and the Status of
869 the message sent to the Marlin IPTV-ES Device SHALL be set to
870 "Message error" (8002h).
 - 871 • *SequenceNumber*
 - 872 ➤ If SequenceNumber is other than the one retaining, or equal to or more
873 than (2²⁴-3), the verification SHALL be deemed as "verification failure"
874 and the Status of the message sent to the Marlin IPTV-ES Device SHALL
875 be set to "Message error" (8002h).
 - 876 • *TransactionFlag*
 - 877 ➤ If TransactionFlag is same as the one retaining, the verification SHALL be
878 deemed as "verification failure" and the Status of the message sent to the
879 Marlin IPTV-ES Device SHALL be set to "Message error" (8002h).
 - 880 • *MessageDigest*
 - 881 ➤ If the hash value of the decrypted Request message excluding this
882 MessageDigest is other than the value of this MessageDigest, the
883 verification SHALL be deemed as "verification failure" and the Status of
884 the message sent to the Marlin IPTV-ES Device SHALL be set to
885 "Authentication error" (8003h).
- 886

887 **4.1.4.6 Reply message parameters**

888 Whenever receiving this Reply message, the Marlin IPTV-ES Device SHALL verify its
889 parameters as shown below.

- 890
- 891 • *EncryptedDataSize*
 - 892 ➤ If EncryptedDataSize is other than (PayloadSize – 4 bytes), the
893 verification SHALL be deemed as "verification failure".
 - 894 • *SequenceNumber*
 - 895 ➤ If SequenceNumber is other than the one retaining, the verification
896 SHALL be deemed as "verification failure".
 - 897 • *MessageDigest*
 - 898 ➤ If the hash value of the decrypted Reply message excluding this
899 MessageDigest is other than the value of this MessageDigest, the

900 verification SHALL be deemed as “verification failure”.

901

902 In addition to the rules described above, the Marlin IPTV-ES Device SHALL verify its
903 parameters as shown below, if it has the capability of handling the Get Permission
904 Request message that requires TransactionFlag Management.

905

906 • *TransactionFlagRecordFlag*

907 ➤ If both of the following two conditions are met, the verification SHALL be
908 deemed as “verification failure”.

909 ✧ Just before receiving the Reply message, the Marlin IPTV-ES Device
910 sent a message (Response & Request message or Request
911 message) which carries either of the following two messages to the
912 Marlin IPTV-ES Server, which is the sender of the Reply message.

913 ○ Get Permission Request message that requires TransactionFlag
914 Management.

915 ○ Packed Message Request message which packs one or more Get
916 Permission Request messages that requires TransactionFlag
917 Management.

918 ✧ TransactionFlagRecordFlag is neither “00h” nor “01h”.

919

920 Conversely, the Marlin IPTV-ES Device MAY interpret the value of
921 TransactionFlagRecordFlag as if it is set to “not to record” (00h), regardless of its
922 value, if it has no capability of handling the Get Permission Request message that
923 requires TransactionFlag Management.

924

925 **4.1.4.7 Plain command message parameters**

926 Whenever receiving this Plain command message, the Marlin IPTV-ES Device
927 SHALL verify its parameters as shown below.

928

929 • *Command*

930 ➤ If Command is other than “ERROR”, the verification SHALL be deemed
931 as “verification failure”.

932 • *Status*

933 ➤ If Status is other than “Error other than below” (8001h), “Message error”
934 (8002h), “Authentication error” (8003h) or “Revoked” (8004h), the
935 verification SHALL be deemed as “verification failure”.

936

937 **4.1.4.8 Encrypted command message parameters**

938 Whenever receiving this Encrypted command message, the Marlin IPTV-ES Device
939 SHALL verify its parameters as shown below.

940

941 • *EncryptedDataSize*

942 ➤ Marlin IPTV-ES Device SHALL verify the EncryptedDataSize as specified
943 in section 4.1.4.6.

944 • *SequenceNumber*

945 ➤ Marlin IPTV-ES Device SHALL verify the SequenceNumber as specified
946 in section 4.1.4.6.

947 • *Command*

948 ➤ If Command of this message received after sending an Encrypted
949 Command message to the Marlin IPTV-ES Server is other than “ERROR”
950 or “ACK”, the verification SHALL be deemed as “verification failure”.

951 ➤ If Command of this message received after sending a Response &

- 952 Commit message to the Marlin IPTV-ES Server is other than “ACK”, the
 953 verification SHALL be deemed as “verification failure”.
- 954 ➤ If Command of this message received before sending an Encrypted
 955 Command message or a Response & Commit message to the Marlin
 956 IPTV-ES Server is other than “ERROR”, the verification SHALL be
 957 deemed as “verification failure”.
- 958 • *Status*
 - 959 ➤ If Status of this message, which its Command is “ERROR”, is other than
 960 “Error other than below” (8001h), “Message error” (8002h) or
 961 “Authentication error” (8003h), the verification SHALL be deemed as
 962 “verification failure”.
 - 963 ➤ If Status of this message, which its Command is “ACK”, is other than
 964 “Success” (0000h), the verification SHALL be deemed as “verification
 965 failure”.
 - 966 • *MessageDigest*
 - 967 ➤ If the hash value of the decrypted Encrypted command message
 968 excluding this MessageDigest is other than the value of this
 969 MessageDigest, the verification SHALL be deemed as “verification failure”.

970
 971
 972 Whenever receiving this Encrypted command message, the Marlin IPTV-ES Server
 973 SHALL verify its parameters as shown below.

- 974
- 975 • *EncryptedDataSize*
 - 976 ➤ If EncryptedDataSize is other than (PayloadSize – 4 bytes), the
 977 verification SHALL be deemed as “verification failure” and the Status of
 978 the message sent to the Marlin IPTV-ES Device SHALL be set to
 979 “Message error” (8002h).
 - 980 • *SequenceNumber*
 - 981 ➤ If SequenceNumber is other than the one retaining, the verification
 982 SHALL be deemed as “verification failure” and the Status of the message
 983 sent to the Marlin IPTV-ES Device SHALL be set to “Message error”
 984 (8002h).
 - 985 • *Command*
 - 986 ➤ If Command is other than “Commit”, the verification SHALL be deemed as
 987 “verification failure” and the Status of the message sent to the Marlin
 988 IPTV-ES Device SHALL be set to “Message error” (8002h).
 - 989 • *Status*
 - 990 ➤ If Status is other than “Success” (0000h), the verification SHALL be
 991 deemed as “verification failure” and the Status of the message sent to the
 992 Marlin IPTV-ES Device SHALL be set to “Message error” (8002h).
 - 993 • *MessageDigest*
 - 994 ➤ If the hash value of the decrypted Encrypted command message
 995 excluding this MessageDigest is other than the value of this
 996 MessageDigest, the verification SHALL be deemed as “verification failure”
 997 and the Status of the message sent to the Marlin IPTV-ES Device SHALL
 998 be set to “Authentication error” (8003h).

1000 4.1.4.9 Response & Commit message parameters

1001 Whenever receiving this Response & Commit message, the Marlin IPTV-ES Server
 1002 SHALL verify its parameters as shown below.

- 1003 • *Signature*

- 1005 ➤ Marlin IPTV-ES Server SHALL verify the Signature as specified in section
- 1006 4.1.4.4.
- 1007 • *EncryptedDataSize*
- 1008 ➤ Marlin IPTV-ES Server SHALL verify the EncryptedDataSize as specified
- 1009 in section 4.1.4.4.
- 1010 • *SequenceNumber*
- 1011 ➤ Marlin IPTV-ES Server SHALL verify the SequenceNumber as specified
- 1012 in section 4.1.4.4.
- 1013 • *TransactionFlag*
- 1014 ➤ If TransactionFlag is neither “00h” nor “01h”, the verification SHALL be
- 1015 deemed as “verification failure” and the Status of the message sent to the
- 1016 Marlin IPTV-ES Device SHALL be set to “Message error” (8002h).
- 1017 • *MessageDigest*
- 1018 ➤ If the hash value of the decrypted Response & Commit message
- 1019 excluding this MessageDigest is other than the value of this
- 1020 MessageDigest, the verification SHALL be deemed as “verification failure”
- 1021 and the Status of the message sent to the Marlin IPTV-ES Device SHALL
- 1022 be set to “Authentication error” (8003h).
- 1023

1024 **4.1.4.10 SAC termination**

1025 **4.1.4.10.1 SAC termination for Marlin IPTV-ES Devices**

1026 The Marlin IPTV-ES Device SHALL terminate SAC connection after sending an

1027 Encrypted command message with a Command of “Commit” or a Response &

1028 Commit message respectively to a Marlin IPTV-ES Server, and receiving an

1029 Encrypted command message with a Command of “ACK” from the Marlin IPTV-ES

1030 Server.

1031

1032 However, in the following four cases, the Marlin IPTV-ES Device SHALL terminate

1033 SAC connection without sending an Encrypted command message with a Command

1034 of “Commit” or a Response & Commit message respectively to the Marlin IPTV-ES

1035 Server.

- 1036
- 1037 • “Verification failure” occurred for parameters of SAC messages received from
- 1038 a Marlin IPTV-ES Server in accordance with the processing rules described in
- 1039 section 4.1.4.
- 1040 • Marlin IPTV-ES Device received a Plain command message or an Encrypted
- 1041 command message, respectively with a Command of “ERROR”, from a Marlin
- 1042 IPTV-ES Server.
- 1043 • “Verification failure” occurred for Service Protocol parameters of a Reply
- 1044 message which carries either of the following two messages, and of which the
- 1045 TransactionFlagRecordFlag is set to “record” (01h).
- 1046 ➤ Get Permission Reply message which corresponds to a Get Permission
- 1047 Request message that requires TransactionFlag Management.
- 1048 ➤ Packed Message Reply message which corresponds to a Packed
- 1049 Message Request message which packs one or more Get Permission
- 1050 Request messages that requires TransactionFlag Management.
- 1051 • Marlin IPTV-ES Device could not receive and quitted waiting for a message
- 1052 from a Marlin IPTV-ES Server.

1053

1054 When Marlin IPTV-ES Device terminated the SAC connection with a Marlin IPTV-ES

1055 Server in events such as accidental power down, the Marlin IPTV-ES Device SHALL

1056 NOT send an Encrypted command message with a Command of “Commit” to the
1057 Marlin IPTV-ES Server.
1058

1059 **4.1.4.10.2 SAC termination for Marlin IPTV-ES Servers**

1060 The Marlin IPTV-ES Server SHALL terminate SAC connection only for the following
1061 four cases:
1062

- 1063 • Marlin IPTV-ES Server received an Encrypted command message with a
1064 Command of “Commit” or a Response & Commit message respectively from
1065 a Marlin IPTV-ES Device, and sent an Encrypted command message with a
1066 Command of “ACK” to the Marlin IPTV-ES Device.
- 1067 • “Verification failure” occurred for parameters of messages received from a
1068 Marlin IPTV-ES Device in accordance with the processing rules described in
1069 section 4.1.4 and sent a Plain command message or an Encrypted command
1070 message, respectively with a Command of “ERROR”, to the Marlin IPTV-ES
1071 Device.
- 1072 • Marlin IPTV-ES Server could not receive and quitted waiting for messages
1073 from Marlin IPTV-ES Device.
- 1074 • Marlin IPTV-ES Server received a second Challenge message from an
1075 already connected Marlin IPTV-ES Device before terminating SAC
1076 connection as shown in the three cases above.
 - 1077 ➤ In this case, Marlin IPTV-ES Server SHALL terminate the SAC connection
1078 initiated by the previously received Challenge message before receiving
1079 the second Challenge message.
1080

1081 **4.1.4.11 TransactionFlag**

1082 The Marlin IPTV-ES Device and Server, which have the capability of handling the
1083 Get Permission Request message that requires TransactionFlag Management,
1084 SHALL process the following procedures after succeeding the verification, i.e.
1085 “verification succeeded” occurred during the verification, in sections 4.1.4.1 through
1086 4.1.4.9.
1087

1088 **4.1.4.11.1 Processing Rules for Marlin IPTV-ES Devices**

1089 (1) Storage of TransactionFlag and ContentKey

1090 When receiving a Reply message, the Marlin IPTV-ES Device SHALL store the
1091 TransactionFlag and the ContentKey if all of the following three conditions are met.
1092

- 1093 • Just before receiving the Reply message, the Marlin IPTV-ES Device sent a
1094 message (Response & Request message or Request message) which carries
1095 either of the following messages to the Marlin IPTV-ES Server, which is the
1096 sender of the Reply message.
 - 1097 ➤ Get Permission Request message that requires TransactionFlag
1098 Management.
 - 1099 ➤ Packed Message Request message which packs one or more Get
1100 Permission Request messages that requires TransactionFlag
1101 Management.
- 1102 • The TransactionFlagRecordFlag of the received Reply message is set to
1103 “record” (01h).
- 1104 • Succeeding the verification, i.e. “verification succeeded” occurred during the
1105 verification, for Service Protocol parameters of the received Reply message
1106 and the Status of the Service Protocol message is set to “Success” (0000h).

1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160

If all of these three conditions are met, the Marlin IPTV-ES Device SHALL store the TransactionFlag and the ContentKey in accordance with the following two procedures.

- Store the TransactionFlag of the message (Response & Request message or Request message) sent just before receiving the Reply message to the Marlin IPTV-ES Server, which is the sender of the Reply message, to Persistent Storage paired with an identifier of the Marlin IPTV-ES Server, and maintain the status of them.
 - When pairing with the TransactionFlag, the attribute value of the subject of the Service Key included in a SourceCertificate in the Response & Challenge message lastly received from the Marlin IPTV-ES Server, SHALL be used as the identifier.
- Store the received ContentKey of either of the following two messages to Persistent Storage as an unavailable state paired with the TransactionFlag stored in the aforementioned procedure, and maintain the status of it.
 - Get Permission Reply message which corresponds to the Get Permission Request message that requires TransactionFlag Management.
 - Get Permission Reply message which corresponds to the Get Permission Request message that requires TransactionFlag Management which is packed in the Packed Message Reply message.

However, if any of these two procedures cannot be completed, the Marlin IPTV-ES Device SHALL set back the TransactionFlag and the ContentKey stored on Persistent Storage to the state before storing them.

(2) Deletion of stored TransactionFlag and status change of stored ContentKey
When receiving an Encrypted command message with a Command of "ACK" or a Reply message, the Marlin IPTV-ES Device SHALL delete the stored TransactionFlag and SHALL change the status of the stored ContentKey if the following condition is met. Note that the Marlin IPTV-ES Device SHALL change the status of the stored ContentKey if and only if the following condition is met while the Marlin IPTV-ES Device MAY delete the stored TransactionFlag even if the following condition is not met. Conditions and processing rules except for the following of when and how the Marlin IPTV-ES Device MAY delete the stored TransactionFlag are not to be specified in this specification.

- When the Marlin IPTV-ES Device receives either of the following messages and a TransactionFlag is stored on Persistent Storage paired with the identifier of the Marlin IPTV-ES Server, which is the sender of the received message.
 - An Encrypted command message with a Command of "ACK".
 - A Reply message.

If this condition is met, the Marlin IPTV-ES Device SHALL delete the stored TransactionFlag and change the status of the stored ContentKey in accordance with the following two procedures.

- Delete the TransactionFlag stored on Persistent Storage paired with an identifier of the Marlin IPTV-ES Server which is the sender of the received message, provided that the Marlin IPTV-ES Device MAY delete the identifier of the Marlin IPTV-ES Server stored on Persistent Storage paired with the deleted TransactionFlag.

- 1161 • Change the status of the ContentKey stored on Persistent Storage paired
1162 with the deleted TransactionFlag from an unavailable state to an available
1163 state.
1164

1165 However, if any of these two procedures cannot be completed, the Marlin IPTV-ES
1166 Device SHALL set back the TransactionFlag and the ContentKey stored on
1167 Persistent Storage to the state before deleting the TransactionFlag and changing the
1168 state of the ContentKey.
1169

1170 If all of the aforementioned conditions in this section are met, in other words, if the
1171 Marlin IPTV-ES Device receives the Reply message that meets all of the three
1172 conditions specified under the subsection of “Storage of TransactionFlag and
1173 ContentKey”, and if the TransactionFlag is stored on Persistent Storage paired with
1174 the identifier of the Marlin IPTV-ES Server which is the sender of the Reply message,
1175 the Marlin IPTV-ES Device SHALL make the status of the TransactionFlag, the
1176 ContentKey and the paired identifier of the Marlin IPTV-ES Server to be in the same
1177 status when the Marlin IPTV-ES Device first completed the set of two procedures
1178 specified under this subsection of “Deletion of stored TransactionFlag and status
1179 change of stored ContentKey” and then completed the set of two procedures
1180 specified under the subsection of “Storage of TransactionFlag and ContentKey”.
1181

1182 (3) Sending of stored TransactionFlag as a Response & Commit message
1183 When receiving a Response & Challenge message, the Marlin IPTV-ES Device
1184 SHALL send the stored TransactionFlag to the Marlin IPTV-ES Server, if the
1185 following condition is met.
1186

- 1187 • A TransactionFlag is stored on Persistent Storage paired with an identifier of
1188 the Marlin IPTV-ES Server, which is the sender of the received Response &
1189 Challenge message.
1190

1191 If this condition is met, the Marlin IPTV-ES Device SHALL send the stored
1192 TransactionFlag with a Response & Commit message to the Marlin IPTV-ES Server.
1193

1194 **4.1.4.11.2 Processing Rules for Marlin IPTV-ES Server**

1195 (1) Storage of TransactionFlag
1196 When sending a Reply message, the Marlin IPTV-ES Server SHALL store the
1197 TransactionFlag if the following condition is met.
1198

- 1199 • The TransactionFlagRecordFlag of the sent Reply message is set to “record”
1200 (01h).
1201

1202 If this condition is met, the Marlin IPTV-ES Server SHALL store the TransactionFlag
1203 in accordance with the following procedure.
1204

- 1205 • Store the TransactionFlag of the message (Response & Request message or
1206 Request message) received just before sending the Reply message from the
1207 Marlin IPTV-ES Device, which is the receiver of the Reply message, to
1208 Persistent Storage paired with an identifier of the Marlin IPTV-ES Device.
1209 When a TransactionFlag is already stored, the stored TransactionFlag SHALL
1210 be updated with the one of the most recently received.
1211 ➤ When pairing with the TransactionFlag, the attribute value of the subject
1212 of the Client Key included in a SinkCertificate in the Challenge message
1213 lastly received from the Marlin IPTV-ES Device, in other words Device ID,

1214 SHALL be used as the identifier.

1215

1216 (2) Deletion of stored TransactionFlag

1217 When sending an Encrypted command message with a Command of "ACK" or a
1218 Reply message, the Marlin IPTV-ES Server SHALL delete the stored
1219 TransactionFlag if either of the following two conditions is met. Note that the Marlin
1220 IPTV-ES Server MAY delete the stored TransactionFlag even if none of the following
1221 two conditions is met. Conditions and processing rules except for the following of
1222 when and how the Marlin IPTV-ES Server MAY delete the stored TransactionFlag
1223 are not to be specified in this specification.

1224

1225 • The TransactionFlagRecordFlag of the sent Reply message is set to "not to
1226 record" (00h).

1227 • The Command of the sent Encrypted command message is set to "ACK".

1228 If either of these two conditions is met, the Marlin IPTV-ES Server SHALL delete the
1229 stored TransactionFlag in accordance with the following procedure.

1230

1231 • Delete the TransactionFlag stored on Persistent Storage paired with an
1232 identifier of the Marlin IPTV-ES Device which is the receiver of the Encrypted
1233 command message or a Reply message, provided that the Marlin IPTV-ES
1234 Server MAY delete the identifier of the Marlin IPTV-ES Device paired with the
1235 TransactionFlag to delete.

1236

1237 (3) Judgement of Reply message reception

1238 After sending a Reply message to a Marlin IPTV-ES Device which a TransactionFlag
1239 is stored on Persistent Storage paired with and after receiving a message from the
1240 same Marlin IPTV-ES Device, the Marlin IPTV-ES Server SHALL judge the reception
1241 of the Reply message, in other words the sent ContentKey, most recently sent to the
1242 Marlin IPTV-ES Device in accordance with the following criteria.

1243

1244 • The Marlin IPTV-ES Server SHALL determine that the sent Reply message is
1245 received by the Marlin IPTV-ES Device, if it receives either of the following
1246 three messages.

1247 ➤ A Request message.

1248 ➤ An Encrypted command message with a Command of "Commit".

1249 ➤ A Response & Commit message which the TransactionFlag of the
1250 received Response & Commit message is same as the stored one.

1251 • The Marlin IPTV-ES Server SHALL determine that the sent Reply message is
1252 not received by the Marlin IPTV-ES Device, if it receives either of the
1253 following two messages.

1254 ➤ A Response & Request message.

1255 ➤ A Response & Commit message which the TransactionFlag of the
1256 received Response & Commit message is other than the stored one.

1257

1258 4.1.4.12 URI signature verification

1259 Before establishment of connection, the Marlin IPTV-ES Device SHALL verify the
1260 signature of the URI that indicates the network location of the Marlin IPTV-ES Server
1261 to whom the Marlin IPTV-ES Device is attempting to connect in accordance with the
1262 following:

1263

1264 • The signature of the URI of the Marlin IPTV-ES Server SHALL be generated
1265 and verified as the same manner as for the signature used in SAC
1266 Authentication, which is defined in section 4.1.2.

- 1267
- 1268
- 1269
- 1270
- 1271
- 1272
- 1273
- 1274
- 1275
- The format of the certificate used for this signature verification of the URI of the Marlin IPTV-ES Server SHALL be specified in PKIPath and SHALL same as the SourceCertificate used in SAC Authentication, which is defined in section 4.1.3.3.
 - The certificate used for this signature verification of the URI of the Marlin IPTV-ES Server SHALL be verified as the same manner as for SourceCertificate of Response & Challenge message described in section 4.1.4.3.

1276 Note that the format of the signature, the format of the information to be signed, and

1277 the behaviour of a Marlin IPTV-ES Device based on the result of this signature

1278 verification are assumed to be defined outside of this specification.

1279

1280 **4.1.4.13 DRL parameters**

1281 The Marlin IPTV-ES Server SHALL perform the following process against a DRL

1282 used for checking the occurrence of revocation.

- 1283
- 1284
- *Signature*
 - Marlin IPTV-ES Server SHALL verify the Signature.
 - *nextUpdate*
 - Marlin IPTV-ES Server SHALL check whether it is passed nextUpdate or not by using its Trusted Time.
- 1285
- 1286
- 1287
- 1288
- 1289

1290 **4.1.4.14 CRL parameters**

1291 The Marlin IPTV-ES Device SHALL perform the following process against a CRL

1292 used for checking the occurrence of revocation.

- 1293
- 1294
- *Signature*
 - Marlin IPTV-ES Device SHALL verify the Signature.
 - *nextUpdate*
 - Marlin IPTV-ES Device SHALL check whether it is passed nextUpdate or not by using its Trusted Time.
- 1295
- 1296
- 1297
- 1298
- 1299

1300 **4.2 Marlin IPTV-ES Service Protocols over SAC**

1301 This section defines Marlin IPTV-ES Service Protocols in SAC. Each of Request

1302 messages defined in this section corresponds to the Request of Response &

1303 Request message defined in section 4.1.3.4 or the Request of Request message

1304 defined in section 4.1.3.5.

1305 Each of Response messages defined in this section corresponds to the Reply of

1306 Reply message defined in section 4.1.3.6.

1307

1308 **4.2.1 Get Permission Protocol**

1309 **4.2.1.1 Overview**

1310 This is a simple REQUEST/RESPONSE protocol via SAC. The Marlin IPTV-ES

1311 Device requests a certain action to the Marlin IPTV-ES Server. When the REQUEST

1312 is permitted, the Marlin IPTV-ES Server returns the Content Key, or the Work Key

1313 and its related Work Key ID, and/or other related information as Status Extension.

1314 The requested permission is identified with UsageRuleReference which is

1315 information delivered to a Marlin IPTV-ES Device in advance of sending the request.
1316 The delivery method is outside of the scope of this specification.

1317

1318 As described in section 2.1, there are cases where the technology defined in this
1319 specification is used for key delivery for conditional access services. In such cases,
1320 Get Permission Request messages and Get Permission Reply messages for
1321 EXPORT and/or RECORD actions with Indirect Key Delivery defined in the following
1322 section 4.2.1 MAY NOT be exchanged between Marlin IPTV-ES Devices and Marlin
1323 IPTV-ES Servers. In cases where those messages are not exchanged, EXPORT or
1324 RECORD action with Indirect Key Delivery MAY be deemed to be granted and
1325 performed based on RenderingObligation instead of ExportInfo or RecordInfo,
1326 respectively. Whether those messages are exchanged or not is assumed to be
1327 determined outside of this specification.

1328

1329 Note that whether or not the TransactionFlag Management is required for the Get
1330 Permission Request message depends on the action requested from the Marlin
1331 IPTV-ES Device. When the Marlin IPTV-ES Device requests for EXPORT action, the
1332 TransactionFlag Management is required for the Get Permission Request message,
1333 while the TransactionFlag Management is not required when RECORD action is
1334 requested. When the Marlin IPTV-ES Device requests for EXTRACT action, there
1335 are cases where the TransactionFlag Management is required and cases where not
1336 required. Whether the TransactionFlag Management is required or not is assumed to
1337 be determined outside of this specification.

1338

1339 4.2.1.2 Get Permission Request parameters

1340 Get Permission Request message SHALL include the parameters defined below.

1341

- 1342 • *ProtocolVersion*: The version identifier of the protocol defined in this
1343 specification.
- 1344 • *MessageID*: The message identifier of the Get Permission Request message
1345 in this specification.
- 1346 • *DeviceInformation*: The Device Information of the Marlin IPTV-ES Device
1347 defined in section 3.2.2.
- 1348 • *UsageRuleReference*: The identifier for the requesting usage rules of the
1349 content.
- 1350 • *ActionID*: The identifier of the requested action in the Marlin IPTV-ES Device.
1351 In this version of Marlin IPTV-ES specification, the following ActionIDs are
1352 defined:
 - 1353 ➤ 01h: This value indicates that the Marlin IPTV-ES Device requests to
1354 “**EXTRACT** with Simple Key Delivery”.
 - 1355 ➤ 02h: This value indicates that the Marlin IPTV-ES Device requests to
1356 “**EXTRACT** with Indirect Key Delivery”. Once RECORD action and
1357 EXTRACT action are permitted for the content, this action is not needed
1358 to render the content which is recorded on the Protected Storage.
 - 1359 ➤ 03h: This value indicates that the Marlin IPTV-ES Device requests to
1360 “**EXTRACT** with Direct Key Delivery”.
 - 1361 ➤ 10h: This value indicates that the Marlin IPTV-ES Device requests to
1362 “**EXPORT** to a certain Media for Copy with Direct Key Delivery”. When
1363 this value is specified, the exporting Media type SHALL be specified by
1364 ActionParameter.
 - 1365 ➤ 11h: This value indicates that the Marlin IPTV-ES Device requests to
1366 “**EXPORT** to a certain Media for Move with Direct Key Delivery”. When
1367 this value is specified, the exporting Media type SHALL be specified by

- 1368 ActionParameter.
- 1369 ➤ 12h: This value indicates that the Marlin IPTV-ES Device requests to
- 1370 “**EXPORT** to a certain Media with Indirect Key Delivery”. When this value
- 1371 is specified, the exporting Media type SHALL be specified by
- 1372 ActionParameter.
- 1373 ➤ 20h: This value indicates that the Marlin IPTV-ES Device requests to
- 1374 “**RECORD** to a certain Media with Indirect Key Delivery”. When this value
- 1375 is specified, the recording Media type SHALL be specified by
- 1376 ActionParameter.
- 1377 • *ActionParameter*: The parameter pertaining to the requested ActionID.
- 1378 ➤ For the case when ActionID is set to “EXTRACT with Simple Key
- 1379 Delivery” (01h), “EXTRACT with Indirect Key Delivery” (02h), or
- 1380 “EXTRACT with Direct Key Delivery” (03h), *ActionParameter* SHALL be
- 1381 set to “FFh”.
- 1382 ➤ For the case when ActionID is set to “EXPORT for Copy with Direct Key
- 1383 Delivery” (10h), “EXPORT for Move with Direct Key Delivery” (11h), or
- 1384 “EXPORT with Indirect Key Delivery” (12h), the parameters defined in
- 1385 [MEXP] SHALL be set, with the exception of the case of Re-Transmission
- 1386 of Digital Broadcasting over IP network in Japan. In such case,
- 1387 *ActionParameter* SHALL be set to the values corresponding to the target
- 1388 copy protection systems in accordance with the rules stated in
- 1389 government or quasi-government regulations.
- 1390 ➤ For the case when ActionID is set to “RECORD with Indirect Key
- 1391 Delivery” (20h), the following parameter is defined in this specification:
- 1392 • 01h: RECORD to Protected Storage.
- 1393 • *SpecificCRID*: The identifier of the Marlin IPTV-ES specific Compliance Rules
- 1394 with which the Marlin IPTV-ES Device complies. If no specific compliance rule
- 1395 is applicable, this value SHALL be set to “FFFFh”. Otherwise, the value
- 1396 specified in each specific compliance rule SHALL be set.
- 1397 • *PrivateDataTag*: The identifier of the usage of PrivateData. In this
- 1398 specification, the following PrivateDataTags are defined:
- 1399 ➤ 00h: PrivateData not in use. When this value is specified, all bytes of
- 1400 PrivateData SHALL be set to “00h”.
- 1401 ➤ 01h: Removable media ID.
- 1402 ➤ 02h–7Fh: Reserved for common use.
- 1403 ➤ 80h–FFh: For private use.
- 1404

Byte index	Description
0-1	ProtocolVersion. “0100h” SHALL be set for ECC 224-bit keys. “0200h” SHALL be set for ECC 384-bit keys.
2-3	MessageID. “0001h” SHALL be set for this specification.
4-15	DeviceInformation.
16-31	UsageRuleReference.
32	ActionID.
33	ActionParameter.
34-35	SpecificCRID.
36	PrivateDataTag.
37-63	PrivateData.

1405

1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432

4.2.1.3 Get Permission Reply parameters

Get Permission Reply message SHALL include the parameters defined below.

- *ProtocolVersion*: The version identifier of the protocol defined in this specification.
- *MessageID*: The message identifier of the Get Permission Reply message in this specification.
- *Status*: When the request is not authorized/failed for some reasons, the status defined in Table 4-6 is returned.
- *StatusExtension*: The data structure used to convey extended information of the status. When the request is failed, i.e. when the Status is other than “Success” (0000h), this parameter SHALL NOT be returned. In this version of Marlin IPTV-ES specification, the following StatusExtensions are defined:
 - For the case when ActionID is set to “EXTRACT with Simple Key Delivery” (01h), the value defined in section 4.2.1.4 is returned.
 - For the case when ActionID is set to “EXTRACT with Indirect Key Delivery” (02h), the value defined in section 4.2.1.5 is returned.
 - For the case when ActionID is set to “EXTRACT with Direct Key Delivery” (03h), the value defined in section 4.2.1.6 is returned.
 - For the case when ActionID is set to “EXPORT for Copy with Direct Key Delivery” (10h) or “EXPORT for Move with Direct Key Delivery” (11h), the value defined in section 4.2.1.7 is returned.
 - For the case when ActionID is set to “EXPORT with Indirect Key Delivery” (12h), the value defined in section 4.2.1.8 is returned.
 - For the case when ActionID is set to “RECORD with Indirect Key Delivery” (20h), the value defined in section 4.2.1.9 is returned.

Byte index	Description
0-1	ProtocolVersion. “0100h” SHALL be set for ECC 224-bit keys. “0200h” SHALL be set for ECC 384-bit keys.
2-3	MessageID. “0002h” SHALL be set for this specification.
4-5	Status.
6-(6+size of StatusExtension-1)	StatusExtension.

1433
1434
1435

Status values are below.

Values	Details
0000h	Success.
8001h	Error other than the below.
8002h	Version error.
8003h	Device Information error.
8004h	UsageRuleReference error.
8005h	ActionID error.
8006h	ActionParameter error.
8007h	Action denied.

Table 4-6: Status value of Get Permission Reply

1436

1437
1438
1439
1440
1441
1442
1443
1444
1445
1446

4.2.1.4 StatusExtension for “EXTRACT with Simple Key Delivery”

StatusExtension of Get Permission Reply message, which is a reply to Get Permission Request message with an ActionID of “EXTRACT with Simple Key Delivery” (01h), SHALL include the parameters defined below.

- *ContentKey*: The Content Key corresponding to the requested Usage Rule Reference is returned.
- *ExtractInfoSize*: Size of ExtractInfo.
- *ExtractInfo*: The value defined in section 4.2.1.4.1 is returned.

Byte index	Description
0-15	ContentKey.
16-17	ExtractInfoSize.
18- (18+ExtractInfoSiz e-1)	ExtractInfo.

1447

1448

4.2.1.4.1 ExtractInfo

1449
1450
1451
1452
1453

ExtractInfo consists of validity period and obligation information for the EXTRACT action. This information is returned from Marlin IPTV-ES Server in the case when Marlin IPTV-ES Device requests EXTRACT action with Simple, Indirect and Direct Key Deliveries.

1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471

- *NotBefore*: Date before which the action is denied is returned. The value is specified as 32-bit unsigned integer value, representing the number of minutes elapsed since January 1, 1970 00:00:00. The value is a UTC date. The value SHALL be smaller than the value of its corresponding NotAfter, except for the special cases defined hereinbelow.
- *NotAfter*: Date after which the action is denied is returned. The value is specified as 32-bit unsigned integer value, representing the number of minutes elapsed since January 1, 1970 00:00:00. The value is a UTC date. The value SHALL be larger than the value of its corresponding NotBefore, except for the special cases defined hereinbelow.
- *RenderingObligation*: Output Control Information is returned. The Output Control Information is essential information of descriptors, which are concatenation of six fields of DTCP_descriptor defined in [DTCP] Appendix B and a field defined in this specification, are returned. “0000h” SHALL be set whenever this RenderingObligation is returned as a reply against a request with ActionID of “EXTRACT with Indirect Key Delivery” (02h), and this parameter SHALL NOT be used as Output Control Information.

1472
1473
1474

The special cases of interpretations for a validity period by combinations of NotBefore and NotAfter are followings:

1475
1476
1477
1478
1479
1480
1481
1482

- When all bytes for NotBefore or NotAfter are set to “FFh”, this means that there is no restriction on the validity period of the Content Key or the Work Key.
- When all bytes for NotBefore and NotAfter are set to “00h”, this means that the Content Key is only allowed to be cached during the rendering of the content. The definition of “during the rendering” above is a subject to be defined in the Marlin compliance rules. “00000000h” SHALL be set whenever this ExtractInfo is returned as a reply against a request with ActionID of

1483 “EXTRACT with Simple Key Delivery” (01h) and SHALL NOT be set
 1484 whenever returned as a reply against a request with ActionID of “EXTRACT
 1485 with Indirect Key Delivery” (02h) or “EXTRACT with Direct Key Delivery” (03h).
 1486

Byte index	Description
0–3	NotBefore.
4–7	NotAfter.
8–9	RenderingObligation.

1487
 1488 The Output Control Information is followings:
 1489

- 1490 • *DigitalRecordingControlData*: Information of control copy generation and
 1491 coded, defined as DTCP_CCI in [DTCP] Appendix B, is returned.
- 1492 • *CopyControlType*: Information whether the output is encoded or not to serial
 1493 interface is returned. The value of this parameter is defined in Table 4-7.
- 1494 • *APSControlData*: Information of control analog output copy, defined as APS
 1495 in [DTCP] Appendix B, is returned.
- 1496 • *ImageConstraintToken*: Information whether the image quality of video signal
 1497 output is constrained is returned. The value of this parameter is defined as
 1498 Image_Constraint_Token in [DTCP] Appendix B.
- 1499 • *RetentionMode*: Information whether temporal accumulation is possible or not
 1500 is returned. The value of this parameter is defined as Retention_Move_mode
 1501 in [DTCP] Appendix B.
- 1502 • *RetentionState*: Information of the allowable time of temporal accumulation
 1503 after the reception of contents is returned. The value of this parameter is
 1504 defined as Retention_State in [DTCP] Appendix B.
- 1505 • *EncryptionMode*: Information whether the output of high-speed digital
 1506 interface is protected or not is returned. The value of this parameter is defined
 1507 as EPN in [DTCP] Appendix B.
 1508

Bit index	Description
0–1	DigitalRecordingControlData.
2–3	CopyControlType.
4–5	APSControlData.
6	ImageConstraintToken.
7	RetentionMode.
8–10	RetentionState.
11	EncryptionMode.
12–15	UserDefined.

1509
 1510 CopyControlType values are below.
 1511

Values	Description
00	Undefined.
01	Output by encoding to serial interface. (Encoding method specified by service provider is used.)
10	Undefined.
11	Output by not encoding to serial interface.

Table 4-7: CopyControlType

1512

1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529

4.2.1.5 StatusExtension for “EXTRACT with Indirect Key Delivery”

StatusExtension of Get Permission Reply message, which is a reply to Get Permission Request message with an ActionID of “EXTRACT with Indirect Key Delivery” (02h), SHALL include the parameters defined below.

- *WorkKey*: The Work Key corresponding to the requested Usage Rule Reference is returned.
- *WorkKeyID*: The identifier of WorkKey. The value defined in section 4.2.1.5.1 is returned.
- *SubscriptionTierBits*: A 64-bits-long bit string that specifies the subscription of a Marlin IPTV-ES Device corresponding to the WorkKeyID is returned. The bits that only correspond to the subscription of a Marlin IPTV-ES Device SHALL be set to “1b”.
- *ExtractInfoSize*: Size of ExtractInfo.
- *ExtractInfo*: The value defined in section 4.2.1.4.1 is returned.

Byte index	Description
0-15	WorkKey.
16-21	WorkKeyID.
22-23	PrivateData.
24-31	SubscriptionTierBits.
32-33	ExtractInfoSize.
34- (34+ExtractInfoSize-1)	ExtractInfo.

1530

4.2.1.5.1 WorkKeyID

WorkKeyID consists of ServiceProviderID, ReservedByte, WorkKeyManagementID and WorkKeyVersion. This information is returned from Marlin IPTV-ES Server in the case when Marlin IPTV-ES Device requests EXTRACT, EXPORT or RECORD action with Indirect Key Delivery.

1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550

- *ServiceProviderID*: An identifier to specify a service provider is returned. The service provider-specific value obtained from MTMO SHALL be set.
- *WorkKeyManagementID*: An identifier of a unit for managing Work Keys is returned. A unique value for each Tier Bits within a single service provider SHALL be set. Therefore, the value of this WorkKeyManagementID SHALL NOT be changed even when values of WorkKey and SubscriptionTierBits, and ExtractInfo, ExportInfo or RecordInfo are to be changed.
- *WorkKeyVersion*: A value that specifies the version of Work Key for a single WorkKeyManagementID is returned. The Work Key whose LSB value of this WorkKeyVersion is “1b” is called “Work Key (odd)” and “0b” called “Work Key (even)”, respectively. The initial value SHALL be “01h” and SHALL be incremented one at a time when the Work Key is renewed. The value SHALL be set to “00h” if the Work Key is renewed when this value is “FFh”.

Byte index	Description
0-1	ServiceProviderID.
2	ReservedByte. “00h” SHALL be set for this specification.
3-4	WorkKeyManagementID.

Byte index	Description
5	WorkKeyVersion.

1551

1552

4.2.1.6 StatusExtension for “EXTRACT with Direct Key Delivery”

1553

StatusExtension of Get Permission Reply message, which is a reply to Get Permission Request message with an ActionID of “EXTRACT with Direct Key Delivery” (03h), SHALL include the parameters defined below.

1554

1555

1556

1557

1558

1559

1560

1561

- *ContentKey*: The Content Key corresponding to the requested Usage Rule Reference is returned.
- *ExtractInfoSize*: Size of ExtractInfo.
- *ExtractInfo*: The value defined in section 4.2.1.4.1 is returned.

Byte index	Description
0-15	ContentKey.
16-17	ExtractInfoSize.
18- (18+ExtractInfoSize-1)	ExtractInfo.

1562

1563

4.2.1.7 StatusExtension for “EXPORT with Direct Key Delivery”

1564

StatusExtension of Get Permission Reply message, which is a reply to Get Permission Request message with an ActionID of “EXPORT for Copy with Direct Key Delivery” (10h) or “EXPORT for Move with Direct Key Delivery” (11h), SHALL include the parameters defined below.

1565

1566

1567

1568

1569

1570

1571

1572

1573

1574

- *ContentKey*: The Content Key corresponding to the requested Usage Rule Reference is returned.
- *ExportInfoSize*: Size of ExportInfo.
- *ExportInfo*: The corresponding Action Result Parameter defined in [MEXP] SHALL be set.

Byte index	Description
0-15	ContentKey.
16-17	ExportInfoSize.
18- (18+ExportInfoSize-1)	ExportInfo.

1575

1576

4.2.1.8 StatusExtension for “EXPORT with Indirect Key Delivery”

1577

StatusExtension of Get Permission Reply message, which is a reply to Get Permission Request message with an ActionID of “EXPORT with Indirect Key Delivery” (12h), SHALL include the parameters defined below.

1578

1579

1580

1581

1582

1583

1584

1585

- *WorkKeyID*: The identifier of WorkKey. The value defined in section 4.2.1.5.1 is returned.
- *ExportInfoSize*: Size of ExportInfo.
- *ExportInfo*: The corresponding Action Result Parameter defined in [MEXP] SHALL be set.

1586

Byte index	Description
0-5	WorkKeyID.
6-15	PrivateData.
16-17	ExportInfoSize.
18- (18+ExportInfoSize-1)	ExportInfo.

1587

1588 **4.2.1.9 StatusExtension for “RECORD with Indirect Key Delivery”**

1589 StatusExtension of Get Permission Reply message, which is a reply to Get
 1590 Permission Request message with an ActionID of “RECORD with Indirect Key
 1591 Delivery” (20h), SHALL include the parameters defined below.

1592

- 1593 • *WorkKeyID*: The identifier of WorkKey. The value defined in section 4.2.1.5.1
 1594 is returned.
- 1595 • *RecordInfoSize*: Size of RecordInfo.
- 1596 • *RecordInfo*: Output Control Information defined in section 4.2.1.4.1 is
 1597 returned.

1598

Byte index	Description
0-5	WorkKeyID.
6-15	PrivateData.
16-17	RecordInfoSize.
18- (18+RecordInfoSize-1)	RecordInfo.

1599

1600 **4.2.2 Get Trusted Time Protocol**

1601 **4.2.2.1 Overview**

1602 This is a simple REQUEST/RESPONSE protocol via SAC. The Marlin IPTV-ES
 1603 Device requests time information to the Marlin IPTV-ES Server. When the REQUEST
 1604 is permitted, the Marlin IPTV-ES Server returns the time information.

1605

1606 **4.2.2.2 Get Trusted Time Request parameters**

1607 Get Trusted Time Request message SHALL include the parameters defined below.

1608

- 1609 • *ProtocolVersion*: The version identifier of the protocol defined in this
 1610 specification.
- 1611 • *MessageID*: The message identifier of the Get Trusted Time Request
 1612 message in this specification.

1613

Byte index	Description
0-1	ProtocolVersion. “0100h” SHALL be set for ECC 224-bit keys. “0200h” SHALL be set for ECC 384-bit keys.
2-3	MessageID. “0003h” SHALL be set for this specification.

1614

1615 **4.2.2.3 Get Trusted Time Reply parameters**

1616 Get Trusted Time Reply message SHALL include the parameters defined below.

1617

- 1618 • *ProtocolVersion*: The version identifier of the protocol defined in this specification.
- 1619
- 1620 • *MessageID*: The message identifier of the Get Trusted Time Reply message in this specification.
- 1621
- 1622 • *Status*: When the request is not authorized/failed for some reasons, the status defined in Table 4-8 is returned.
- 1623
- 1624 • *Datetime*: When the request is authorized, the time information is returned. The value is specified as 32-bit unsigned integer value, representing the number of minutes elapsed since January 1, 1970 00:00:00. The value is a UTC date. When the request is failed, NULL value (00h) is returned.
- 1625
- 1626
- 1627
- 1628

Byte index	Description
0-1	ProtocolVersion. "0100h" SHALL be set for ECC 224-bit keys. "0200h" SHALL be set for ECC 384-bit keys.
2-3	MessageID. "0004h" SHALL be set for this specification.
4-5	Status.
6-9	Datetime.

1629

1630 Status values are below.

1631

Values	Details
0000h	Success.
8001h	Error other than the below.
8002h	Version error.
8008h	Action failed.

Table 4-8: Status value of Get Trusted Time Reply

1632

1633 **4.2.3 Packed Message Protocol**

1634 **4.2.3.1 Overview**

1635 This is a simple REQUEST/RESPONSE protocol via SAC. The Marlin IPTV-ES
 1636 Device packs some Request messages defined section 4.2.1 and section 4.2.2 into a
 1637 Packed Message Request, and sends to the Marlin IPTV-ES Server. Then the Marlin
 1638 IPTV-ES Server also packs corresponding Replies into a Packed Message Reply
 1639 and returns to the Marlin IPTV-ES Device.
 1640

1641 **4.2.3.2 Packed Message Request parameters**

1642 Packed Message Request message SHALL include the parameters defined below.

1643

- 1644 • *ProtocolVersion*: The version identifier of the protocol defined in this specification.
- 1645

1646
1647
1648
1649
1650
1651

- *MessageID*: The message identifier of the Packed Message Request message in this specification.
- *NumberOfRequestMessageBoxes*: Number of RequestMessageBoxes in the Packed Message Request message.
- *RequestMessageBoxList*: The list of RequestMessageBox.

Byte index	Description
0-1	ProtocolVersion. "0100h" SHALL be set for ECC 224-bit keys. "0200h" SHALL be set for ECC 384-bit keys.
2-3	MessageID. "0101h" SHALL be set for this specification.
4-5	NumberOfRequestMessageBoxes.
6-(6+ size of RequestMessageBoxList -1)	RequestMessageBoxList.

1652
1653
1654

The RequestMessageBox is defined as follows.

Byte index	Description
0-1	RequestMessageSize.
2-(2+RequestMessageSize-1)	RequestMessage.

1655

4.2.3.3 Packed Message Reply parameters

1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671

Packed Message Reply message SHALL include the parameters defined below.

- *ProtocolVersion*: The version identifier of the protocol defined in this specification.
- *MessageID*: The message identifier of the Packed Message Reply message in this specification.
- *Status*: When the Packed Message Request fails, for example message format error, the error status defined in Table 4-9 is returned.
- *NumberOfReplyMessageBoxes*: Number of ReplyMessageBoxes in the Packed Message Reply message. When the request is failed, i.e. when the Status is other than "Success" (0000h), zero (0000h) is returned.
- *ReplyMessageBoxList*: The list of ReplyMessageBox. When the request is failed, i.e. when the Status is other than "Success" (0000h), this parameter SHALL NOT be returned.

Byte index	Description
0-1	ProtocolVersion. "0100h" SHALL be set for ECC 224-bit keys. "0200h" SHALL be set for ECC 384-bit keys.
2-3	MessageID. "0102h" SHALL be set for this specification.
4-5	Status.
6-7	NumberOfReplyMessageBoxes.

Byte index	Description
8-(8+ size of ReplyMessageBox List -1)	ReplyMessageBoxList. The order of ReplyMessageBox SHALL correspond with that of RequestMessageBox in the Packed Message Request.

1672
1673
1674

The ReplyMessageBox is defined as follows.

Byte index	Description
0-1	ReplyMessageSize.
2-(2+ReplyMessage Size-1)	ReplyMessage.

1675
1676
1677
1678
1679
1680
1681
1682

Status values are below.

Values	Details
0000h	Success.
8001h	Error other than the below.
8002h	Version error.
8009h	Message format error.

Table 4-9: Status value of Packed Message Reply

1683

1684 **4.2.4 Processing Rules**

1685 The Marlin IPTV-ES Server SHALL send a Reply message that corresponds to the
1686 MessageID of the Request message received from a Marlin IPTV-ES Device.
1687 When the Marlin IPTV-ES Server has only the capability of handling requests for
1688 Simple Key Delivery, the Marlin IPTV-ES Server SHALL send a Get Permission
1689 Reply message with the Status of "Error other than below" (8001h) if the MessageID
1690 of the Request message is other than "Get Permission Request message" (0001h)
1691 and "Packed Message Request message" (0101h). Otherwise, the Marlin IPTV-ES
1692 Server SHALL send a Get Permission Reply message with the Status of "Error other
1693 than below" (8001h) if the MessageID of the Request message is other than the
1694 values defined in sections 4.2.1.2, 4.2.2.2 and 4.2.3.2.

1695
1696
1697
1698
1699
1700
1701
1702
1703
1704

The following subsections define the how Marlin IPTV-ES Service Protocol messages are verified by the Marlin IPTV-ES Server and/or the Marlin IPTV-ES Device. Whenever receiving a Marlin IPTV-ES Service Protocol message, the Marlin IPTV-ES Server and the Marlin IPTV-ES Device SHALL verify them in accordance with the processing rules defined in the following subsections. Note that, if not explicitly stated, verifications defined in this section SHALL be deemed as "verification succeeded" when it does not fall under the condition of "verification failure".

1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757

4.2.4.1 Get Permission Request parameters

Whenever receiving this Get Permission Request message, the Marlin IPTV-ES Server SHALL verify its parameters as shown below.

- *ProtocolVersion*
 - If ProtocolVersion is other than “0100h” or “0200h”, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Version error” (8002h).
- *DeviceInformation*
 - If Marlin IPTV-ES SpecificationVersionMajor is other than “01h” or if Marlin IPTV-ES SpecificationVersionMinor is other than “00h”, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Device Information error” (8003h).
- *UsageRuleReference*
 - If Marlin IPTV-ES Server does not permit the request for Content Key or Work Key, which corresponds to this UsageRuleReference, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Action denied” (8007h).
- *ActionID*
 - If ActionID is other than the value defined section 4.2.1.2 or if it is a value of request which the Marlin IPTV-ES Server does not have the capability of handling, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “ActionID error” (8005h).
- *ActionParameter*
 - Except for the case when ActionID is set to “EXTRACT with Simple Key Delivery” (01h), “EXTRACT with Indirect Key Delivery” (02h), or “EXTRACT with Direct Key Delivery” (03h), the Marlin IPTV-ES Server SHALL verify its corresponding ActionParameter. If ActionParameter other than the value defined in section 4.2.1.2 or if it is a value of request which the Marlin IPTV-ES Server does not have the capability of handling, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “ActionParameter error” (8006h).
- *SpecificCRID*
 - The Marlin IPTV-ES Server SHALL verify this SpecificCRID in accordance with Marlin IPTV-ES specific Compliance Rules, and if “verification failure” occurs, the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Error other than below” (8001h).
- *PrivateDataTag*
 - The Marlin IPTV-ES Server SHALL verify this PrivateDataTag in accordance with Marlin IPTV-ES specific Compliance Rules, and if “verification failure” occurs, the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Error other than below” (8001h).
- *PrivateData*
 - When PrivateDataTag is set to “00h”, the Marlin IPTV-ES Server SHALL verify this PrivateData, and if any byte of PrivateData is other than “00h”, the verification SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Error other than below” (8001h).

1758 **4.2.4.2 Get Permission Reply parameters**

1759 Whenever receiving this Get Permission Reply message, the Marlin IPTV-ES Device
1760 SHALL verify its parameters as shown below.

- 1761
- 1762 • *ProtocolVersion*
 - 1763 ➤ If ProtocolVersion is other than “0100h” or “0200h”, the verification SHALL
 - 1764 be deemed as “verification failure”.
 - 1765 • *Status*
 - 1766 ➤ If Status is other than the value defined in Table 4-6, the verification
 - 1767 SHALL be deemed as “verification failure”.
- 1768

1769 **4.2.4.3 StatusExtension for “EXTRACT with Simple Key Delivery”**

1770 Whenever receiving a Reply message with this StatusExtension, the Marlin IPTV-ES
1771 Device SHALL verify its parameters as shown below.

- 1772
- 1773 • *ExtractInfoSize*
 - 1774 ➤ If ExtractInfoSize is other than “000Ah”, the verification SHALL be
 - 1775 deemed as “verification failure”.
- 1776

1777 **4.2.4.4 StatusExtension for “EXTRACT with Indirect Key Delivery”**

1778 Whenever receiving a Get Permission Reply message with this StatusExtension, the
1779 Marlin IPTV-ES Device SHALL verify its parameters as shown below.

- 1780
- 1781 • *WorkKeyID*
 - 1782 ➤ If the third significant byte of WorkKeyID, i.e. ReservedByte, is other than
 - 1783 “00h”, the verification SHALL be deemed as “verification failure”.
 - 1784 • *ExtractInfoSize*
 - 1785 ➤ Marlin IPTV-ES Device SHALL verify the ExtractInfoSize as specified in
 - 1786 section 4.2.4.3.
- 1787

1788 In addition to the verification described above, the Marlin IPTV-ES Device SHALL
1789 update its retaining set of WorkKey, WorkKeyID, SubscriptionTierBits, PrivateData
1790 and ExtractInfo with the received set of them when receiving a Reply message with a
1791 Work Key which the values of the following parameters are the same as the retaining
1792 ones, provided that Marlin IPTV-ES Device MAY skip to update the retaining
1793 parameters when they are equivalent to the received ones.

- 1794
- 1795 • ServiceProviderID
 - 1796 • ReservedByte
 - 1797 • WorkKeyManagementID
 - 1798 • Odd/Even of Work Key (the LSB value of WorkKeyVersion)
- 1799

1800 **4.2.4.4.1 ExtractInfo**

1801 Whenever receiving this ExtractInfo within a reply against a request with ActionID of
1802 “EXTRACT with Indirect Key Delivery” (02h), the Marlin IPTV-ES Device SHALL
1803 verify its parameters as shown below.

- 1804
- 1805 • *NotBefore/NotAfter*
 - 1806 ➤ If NotBefore is equal to or larger than NotAfter, except for cases when
 - 1807 NotBefore is “FFFFFFFFh”, the verification SHALL be deemed as

1808 “verification failure”.
1809 ➤ If NotBefore or NotAfter is “00000000h”, the verification SHALL be
1810 deemed as “verification failure”.
1811

1812 **4.2.4.5 StatusExtension for “EXTRACT with Direct Key Delivery”**

1813 Whenever receiving a Get Permission Reply message with this StatusExtension, the
1814 Marlin IPTV-ES Device SHALL verify its parameters as shown below.
1815

- 1816 • *ExtractInfoSize*
 - 1817 ➤ Marlin IPTV-ES Device SHALL verify the ExtractInfoSize as specified in
1818 section 4.2.4.3.
1819

1820 **4.2.4.5.1 ExtractInfo**

1821 Whenever receiving this ExtractInfo within a reply against a request with ActionID of
1822 "EXTRACT with Direct Key Delivery" (03h), the Marlin IPTV-ES Device SHALL verify
1823 its parameters as shown below.
1824

- 1825 • *NotBefore/NotAfter*
 - 1826 ➤ Marlin IPTV-ES Device SHALL verify NotBefore and NotAfter as specified
1827 in section 4.2.4.4.1.

1828 In addition to the verification described above, the Marlin IPTV-ES Device SHALL
1829 verify whether values of NotBefore and NotAfter satisfy the following conditions by
1830 using its Trusted Time. If and only if this verification succeeds, the Content Key
1831 related to the set of NotBefore and NotAfter is deemed to be valid.
1832

- 1833 • The value of NotBefore is no larger than the Trusted Time that the Marlin
1834 IPTV-ES Device retains, except for when this value is “FFFFFFFFh”.
- 1835 • The value of NotAfter is no smaller than the Trusted Time that the Marlin
1836 IPTV-ES Device retains, except for when this value is “FFFFFFFFh”.
1837

1838 **4.2.4.6 StatusExtension for “EXPORT with Direct Key Delivery”**

1839 Whenever receiving a Get Permission Reply message with this StatusExtension, the
1840 Marlin IPTV-ES Device SHALL verify its parameters as shown below.
1841

- 1842 • *ExportInfoSize*
 - 1843 ➤ If ExportInfoSize is other than (Get Permission Reply size – 24 bytes), the
1844 verification SHALL be deemed as “verification failure”.
1845

1846 **4.2.4.7 StatusExtension for “EXPORT with Indirect Key Delivery”**

1847 Whenever receiving a Get Permission Reply message with this StatusExtension, the
1848 Marlin IPTV-ES Device SHALL verify its parameters as shown below.
1849

- 1850 • *WorkKeyID*
 - 1851 ➤ Marlin IPTV-ES Device SHALL verify the third significant byte of
1852 WorkKeyID, i.e. ReservedByte, as specified in section 4.2.4.4.
- 1853 • *ExportInfoSize*
 - 1854 ➤ Marlin IPTV-ES Device SHALL verify the ExportInfoSize as specified in
1855 section 4.2.4.6.
1856

1857 In addition to the verification described above, the Marlin IPTV-ES Device SHALL
1858 update its retaining set of WorkKeyID, PrivateData and ExportInfo for exporting to a
1859 certain media system with the received one for exporting to the same media system
1860 when receiving a Reply message with an ExportInfo which the values of the following
1861 parameters are the same as the retaining ones, provided that Marlin IPTV-ES Device
1862 MAY skip to update the retaining parameters when they are equivalent to the
1863 received ones.

1864
1865
1866
1867
1868
1869

- ServiceProviderID
- ReservedByte
- WorkKeyManagementID
- Odd/Even of Work Key (the LSB value of WorkKeyVersion)

1870 **4.2.4.8 StatusExtension for “RECORD with Indirect Key Delivery”**

1871 Whenever receiving a Get Permission Reply message with this StatusExtension, the
1872 Marlin IPTV-ES Device SHALL verify its parameters as shown below.

1873
1874
1875
1876
1877
1878
1879
1880
1881

- *WorkKeyID*
 - Marlin IPTV-ES Device SHALL verify the third significant byte of WorkKeyID, i.e. ReservedByte, as specified in section 4.2.4.4.
- *RecordInfoSize*
 - If RecordInfoSize is other than “0002h”, the verification SHALL be deemed as “verification failure”.

1882 In addition to the verification described above, the Marlin IPTV-ES Device SHALL
1883 update its retaining set of WorkKeyID, PrivateData and RecordInfo with the received
1884 one when receiving a Reply message with an RecordInfo which the values of the
1885 following parameters are the same as the retaining ones, provided that Marlin IPTV-
1886 ES Device MAY skip to update the retaining parameters when they are equivalent to
1887 the received ones.

1888
1889
1890
1891
1892
1893

- ServiceProviderID
- ReservedByte
- WorkKeyManagementID
- Odd/Even of Work Key (the LSB value of WorkKeyVersion)

1894 **4.2.4.9 Get Trusted Time Request parameters**

1895 Whenever receiving this Get Trusted Time Request message, the Marlin IPTV-ES
1896 Server SHALL verify its parameters as shown below.

1897
1898
1899
1900
1901

- *ProtocolVersion*
 - Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in section 4.2.4.1.

1902 **4.2.4.10 Get Trusted Time Reply parameters**

1903 Whenever receiving this Get Trusted Time Reply message, the Marlin IPTV-ES
1904 Device SHALL verify its parameters as shown below.

1905
1906

- *ProtocolVersion*

- 1907 ➤ Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in
1908 section 4.2.4.2.
1909 • *Status*
1910 ➤ If Status is other than the value defined in Table 4-8, the verification
1911 SHALL be deemed as “verification failure”.
1912

1913 **4.2.4.11 Packed Message Request parameters**

1914 Whenever receiving this Packed Message Request message, the Marlin IPTV-ES
1915 Server SHALL verify its parameters as shown below.
1916

- 1917 • *ProtocolVersion*
 - 1918 ➤ Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in
1919 section 4.2.4.1.
- 1920 • *NumberOfRequestMessageBoxes*
 - 1921 ➤ If NumberOfRequestMessageBoxes is other than the number of
1922 messages packed, the verification SHALL be deemed as “verification
1923 failure” and the Status of the message sent to the Marlin IPTV-ES Device
1924 SHALL be set to “Message format error” (8009h).
- 1925 • *RequestMessageBoxList*
 - 1926 ➤ If RequestMessageBoxList consists of Request messages which the
1927 Marlin IPTV-ES Server has no capability of handling, the verification
1928 SHALL be deemed as “verification failure” and the Marlin IPTV-ES Server
1929 SHALL set the Status of the Reply message to “Message format error”
1930 (8009h).
 - 1931 ➤ If the verification of one or more parameters of Request messages
1932 packed in this RequestMessageBoxList has been deemed to be
1933 “verification failure”, the verification of this RequestMessageBoxList
1934 SHALL be deemed as “verification failure” and the Status of the message
1935 sent to the Marlin IPTV-ES Device SHALL be set to “Message format
1936 error” (8009h). In this case, Marlin IPTV-ES Server SHALL NOT pack
1937 individual replies as ReplyMessageBoxList and SHALL NOT set Status
1938 for individual requests.
- 1939 • *RequestMessageSize*
 - 1940 ➤ If the sum of all RequestMessageSize is other than (Packed Message
1941 Request size – (6 + 2 * NumberOfRequestMessageBoxes)), the
1942 verification SHALL be deemed as “verification failure” and the Status of
1943 the message sent to the Marlin IPTV-ES Device SHALL be set to
1944 “Message format error” (8009h).
1945

1946 **4.2.4.12 Packed Message Reply parameters**

1947 Whenever receiving this Packed Message Reply message, the Marlin IPTV-ES
1948 Device SHALL verify its parameters as shown below.
1949

- 1950 • *ProtocolVersion*
 - 1951 ➤ Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in
1952 section 4.2.4.2.
- 1953 • *Status*
 - 1954 ➤ If Status is other than the value defined in Table 4-9, the verification
1955 SHALL be deemed as “verification failure”.
- 1956 • *NumberOfReplyMessageBoxes*
 - 1957 ➤ If NumberOfReplyMessageBoxes is other than the number of messages
1958 packed, the verification SHALL be deemed as “verification failure”.

- 1959
 - 1960
 - 1961
 - 1962
 - 1963
 - 1964
 - 1965
 - 1966
 - 1967
 - 1968
- *ReplyMessageBoxList*
 - If the verification of one or more parameters of Reply messages packed in this ReplyMessageBoxList has been deemed to be “verification failure”, the verification of this ReplyMessageBoxList SHALL be deemed as “verification failure”.
 - *ReplyMessageSize*
 - If the sum of all ReplyMessageSize is other than (Packed Message Reply size – (8 + 2 * NumberOfReplyMessageBoxes)), the verification SHALL be deemed as “verification failure”.

1969 **5 Marlin IPTV-ES Trust Management**

1970 **5.1 Certificates**

1971 Certificates assert a binding between an identity and a public key. The format of the
1972 certificates used in Marlin IPTV-ES is X.509 v3 defined in [X509]. Except where
1973 otherwise noted the certificate fields SHALL comply with the X.509 specification
1974 defined in [X509] and the IETF PKIX profile defined in [PKIX].
1975

1976 **5.1.1 Certificate Contents**

1977 Typical contents of X.509 certificates used in Marlin IPTV-ES consist of the following
1978 fields:
1979

- 1980 • Version.
- 1981 • Serial Number.
- 1982 • Signature.
- 1983 • Issuer.
- 1984 • Validity.
- 1985 • Subject.
- 1986 • Subject Public Key Information.
- 1987 • Extensions:
 - 1988 ○ Authority Key Identifier
 - 1989 ○ Subject Key Identifier
 - 1990 ○ Key Usage.
 - 1991 ○ Basic Constraints.
 - 1992 ○ CRL Distribution Points
- 1993

1994 **5.1.1.1 Version**

1995 The value of this field MUST be 2, which corresponds to X.509 version 3 Certificates.

1996
1997 `Version ::= INTEGER { v3(2) }`
1998

1999 **5.1.1.2 Signature**

2000 The value of this field SHALL be *EC-DSA with SHA-256*.
2001

2002 **5.1.1.3 Issuer**

2003 The distinguished name of the Issuer MUST be represented with a single directory
2004 name attribute. The attribute type MUST be either a X.500 commonName or a
2005 directory name attribute whose syntax adheres to a URN and is identified by the
2006 object identifier *id-nat-uri*. The latter is the preferred attribute type and it MUST be
2007 used for all CA or end-entity certificates managed outside of the trust authority. This
2008 attribute SHALL be encoded using UTF-8.

2009 cf.
2010 `id-marlin OBJECT IDENTIFIER ::= { iso(1) identified-organization(3) dod(6)`
2011 `internet(1) private(4) enterprise(1) marlin(23727) }`
2012 `id-nemo OBJECT IDENTIFIER ::= { id-marlin nemo(1) }`
2013 `id-nemo-nat OBJECT IDENTIFIER ::= { id-nemo nameAttribute(1) }`
2014 `id-nat-uri OBJECT IDENTIFIER ::= { id-nemo-nat 1 }`
2015
2016

2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036

5.1.1.4 Subject

The distinguished name of the Subject field MUST be represented with a single attribute.

The attribute type for Marlin IPTV-ES Server MUST be either a X.500 commonName or a URI attribute whose object identifier is *id-nat-uri*. The latter is the preferred attribute type and it MUST be used for all CA or end-entity certificates managed outside of the trust authority. This attribute SHALL be encoded using UTF-8.

The attribute type for Marlin IPTV-ES Device MUST be a devid attribute whose syntax adheres to a Device ID defined in [Starfish] §3.2.2 and is identified by the object identifier *id-nat-devid*. This attribute SHALL be encoded using UTF-8.

```
cf.  
id-marlin OBJECT IDENTIFIER ::= {iso(1) identified-organization(3) dod(6)  
internet(1) private(4) enterprise(1) marlin(23727)}  
id-starfish OBJECT IDENTIFIER ::= {id-marlin starfish(3)}  
id-starfish-nat OBJECT IDENTIFIER ::= {id-starfish nameAttribute(1)}  
id-nat-devid OBJECT IDENTIFIER ::= {id-starfish-nat 1}
```

2037
2038
2039
2040

5.1.1.5 Subject Public Key Info

This field carries the public key of the subject and identifies the algorithm with which the key is used. Presently the only supported algorithm is *eccEncryption*.

2041
2042
2043
2044

5.1.2 Certificate Extensions

Marlin IPTV-ES implementation certificate extension fields may include CRL Distribution Points.

2045
2046
2047
2048
2049
2050
2051

5.1.2.1 Authority Key Identifier

This field contains a hash of the issuer's public key.

```
extnID : id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 35 }  
critical : FALSE  
value : hash(PublicKey)
```

The key identifier SHOULD be composed of the 160-bit SHA-1 hash (as defined in [PKIX] §4.2.1.2 method 1) of the value of the bit string *issuerPublicKey* (excluding the tag, length, and number of unused bits). This field is used to enable key changeover.

2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062

5.1.2.2 Subject Key Identifier

This field contains a hash of the subject's public key.

```
extnID : id-ce-subjectKeyIdentifier OBJECT IDENTIFIER ::= { id-ce 14 }  
critical : FALSE  
value : hash(PublicKey)
```

The key identifier is composed of the 160-bit SHA-1 hash (as defined in [PKIX] §4.2.1.2 method 1) of the value of the bit string *subjectPublicKey* (excluding the tag, length, and number of unused bits).

2063
2064
2065
2066
2067
2068
2069

5.1.2.3 Key Usage

The key usage extension defines the purpose for which the key has been certified. For example, it specifies whether a key can be used for signature, certificate signing

2070 and key or data encipherment. The key usage field contains a bit string consisting of
2071 a series of flags, as indicated in [PKIX] §4.2.1.3.

```
2072 extnID : id-ce-keyUsage OBJECT IDENTIFIER ::= { id-ce 15 }  
2073  
2074
```

2075 **5.1.2.4 Basic Constraints**

2076 This field contains the value of the certificate's basic constraints. The basic
2077 constraints extension specifies whether the subject of the certificate is a Certificate
2078 Authority (CA) and in that case the maximum number of CA certificates that can
2079 follow this certificate in a certification path. This profile MUST adhere to the definition
2080 provided in [PKIX] §4.2.1.10.

```
2081 extnID : id-ce-basicConstraints OBJECT IDENTIFIER ::= { id-ce 19 }  
2082  
2083
```

2084 **5.1.2.5 CRL Distribution Points Field**

2085 This field identifies how CRL information is obtained. This profile relies upon an
2086 indirect CRL as described in [PKIX] §5. The CRL Distribution Points field MUST
2087 contain a *DistributionPointName*. This name MUST contain a general name of type
2088 URI. This URI is a pointer to the current CRL and is issued by the entity identified in
2089 *cRLIssuer*.

```
2090  
2091 extnID : id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::= { id-ce 31 }  
2092
```

2093 All implementations SHALL be prepared to resolve an HTTP URL as the URI pointer.
2094 The following is an (non-normative) example of how the cRLDistributionPoints field is
2095 populated:

```
2096  
2097 cRLDistributionPoints:  
2098   DistributionPoint:  
2099     distributionPoint: fullName: uniformResourceIdentifier: http://marlin-  
2101 tmo.com/crl/iptvescrl.crl  
2102     cRLIssuer: directoryName: URI=urn:marlin:datacertification:revocation
```

2103 **5.2 Certificate Revocation List**

2104 A Certificate Revocation List (CRL) is used to convey to a certificate user the set of
2105 revoked certificates. The format of the CRL used in Marlin IPTV-ES is X.509 v2 CRL
2106 defined in [X509]. Except where otherwise noted the CRL fields SHALL comply with
2107 the X.509 specification defined in [X509] and the IETF PKIX profile defined in [PKIX].
2108

2109 **5.2.1 CRL Contents**

2110 Contents of X.509 CRLs used in Marlin IPTV-ES consist of the following fields:

- 2111
- 2112 • Version.
- 2113 • Signature.
- 2114 • Issuer.
- 2115 • ThisUpdate
- 2116 • NextUpdate
- 2117 • Revoked Certificates.
 - 2118 ○ User Certificate.
 - 2119 ○ Revocation Date

- 2120 ○ CRL Entry Extension:
- 2121 ○ Certificate Issuer
- 2122 ● CRL Extensions:
- 2123 ○ Authority Key Identifier
- 2124 ○ CRL Number.
- 2125 ○ Issuing Distribution Point
- 2126

2127 **5.2.1.1 Version**

2128 The value of this field MUST be 1, which corresponds to X.509 version 2 CRL.

2129
2130 `Version ::= INTEGER { v2(1) }`
2131

2132 **5.2.1.2 Signature**

2133 The value of this field SHALL be *EC-DSA with SHA-256*.
2134

2135 **5.2.1.3 Issuer**

2136 The distinguished name of the CRL Issuer MUST be represented with a single
2137 directory name attribute. The attribute type MUST be either a X.500 commonName
2138 or a directory name attribute whose syntax adheres to a URN and is identified by the
2139 object identifier *id-nat-uri*.
2140

2141 **5.2.1.4 CRL Entry Extension**

2142 All mandatory fields for this extension must be present and follow the guidance given
2143 in [PKIX] §5.3.
2144

2145 **5.2.1.5 CRL Extensions**

2146 As previously mentioned, the specification adheres to the [PKIX] CRL profile which
2147 mandates that the fields must be present in the CRL Extensions. Specifically the
2148 *AuthorityKeyIdentifier* and the *CRLNumber* MUST be present. This specification MAY
2149 rely upon an indirect CRL. When an indirect CRL is used, the *issuingDistributionPoint*
2150 extension MUST be present.
2151

2152 **5.2.1.5.1 Issuing Distribution Point**

2153 This field MUST follow the guidance given in [PKIX] §5.2.5. Specifically since the
2154 CRL is an indirect CRL, the *indirectCRL* field MUST be present and MUST have a
2155 value of TRUE.
2156

2157 **5.3 DRL**

2158 A Device Revocation List (DRL) is used to convey to a Service the set of revoked
2159 devices. The format of the DRL is based on X.509 v2 CRL defined in [X509].
2160

2161 **5.3.1 Node and Device IDs**

2162 Nodes and devices are identified by a sequence of npid's, as in [Starfish]. A Device
2163 ID is the identifier of a node at the bottom layer of the HBES tree (layer 15). A node
2164 ID is encoded as a character string whose length is one more than the layer of the

2165 node. That is, nodes at layer 0 are encoded with a single character, while nodes at
2166 layer 15 are encoded with a string of 16 characters. The n^{th} character of the node ID
2167 encoding is a hexadecimal digit ('0'-'f') that encodes the node's layer $n-1$ npid. The
2168 node ID encoding is not case sensitive, so that characters 'A'-'F' may also appear.

2169
2170 For example, a node at layer three whose npid sequence is 1, 1, 10, 5 has the id
2171 encoding: 11a5.
2172

2173 5.3.2 DRL Fields

2174 A Device Revocation List contains the following fields.
2175

Key Tree Name	This identifier MUST be the same as the corresponding field in Marlin Starfish BKB's [Starfish].
Revocation Version	A revocation number. DRLs are issued as an ordered series. Each time one or more additional Nodes are revoked, the Revocation Version of the DRL is incremented by one.
Issued On	The time at which the DRL was issued.
Next Update	The date by which the next DRL will be issued.
Revoked Node Ids	A sequence of the Maximal Completely Revoked Node IDs.
Signature	A signature covering all the fields in DRL.

2176

2177 5.3.3 DRL Format

2178 A Device Revocation List is encoded by X.509 CRL v2 format defined in [X509].
2179

2180 Contents of DRLs consist of the following fields:

- 2181 • Version
 - 2182 • Signature
 - 2183 • Issuer
 - 2184 • ThisUpdate
 - 2185 • NextUpdate
 - 2186 • Revoked Certificates
 - 2187 ○ User Certificate
 - 2188 ○ Revocation Date
 - 2189 ○ CRL Entry Extension:
 - 2190 ○ MaskBitCount
 - 2191 • CRL Extensions:
 - 2192 ○ Authority Key Identifier
 - 2193 ○ CRL Number
 - 2194 ○ Issuing Distribution Point
 - 2195 ○ Key Tree Name
- 2196

2197 5.3.3.1 Version

2198 The value of this field MUST be 1.

2199
2200 `Version ::= INTEGER { v2(1) }`
2201

2202 5.3.3.2 Signature

2203 The value of this field SHALL be *ECDSA with SHA256*. This field corresponds to
2204 Signature field in section 5.3.2.

2205
2206
2207
2208
2209

```
ecdsa-with-Sha256 OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-
    Specified(3) sha256(2) }
```

2210 **5.3.3.3 Issuer**

2211 The distinguished name of the DRL Issuer MUST be represented with a single
2212 directory name attribute. The attribute type MUST be either a X.500 commonName
2213 or a directory name attribute whose syntax adheres to a URN and is identified by the
2214 object identifier *id-nat-uri*.
2215

2216 **5.3.3.4 ThisUpdate**

2217 This field corresponds to Issued On field in section 5.3.2.
2218

2219 **5.3.3.5 NextUpdate**

2220 This field corresponds to Next Update field in section 5.3.2.
2221

2222 **5.3.3.6 User Certificate**

2223 This field corresponds to Revoked Node Ids field in section 5.3.2.
2224

2225 **5.3.3.7 MaskBitCount**

2226 The MaskBitCount extension is a non-standard CRL entry extension. This extension
2227 is used to revoke a batch of devices identified with a HBES Node ID range.

2228
2229
2230

```
id-iptves-drl-maskbitcount OBJECT IDENTIFIER ::= 1.3.6.1.4.1.23727.4.1.1
maskBitCount ::= INTEGER
```

2231
2232
2233

The extension SHALL be marked as critical.

2234
2235
2236

This mask bit count SHALL be a value between 1 and 64 and describes the bit count
of a Device ID that is to be considered in device revocation.

2237

The device revocation proceeds as follows.

2238
2239
2240
2241
2242
2243
2244

1. Convert a mask bit count to a 64-bit mask.
2. Perform a bit-product (AND) operation on the mask and the Device ID in Subject field of a certificate to be checked.
3. Compare the value produced by the operation and the Revoked Node ID contained in the userCertificate field of DRL. If the two values match, the device with this certificate is revoked.

2245
2246

The mask bit format is depicted in Table 5-1.

Mask bit	Mask	Number of Revoked Devices
1	8000000000000000h	8000000000000000h
2	C000000000000000h	4000000000000000h
...
64	FFFFFFFFFFFFFFFFh	1h

Table 5-1 Mask bit format

2247

2248
2249
2250

5.3.3.8 CRL Number

This field corresponds to Revocation Version field in section 5.3.2.

2251

5.3.3.9 Key Tree Name

2252
2253

The KeyTreeName extension is a non-standard CRL extension. This extension is used to identify the key tree name by specifying a URI.

2254
2255
2256

```
id-iptves-drl-keytreename OBJECT IDENTIFIER ::= 1.3.6.1.4.1.23727.4.1.2  
keyTreeName ::= UTF8String
```

2257

2258
2259
2260

The extension SHALL be marked as non-critical. This field corresponds to Key Tree Name field in section 5.3.2.

2261 **6 File Format for Marlin IPTV-ES Content**

2262 **6.1 Standalone Format**

2263 The Standalone Format is compatible with MPEG-2 TS defined in [MP2S]. The
2264 stream is Full or Partial Transport Stream. When an ECM is multiplexed into the
2265 stream, it contains one or more programs. Otherwise, it SHALL contain only one
2266 program. The Timed TS (TTS) which consists of TS packets each of which is
2267 preceded by a 32-bit timestamp is also supported. The time stamp is a binary counter
2268 value counted at 27 MHz frequency. Then the TTS packet size is always 192-byte.
2269

2270 **6.1.1 Stream encryption**

2271 The TS/TTS is encrypted partially as is the case in Conditional Access System (CAS)
2272 defined in [MP2S]. Whether a TS/TTS packet is encrypted is signaled with
2273 transport_scrambling_control bits in the TS packet header. For Simple Key Delivery
2274 or Direct Key Delivery, when a TS/TTS packet is encrypted, the
2275 transport_scrambling_control bits SHALL be set to "10b", otherwise "00b". For
2276 Indirect Key Delivery, the transport_scrambling_control bits SHALL be set to "10b"
2277 when a TS/TTS packet is encrypted with the Scramble Key (even), to "11b" when
2278 encrypted with the Scramble Key (odd), or to "00b" when not encrypted.
2279

2280 The encryption algorithm SHALL be AES with a 128-bit key. The encryption mode is
2281 Cipher Block Chaining (CBC) mode with the residual termination block process
2282 specified in [SCTE52].
2283

2284 The encryption is performed per TS/TTS packet. The IV for the CBC mode SHALL be
2285 a value with all bits equal to zero and this single IV SHALL be applied for all of
2286 TS/TTS packets in the stream.
2287

2288 **6.1.2 ECM format**

2289 The ECM that contains Scramble Keys is typically multiplexed into MPEG-2 TS as
2290 the private section defined in [MP2S], but may also be transmitted alone without
2291 being multiplexed.
2292

2293 When the ECM is multiplexed, the table_id and the section_syntax_indicator of the
2294 private section that carries the ECM SHALL be set to "82h" and "1b", respectively.
2295 The CA_descriptor is always set in the PMT, and the CA_system_ID and the CA_PID
2296 of the CA_descriptor are used to designate the CA system identifier of the Marlin
2297 IPTV-ES and the PID of TS/TTS packets of the ECM.
2298

2299 The ECM contains the following parameters.
2300

- 2301 • *ProtocolVersion*: The identification of a protocol that processes the ECM.
- 2302 • *WorkKeyID*: The identifier of the Work Key that decrypts the ECM. The value
2303 of WorkKeyID of the Work Key used to decrypt the ECM SHALL be set.
- 2304 • *Datetime*: Current date and time. The value is specified as 32-bit unsigned
2305 integer value, representing the number of minutes elapsed since January 1,
2306 1970 00:00:00. The value is a UTC date. The first Datetime (byte index of 18
2307 to 21) SHALL be set to the time and date of ECM delivery. The second
2308 Datetime (byte index of 50 to 53) SHALL be set to the same value of the first
2309 Datetime.

- 2310 • *ChannelTierBits*: The first ChannelTierBits (byte index of 22 to 29) SHALL be
- 2311 set to the value of bit strings that specify the subscription to which the
- 2312 Channel carrying the ECM belongs. The second ChannelTierBits (byte index
- 2313 of 54 to 61) SHALL be set to the same value of the first ChannelTierBits.
- 2314 • *RenderingObligation*: The first RenderingObligation (byte index of 30 to 31)
- 2315 SHALL be set to the value of Output Control Information defined in section
- 2316 4.2.1.4.1. The second RenderingObligation (byte index of 62 to 63) SHALL be
- 2317 set to the same value of the first RenderingObligation.
- 2318 • *ScrambleKey (odd/even)*: ScrambleKeys (odd/even) that decrypt TS/TTS
- 2319 packets.
- 2320

2321 The Datetime, ChannelTierBits, RenderingObligation, PrivateData (except the first
 2322 one with byte index of 8 to 17) and both ScrambleKey (odd and even) are encrypted
 2323 with the Work Key. The encryption algorithm SHALL be AES (128bits), and the
 2324 encryption mode used is the CBC mode. The IV for the CBC mode SHALL be a
 2325 value with all bits equal to zero. If a fraction is produced, the OFB mode SHALL be
 2326 used as in [SCTE52].
 2327

Byte index	Description
0-1	ProtocolVersion. "0100h" SHALL be set for ECC 224-bit keys. "0200h" SHALL be set for ECC 384-bit keys
2-7	WorkKeyID.
8-17	PrivateData. "00h" SHALL be set to all 10 bytes for this specification.
18-21	Datetime.
22-29	ChannelTierBits.
30-31	RenderingObligation.
32-33	PrivateData. "00h" SHALL be set to all 2 bytes for this specification.
34-49	ScrambleKey (odd).
50-53	Datetime.
54-61	ChannelTierBits.
62-63	RenderingObligation.
64-65	PrivateData. "00h" SHALL be set to all 2 bytes for this specification.
66-81	ScrambleKey (even).
82-97	PrivateData. "00h" SHALL be set to all 16 bytes for this specification.

2328

2329 6.1.3 Processing Rules of ECM

2330 If not explicitly stated, verifications defined in this section SHALL be deemed as
 2331 "verification succeeded" when it does not fall under the condition of "verification
 2332 failure".
 2333

2334 Whenever receiving this ECM, the Marlin IPTV-ES Device SHALL process the ECM
 2335 as following:
 2336

- 2337 • The Marlin IPTV-ES Device SHALL verify whether values of NotBefore and
- 2338 NotAfter of the set of WorkKey, WorkKeyID, SubscriptionTierBits, PrivateData
- 2339 and ExtractInfo, which is specified by the WorkKeyID within the ECM,

2340 satisfies the following conditions by using its Trusted Time. If and only if this
 2341 verification succeeds, the Marlin IPTV-ES Device MAY decrypt the ECM
 2342 using the relative Work Key.

- 2343 ➤ The value of NotBefore is no larger than the Trusted Time that the Marlin
 2344 IPTV-ES Device retains, except for when this value is "FFFFFFFFh".
- 2345 ➤ The value of NotAfter is no smaller than the Trusted Time that the Marlin
 2346 IPTV-ES Device retains, except for when this value is "FFFFFFFFh".
- 2347 • After decrypting the ECM, the Marlin IPTV-ES Device SHALL verify its
 2348 parameters as shown below. If "verification failure" occurs, the Marlin IPTV-
 2349 ES Device SHALL NOT proceed the process described hereinafter.
 - 2350 ➤ *ProtocolVersion*
 - 2351 ✧ Marlin IPTV-ES Device SHALL verify the ProtocolVersion as specified
 2352 in section 4.2.4.2.
 - 2353 ➤ *Datetime*
 - 2354 ✧ If Datetime is other than the other (byte index of 18 to 21 and 50 to 53,
 2355 respectively), the verification SHALL be deemed as "verification
 2356 failure".
 - 2357 ➤ *ChannelTierBits*
 - 2358 ✧ If ChannelTierBits is other than the other (byte index of 22 to 29 and
 2359 54 to 61, respectively), the verification SHALL be deemed as
 2360 "verification failure".
 - 2361 ➤ *RenderingObligation*
 - 2362 ✧ If RenderingObligation is other than the other RenderingObligation
 2363 (byte index of 30 to 31 and 62 to 63, respectively), the verification
 2364 SHALL be deemed as "verification failure".
- 2365 • After succeeding the verification of each parameters of the ECM, the Marlin
 2366 IPTV-ES Device SHALL logically multiply (perform AND operation) each bit of
 2367 ChannelTierBits within the ECM and SubscriptionTierBits of the Work Key
 2368 used to decrypt the ECM. If one or more logical multiplications of each bit are
 2369 "1b", the Marlin IPTV-ES Device MAY decrypt the content using the Scramble
 2370 Keys.
- 2371 • After decrypting the content, the Marlin IPTV-ES Device SHALL control the
 2372 consumption of the content in accordance with RenderingObligation obtained
 2373 as an ECM, or with ExportInfo or RecordInfo corresponding to the WorkKeyID
 2374 of the Work Key used to decrypt the ECM.

2376 **6.2 Interoperable Format**

2377 The Interoperable Format is a subset of the Marlin BC Content Format defined in
 2378 [MFF]. The Interoperable Format applies the following restrictions on the Marlin BC
 2379 Content Format:

- 2380
- 2381 • Support for CBC mode only.
- 2382 • Support for only one content ID/License per file.
- 2383 • No IV update through a content stream.
- 2384 • No PI packet included in the content stream.
- 2385

2386 In the interoperable format, the TS/TTS is carried in the hierarchical box structure as
 2387 defined in the Marlin BC Content Format. The box syntax is defined in [ISOMFF].
 2388 Marlin IPTV-ES Devices MAY recognize and process all kinds of boxes and their
 2389 content. However, they SHOULD be able to process *size* and *type* fields of top-level
 2390 boxes to access the TS/TTS in the box structure. The top-level box carrying a
 2391 TS/TTS is the Media Data box, which box type is 'mdat'. The Marlin IPTV-ES
 2392 Devices can reach the box by skipping other top-level boxes which precede the

2393 Media Data box. When the Media Data box is found, the TS/TTS is located after *size*,
2394 *type* and occasionally *largesize* fields in the box. The TS/TTS length is calculated
2395 with the box size.
2396

2397 **Appendix A Profiles (Normative)**

2398 This section specifies profiles for Marlin IPTV-ES Devices and Marlin IPTV-ES
 2399 Servers based on ActionID of Get Permission Request message which they have the
 2400 capability of handling. The profiles are defined by using mandatory/optional tables. In
 2401 the table, 'M', 'O', and 'N/A' represent mandatory, optional, and not applicable
 2402 function, respectively. When the Marlin IPTV-ES Devices or Servers have the
 2403 capability of handling an ActionID specified on the horizontal axis (e.g. "01h"), they
 2404 SHALL support the mandatory functions specified on the vertical axis (e.g. "Protocol
 2405 Sequence" defined in section 4.1.1), and also support the optional functions where
 2406 applicable (e.g. "Request message" defined in section 4.1.3.5). In the case that the
 2407 Marlin IPTV-ES Devices/Servers support multiple ActionID, their profile is applied in
 2408 logical disjunctive manner ("OR operation"), which means a function is mandatory if
 2409 the function is specified as mandatory for at least one supported ActionID.
 2410 Note that "w/o TM" indicates that supporting of TransactionFlag Management is not
 2411 required while "w/ TM" indicates that supporting of TransactionFlag Management is
 2412 required.

2413 **A.1 SAC Protocol**

2414 **A.1.1 Profile for Marlin IPTV-ES Devices**

2415

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Protocol sequence	4.1.1	M	M	M	M	M	M	M	M	M	M
Crypto algorithm	4.1.2	M	M	M	M	M	M	M	M	M	M
Message header and payload	4.1.3.1, 4.1.4.1	M	M	M	M	M	M	M	M	M	M
Challenge message	4.1.3.2	M	M	M	M	M	M	M	M	M	M
Response & Challenge message	4.1.4.3	M	M	M	M	M	M	M	M	M	M
Response & Request message	4.1.3.4	M	M	M	M	M	M	M	M	M	M
Request message	4.1.3.5	O ¹	O ¹	O ¹	O ¹	O ¹	O ¹	O ¹	O ¹	O ¹	O ¹
Reply message	4.1.4.6	M	M	M	M	M	M	M	M	M	M
Plain command message	4.1.4.7	M	M	M	M	M	M	M	M	M	M

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Encrypted command message	4.1.3.8, 4.1.4.8	M	M	M	M	M	M	M	M	M	M
Response & Commit message	4.1.3.9	N/A	M	N/A	M	M	N/A	N/A	M	M	M
Transaction Flag Processing	4.1.4.11.1	N/A	M	N/A	M	M	N/A	N/A	M	M	M
URI signature verification	4.1.4.12	M	M	M	M	M	M	M	M	M	M
CRL Processing	4.1.4.14	M	M	M	M	M	M	M	M	M	M

2416 ¹ Marlin IPTV-ES Devices SHALL be able to send Request message if and only if it
2417 needs to send two or more requests over one Marlin IPTV-ES SAC session.
2418

2419 **A.1.2 Profile for Marlin IPTV-ES Servers**

2420

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Protocol sequence	4.1.1	M	M	M	M	M	M	M	M	M	M
Crypto algorithm	4.1.2	M	M	M	M	M	M	M	M	M	M
Message header and payload	4.1.3.1, 4.1.4.1	M	M	M	M	M	M	M	M	M	M
Challenge message	4.1.4.2	M	M	M	M	M	M	M	M	M	M
Response & Challenge message	4.1.3.3	M	M	M	M	M	M	M	M	M	M
Response & Request message	4.1.4.4	M	M	M	M	M	M	M	M	M	M
Request message	4.1.4.5	M	M	M	M	M	M	M	M	M	M
Reply message	4.1.3.6	M	M	M	M	M	M	M	M	M	M

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Plain command message	4.1.3.7	M	M	M	M	M	M	M	M	M	M
Encrypted command message	4.1.3.8, 4.1.4.8	M	M	M	M	M	M	M	M	M	M
Response & Commit message	4.1.4.9	N/A	M	N/A	M	M	N/A	N/A	M	M	M
Transaction Flag Processing	4.1.4.11.2	N/A	M	N/A	M	M	N/A	N/A	M	M	M
URI signature verification	4.1.4.12	M	M	M	M	M	M	M	M	M	M
DRL Processing	4.1.4.13	M	M	M	M	M	M	M	M	M	M

2421

2422 **A.2 Service Protocol**

2423 **A.2.1 Profile for Marlin IPTV-ES Devices**

2424

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Get Permission support (EXTRACT with Simple Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.3	M	M	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Get Permission support (EXTRACT with Indirect Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.4, 4.2.4.4.1	N/A	N/A	M	M	M	M	N/A	N/A	N/A	N/A

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Get Permission support (EXTRACT with Direct Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.5, 4.2.4.5.1	N/A	N/A	N/A	N/A	N/A	N/A	M	M	N/A	N/A
Get Permission support (EXPORT for Copy with Direct Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.6	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	M	N/A
Get Permission support (EXPORT for Move with Direct Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.6	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	M
Get Permission support (EXPORT with Indirect Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.7	N/A	N/A	N/A	N/A	M	N/A	N/A	N/A	N/A	N/A
Get Permission support (RECORD with Indirect Key Delivery)	4.2.1.2, 4.2.4.2, 4.2.4.8	N/A	N/A	N/A	N/A	N/A	M	N/A	N/A	N/A	N/A
Get Trusted Time support	4.2.2.2, 4.2.4.10	O ²	O ²	O	O	O	O	O	O	O	O
Packed Message support	4.2.3.2, 4.2.4.12	O	O	O	O	O	O	O	O	O	O

2425
2426
2427
2428

² Marlin IPTV-ES Devices SHALL be able to send Get Trusted Time Request message if and only if the message is packed in Packed Message Request message.

2429
2430

A.2.2 Profile for Marlin IPTV-ES Servers

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Get Permission support (EXTRACT with Simple Key Delivery)	4.2.1.3, 4.2.1.4, 4.2.1.4.1, 4.2.4.1	M	M	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Get Permission support (EXTRACT with Indirect Key Delivery)	4.2.1.3, 4.2.1.4.1, 4.2.1.5, 4.2.1.5.1, 4.2.4.1	N/A	N/A	M	M	M	M	N/A	N/A	N/A	N/A
Get Permission support (EXTRACT with Direct Key Delivery)	4.2.1.3, 4.2.1.4.1, 4.2.1.6, 4.2.4.1	N/A	N/A	N/A	N/A	N/A	N/A	M	M	N/A	N/A
Get Permission support (EXPORT for Copy with Direct Key Delivery)	4.2.1.3, 4.2.1.7, 4.2.4.1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	M	N/A
Get Permission support (EXPORT for Move with Direct Key Delivery)	4.2.1.3, 4.2.1.7, 4.2.4.1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	M

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Get Permission support (EXPORT with Indirect Key Delivery)	4.2.1.3, 4.2.1.5.1, 4.2.1.8, 4.2.4.1	N/A	N/A	N/A	N/A	M	N/A	N/A	N/A	N/A	N/A
Get Permission support (RECORD with Indirect Key Delivery)	4.2.1.3, 4.2.1.4.1, 4.2.1.5.1, 4.2.1.9, 4.2.4.1	N/A	N/A	N/A	N/A	N/A	M	N/A	N/A	N/A	N/A
Get Trusted Time support	4.2.2.3, 4.2.4.9	M ³	M ³	M	M	M	M	M	M	M	M
Packed Message support	4.2.3.3, 4.2.4.11	M	M	M	M	M	M	M	M	M	M
General message processing rules	4.2.4	M	M	M	M	M	M	M	M	M	M

2431 ³ Servers SHALL accept Get Trusted Time Request message if and only if the
2432 message is packed in Packed Message Request message.
2433

2434 **A.3 File Format**

2435 **A.3.1 Profile for Marlin IPTV-ES Devices**

2436

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Standalone Format	6.1	M	M	M	M	M	M	M	M	M	M
Stream encryption	6.1.1	M	M	M	M	M	M	M	M	M	M
ECM format	6.1.3	N/A	N/A	M	M	M	M	N/A	N/A	N/A	N/A

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Interoperable Format	6.2	O	O	O	O	O	O	O	O	O	O

2437

2438 **A.3.2 Profile for Marlin IPTV-ES Servers**

2439

Function	Reference	Key Delivery Scheme/Action ID									
		Simple Key Delivery		Indirect Key Delivery				Direct Key Delivery			
		01h w/o TM	01h w/ TM	02h w/o TM	02h w/ TM	12h	20h	03h w/o TM	03h w/ TM	10h	11h
Standalone Format	6.1	M	M	M	M	M	M	M	M	M	M
Stream encryption	6.1.1	M	M	M	M	M	M	M	M	M	M
ECM format	6.1.2	N/A	N/A	M	M	M	M	N/A	N/A	N/A	N/A
Interoperable Format	6.2	O	O	O	O	O	O	O	O	O	O

2440

2441

2442

Note that the profiles in this section MAY apply to servers other than Marlin IPTV-ES Server if they process these functions in this section.