# Marlin IPTV End-point Service Specification

Version 2.0.0

Final

| | |
|---|---|
| Source | Marlin Developer Community |
| Date | January 16, 2026 |


Refer to Notices on page 2 for important legal information

## Notice

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

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

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



## Intellectual Property

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

## Contact Information

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

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

# Contents


1 Introduction ..... 4
1.1 Document Organization ..... 4
1.2 Terminology and Conventions ..... 4
1.3 Abbreviations ..... 4
1.4 Terms and Definitions ..... 5
1.5 References ..... 6
1.5.1 Normative References ..... 6
1.6 Bit/Byte ordering ..... 7
2 Marlin IPTV-ES System entities (Informative) ..... 8
2.1 Marlin IPTV-ES Device ..... 8
2.2 Marlin IPTV-ES Server ..... 8
3 Architecture of Marlin IPTV End-point Service ..... 9
3.1 Architecture (Informative) ..... 9
3.2 Marlin IPTV-ES Device ..... 9
3.2.1 Functions ..... 9
3.2.2 Credentials and Device Information ..... 9
3.3 Marlin IPTV-ES Server ..... 10
3.3.1 Functions ..... 10
3.3.2 Credentials ..... 11
4 Marlin IPTV-ES SAC and Marlin IPTV-ES Service Protocols ..... 12
4.1 Secure Authenticated Channel (SAC) Protocol ..... 12
4.1.1 Protocol overview ..... 12
4.1.2 Crypto Algorithm ..... 15
4.1.3 Protocol ..... 16
4.1.4 Processing Rules ..... 25
4.2 Marlin IPTV-ES Service Protocols over SAC ..... 38
4.2.1 Get Permission Protocol ..... 38
4.2.2 Get Trusted Time Protocol ..... 46
4.2.3 Packed Message Protocol ..... 47
4.2.4 Processing Rules ..... 49
5 Marlin IPTV-ES Trust Management ..... 56
5.1 Certificates ..... 56
5.1.1 Certificate Contents ..... 56
5.1.2 Certificate Extensions ..... 57
5.2 Certificate Revocation List ..... 58
5.2.1 CRL Contents ..... 58
5.3 DRL ..... 59
5.3.1 Node and Device IDs ..... 59
5.3.2 DRL Fields ..... 60
5.3.3 DRL Format ..... 60
6 File Format for Marlin IPTV-ES Content ..... 63
6.1 Standalone Format ..... 63
6.1.1 Stream encryption ..... 63
6.1.2 ECM format ..... 63
6.1.3 Processing Rules of ECM ..... 64
6.2 Interoperable Format ..... 65
Appendix A Profiles (Normative) ..... 67
A.1 SAC Protocol ..... 67
A.2 Service Protocol ..... 69
A.3 File Format ..... 72

# 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. |

## *1.5 References*

### 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, *Key words for use in RFCs to Indicate Requirement Levels*, 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. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. 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] | *ITU-T Recommendation X.509* (1997 E): *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework,* June 1997. |
| [X509Cor1] | *ITU-T Recommendation X.509 (2000) Corrigendum 1: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks Technical Corrigendum 1* |

## *1.6 Bit/Byte ordering*

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.
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.

# 2 Marlin IPTV-ES System entities (Informative)

## *2.1 Marlin IPTV-ES Device*

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

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

## *2.2 Marlin IPTV-ES Server*

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

# 3 Architecture of Marlin IPTV End-point Service

## *3.1 Architecture (Informative)*

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

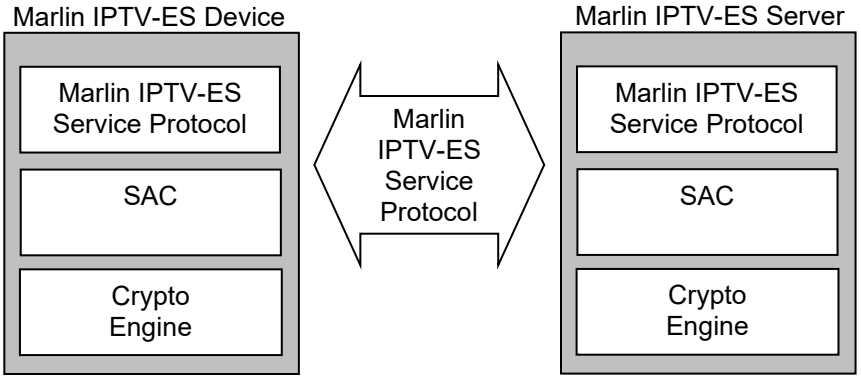


*Figure 1: Architecture of Marlin IPTV-ES Device and Marlin IPTV-ES Server*

## *3.2 Marlin IPTV-ES Device*

### 3.2.1 Functions

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

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

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

### 3.2.2 Credentials and Device Information

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

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

- ✧ bit0: This bit indicates that the client implements and has the secure clock function using the Trusted Time.
- ✧ bit1-bit7: Reserved.

➢ *Manufacturer*: Manufacturer indicates a unique identity of each of manufacturers. The manufacturer-specific value obtained from MTMO SHALL be set.

➢ *ManufacturerModel*: ManufacturerModel indicates an identity of a model in the specified Manufacturer.

➢ *ManufacturerModelVersion*: ManufacturerModelVersion represents the major and minor versions of the specified ManufacturerModel.

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

*Table 3-1: Device Information Encoding*

## *3.3 Marlin IPTV-ES Server*

### 3.3.1 Functions

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

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

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

[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.

### 3.3.2 Credentials

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

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

# 4 Marlin IPTV-ES SAC and Marlin IPTV-ES Service Protocols

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

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

### 4.1.1 Protocol overview

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

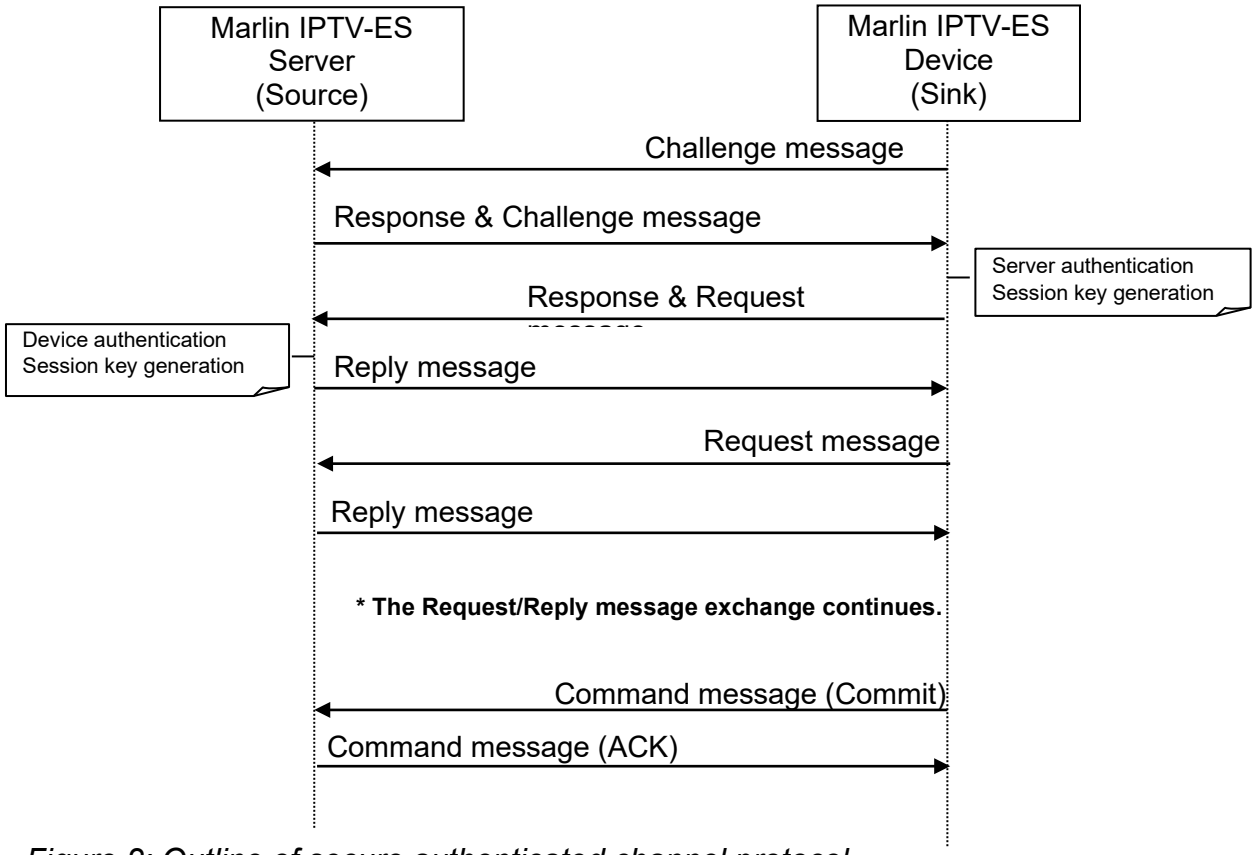


*Figure 2: Outline of secure authenticated channel protocol*

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

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

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

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

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

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

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

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

#### 4.1.1.1 Transaction Flag Management

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

Timings of the status transition are shown in Figure 3.

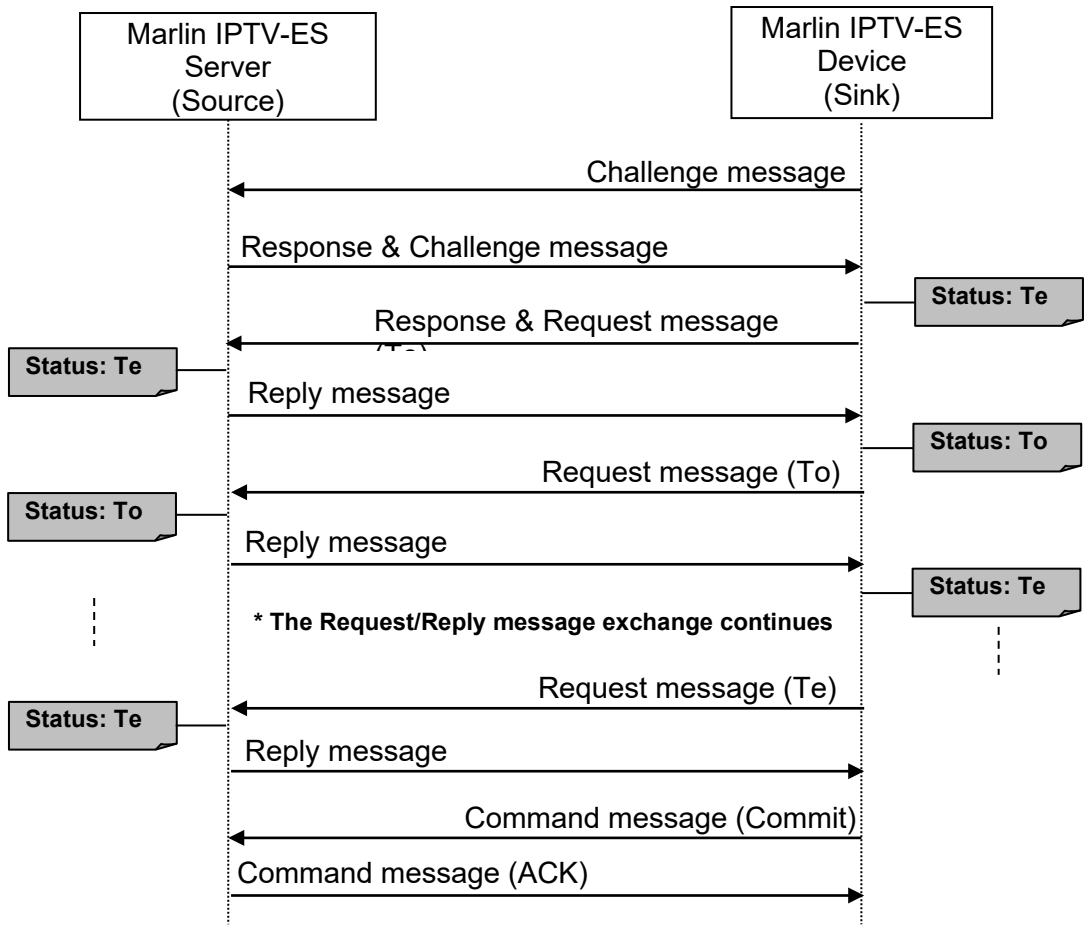


*Figure 3: Status transition of secure authenticated communications*

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

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

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

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

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

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

- Storage of TransactionFlag and ContentKey.
- Deletion of stored TransactionFlag and status change of stored ContentKey.
- Sending of stored TransactionFlag as a Response & Commit message.

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

- Storage of TransactionFlag.
- Deletion of stored TransactionFlag.
- Judgement of Reply message reception.

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

#### 4.1.1.2 URI signature verification

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

### 4.1.2 Crypto Algorithm

#### 4.1.2.1 Crypto Algorithm for ECC 224-bit keys

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

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

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

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

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

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

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

#### 4.1.2.2 Crypto Algorithm for ECC 384-bit keys

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

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

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

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

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

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

## 4.1.3 Protocol

#### 4.1.3.1 Message header and payload

Each message SHALL include the following header.

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

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |

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

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

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

### 4.1.3.2 Challenge message parameters

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

- *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.

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

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

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

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0200h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“0001h” SHALL be set for this specification. |
| 16-19 | PayloadSize. |
| 20-51 | SinkRandomNumber. |
| 52-53 | SinkCertificateSize. |
| 54-(54+SinkCertificateSize-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.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| *4-5* | *ProtocolVersion.*<br>*“0100h” SHALL be set for this specification.* |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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+SourceCertificateSize-1) | SourceCertificate. |

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| *4-5* | *ProtocolVersion.*<br>*“0200h” SHALL be set for this specification.* |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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+SourceCertificateSize-1) | SourceCertificate. |

#### 4.1.3.4 Response & Request message parameters

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

- *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.

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

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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. |

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0200h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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. |

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.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0200h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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.

- *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.

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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. |

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0200h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for ECC 224-bit keys.<br>“0200h” SHALL be set for ECC 384-bit keys. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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.

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

SequenceNumber, TransactionFlag, Command, Status and MessageDigest SHALL be encrypted with the session key as described in section 4.1.2.

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for ECC 224-bit keys.<br>“0200h” SHALL be set for ECC 384-bit keys. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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-<br>(32+MessageDigestSize-1) | MessageDigest.<br>MessageDigestSize = 32 for ECC 224-bit keys.<br>MessageDigestSize = 48 for ECC 384-bit keys. |

Status values are below.

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

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

#### 4.1.3.9 Response & Commit message parameters

This message is used to send Commit command following SAC establishment. Response & Commit message SHALL include the header defined in section 4.1.3.1 followed by the parameters defined below. Definitions of all parameters are the same as section 4.1.3.4 with the exception of parameters defined below.

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

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

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0100h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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. |

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

| Byte index | Description |
|---|---|
| 0-3 | ProtocolID.<br>ASCII value of “IPTV” SHALL be set for this specification. |
| 4-5 | ProtocolVersion.<br>“0200h” SHALL be set for this specification. |
| 6-13 | SenderID. |
| 14-15 | PayloadType.<br>“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. |

## 4.1.4 Processing Rules

A Marlin IPTV-ES Server and a Marlin IPTV-ES Device SHALL verify SAC messages in accordance with the processing rules defined in the following subsections. If the verification defined in sections 4.1.4.1 through 4.1.4.9 resulted in a “verification failure”, the Marlin IPTV-ES Server or the Marlin IPTV-ES Device SHALL terminate the SAC connection as defined in section 4.1.4.10.
Note that verification is considered successful unless it is defined otherwise in the processing rules to follow.

### 4.1.4.1 Message header and payload

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

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

| 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*

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

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

| Combination | 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*

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.

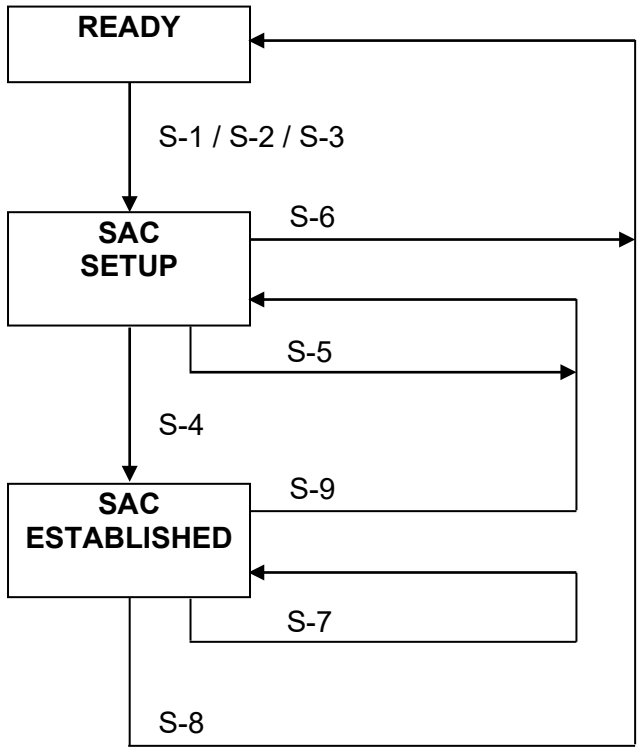


*Figure 4: Valid message combinations of Marlin IPTV-ES Server (Informative)*

| Combination | 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*

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.

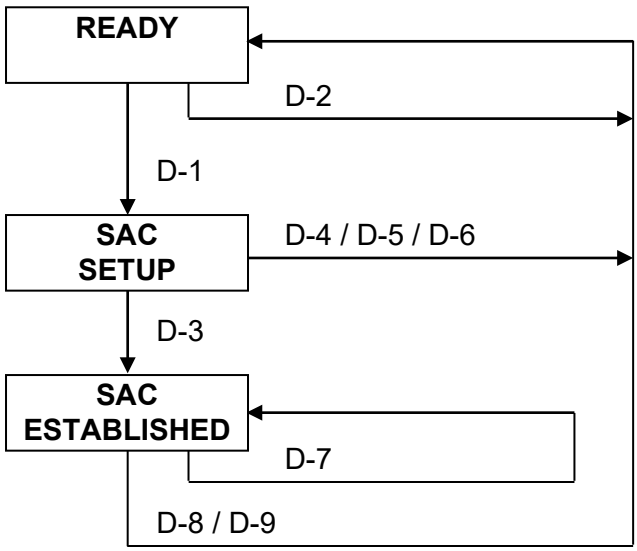


*Figure 5: Valid message combinations of Marlin IPTV-ES Device (Informative)*

#### 4.1.4.2 Challenge message parameters

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

- *SinkCertificateSize*
  - for ECC 224-bit keys
    - If Sink CertificateSize is other than (PayloadSize – 18 bytes), 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 “Message error” (8002h).
  - for ECC 384-bit keys
    - If Sink CertificateSize is other than (PayloadSize – 34 bytes), 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 “Message error” (8002h).
- *SinkCertificate*
  - Using the public keys of the Issuers of the certificates, Marlin IPTV-ES Server SHALL verify the signature of SinkCertificate. If failed, 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 “Authentication error” (8003h).
  - If path length validation is failed, 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 “Authentication error” (8003h).
  - Using a DRL, Marlin IPTV-ES Server SHALL check whether the Marlin IPTV-ES Device has been revoked or not. If the Marlin IPTV-ES Device is determined to have been revoked, 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 “Revoked” (8004h). The Marlin IPTV-ES Server SHALL use a DRL whose Signature is successfully verified in accordance with section 4.1.4.13.

#### 4.1.4.3 Response & Challenge message parameters

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

- *Signature*
  - Marlin IPTV-ES Device SHALL verify the Signature using the SourceCertificate. If failed, the verification SHALL be deemed as "verification failure".
- *SourceCertificateSize*
  - for ECC 224-bit keys
    - If SourceCertificateSize is other than (PayloadSize – 130 bytes), the verification SHALL be deemed as "verification failure".
  - for ECC 384-bit keys
    - If SourceCertificateSize is other than (PayloadSize – 226 bytes), the verification SHALL be deemed as "verification failure".
- *SourceCertificate*
  - Using the public keys of the Issuers of the certificates, Marlin IPTV-ES Device SHALL verify the signature of SourceCertificate. If failed, the verification SHALL be deemed as "verification failure".
  - Marlin IPTV-ES Device SHALL verify the validity of SourceCertificate. If it is expired, the verification SHALL be deemed as "verification failure".
  - If path length validation is failed, the verification SHALL be deemed as "verification failure".
  - Using a CRL, Marlin IPTV-ES Device SHALL check whether the Marlin IPTV-ES Server has been revoked or not. If the Issuer and the serial number of the Source Certificate matches to those listed in the CRL, the Marlin IPTV-ES Server is determined to be revoked, and, therefore, the verification SHALL be deemed as "verification failure". The Marlin IPTV-ES Device SHALL use a CRL whose Signature is successfully verified in accordance with section 4.1.4.14.

#### 4.1.4.4 Response & Request message parameters

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

- *Signature*
  - Marlin IPTV-ES Server SHALL verify the Signature using the SinkCertificate. If failed, 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 "Authentication error" (8003h).
- *EncryptedDataSize*
  - for ECC 224-bit keys
    - If EncryptedDataSize is other than (PayloadSize – 116 bytes), 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 "Message error" (8002h).
  - for ECC 384-bit keys
    - If EncryptedDataSize is other than (PayloadSize – 196 bytes), 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 "Message error" (8002h).
- *SequenceNumber*

- If SequenceNumber is other than 1, 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 "Message error" (8002h).

- *TransactionFlag*
  - If TransactionFlag is not even (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 "Message error" (8002h).
- *MessageDigest*
  - If the hash value of the decrypted Response & Request message excluding this MessageDigest is other than the value of this MessageDigest, 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 "Authentication error" (8003h).

#### 4.1.4.5 Request message parameters

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

- *EncryptedDataSize*
  - If EncryptedDataSize is other than (PayloadSize – 4 bytes), 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 "Message error" (8002h).
- *SequenceNumber*
  - If SequenceNumber is other than the one retaining, or equal to or more than (2^24-3), 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 "Message error" (8002h).
- *TransactionFlag*
  - If TransactionFlag is same as the one retaining, 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 "Message error" (8002h).
- *MessageDigest*
  - If the hash value of the decrypted Request message excluding this MessageDigest is other than the value of this MessageDigest, 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 "Authentication error" (8003h).

#### 4.1.4.6 Reply message parameters

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

- *EncryptedDataSize*
  - If EncryptedDataSize is other than (PayloadSize – 4 bytes), the verification SHALL be deemed as "verification failure".
- *SequenceNumber*
  - If SequenceNumber is other than the one retaining, the verification SHALL be deemed as "verification failure".
- *MessageDigest*
  - If the hash value of the decrypted Reply message excluding this MessageDigest is other than the value of this MessageDigest, the

verification SHALL be deemed as “verification failure”.

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

- *TransactionFlagRecordFlag*
  - ➢ If both of the following two conditions are met, the verification SHALL be deemed as “verification failure”.
    - ✧ Just before receiving the Reply message, the Marlin IPTV-ES Device sent a message (Response & Request message or Request message) which carries either of the following two messages to the Marlin IPTV-ES Server, which is the sender of the Reply message.
      - o Get Permission Request message that requires TransactionFlag Management.
      - o Packed Message Request message which packs one or more Get Permission Request messages that requires TransactionFlag Management.
    - ✧ TransactionFlagRecordFlag is neither “00h” nor “01h”.

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

#### 4.1.4.7 Plain command message parameters

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

- *Command*
  - ➢ If Command is other than “ERROR”, the verification SHALL be deemed as “verification failure”.
- *Status*
  - ➢ If Status is other than “Error other than below” (8001h), “Message error” (8002h), “Authentication error” (8003h) or “Revoked” (8004h), the verification SHALL be deemed as “verification failure”.

#### 4.1.4.8 Encrypted command message parameters

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

- *EncryptedDataSize*
  - ➢ Marlin IPTV-ES Device SHALL verify the EncryptedDataSize as specified in section 4.1.4.6.
- *SequenceNumber*
  - ➢ Marlin IPTV-ES Device SHALL verify the SequenceNumber as specified in section 4.1.4.6.
- *Command*
  - ➢ If Command of this message received after sending an Encrypted Command message to the Marlin IPTV-ES Server is other than “ERROR” or “ACK”, the verification SHALL be deemed as “verification failure”.
  - ➢ If Command of this message received after sending a Response &

Commit message to the Marlin IPTV-ES Server is other than “ACK”, the verification SHALL be deemed as “verification failure”.

- If Command of this message received before sending an Encrypted Command message or a Response & Commit message to the Marlin IPTV-ES Server is other than “ERROR”, the verification SHALL be deemed as “verification failure”.

- *Status*
  - If Status of this message, which its Command is “ERROR”, is other than “Error other than below” (8001h), “Message error” (8002h) or “Authentication error” (8003h), the verification SHALL be deemed as “verification failure”.
  - If Status of this message, which its Command is “ACK”, is other than “Success” (0000h), the verification SHALL be deemed as “verification failure”.

- *MessageDigest*
  - If the hash value of the decrypted Encrypted command message excluding this MessageDigest is other than the value of this MessageDigest, the verification SHALL be deemed as “verification failure”.

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

- *EncryptedDataSize*
  - If EncryptedDataSize is other than (PayloadSize – 4 bytes), 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 “Message error” (8002h).
- *SequenceNumber*
  - If SequenceNumber is other than the one retaining, 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 “Message error” (8002h).
- *Command*
  - If Command is other than “Commit”, 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 “Message error” (8002h).
- *Status*
  - If Status is other than “Success” (0000h), 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 “Message error” (8002h).
- *MessageDigest*
  - If the hash value of the decrypted Encrypted command message excluding this MessageDigest is other than the value of this MessageDigest, 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 “Authentication error” (8003h).

#### 4.1.4.9 Response & Commit message parameters

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

- *Signature*

- Marlin IPTV-ES Server SHALL verify the Signature as specified in section 4.1.4.4.
- *EncryptedDataSize*
  - Marlin IPTV-ES Server SHALL verify the EncryptedDataSize as specified in section 4.1.4.4.
- *SequenceNumber*
  - Marlin IPTV-ES Server SHALL verify the SequenceNumber as specified in section 4.1.4.4.
- *TransactionFlag*
  - If TransactionFlag is neither “00h” nor “01h”, 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 “Message error” (8002h).
- *MessageDigest*
  - If the hash value of the decrypted Response & Commit message excluding this MessageDigest is other than the value of this MessageDigest, 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 “Authentication error” (8003h).

### 4.1.4.10 SAC termination

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

The Marlin IPTV-ES Device SHALL terminate SAC connection after sending an Encrypted command message with a Command of “Commit” or a Response & Commit message respectively to a Marlin IPTV-ES Server, and receiving an Encrypted command message with a Command of “ACK” from the Marlin IPTV-ES Server.

However, in the following four cases, the Marlin IPTV-ES Device SHALL terminate SAC connection without sending an Encrypted command message with a Command of “Commit” or a Response & Commit message respectively to the Marlin IPTV-ES Server.

- “Verification failure” occurred for parameters of SAC messages received from a Marlin IPTV-ES Server in accordance with the processing rules described in section 4.1.4.
- Marlin IPTV-ES Device received a Plain command message or an Encrypted command message, respectively with a Command of “ERROR”, from a Marlin IPTV-ES Server.
- “Verification failure” occurred for Service Protocol parameters of a Reply message which carries either of the following two messages, and of which the TransactionFlagRecordFlag is set to “record” (01h).
  - 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.
- Marlin IPTV-ES Device could not receive and quitted waiting for a message from a Marlin IPTV-ES Server.

When Marlin IPTV-ES Device terminated the SAC connection with a Marlin IPTV-ES Server in events such as accidental power down, the Marlin IPTV-ES Device SHALL

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

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

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

- Marlin IPTV-ES Server received an Encrypted command message with a Command of “Commit” or a Response & Commit message respectively from a Marlin IPTV-ES Device, and sent an Encrypted command message with a Command of “ACK” to the Marlin IPTV-ES Device.
- “Verification failure” occurred for parameters of messages received from a Marlin IPTV-ES Device in accordance with the processing rules described in section 4.1.4 and sent a Plain command message or an Encrypted command message, respectively with a Command of “ERROR”, to the Marlin IPTV-ES Device.
- Marlin IPTV-ES Server could not receive and quitted waiting for messages from Marlin IPTV-ES Device.
- Marlin IPTV-ES Server received a second Challenge message from an already connected Marlin IPTV-ES Device before terminating SAC connection as shown in the three cases above.
  - In this case, Marlin IPTV-ES Server SHALL terminate the SAC connection initiated by the previously received Challenge message before receiving the second Challenge message.

### 4.1.4.11 TransactionFlag

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

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

(1) Storage of TransactionFlag and ContentKey

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

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

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.

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

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 deleting the TransactionFlag and changing the state of the ContentKey.

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

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

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

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

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

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

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

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

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

SHALL be used as the identifier.

(2) Deletion of stored TransactionFlag

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

- The TransactionFlagRecordFlag of the sent Reply message is set to “not to record” (00h).
- The Command of the sent Encrypted command message is set to “ACK”.

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

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

(3) Judgement of Reply message reception

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

- The Marlin IPTV-ES Server SHALL determine that the sent Reply message is received by the Marlin IPTV-ES Device, if it receives either of the following three messages.
  - A Request message.
  - An Encrypted command message with a Command of “Commit”.
  - A Response & Commit message which the TransactionFlag of the received Response & Commit message is same as the stored one.
- The Marlin IPTV-ES Server SHALL determine that the sent Reply message is not received by the Marlin IPTV-ES Device, if it receives either of the following two messages.
  - A Response & Request message.
  - A Response & Commit message which the TransactionFlag of the received Response & Commit message is other than the stored one.

#### 4.1.4.12 URI signature verification

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

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

- 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.

Note that the format of the signature, the format of the information to be signed, and the behaviour of a Marlin IPTV-ES Device based on the result of this signature verification are assumed to be defined outside of this specification.

#### 4.1.4.13 DRL parameters

The Marlin IPTV-ES Server SHALL perform the following process against a DRL used for checking the occurrence of revocation.

- *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.

#### 4.1.4.14 CRL parameters

The Marlin IPTV-ES Device SHALL perform the following process against a CRL used for checking the occurrence of revocation.

- *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.

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

This section defines Marlin IPTV-ES Service Protocols in SAC. Each of Request messages defined in this section corresponds to the Request of Response & Request message defined in section 4.1.3.4 or the Request of Request message defined in section 4.1.3.5.
Each of Response messages defined in this section corresponds to the Reply of Reply message defined in section 4.1.3.6.

### 4.2.1 Get Permission Protocol

#### 4.2.1.1 Overview

This is a simple REQUEST/RESPONSE protocol via SAC. The Marlin IPTV-ES Device requests a certain action to the Marlin IPTV-ES Server. When the REQUEST is permitted, the Marlin IPTV-ES Server returns the Content Key, or the Work Key and its related Work Key ID, and/or other related information as Status Extension. The requested permission is identified with UsageRuleReference which is

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

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

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

#### 4.2.1.2 Get Permission Request parameters

Get Permission Request 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 Request message in this specification.
- *DeviceInformation*: The Device Information of the Marlin IPTV-ES Device defined in section 3.2.2.
- *UsageRuleReference*: The identifier for the requesting usage rules of the content.
- *ActionID*: The identifier of the requested action in the Marlin IPTV-ES Device. In this version of Marlin IPTV-ES specification, the following ActionIDs are defined:
  - 01h: This value indicates that the Marlin IPTV-ES Device requests to "**EXTRACT** with Simple Key Delivery".
  - 02h: This value indicates that the Marlin IPTV-ES Device requests to "**EXTRACT** with Indirect Key Delivery". Once RECORD action and EXTRACT action are permitted for the content, this action is not needed to render the content which is recorded on the Protected Storage.
  - 03h: This value indicates that the Marlin IPTV-ES Device requests to "**EXTRACT** with Direct Key Delivery".
  - 10h: This value indicates that the Marlin IPTV-ES Device requests to "**EXPORT** to a certain Media for Copy with Direct Key Delivery". When this value is specified, the exporting Media type SHALL be specified by ActionParameter.
  - 11h: This value indicates that the Marlin IPTV-ES Device requests to "**EXPORT** to a certain Media for Move with Direct Key Delivery". When this value is specified, the exporting Media type SHALL be specified by

ActionParameter.

- 12h: This value indicates that the Marlin IPTV-ES Device requests to "**EXPORT** to a certain Media with Indirect Key Delivery". When this value is specified, the exporting Media type SHALL be specified by ActionParameter.
- 20h: This value indicates that the Marlin IPTV-ES Device requests to "**RECORD** to a certain Media with Indirect Key Delivery". When this value is specified, the recording Media type SHALL be specified by ActionParameter.

- *ActionParameter*: The parameter pertaining to the requested ActionID.
  - 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), *ActionParameter* SHALL be set to "FFh".
  - For the case when ActionID is set to "EXPORT for Copy with Direct Key Delivery" (10h), "EXPORT for Move with Direct Key Delivery" (11h), or "EXPORT with Indirect Key Delivery" (12h), the parameters defined in [MEXP] SHALL be set, with the exception of the case of Re-Transmission of Digital Broadcasting over IP network in Japan. In such case, *ActionParameter* SHALL be set to the values corresponding to the target copy protection systems in accordance with the rules stated in government or quasi-government regulations.
  - For the case when ActionID is set to "RECORD with Indirect Key Delivery" (20h), the following parameter is defined in this specification:
    - 01h: RECORD to Protected Storage.
- *SpecificCRID*: The identifier of the Marlin IPTV-ES specific Compliance Rules with which the Marlin IPTV-ES Device complies. If no specific compliance rule is applicable, this value SHALL be set to "FFFFh". Otherwise, the value specified in each specific compliance rule SHALL be set.
- *PrivateDataTag*: The identifier of the usage of PrivateData. In this specification, the following PrivateDataTags are defined:
  - 00h: PrivateData not in use. When this value is specified, all bytes of PrivateData SHALL be set to "00h".
  - 01h: Removable media ID.
  - 02h–7Fh: Reserved for common use.
  - 80h–FFh: For private use.

| Byte index | Description |
|---|---|
| 0-1 | ProtocolVersion.<br>"0100h" SHALL be set for ECC 224-bit keys.<br>"0200h" SHALL be set for ECC 384-bit keys. |
| 2-3 | MessageID.<br>"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. |

#### 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.<br>"0100h" SHALL be set for ECC 224-bit keys.<br>"0200h" SHALL be set for ECC 384-bit keys. |
| 2-3 | MessageID.<br>"0002h" SHALL be set for this specification. |
| 4-5 | Status. |
| 6-(6+size of StatusExtension-1) | StatusExtension. |

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*

#### 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+ExtractInfoSize-1) | ExtractInfo. |

##### *4.2.1.4.1 ExtractInfo*

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.

- *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.

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

- 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

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

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

The Output Control Information is followings:

- *DigitalRecordingControlData*: Information of control copy generation and coded, defined as DTCP_CCI in [DTCP] Appendix B, is returned.
- *CopyControlType*: Information whether the output is encoded or not to serial interface is returned. The value of this parameter is defined in Table 4-7.
- *APSControlData*: Information of control analog output copy, defined as APS in [DTCP] Appendix B, is returned.
- *ImageConstraintToken*: Information whether the image quality of video signal output is constrained is returned. The value of this parameter is defined as Image_Constraint_Token in [DTCP] Appendix B.
- *RetentionMode*: Information whether temporal accumulation is possible or not is returned. The value of this parameter is defined as Retention_Move_mode in [DTCP] Appendix B.
- *RetentionState*: Information of the allowable time of temporal accumulation after the reception of contents is returned. The value of this parameter is defined as Retention_State in [DTCP] Appendix B.
- *EncryptionMode*: Information whether the output of high-speed digital interface is protected or not is returned. The value of this parameter is defined as EPN in [DTCP] Appendix B.

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

CopyControlType values are below.

| 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*

#### 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. |

##### *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.

- *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.<br>“00h” SHALL be set for this specification. |
| 3-4 | WorkKeyManagementID. |

| Byte index | Description |
|---|---|
| 5 | WorkKeyVersion. |

#### 4.2.1.6 StatusExtension for "EXTRACT with Direct Key Delivery"

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.

- *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. |

#### 4.2.1.7 StatusExtension for "EXPORT with Direct Key Delivery"

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.

- *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. |

#### 4.2.1.8 StatusExtension for "EXPORT with Indirect Key Delivery"

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.

- *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.

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

#### 4.2.1.9 StatusExtension for "RECORD with Indirect Key Delivery"

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

- *WorkKeyID*: The identifier of WorkKey. The value defined in section 4.2.1.5.1 is returned.
- *RecordInfoSize*: Size of RecordInfo.
- *RecordInfo*: Output Control Information defined in section 4.2.1.4.1 is returned.

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

## 4.2.2 Get Trusted Time Protocol

#### 4.2.2.1 Overview

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

#### 4.2.2.2 Get Trusted Time Request parameters

Get Trusted Time Request 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 Trusted Time Request message in this specification.

| Byte index | Description |
|---|---|
| 0-1 | ProtocolVersion.<br>"0100h" SHALL be set for ECC 224-bit keys.<br>"0200h" SHALL be set for ECC 384-bit keys. |
| 2-3 | MessageID.<br>"0003h" SHALL be set for this specification. |

#### 4.2.2.3 Get Trusted Time Reply parameters

Get Trusted Time 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 Trusted Time Reply message in this specification.
- *Status*: When the request is not authorized/failed for some reasons, the status defined in Table 4-8 is returned.
- *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.

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

Status values are below.

| 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*

### 4.2.3 Packed Message Protocol

#### 4.2.3.1 Overview

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

#### 4.2.3.2 Packed Message Request parameters

Packed Message Request 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 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.<br>“0100h” SHALL be set for ECC 224-bit keys.<br>“0200h” SHALL be set for ECC 384-bit keys. |
| 2-3 | MessageID.<br>“0101h” SHALL be set for this specification. |
| 4-5 | NumberOfRequestMessageBoxes. |
| 6-(6+ size of RequestMessageBoxList -1) | RequestMessageBoxList. |

The RequestMessageBox is defined as follows.

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

#### 4.2.3.3 Packed Message Reply parameters

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.<br>“0100h” SHALL be set for ECC 224-bit keys.<br>“0200h” SHALL be set for ECC 384-bit keys. |
| 2-3 | MessageID.<br>“0102h” SHALL be set for this specification. |
| 4-5 | Status. |
| 6-7 | NumberOfReplyMessageBoxes. |

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

The ReplyMessageBox is defined as follows.

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

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*

### 4.2.4 Processing Rules

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

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”.

#### 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).

#### 4.2.4.2 Get Permission Reply parameters

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

- *ProtocolVersion*
  - If ProtocolVersion is other than “0100h” or “0200h”, the verification SHALL be deemed as “verification failure”.
- *Status*
  - If Status is other than the value defined in Table 4-6, the verification SHALL be deemed as “verification failure”.

#### 4.2.4.3 StatusExtension for “EXTRACT with Simple Key Delivery”

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

- *ExtractInfoSize*
  - If ExtractInfoSize is other than “000Ah”, the verification SHALL be deemed as “verification failure”.

#### 4.2.4.4 StatusExtension for “EXTRACT with Indirect Key Delivery”

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

- *WorkKeyID*
  - If the third significant byte of WorkKeyID, i.e. ReservedByte, is other than “00h”, the verification SHALL be deemed as “verification failure”.
- *ExtractInfoSize*
  - Marlin IPTV-ES Device SHALL verify the ExtractInfoSize as specified in section 4.2.4.3.

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

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

##### *4.2.4.4.1 ExtractInfo*

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

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

"verification failure".
- If NotBefore or NotAfter is "00000000h", the verification SHALL be deemed as "verification failure".

#### 4.2.4.5 StatusExtension for "EXTRACT with Direct Key Delivery"

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

- *ExtractInfoSize*
  - Marlin IPTV-ES Device SHALL verify the ExtractInfoSize as specified in section 4.2.4.3.

##### *4.2.4.5.1 ExtractInfo*

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

- *NotBefore/NotAfter*
  - Marlin IPTV-ES Device SHALL verify NotBefore and NotAfter as specified in section 4.2.4.4.1.

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

- The value of NotBefore is no larger than the Trusted Time that the Marlin IPTV-ES Device retains, except for when this value is "FFFFFFFFh".
- The value of NotAfter is no smaller than the Trusted Time that the Marlin IPTV-ES Device retains, except for when this value is "FFFFFFFFh".

#### 4.2.4.6 StatusExtension for "EXPORT with Direct Key Delivery"

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

- *ExportInfoSize*
  - If ExportInfoSize is other than (Get Permission Reply size – 24 bytes), the verification SHALL be deemed as "verification failure".

#### 4.2.4.7 StatusExtension for "EXPORT with Indirect Key Delivery"

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

- *WorkKeyID*
  - Marlin IPTV-ES Device SHALL verify the third significant byte of WorkKeyID, i.e. ReservedByte, as specified in section 4.2.4.4.
- *ExportInfoSize*
  - Marlin IPTV-ES Device SHALL verify the ExportInfoSize as specified in section 4.2.4.6.

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

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

#### 4.2.4.8 StatusExtension for "RECORD with Indirect Key Delivery"

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

- *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".

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

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

#### 4.2.4.9 Get Trusted Time Request parameters

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

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

#### 4.2.4.10 Get Trusted Time Reply parameters

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

- *ProtocolVersion*

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

#### 4.2.4.11 Packed Message Request parameters

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

- *ProtocolVersion*
  - Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in section 4.2.4.1.
- *NumberOfRequestMessageBoxes*
  - If NumberOfRequestMessageBoxes is other than the number of messages packed, 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 “Message format error” (8009h).
- *RequestMessageBoxList*
  - If RequestMessageBoxList consists of Request messages which the Marlin IPTV-ES Server has no capability of handling, the verification SHALL be deemed as “verification failure” and the Marlin IPTV-ES Server SHALL set the Status of the Reply message to “Message format error” (8009h).
  - If the verification of one or more parameters of Request messages packed in this RequestMessageBoxList has been deemed to be “verification failure”, the verification of this RequestMessageBoxList SHALL be deemed as “verification failure” and the Status of the message sent to the Marlin IPTV-ES Device SHALL be set to “Message format error” (8009h). In this case, Marlin IPTV-ES Server SHALL NOT pack individual replies as ReplyMessageBoxList and SHALL NOT set Status for individual requests.
- *RequestMessageSize*
  - If the sum of all RequestMessageSize is other than (Packed Message Request size – (6 + 2 * NumberOfRequestMessageBoxes)), 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 “Message format error” (8009h).

#### 4.2.4.12 Packed Message Reply parameters

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

- *ProtocolVersion*
  - Marlin IPTV-ES Server SHALL verify the ProtocolVersion as specified in section 4.2.4.2.
- *Status*
  - If Status is other than the value defined in Table 4-9, the verification SHALL be deemed as “verification failure”.
- *NumberOfReplyMessageBoxes*
  - If NumberOfReplyMessageBoxes is other than the number of messages packed, the verification SHALL be deemed as “verification failure”.

- *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”.

# 5 Marlin IPTV-ES Trust Management

## *5.1 Certificates*

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

### 5.1.1 Certificate Contents

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

- Version.
- Serial Number.
- Signature.
- Issuer.
- Validity.
- Subject.
- Subject Public Key Information.
- Extensions:
  - Authority Key Identifier
  - Subject Key Identifier
  - Key Usage.
  - Basic Constraints.
  - CRL Distribution Points

#### 5.1.1.1 Version

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

```
Version ::= INTEGER { v3(2) }
```

#### 5.1.1.2 Signature

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

#### 5.1.1.3 Issuer

The distinguished name of the Issuer MUST be represented with a single directory name attribute. The attribute type MUST be either a X.500 commonName or a directory name attribute whose syntax adheres to a URN and is identified by the object identifier *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.

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

#### 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}
```

#### 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*.

### 5.1.2 Certificate Extensions

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

#### 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.

#### 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).

#### 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

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

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

#### 5.1.2.4 Basic Constraints

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

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

#### 5.1.2.5 CRL Distribution Points Field

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

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

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

```
cRLDistributionPoints:
    DistributionPoint:
        distributionPoint: fullName: uniformResourceIdentifier: http://marlin-tmo.com/crl/iptvescrl.crl
        cRLIssuer: directoryName: URI=urn:marlin:datacertification:revocation
```

## *5.2 Certificate Revocation List*

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

### 5.2.1 CRL Contents

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

- Version.
- Signature.
- Issuer.
- ThisUpdate
- NextUpdate
- Revoked Certificates.
  - User Certificate.
  - Revocation Date

- CRL Entry Extension:
  - Certificate Issuer
- CRL Extensions:
  - Authority Key Identifier
  - CRL Number.
  - Issuing Distribution Point

#### 5.2.1.1 Version

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

```
Version ::= INTEGER { v2(1) }
```

#### 5.2.1.2 Signature

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

#### 5.2.1.3 Issuer

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

#### 5.2.1.4 CRL Entry Extension

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

#### 5.2.1.5 CRL Extensions

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

##### *5.2.1.5.1 Issuing Distribution Point*

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

## *5.3 DRL*

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

### 5.3.1 Node and Device IDs

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

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

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

### 5.3.2 DRL Fields

A Device Revocation List contains the following fields.

| | |
|---|---|
| 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. |

### 5.3.3 DRL Format

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

Contents of DRLs consist of the following fields:

- Version
- Signature
- Issuer
- ThisUpdate
- NextUpdate
- Revoked Certificates
  - User Certificate
  - Revocation Date
  - CRL Entry Extension:
    - MaskBitCount
- CRL Extensions:
  - Authority Key Identifier
  - CRL Number
  - Issuing Distribution Point
  - Key Tree Name

#### 5.3.3.1 Version

The value of this field MUST be 1.

```
Version ::= INTEGER { v2(1) }
```

#### 5.3.3.2 Signature

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

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

#### 5.3.3.3 Issuer

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

#### 5.3.3.4 ThisUpdate

This field corresponds to Issued On field in section 5.3.2.

#### 5.3.3.5 NextUpdate

This field corresponds to Next Update field in section 5.3.2.

#### 5.3.3.6 User Certificate

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

#### 5.3.3.7 MaskBitCount

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

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

The extension SHALL be marked as critical.

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.

The device revocation proceeds as follows.

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.

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*

#### 5.3.3.8 CRL Number

This field corresponds to Revocation Version field in section 5.3.2.

#### 5.3.3.9 Key Tree Name

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

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

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

# 6 File Format for Marlin IPTV-ES Content

## *6.1 Standalone Format*

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

### 6.1.1 Stream encryption

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

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

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

### 6.1.2 ECM format

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

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

The ECM contains the following parameters.

- *ProtocolVersion*: The identification of a protocol that processes the ECM.
- *WorkKeyID*: The identifier of the Work Key that decrypts the ECM. The value of WorkKeyID of the Work Key used to decrypt the ECM SHALL be set.
- *Datetime*: Current date and time. 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 first Datetime (byte index of 18 to 21) SHALL be set to the time and date of ECM delivery. The second Datetime (byte index of 50 to 53) SHALL be set to the same value of the first Datetime.

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

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

| Byte index | Description |
|---|---|
| 0-1 | ProtocolVersion.<br>“0100h” SHALL be set for ECC 224-bit keys.<br>“0200h” SHALL be set for ECC 384-bit keys |
| 2-7 | WorkKeyID. |
| 8-17 | PrivateData.<br>“00h” SHALL be set to all 10 bytes for this specification. |
| 18-21 | Datetime. |
| 22-29 | ChannelTierBits. |
| 30-31 | RenderingObligation. |
| 32-33 | PrivateData.<br>“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.<br>“00h” SHALL be set to all 2 bytes for this specification. |
| 66-81 | ScrambleKey (even). |
| 82-97 | PrivateData.<br>“00h” SHALL be set to all 16 bytes for this specification. |

### 6.1.3 Processing Rules of ECM

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”.

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

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

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

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

- After decrypting the ECM, the Marlin IPTV-ES Device SHALL verify its parameters as shown below. If “verification failure” occurs, the Marlin IPTV-ES Device SHALL NOT proceed the process described hereinafter.
  - *ProtocolVersion*
    - Marlin IPTV-ES Device SHALL verify the ProtocolVersion as specified in section 4.2.4.2.
  - *Datetime*
    - If Datetime is other than the other (byte index of 18 to 21 and 50 to 53, respectively), the verification SHALL be deemed as “verification failure”.
  - *ChannelTierBits*
    - If ChannelTierBits is other than the other (byte index of 22 to 29 and 54 to 61, respectively), the verification SHALL be deemed as “verification failure”.
  - *RenderingObligation*
    - If RenderingObligation is other than the other RenderingObligation (byte index of 30 to 31 and 62 to 63, respectively), the verification SHALL be deemed as “verification failure”.
- After succeeding the verification of each parameters of the ECM, the Marlin IPTV-ES Device SHALL logically multiply (perform AND operation) each bit of ChannelTierBits within the ECM and SubscriptionTierBits of the Work Key used to decrypt the ECM. If one or more logical multiplications of each bit are “1b”, the Marlin IPTV-ES Device MAY decrypt the content using the Scramble Keys.
- After decrypting the content, the Marlin IPTV-ES Device SHALL control the consumption of the content in accordance with RenderingObligation obtained as an ECM, or with ExportInfo or RecordInfo corresponding to the WorkKeyID of the Work Key used to decrypt the ECM.

## 6.2 Interoperable Format

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

- Support for CBC mode only.
- Support for only one content ID/License per file.
- No IV update through a content stream.
- No PI packet included in the content stream.

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

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

# Appendix A Profiles (Normative)

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

## *A.1 SAC Protocol*

### A.1.1 Profile for Marlin IPTV-ES Devices

| 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 |

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

## A.1.2 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 |
| 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 |

## *A.2 Service Protocol*

### A.2.1 Profile for Marlin IPTV-ES Devices

| 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 |

[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.

## 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 |

[3] Servers SHALL accept Get Trusted Time Request message if and only if the message is packed in Packed Message Request message.

## *A.3 File Format*

### A.3.1 Profile for Marlin IPTV-ES Devices

| 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 Note that the profiles in this section MAY apply to servers other than Marlin IPTV-ES
2442 Server if they process these functions in this section.