Forward Link Only Open Conditional Access (OpenCA) Specification

ARIB STANDARD

ARIB STD-B50 Version 1.0

Version 1.0 November 5, 2010

Association of Radio Industries and Businesses

Forward

1. Introduction

With participation of radio communication equipment manufacturers, broadcasting equipment manufacturers, telecommunication operators, broadcasters and general equipment users, Association of Radio Industries and Businesses (ARIB) defines basic technical requirements for standard specifications of radio equipment, etc. as an "ARIB STANDARD" in the field of various radio systems.

In conjunction with national technical standards which are intended for effective spectrum utilization and avoidance of interference with other spectrum users, an ARIB STANDARD is intended as a standard for use by a private sector compiling various voluntary standards regarding the adequate quality of radio and broadcasting service, compatibility issues, etc., and aims to enhance conveniences for radio equipment manufacturers, telecommunication operators, broadcasting equipment manufacturers, broadcasters and general users.

An ARIB STANDARD herein is published as "Forward Link Only Open Conditional Access (OpenCA) Specification." In order to ensure fairness and transparency in the defining stage, the standard was set by consensus of the standard council with participation of interested parties including radio equipment manufacturers, telecommunication operators, broadcasters, testing organizations, general users, etc. with impartiality.

It is our sincere hope that the standard would be widely used by radio equipment manufacturers, testing organizations, general users, etc.

2. Scope

This standard applies to the multimedia broadcasting defined in Section 2 of Chapter 3-2, Ordinance No.26 of the Ministry of Internal Affairs and Communications, 2003.

3. Standard References for Forward Link Only

The following list identifies the current version of the standards in the FLO family of standards.

Standard#	Title
STD-B47	Forward Link Only Air Interface Specification for Terrestrial Mobile Multimedia Multicast
STD-B48	Forward Link Only Transport Specification
STD-B49	Forward Link Only Media Adaptation Layer Specification
STD-B50	Forward Link Only Open Conditional Access (OpenCA) Specification
STD-B51	Forward Link Only System Information Specification
STD-B52	Forward Link Only Messaging Transport Specification
STD-B32	Video Coding, Audio Coding and Multiplexing Specifications for Digital Broadcasting*

*NOTE: The original document of this standard is Japanese version. Part 3 of this standard is not applicable to Forward Link Only system.

4. Industrial Property Rights

This standard does not describe industrial property rights mandatory to this standard. However, the right proprietor of the industrial property rights has expressed that "Industrial property rights related to this standard, listed in the annexed table below, are possessed by the applicator shown in the list. However, execution of the right listed in the annexed table below is permitted indiscriminately, without exclusion, under appropriate condition, to the user of this standard. In the case when the user of this standard possesses the mandatory industrial property rights for all or part of the contents specified in this standard, and when he asserts his rights, it is not applied."

Annexed Table (Selection of Option 2)

Patent Applicant/Holder	Name of Patent	Registration No./ Application No.	Remarks
QUALCOMM Inc.	A comprehensive confirmation form has been submitted with regard to ARIB STD-B50 Ver.1.0.		
JVC KENWOOD Holdings, Inc.	A comprehensive confirmation form has been submitted with regard to ARIB STD-B50 Ver.1.0.		

Contents

1. SCOPE AND ORGANIZATION

This document describes a framework for enabling Service Purchase and Protection for Flows, as defined in [2], Clause 1.11, over the Forward Link Only network using customer selected upgradeable Key Management Systems. It describes the OpenCA framework that includes necessary blocks for ensuring interoperability between any Key Management System.

1.1. Organization of the document

This document is organized as follows:

Clause 1: An informative clause describing the scope and the organization of the document.

Clause 2: A normative clause defining compliance terminology and references.

Clause 3: A normative clause giving acronyms and definitions of terms.

Clause 4: An informative clause giving an introduction.

Clause 5: A normative clause defining the Entitlement Management Message signaling.

Clause 6: A normative clause defining the Entitlement Control Message signaling.

Clause 7: A normative clause defining the signaling for real-time services.

Clause 8: A normative clause giving details on the Secure Container when implemented as a UICC.

Clause 9: A normative clause defining an optional Secure Authenticated Channel protocol between the Secure Container and the Descrambler in the Device.

Clause 10: A normative clause extending the Simulcrypt head-end interfaces to support Forward Link Only networks.

Annex A: An informative annex giving an overview of the architecture of the system. It also presents the key hierarchy supported by any KMS compliant to this specification.

Annex B: An informative annex explaining the even/odd mechanism used to synchronize keys between a scrambler and a descrambler.

2. APPARATUS

2.1. Compliance Terminology

The key words “shall”, “shall not”, “should”, “should not”, “may”, “need not”, “can” and “cannot”, when used in this Specification, are to be interpreted as specified in the TIA Style Manual [1].

2.2. Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this Specification. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Specification are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

[1]. TIA Engineering Committee Recommendation. *TIA style manual (Internet Version),* 1992.

[2]. ARIB STD-B47, *Forward Link Only: Air Interface Specification for Terrestrial Mobile Multimedia Multicast*.

[3]. ARIB STD-B48, *Forward Link Only: Transport Layer Specification*.

[4]. ARIB STD-B49, *Forward Link Only: Media Adaptation Layer Specification*.

[5]. ETSI[1)] TS 103 197 (V1.5.1): *Digital Video Broadcasting (DVB); Head-end implementation of DVB SimulCrypt*.

[6]. ETSI TS 101 220 Smart-cards; *ETSI numbering system for telecommunication application providers*.

[7]. ETSI TS 102 221 Smart-cards; *UICC-Terminal interface; Physical and logical characteristics*.

[8]. ISO[2)] 11770-3: *Information technology – Security techniques – key management – Part 3: Mechanisms using asymmetric techniques*.

[9]. ISO 7816-4:2005, *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*.

[10]. IETF[3)] RFC 1112, *Host Extensions for IP Multicasting*.

[11]. IETF RFC 3280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

[12]. IETF RFC 3447, *Public-Key Cryptography Standards (PKCS)#1; RSA Cryptography Specifications Version 2.1*.

[13]. ITU-T *Recommendation X.509, Information Technology – Open Systems Interconnection – The Directory: Authentication Framework ITU-T X.509 standard*.

[14]. FIPS PUB 197. *Specification for the advanced encryption standard (AES)*, 2001.

[15]. Recommendation for Block Cipher Modes of Operation, NIST Special Publication 800-38A, 2001 Edition.

[16]. Ordinance No.26 of the Ministry of Internal Affairs and Communications, 2003.

[17]. Notification No.40 of the Ministry of Internal Affairs and Communications, 2003.

[18]. Notification No.88 of the Ministry of Internal Affairs and Communications, 2009.

1) ETSI publications are available from http://www.etsi.org.

2) ISO publications are available from http://www.iso.org.

3) RFCs are issued by the Internet Engineering Task Force (IETF). The address of the IETF is: IETF Secretariat, c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA.

3. DEFINITIONS AND ABBREVIATIONS

3.1. Definitions

All definitions from [2], [3], and [4] apply to this specification.

Content Owner: The Entity that owns the rights to the content.

Content Protection: Content protection deals with post-delivery usage rights which specify how content can be used according to permissions and constraints.

Content Provider: The Entity that provides and transmits the content via the Forward Link Only network.

Control Word: Key used to decrypt services.

Key Management System: An end-to-end system to authorize users and provide them the necessary means to access protected content.

KMS Device Agent: An entity that contains the specific logic required to control the descrambling process for a specific KMS.

Secure Container: Secure hardware (e.g. a UICC) allowing performing secure generation of the Control Words from the received Entitlement Control Messages.

Service Protection: It refers to controlling consumer access to content on a service provider's network at the moment of broadcast. It does not define what happens to content once delivered to the client.

3.2. Symbols and abbreviations

APDU	Application Protocol Data Unit
CW	Control Word
ECM	Entitlement Control Message
EMM	Entitlement Management Message
KDA	KMS Device Agent
KMS	Key Management System
MLC	Multicast Logical Channel
SAC	Secure Authenticated Channel
SEK	Service Encryption Key
SMS	Short Message Service
UICC	Universal Integrated Circuit Card
URL	Uniform Resource Locator
USI	Usage State Information

4. INTRODUCTION

The OpenCA framework is designed to provide commercial and security benefits to operators, horizontal-market channel providers and device manufacturers, and end-users.

Commercial benefits include:

- **Adaptability**: The ability to download updates of key security features and new business models to Devices in the field. Thus a flaw in the security system can be fixed by the security provider without waiting for a standard to be agreed upon. Moreover, new business models can be developed by operators and content providers and rapidly provided to end-users.
- **Vendor independence**: Operators have the freedom to seamlessly switch between security providers or even to simultaneously use two different vendors without the need to replace the Device. Moreover, the framework is ideally suited for implementation in horizontal-market (standard) devices, free of integration cost or customization.
- **Proven approach**: The framework is modeled after the proven pay-TV content security paradigms that protect high quality content world-wide.
- **Control of Key Management System**: The Key Management System (KMS) is the security component responsible for the generation of Entitlement Control Messages (ECMs) and business model enforcement. For better security and control in the device, the KMS can be implemented inside a Secure Container under the control of the operator.
- **Interoperability**: The framework enables the deployment of Simulcrypt, which provides secure sharing of content amongst a set of operators, each with its own independent security system. Using Simulcrypt, the effect of one security system being compromised can be negated and has no impact on the other security systems.

Security benefits include:

- **Renewable security**: The main element of any security system is its KMS; the ability of a security system to renew its KMS is crucial for long-term maintenance of security.
- **KMS compartmentalization**: Having a choice from many independent KMSs enhances security as a compromised KMS can be switched off in favor of a new one. The framework enables seamless transition to another KMS provider. A KMS provider can use variants of their security solution in different markets to minimize the likelihood and impact of any security compromise.
- **Support for multiple security systems**: The framework is designed to support any number of security providers' technologies by allowing seamless replacement of one security provider's KMS with another.

The proven approach for pay-TV systems has been adopted for broadcasting over satellite, terrestrial, cable and mobile. It is extended by the OpenCA framework. The functional relationships provided by the framework are illustrated in Figure 1. The framework sits above the common content scrambling/descrambling mechanism, and allows any KMS to be plugged. For example, Figure 1 illustrates that a KMS solution that plugs into the framework could be from a provider "A", "B", or "C".

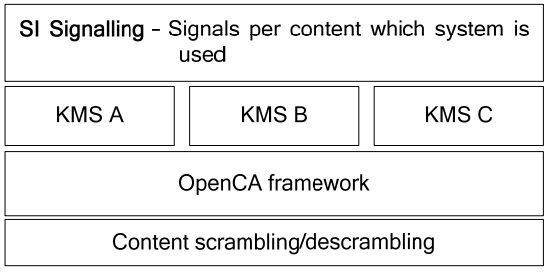

Figure 1: OpenCA framework concept.

Annex A contains further descriptions and details of the different elements making up the system.

4.1. General requirement for OpenCA compliant systems

OpenCA solutions shall support the simultaneous generation and decryption of at least three Control Words (CWs). For example, as would be required to support picture-in-picture with simultaneous recording of a third channel.

5. ENTITLEMENT MANAGEMENT MESSAGE

Entitlement Management Messages (EMMs) convey the necessary authorizations/privileges for the reception of broadcast media content for a single end user or group of end users. EMMs encompass the different business rules that allow operators to implement multiple business models, such as subscription, pay-per-view, pay-per-time, etc. In addition, EMMs may provide the secure mechanisms for initiating control actions on a Device, for example to control update to the KMS.

EMMs can be broadcast using the Forward Link Only network or retrieved from a URL using the unicast network. Other delivery methods for EMMs are not precluded by this specification (e.g. SMS if the unicast network supports it). In the case of the Forward Link Only network, the messages may be sent on an operator configured channel, so as to optimize delivery of the message stream to Devices. During the process of receiving the EMMs, the Device will typically be authenticated, authorized, and accounted to the requested content or service.

The format of the EMMs is specific to the individual KMS and is out of the scope of this specification.

5.1. Recommended Means of Delivering EMMs

For efficient use of the bandwidth and reliable delivery, it is recommended that EMMs be delivered using the unicast network where available. Delivery on the Forward Link Only network is not considered as the main delivery network for EMMs.

6. ENTITLEMENT CONTROL MESSAGE

6.1. Signaling and delivery of ECMs

Entitlement Control Messages (ECMs) transmit information that allows the Device to reconstruct the CW with which the media stream is scrambled. They shall be broadcast in the Stream 0 associated to the Flow [3]. ECMs may also include any access privileges definitions associated with the protected Flow. The ECM is processed by the KMS Device Agent (KDA) to retrieve the CWs required by the descrambler to process the Stream (See Annex A for more information).

At least one ECM shall be in each MLC carrying a scrambled Stream for each Superframe (defined in [2], Clause 1.11.) ECM information in Stream 0 enables descrambling of the Stream Packet of Stream 1 and 2 in the same MLC. There may be as many ECMs as there are scrambled Flows in the MLC multiplied by the number of KMSs used to protect those Flows. Clause 6 of [3] gives the general structure for messages in Stream 0. Additional fields are added as shown in Table 1.

Table 1: Format of an ECM.

Field Name	Field Type
MESSAGE_ID	UINT(8)
CA_SYSTEM_ID	UINT(16)
OPERATOR_ID	UINT(16)
CW_SEQUENCE_ID	UINT(16)
ECM_MESSAGE_BODY	Variable

The MESSAGE_ID field for ECM is equal to 0x01.

The CA_SYSTEM_ID field identifies the KMS provider generating the contents of the ECM. It shall be globally unique.

The OPERATOR_ID field is managed by each KMS provider, and allows identifying the operator operating the KMS..

The CW_SEQUENCE_ID (Control Word Sequence ID) identifies the sequence of control words used to scramble a set of Flows. Further details on this ID are given in Clause 7.2.

The ECM_MESSAGE_BODY is of variable length and carries KMS-specific data. The length of the ECM can be obtained from the Transport Layer.

The data to be inserted in Stream 0 is shown in Figure 2. In this simplified example where only some possible messages are shown, in addition to the EIM (Encryption Information Message, see in Clause 7.2), ECMs for two different KMSs are to be multiplexed. KMS A inserts, in this example, two ECMs (for different Flows for example) while KMS B inserts only one ECM.

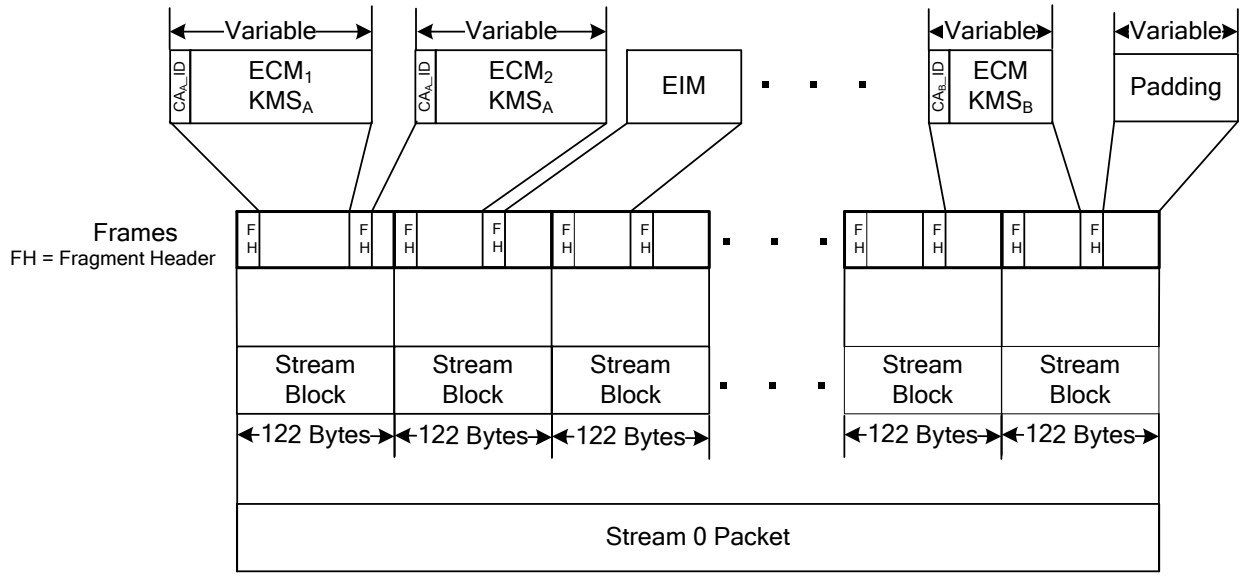

Figure 2: Framing layer.

6.2. Crypto-period and Superframes

A crypto-period defines the period when the same CW is applied by the scrambler on a Flow. The crypto-period for a Flow is equal to an integer number of Superframes. This means that for a given Flow, only one CW can be used in a Superframe. The ECM applying to this Flow in Stream 0 may be the same for consecutive Superframes during the crypto-period.

6.3. Recommendation on Bandwidth Allocation for ECMs

The total bandwidth allocated to Stream 0 messages is limited according to the restrictions specified in [2], Clause 4.2.5.1. In summary, it cannot exceed around 4 kbps if there are two other streams present in the MLC, and cannot exceed around 255 kbps if there is only one other stream present in the MLC. As a consequence, the total amount of data transported in all ECMs should remain reasonable, especially when simulcrypting is considered.

7. REAL-TIME SERVICES

7.1. Encryption and transport settings

Scrambling of media streams is as defined in [3]. The IV construction is as described in [3]. The CW is used to scramble Stream Packets using the Flow Cipher before they are transmitted. The Flow Cipher preserves the length of the data, and each Stream Packet is separately scrambled. If the Device is required to process a scrambled Stream, it first recovers the encrypted CW from the ECM delivered in Stream 0 of the MLC transporting the Stream. The device decrypts the CW under the associated EMM. The CW is then used to descramble data. Once the Stream Packets are descrambled, they are delivered to the Framing Layer for further processing.

Configuration options in the FlowBLOB follows the policy defined in [4].

7.2. Encryption Information Message

Information regarding the scrambling of the media streams is common to all compliant KMSs. Therefore, a message carrying any information not KMS-specific is defined. The Encryption Information Message (EIM) shall be carried in Stream 0. It allows a single ECM to be transported in an MLC when more than one Flow is scrambled by the same CW in this MLC. An EIM only applies to Flows which are carried in its MLC. A single EIM shall be in the stream 0 for a single MLC,

The EIM has the format shown in Table 2.

Table 2: Format of the EIM.

Field Name	Field Type
MESSAGE_ID	UINT(8)
List of EIM_records {	
FLOW_ID	UINT(20)
RESERVED	UINT(4)
CW_SEQUENCE_ID	UINT(16)
EVEN_ODD_INDICATOR	UINT(1)
FLOW_CIPHER_TYPE	UINT(6)
MORE_FLOW_NEXT	UINT(1)
}	

The MESSAGE_ID for ENCRYPTION_INFO_MESSAGE is equal to 0x05.

The FLOW_ID field shall be set to the ID of the Flow to be considered for the record.

The RESERVED field will serve byte alignment and possible future extension.

The CW_SEQUENCE_ID field shall be set to the CW_SEQUENCE_ID field in the ECM associated with the Flow.

The EVEN_ODD_INDICATOR field allows the scrambler and descrambler to synchronize the actual CW to be used. The EVEN_ODD_SELECTOR allows selection between two CW in the CW storage of the descrambler. The possible values for the EVEN_ODD_SELECTOR are given in Table 3.

Table 3: Defined values for EVEN_ODD_SELECTOR.

Value	Meaning
0	The Stream Packet is scrambled with Even CW
1	The Stream Packet is scrambled with Odd CW
All other values are reserved.	

Additional details on the even/odd synchronization mechanism are given in Annex B.

The FLOW_CIPHER_TYPE field identifies the algorithm used to scramble the Flow data. The defined values are given in

Table 4.

Table 4: Defined values for FLOW_CIPHER_TYPE.

Value	Meaning
0	UNSCRAMBLED
1	AES_CTR_128 ([13], [14])
All other values are reserved.	

Regardless of whether a Flow is scrambled or not, the EIM message shall contain the EIM_record for the Flow. When a Flow is unscrambled (clear to air), the FLOW_CIPHER_TYPE of the unscrambled Flow shall be set to 0, and CW_SEQUENCE_ID and EVEN_ODD_INDICATOR fields shall be ignored at the device.

The MORE_FLOW_NEXT field indicates the last record in the message. The possible values are defined in Table 5.

Table 5: Defined values for MORE_FLOW_NEXT.

Value	Meaning
0	There is no more EIM_record
1	There is another EIM_record
All other values are reserved.	

7.3. Examples of EIM use

Figure 3 shows an MLC made of one audio and one video protected by two different KMSs (KMS_A and KMS_B), each Flow is scrambled with the same CW. Therefore, only one ECM in Stream 0 per KMS is needed. The video Flow has the ID "$FLOW\_ID_1$" and the audio Flow has the ID "$FLOW\_ID_2$". Both are scrambled with the same CW. The EIM message in Stream 0 contains then 2 records. The first record states that for descrambling the Flow with ID "$FLOW\_ID_1$", one has to find the ECM with sequence ID equal to "CW_SEQUENCE_ID". The second record states that for descrambling the Flow with ID "$FLOW\_ID_2$", one has to find the ECM with sequence ID equal to "CW_SEQUENCE_ID". The choice between KMS_A and KMS_B to obtain the adequate ECM is done on the {CA_SYSTEM_ID, OPERATOR_ID} couple.

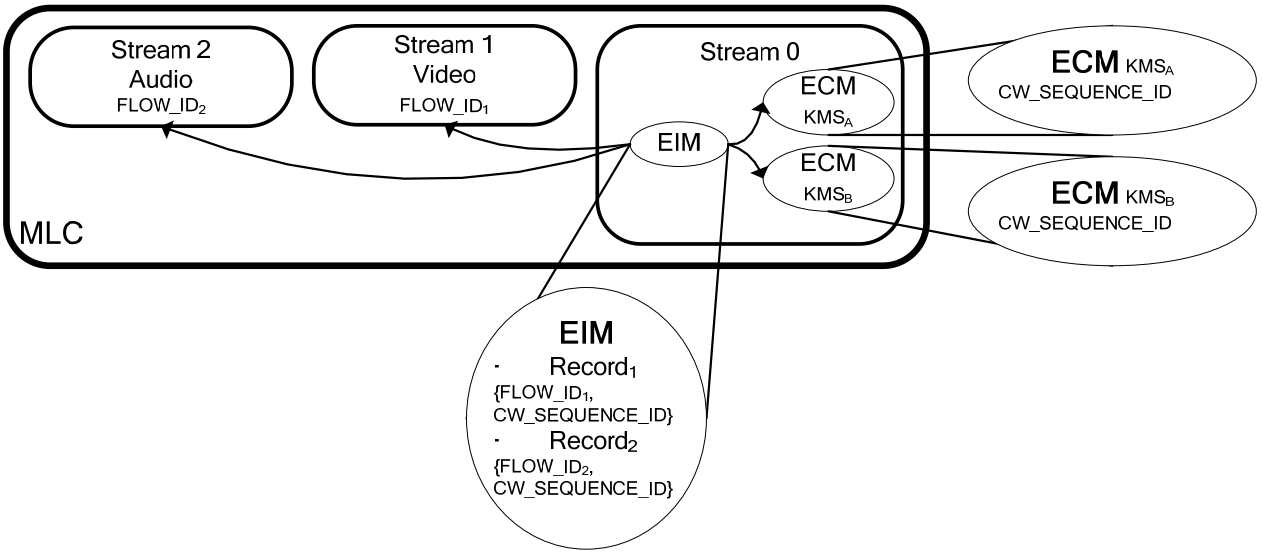

Figure 3: Example of a MLC with one audio and one video scrambled with the same CW (multiple KMS case).

Figure 4 shows an MLC made of one audio and one video protected by two different KMSs (KMS_A and KMS_B), each Flow is scrambled with a different CW. Therefore, two ECMs in Stream 0 per KMS are needed, one for the video Flow and one for the audio Flow. The video Flow has the ID "$FLOW\_ID_1$"

and the audio Flow has the ID "FLOW_ID$_2$". The EIM message in Stream 0 contains then 2 records. The first record states that for descrambling the Flow with ID "FLOW_ID$_1$", one has to find the ECM with sequence ID equal to "CW_SEQUENCE_ID$_1$". The second record states that for descrambling the Flow with ID "FLOW_ID$_2$", one has to find the ECM with sequence ID equal to "CW_SEQUENCE_ID$_2$". The choice between KMS$_A$ and KMS$_B$ to obtain the adequate ECM is done on the {CA_SYSTEM_ID, OPERATOR_ID} couple.

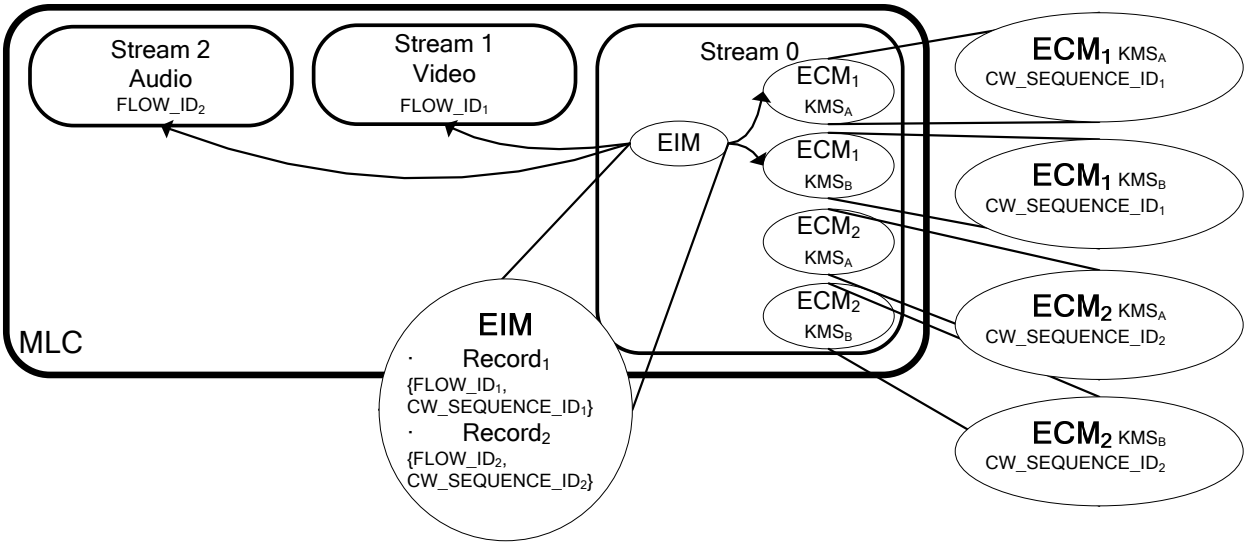

Figure 4: Example of a MLC with one audio and one video scrambled with different CWs (multiple KMS case).

7.4. Copy protection

In case content needs to be securely exported to a separate post-delivery copy-protection system, the Usage State Information (USI) needed by such a system can be carried by the KMS. A simple example is a 2-bit flag with four different values. These values define the restrictions that are applied on the content:

- Copy-freely: Content can be copied-freely.
- Copy-one-generation: One generation of copies can be made, the copy becomes then "No-more-copies".
- No-more-copies: Content has been copied once and can not be be copied anymore.
- Copy-Never: Content can never be copied.

USI could also describe more complex export rules, depending on the supported copy-protection system. USI could be carried, for example in ECMs or as an additional message in Stream 0, this is then part of the broadcast. The USI values are obtained with content and are defined by content owners.

8. THE SECURE CONTAINER AS A UICC

The KDA may communicate with secure hardware, called the Secure Container, (e.g. a UICC) to perform secure generation of the CWs from the received ECMs. If no Secure Container is present in the Device, the KDA is responsible for the generation of the CWs from the received ECMs.

If the Secure Container takes the form of a UICC, the interface between the device and the UICC is specified by [9] with the following constraints. The content of the messages exchanged on this communication link is outside the scope of this specification.

8.1. Application IDentifier (AID)

All UICCs offering KMS functionality are identified as such via the Application IDentifier (AID). The structure of the AID is given in [6].

For any compliant KMS Application:

- The Registered application provider IDentifier (RID) SHALL be 'A000000009';
- The Application code SHALL be “0101”;
- The Application provider code SHALL be the CA_System_Id assigned to the KMS encoded as 4 hexadecimal digits. The coding is right justified and padded with "FF" on the left;
- The Application provider field is reserved for the KMS provider's use.

8.2. KMS application selection

The application and logical structure for the UICC is defined in [7].

All UICCs contain, in the EF_{DIR} file at the MF level, the list of AIDs installed in the cards. For a UICC offering KMS functionality, the EF_{DIR} file contains a record with an AID corresponding to the KMS application (i.e. with the Application code equal to “0101” and the Application provider code equal to the KMS provider's CA_System_Id).

A UICC may contain several applications. Typically the UICC would contain a USIM application and one or several KMS applications. Each application will have a specific AID referenced in a record of this EF_{DIR} file.

After UICC activation, the KDA creates an APDU connection to communicate with the KMS Application using, as a parameter, the corresponding AID of the application found in the EF_{DIR}.

9. SECURE AUTHENTICATED CHANNEL

This clause describes an optional mechanism used to establish a Secure Authenticated Channel (SAC) between the KMS application residing on the UICC and the descrambler when implemented in hardware.

The KDA may establish a SAC between the Secure Container and the descrambler to control the exchange of CWs. Any secure KMS device function will be contained as an application inside this multi-application Secure Container.

The SAC ensures the secrecy of the communication between the KMS application and the descrambler and prevents CW extraction by an eavesdropper. The SAC also provides authentication of the descrambler by the KMS application, thus preventing unauthorized devices, such as a PC, from extracting CWs from the KMS. The SAC does not, however, provide authentication of the KMS by the descrambler, as protection of the descrambler against misuse is not the primary goal of the SAC. The authentication of the KMS application may be done by the KDA.

9.1. High level description of the SAC

The SAC is based on asymmetric cryptography. Each hardware descrambler is assigned a unique triplet:

- A Unique Identifier.
- A Private Key kept secret in the descrambler.
- A Public Key, made available to the KMS.

This unique triplet is provided to the descrambler manufacturer for adding it at the manufacturing moment. A certification authority provides this triplet. How this certification authority provides the triplets to manufacturers is out of the scope of this document. The descrambler makes its Unique Identifier available to the KDA through a public interface. This Unique ID may then be used by the KMS to request from its head-end (server) the corresponding Public Key of the descrambler. The manner in which the KMS communicates with its head-end and retrieves the Public Key of the descrambler is out of scope of this document, but it may involve using the interactivity channel.

Once in possession of the Public Key of the descrambler, the KMS application generates a random session key seed, encrypts it with the descrambler's public key and provides the result to the descrambler. Both the KMS application and the descrambler then derive a session key from the session key seed.

Once both parties share this session key, all CWs are encrypted by the KMS application before they are passed to the descrambler, which decrypts them before use.

9.2. The cryptographic keys and parameters

The cryptographic protocol is based on the El Gamal Key Agreement (half-certified Diffie-Helman), specified in [8].

9.2.1. The descrambler's keys

Each descrambler is assigned the following values:

p A 1536-bit prime modulus

q A 160-bit prime divisor of **p-1**

g A generator of order **q**

x A randomly or pseudo-randomly generated integer with **0** < **x** < **q**

The public key is composed of **(p, q, g, g^x mod p)**, and the secret key is **x**.

9.3. The SAC protocol

9.3.1. Session key establishment

The KMS application picks a random value y so that 0 < y < q, sends g^y mod p to the descrambler and computes

$$s = (g^x \bmod p)^y \bmod p \quad (1)$$

The descrambler computes

$$s' = (g^x \bmod p)^y \bmod p = (g^y \bmod p)^x \bmod p = s \quad (2)$$

At the end of this protocol the KMS application and the descrambler share s, a 1536-bit value. This is hashed into a 160-bit value and the 128 high order bits of the hash are extracted to obtain the session key

$$k = [\text{SHA-1}(s)]_{\text{MSB-128}} \quad (3)$$

9.3.2. Secure key exchange

Following the session key establishment, the KMS application may load keys in the descrambler, encrypting them with the session key k

$$m = \text{AES-ECB-128}(CW, k) \quad (4)$$

In this version of the specification, both the CW and the session key are 128-bit keys and AES is used directly on the CW in ECB mode.

10. HEAD-END INTERFACES

10.1. Adaptation of Simulcrypt head-end interfaces

Simulcrypting, from a Device perspective, means that signaling is available that allows the Device to acquire the necessary information based on the supported KMS.

In the head-end, supporting several KMSs has implications on the architecture. The ECMs delivered in Stream 0 shall include all necessary information for all KMSs. Some data (such as CW) must be shared among the various KMSs. Simulcrypt [5] defines interfaces between various entities in the head-end. It is designed for systems dealing with MPEG-2 Transport Streams (TS) with scrambling of the MPEG-2 TS payload. This clause describes adaptation of the head-end interfaces to the Forward Link Only environment.

10.1.1. Reference head-end architecture and interfaces and adaptations

From Figure 1 of [5], the elements taken into account in the Forward Link Only head-end system architecture are:

- The SCS – SimulCrypt Synchronizer.
- The ECMG - Entitlement Control Message Generator.
- The EMMG - Entitlement Management Message Generator.
- The CWG - Control Word Generator.

Description and role of these elements can be found in [5].

The head-end architecture with these elements is shown in Figure 5. The Multiplexing Subsystem incorporates the SCS, CWG, and Scrambler shown in Figure 1 of [5]. The role of the Multiplexing Subsystem component is to perform time multiplexing of input data, and to output Superframes. The input data can be Flows, or IP data.

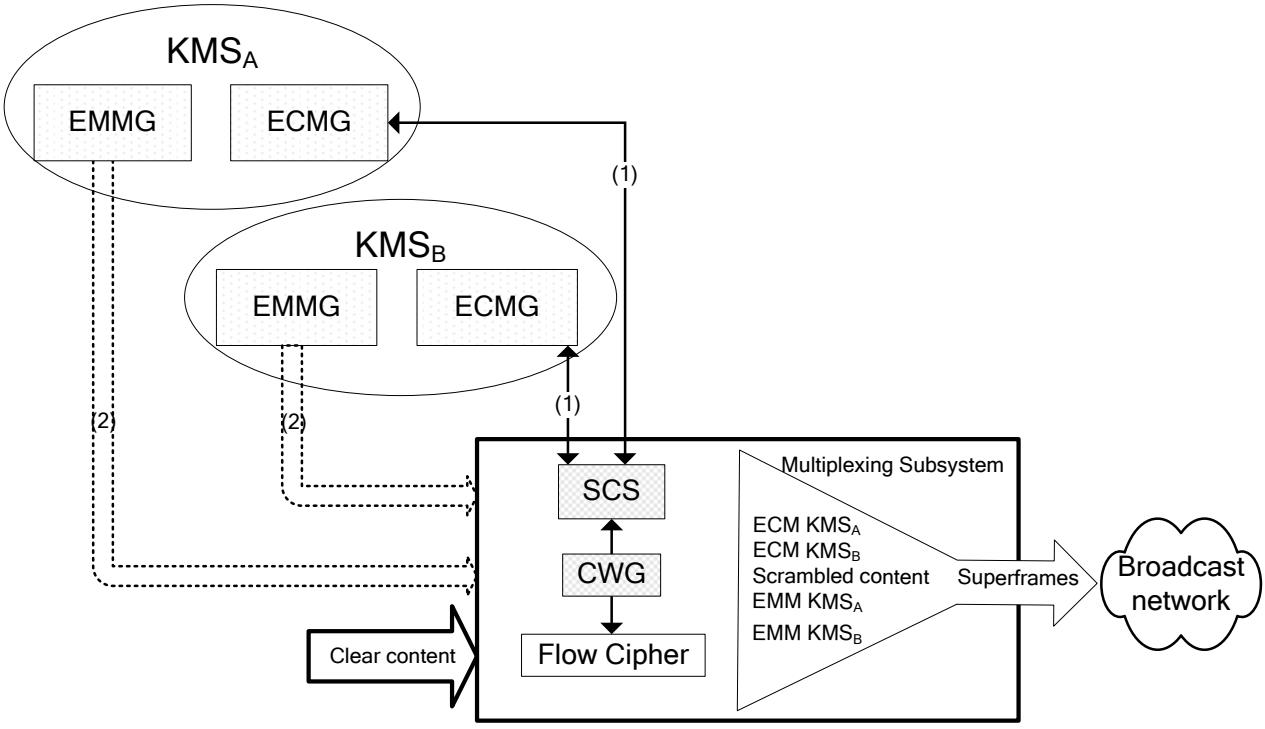

Figure 5: Simulcrypt compliant Forward Link Only head-end architecture.

The reference architecture from [5] can be mapped to Forward Link Only head-end architecture as follows (interface reference numbers correspond with those in Figure 5):

1. Interface ECMG ⇔ SCS should be implemented according to [5], Clauses 4.4 and 5, and the modifications in this specification in Clause 10.1.4.

2. Injecting EMM may be implemented according to Clause 10.2. This applies only if the system supports delivery of EMM over the broadcast channel.

This specification is limited to these two interfaces; any other interface is out of the scope and may be specified in a further version.

10.1.2. Definitions

Within the Simulcrypt specification, channel and stream are defined as follows:

Channel Application specific representation of an open TCP connection, allowing the association of application specific parameters with such a connection.

Stream Independent bi-directional data flow across a channel.

To avoid confusion with Forward Link Only defined Channel and Stream, the terms are specialized on the ECMG ⇔ SCS interface to ECM_Channel and ECM_Stream respectively.

10.1.3. Message Protocol Version

The protocol_version field in the generic_message structure defined in [5], Clause 4.4.1, shall be set to 0x06.

10.1.4. Interface ECMG ⇔ SCS

This interface allows a KMS to provide a MUX with ECMs under the control of the SCS. The messages for ECM_Channel and ECM_Stream interfaces are as defined in [5], Clause 5 with the following extensions.

The combination {Super_CAS_id + ECM_id} identifies uniquely an ECM stream in the system. On this interface, the SCS sends CWs to an ECMG to allow it to generate ECMs. The SCS identifies which ECMG to contact by using the Super_CAS_id. The SCS indicates to the ECMG for which ECM stream the CWs are used for ECM generation by using the ECM_id.

The **Super_CAS_id** is a 4-byte identifier that uniquely identifies a KMS Provider. It is formed by the concatenation of the CA_System_Id (as defined in Clause 6) for the first 2 bytes and the CA_subsystem_id for the last 2 bytes. The CA_subsystem_id is defined by the KMS Provider, it is private and may be equal to the Operator_Id.

The **CP_CW_combination** used in CW_provision message, uncludea the KSI. The KSI is not used, thus the CP_CW_combination only contains the CP and TKM values and the TKM value is equal to the CW.

The compliance to ETR289 for the **ECM_datagram** defined in Clause 5.3 of [5] does not apply. The **ECM_datagram** consists of the ECM_MESSAGE_BODY specified in Clause 6.1. Its format is KMS-specific. Once received, the Multiplexing Subsystem shall update the header to create the ECM message.

10.1.5. Using ECMG ⇔ SCS in a Forward Link Only network

This clause shows a typical set-up of connection between an ECMG and a SCS. A more general description of setting-up this connection can be found in [5], Clause 5.1.

There is one instantiation of the SCS for each scrambled Service. The connection between an SCS and an ECMG is established in three steps:

- A TCP connection is established as described in [5], Clause 5.1.2. This requires that the SCS knows the IP address of the ECMG before any communication can take place.
- An ECM_Channel is then set-up using the messages defined in Clause 10.1.4. The SCS connects to an ECMG over a TCP connection and assigns to this connection an ECM_channel_id value. This value allows the SCS to uniquely associate the ECM_Channel to the ECMG. This first message contains the Super_CAS_id that states which KMS the SCS expects the ECMG supports on this ECM_Channel. The ECMG replies with a status message that contains some

information, including its optimal crypto-period, the maximum number of ECM streams it can concurrently create. It can also reply with a channel_error message.

- On top of this ECM_Channel, for each ECM stream that has to be created for the Service, an ECM_Stream is created. The SCS sends a set-up message to the ECMG that contains the effective crypto-period duration to use for this ECM_Stream, and the unique ECM_stream_id. The ECMG replies with a stream_status message or a stream_error message.

Once these three steps are completed, the ECMG can provision the SCS with ECMs on the established ECM_Stream. The CW_provision message allows the SCS to send to the ECMG the necessary material to create ECMs that are send back in ECM_message response. Sending of CWs in the CW_provision message can be optionally encrypted with a selectable algorithm. As already mentioned, the combination {Super_CAS_id + ECM_id} uniquely identifies an ECM stream in the system, therefore the ECMG needs to know which ECMs it identifies. This identification mechanism is out of the scope of this interface.

The SCS then receives one ECM (the body of the message) for each crypto-period (a given number of Superframes). It can then multiplex this ECM, once created with the header information with all other information, such as EIM, part of Stream 0 of the relevant Superframes, i.e. the Superframe containing the MLC scrambled with the Control Word securely delivered in this ECM.

10.2. Injecting EMM IP streams

EMM are carried in IP packets as specified in Clause 5. Injecting these IP packets in the Forward Link Only network and more specifically in the multiplexing subsystem shall be done following the interface defined for injecting any IP Datacast packets. This interface shall be compliant to [10].

No Text

ANNEX A. (INFORMATIVE)

System architecture

A.1 High-level view of the system

A.1.1 End-to-end system

The framework takes advantage of the transport layer system defined in [1] and [3]. It extends it where necessary to support multiple KMSs. Figure 6 shows the overall high-level view of the end-to-end system.

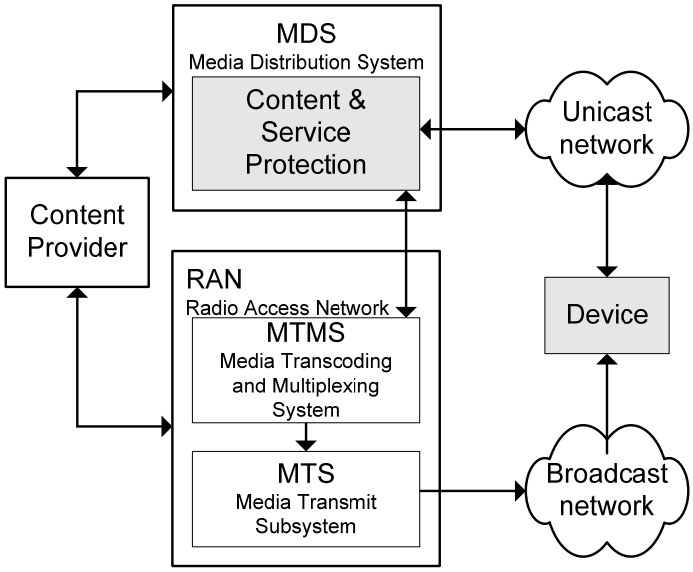

Figure 6: Overview of the OpenCA framework architecture.

In this high-level view the following segments are presented:

	Function
Content Provider	Controls what content is being played at a given time, and using what access criteria.
Content & Service Protection	Responsible for: • Maintaining containers used to carry entitlements and access privileges. • Communication with the device via the unicast channel when such channel is available.
Media Transcoding and Multiplexing System	Responsible for: • Creation of the Forward Link Only Streams, including combining any data presented by the Content and Service Protection Segment. This includes the Streams defined in [3]. • Presenting this data to the MFTS Segment for transmission.
Media Transmit Subsystem	Responsible for Transmission of data to the device.
Device	This consists of the viewer device and optional Secure Container. Receives the Forward Link Only signal, parses the Flows, and is responsible for descrambling and rendering.

A.1.2 Content & service protection segment

This specification concentrates on the Device and the Content & Service Protection Segments. Figure 7 provides an overview of the architecture for these specific elements.

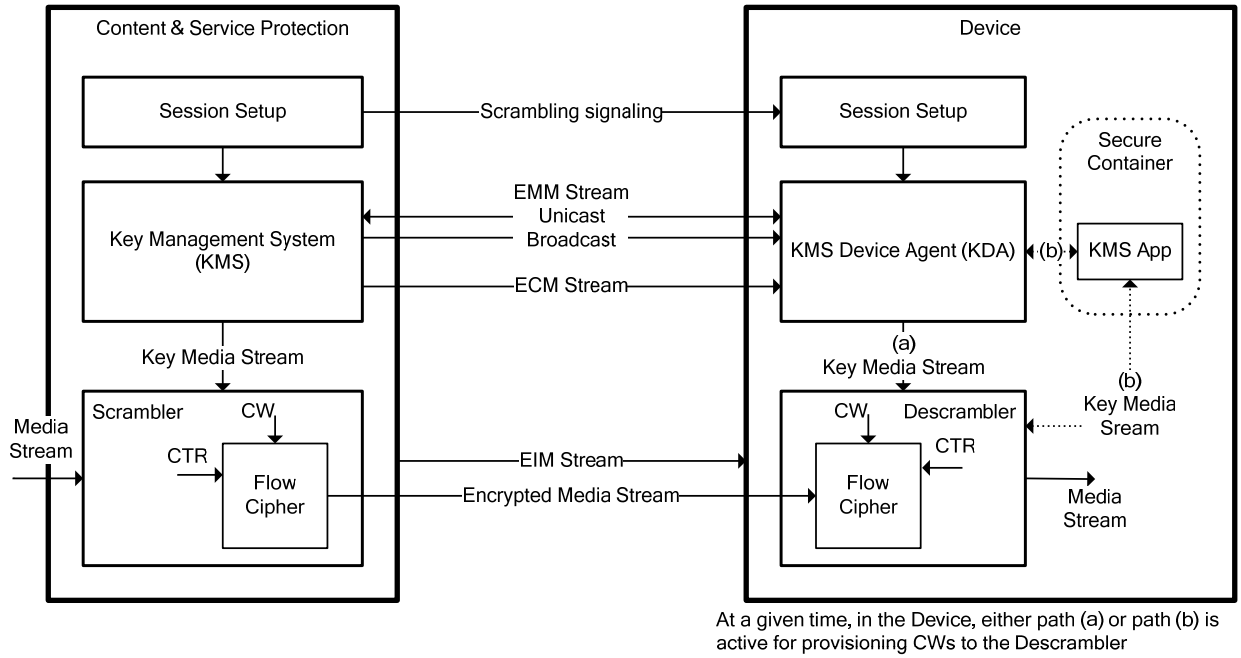

Figure 7: Content and Service Protection Architecture.

Within the Device, the KDA contains vendor-specific logic required to control the descrambling process for a specific KMS. The KDA performs the following security functions:

1. Reception of the EMMs from the KMS by means of the broadcast or unicast networks.
2. Secure generation of Device authorizations from the received EMMs using, if necessary, a KMS-specific application inside a Secure Container.
3. Reception of the ECM Stream for the selected service from the broadcast network.
4. Secure generation of the CWs from the received ECM stream using, if necessary, a KMS-specific application inside a Secure Container.
5. Application of the CWs to the descrambler, controlling the descrambling of the scrambled media stream. The EIM is then used to ensure that the Device finds the relevant ECMs and that the descrambler is synchronized with the scrambler.

A.1.3 The Secure Container

The KDA may communicate with secure hardware, called the Secure Container, (e.g. a UICC) to perform secure generation of the CWs from the received ECMs. If no Secure Container is present in the Device, the KDA is responsible for the generation of the CWs from the received ECMs.

The KMS application residing inside the Secure Container performs the following security functions under control of the KDA:

1. Secure generation of Device authorizations from the received EMMs.
2. Secure generation of the CWs from the received ECM stream.

The KDA and the Secure Container may optionally establish a SAC depending on the lifespan of the keys and the commercial value of the content.

A.2 Hierarchical model for content & service protection

The framework is based on a common key hierarchy model supported by all KMSs. KMSs refer to Stream Encryption mechanism described in [3] that is to be implemented by all Devices and therefore they can all be used by any of the KMSs.

Both the ECM and EMM systems are private, allowing the possibility to deploy a KMS that defines the Key Stream, Rights Management and Registration layers (as shown in Figure 8). At the content & service protection layer the mechanism specified in [3] is used. The hierarchical model for the content & service protection mechanisms is outlined in Figure 8.

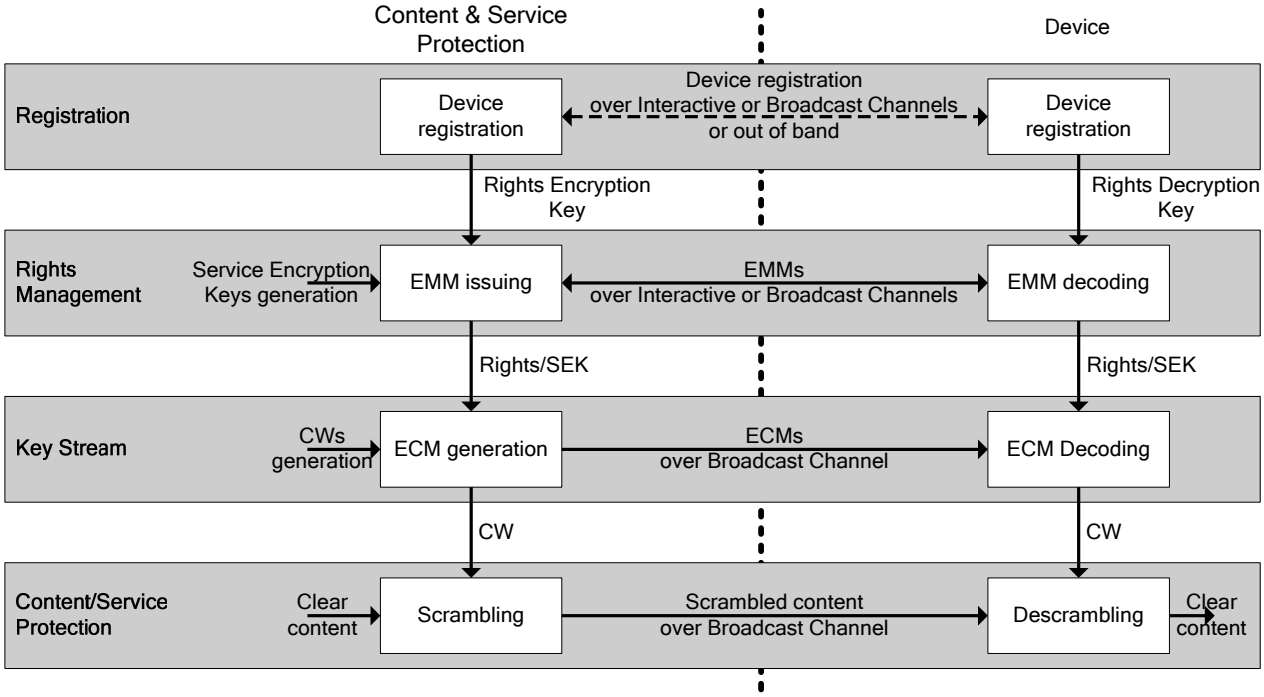

Figure 8: Hierarchical Model for Content & Service Protection.

The following provides a short description of the levels:

A.2.1 Registration

Key material and metadata are exchanged during the registration phase. This will enable Devices to decrypt and authenticate rights and subsequently access content/services. This registration layer is out of the scope of this specification as this is private to the KMS. Device registration shall take advantage of the unicast network. The use of the unicast network for Device registration shall be defined as private to the KMS. This phase is optional and Registration can be done out of band.

A.2.2 Rights Management/Authorization and Rights Issuing

Content/service access privileges are delivered to the Device using the EMMs. EMMs are typically exchanged as a result of a purchase transaction and transferred to Devices via the broadcast or unicast networks. The content/service access privileges typically consist of a Service Encryption Key (SEK) or an “authorization”, used to access the ECMs, and/or information such as entitlements. The format and content of the EMM is private to the KMS.

A.2.3 Key Stream

The Key Stream layer implements the delivery of CWs by transmission of ECMs to the Device on the broadcast network. These messages, in essence, contain information that allows the Device to reconstruct the CW needed to descramble the content/service. ECMs may contain additional information to control access to the content/service, such as access criteria. The format and content of the ECM is private to the KMS.

Because of the specific structure of a Superframe, ECMs are to be found in every Superframe containing scrambled content. As a consequence, just-in-time delivery is easily achieved. The scrambling mechanism supports the concept of crypto-periods, allowing a time-varying sequence of

CWs to be applied to the scrambler. The length of a crypto-period is equal to an integer number of Superframes.

A.2.4 Content/Service Protection

The content/service is scrambled by a symmetric encryption algorithm using a CW. The scrambling is performed as defined in [3]. CWs change frequently to prevent real-time key distribution attacks.

ANNEX B. (INFORMATIVE)

Scrambler-descrambler synchronization with the even/odd indicator

During a crypto-period (equal to N Superframes), the Flow Cipher engine uses a CW. A different CW is used for the each crypto-period. The scrambler in the head-end and the descramblers in the terminals need a mechanism to always use the right CW. This is critical at crypto-period boundaries. The even/odd indicator is used to ensure that the scrambler and de-scrambler are synchronized and always use the same CW for scrambling and de-scrambling respectively.

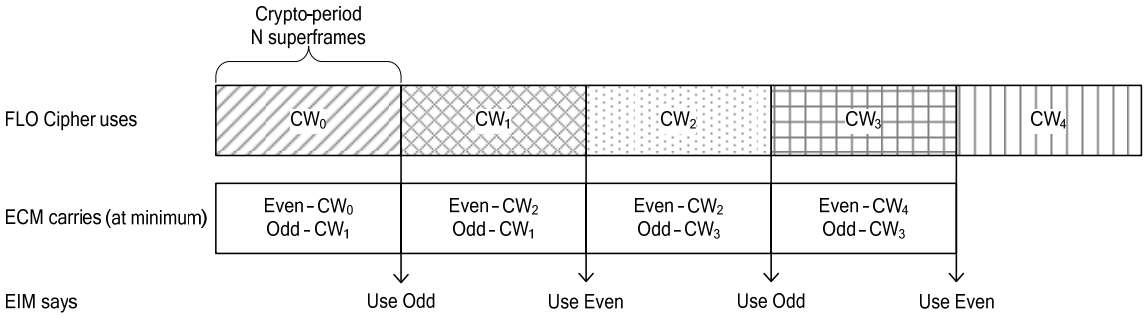

Figure 9: Even/odd scrambler-descrambler synchronization mechanism.

As shown in Figure 9, in a typical implementation, an ECM carries a minimum of 2 CWs, each one has a even/odd indicator. In the Head-end, the Simulcrypt interface ECMG ⇔ SCS allows the SCS to provide necessary information to the ECMG for creating the ECMs with this indicator. As an example, on this interface, the first CW in the “CW_provision” message could be the even CW and the second CW could be the odd CW In the terminal, both CWs are provided to the descrambler with their respective even/odd indicator. The synchronization is then ensured by the EIM that carries the current even/odd indicator defining which CW the descrambler shall use for the Superframe.

No Text

Forward Link Only Open Conditional Access
(OpenCA) Specification

ARIB STANDARD

ARIB STD-B50 Version 1.0

Version 1.0 November 5th 2010

Published by

Association of Radio Industries and Businesses

11F, Nittochi Building,
1-4-1 Kasumigaseki, Chiyoda-ku, Tokyo 100-0013,Japan

TEL 03-5510-8590
FAX 03-3592-1103

Printed in Japan