

International Telecommunication Union

**ITU-T**

TELECOMMUNICATION  
STANDARDIZATION SECTOR  
OF ITU

**X.1195**

(02/2011)

SERIES X: DATA NETWORKS, OPEN SYSTEM  
COMMUNICATIONS AND SECURITY

Secure applications and services – IPTV security

---

**Service and content protection interoperability  
scheme**

Recommendation ITU-T X.1195



ITU-T X-SERIES RECOMMENDATIONS  
**DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY**

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
<b>IPTV security</b>	<b>X.1180–X.1199</b>
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589

*For further details, please refer to the list of ITU-T Recommendations.*

# Recommendation ITU-T X.1195

## Service and content protection interoperability scheme

### Summary

Recommendation ITU-T X.1195 develops a complete set of requirements for interoperable service and content protection (SCP) to support interoperability between multiple SCP mechanisms. This includes interoperable SCP scenarios, interoperable SCP architecture and interoperable SCP process.

### History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.1195	2011-02-13	17

### Keywords

Content protection, interoperability, interoperable SCP, security, service protection.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2012

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

## Table of Contents

	<b>Page</b>
1	Scope ..... 1
2	References..... 1
3	Terms and definitions ..... 1
3.1	Terms defined elsewhere ..... 1
3.2	Terms defined in this Recommendation..... 2
4	Abbreviations and acronyms ..... 2
5	Conventions ..... 2
6	Introduction ..... 3
6.1	IPTV general architecture and content protection architecture ..... 3
6.2	Overview of the interoperable SCP framework ..... 5
7	Interoperable SCP framework ..... 6
7.1	Interoperable SCP architecture..... 6
7.2	Interworking processes ..... 7
8	Interworking process protocol specification..... 14
8.1	Overview of the protocol message ..... 14
8.2	Authentication protocol ..... 15
8.3	Negotiation protocol..... 16
8.4	Transmission protocol ..... 18
8.5	Transmission packet format ..... 21
Annex A	– Scenarios of SCP-B or SCP-IX deployed in IPTV TD ..... 23
A.1	Definitions of terms used in Figure A.1 ..... 23
A.2	Scenario 1: SCP with SCP-IX ..... 23
A.3	Scenario 2: SCP with optional SCP-B and storage ..... 23
A.4	Scenario 3: SCP with storage and SCP-IX..... 24
Annex B	– Basic type and premises of the interworking process protocol ..... 25
B.1	Request type ..... 25
B.2	Response type..... 25
B.3	Status type ..... 25
B.4	Extension type ..... 26
B.5	Nonce type..... 26
B.6	Canonicalization and digital signature ..... 26
Annex C	– Interworking process protocol schema ..... 27
Appendix I	– Interoperable SCP types..... 31
I.1	SCP end-to-end (SCP-EE)..... 31
I.2	SCP bridging (SCP-B)..... 31
I.3	SCP interchange (SCP-IX) ..... 31

	<b>Page</b>
Appendix II – Security considerations.....	32
II.1    Channel security .....	32
II.2    Possible attacks.....	32
Appendix III – SCP interoperability and legacy scheme .....	33
Appendix IV – Proprietary authentication scheme .....	34
Appendix V – Sample interworking process protocol messages.....	37
V.1    AuthenticationHelloRequest.....	37
V.2    AuthenticationHelloResponse .....	37
V.3    NegotiationRequest .....	37
V.4    NegotiationResponse .....	38
V.5    TransmissionHelloRequest.....	38
V.6    TransmissionHelloResponse .....	38
V.7    IdentificationRequest.....	38
V.8    IdentificationResponse .....	39
V.9    TransmissionRequest.....	39
V.10   TransmissionResponse .....	39
Appendix VI – Calculation of ITU-T X.509 certificate.....	40
Bibliography.....	41

# Recommendation ITU-T X.1195

## Service and content protection interoperability scheme

### 1 Scope

This Recommendation develops a complete set of requirements for the interoperable service and content protection (SCP) to support interoperability between multiple SCP mechanisms. This includes interoperable SCP scenarios, interoperable SCP architecture and interoperable SCP process. This Recommendation describes general requirements of the SCP interoperability. This Recommendation provides a complete set of requirements for a specific approach and does not include any other legacy approaches to solve SCP interoperability in [ITU-T X.1191].

### 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- [ITU-T X.509] Recommendation ITU-T X.509 (2008) | ISO/IEC 9594-8:2008, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.
- [ITU-T X.800] Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.
- [ITU-T X.1191] Recommendation ITU-T X.1191 (2009), *Functional requirements and architecture for IPTV security aspects*.

### 3 Terms and definitions

#### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

**3.1.1 authentication** [ITU-T X.800]: See data origin authentication and peer-entity authentication.

**3.1.2 authorization** [ITU-T X.800]: The granting of rights, which includes the granting of access based on access rights.

**3.1.3 content protection** [ITU-T X.1191]: Ensuring that an end user can only use the content that he/she already acquired in accordance with the rights granted to him/her by the rights holder; content protection includes protecting contents from illegal copying and distribution, interception, tampering, unauthorized use, etc.

**3.1.4 rights** [ITU-T X.1191]: Referring to the ability to perform a predefined set of utilization functions for a content item; these utilization functions include permissions (e.g., to view/hear, copy, modify, record, excerpt, sample, keep for a certain period, distribute), restrictions (e.g., play/view/hear for multiple number of times, play/view/hear for certain number of hours), and obligations (e.g., payment, content tracing) that apply to the content and provide the liberty of use as granted to the end user.

**3.1.5 rights expression** [ITU-T X.1191]: Syntactic embodiment of rights in concrete, formal form.

**3.1.6 SCP end-to-end** [ITU-T X.1191]: Service and content protection operating mode wherein content is accessed or exchanged by end devices according to the granted rights using a single service and content protection system.

**3.1.7 SCP bridging** [ITU-T X.1191]: Service and content protection operating mode wherein two or more service and content protection systems are operational on a single device acting as a bridge between these service and content protection systems; content acquired via one service and content protection system can be accessed via another service and content protection system on the bridge according to the granted rights.

**3.1.8 SCP interchange** [ITU-T X.1191]: A more general service and content protection operating mode involving two or more devices, with each device having one or more operational service and content protection systems; the content acquired by one device through one of its service and content protection systems can be securely transferred to and accessed on another device through a different service and content protection system according to the granted rights.

**3.1.9 service protection** [ITU-T X.1191]: Ensuring that an end user can only acquire a service and the content hosted therein by extension as what he/she is entitled to receive; service protection includes protecting service from unauthorized access as IPTV contents traverse through the IPTV service connections.

**3.1.10 service and content protection** [ITU-T X.1191]: A combination of service protection and content protection, or the system or implementation thereof.

## **3.2 Terms defined in this Recommendation**

This Recommendation defines the following term:

**3.2.1 SCP interworking**: General SCP interoperable operation including SCP bridging and SCP interchange

**3.2.2 service**: A set of functionality enabled by a provider for end-users; for example, providing IP connectivity with managed quality of service, providing an IPTV Service, providing a content on demand service, etc.

## **4 Abbreviations and acronyms**

This Recommendation uses the following abbreviations and acronyms:

IPTV	Internet Protocol Television
PVR	Personal Video Recorder
SCP	Service and Content Protection
SCP-B	SCP Bridging
SCP-EE	SCP End-to-End
SCP-IX	SCP Interchange
TD	Terminal Device
WM	Watermark(ing)

## **5 Conventions**

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus this requirement need not be present to claim conformance.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

In the body of this Recommendation and its annexes, the words *shall*, *shall not*, *should*, and *may* sometimes appear, in which case they are to be interpreted, respectively, as *is required to*, *is prohibited from*, *is recommended*, and *can optionally*. The appearance of such phrases or keywords in an appendix or in material explicitly marked as *informative* are to be interpreted as having no normative intent.

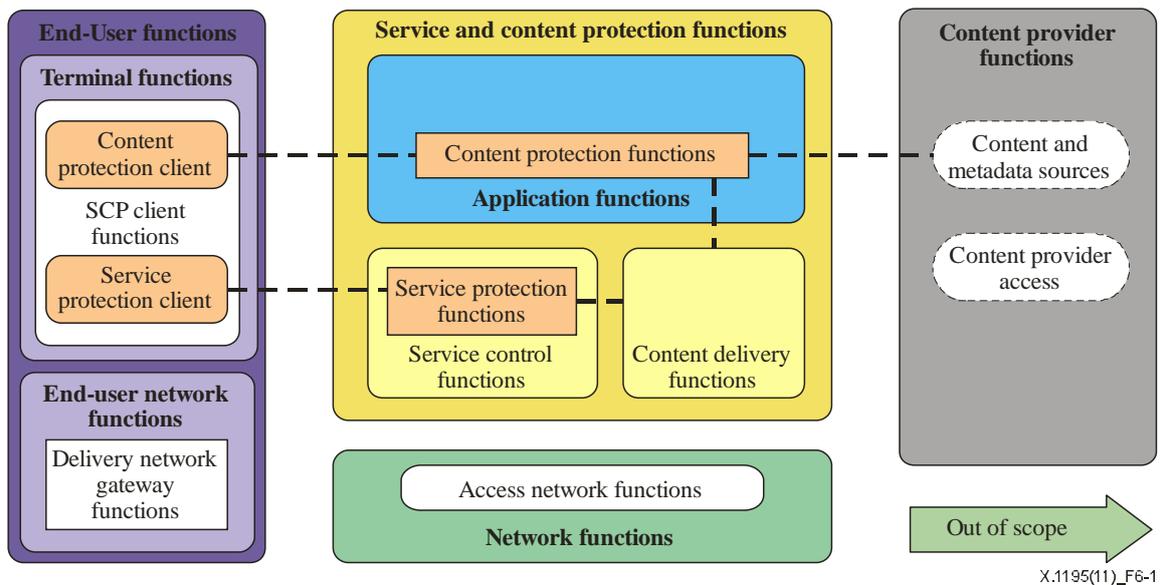
## **6 Introduction**

In general, two or more SCPs are deployed on a single TD. Content acquired via one SCP system (e.g., from a network) can be accessed via another SCP residing on the same device according to the granted rights. It is necessary to support SCP interoperability among multiple security systems using different security mechanisms for the purpose of supporting the seamless time-shifting service (subscribers can store the content and retrieve it later) and place-shifting service (subscribers can view the content anywhere) even with different security mechanisms.

The objective of this Recommendation is to develop a complete set of requirements for the interoperable SCP to support interoperability between multiple SCP mechanisms. This includes interoperable SCP scenarios, interoperable SCP architecture, and interoperable SCP process.

### **6.1 IPTV general architecture and content protection architecture**

The general security architecture for IPTV is depicted in Figure 6-1 below. The general architecture is divided into two primary areas: one considered in-scope for the purpose of considering interoperability based on this Recommendation, and the other considered out-of-scope. The first area encompasses end-user, network provider, and service provider domains, whereas the second area encompasses the content provider domain.

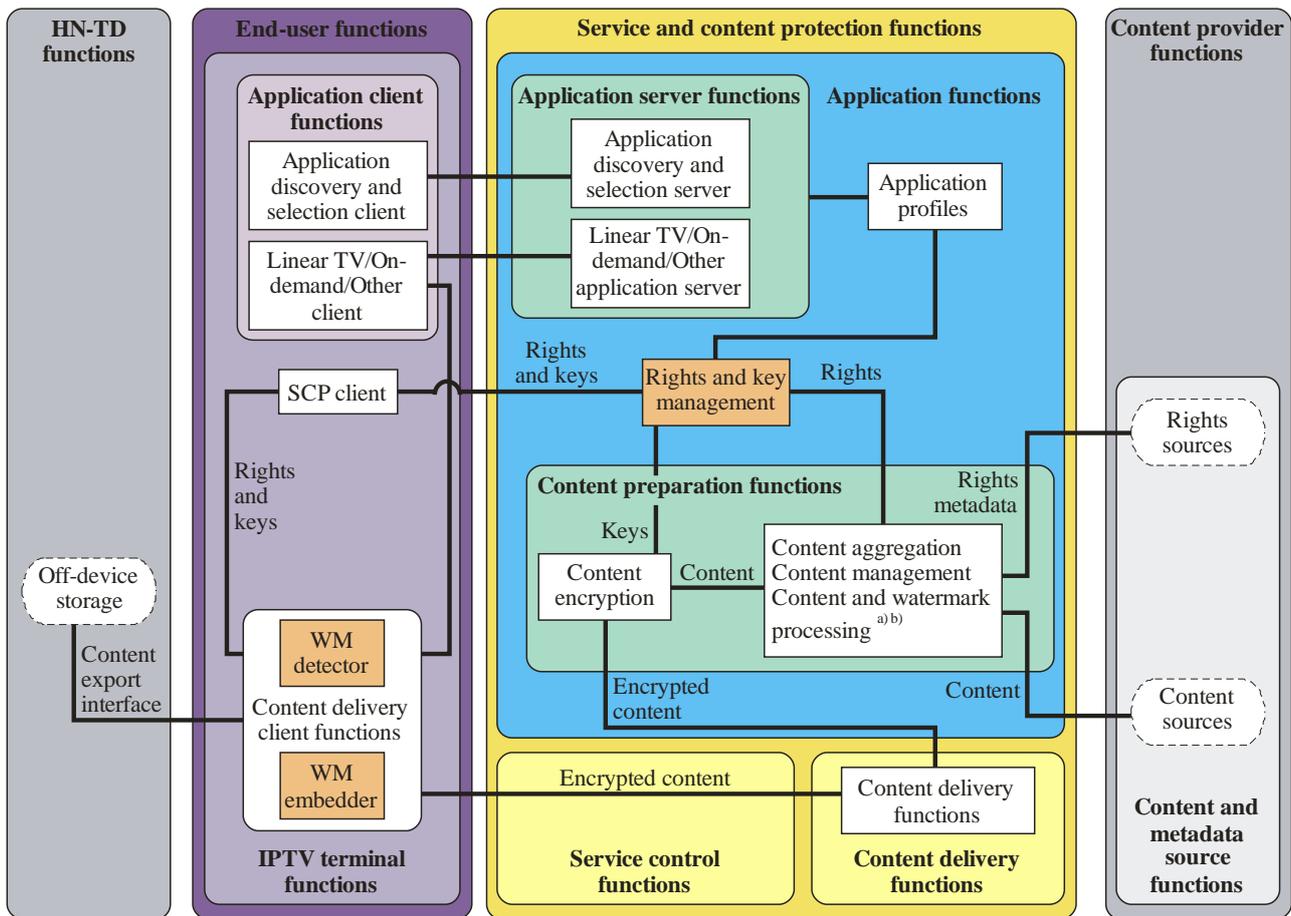


**Figure 6-1 – IPTV general security architecture**

The content protection architecture for IPTV is depicted in Figure 6-2 below.

The primary function of the content protection architecture is to delineate the flow and process of information related to content usage rights and information required to manage and facilitate such rights.

Ultimately, the rights of content use originate with the content provider(s); however, such rights may be modified (e.g., narrowed, or perhaps even widened) by service provider(s) according to their agreements with content providers and their operational and business policies. From an operational and typical legal perspective, an end-user's access and use of content is with the service provider, and not with a content provider.



X.1195(11)\_F6-2

- a) Optional watermark metadata generation to facilitate downstream watermark embedding.
- b) Optional watermark embedder to individuate content to networks, servers, and unicast deliveries.
- c) Optional watermark embedder to individuate multi-cast content instances.
- d) Optional off-device storage: a storage device inside HN-TD.
- e) Optional detector for copy protection watermarks.

NOTE – Objects in grey colour are out of scope of the IPTV security architecture.

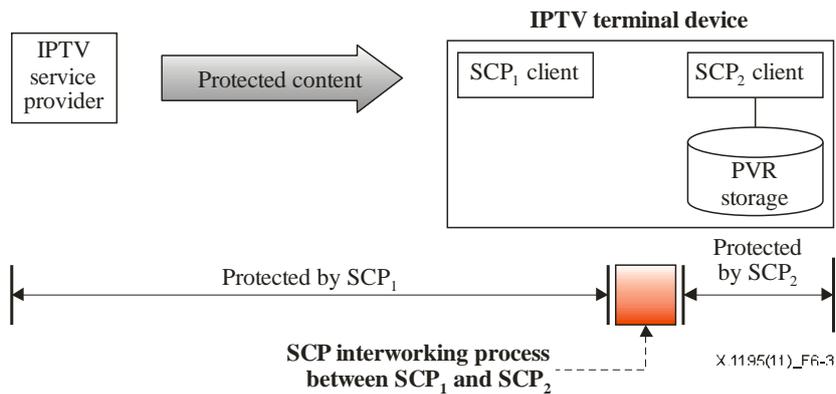
**Figure 6-2 – IPTV content protection architecture**

## 6.2 Overview of the interoperable SCP framework

An IPTV terminal device can have two different SCPs, one for IPTV service and the other for personal video recorder (PVR) function. In general, SCPs have their own proprietary protected content format, encryption scheme, rights expression languages, and metadata description language and protection policies.

If a content which is provided by a certain IPTV service provider and protected by SCP<sub>1</sub> is about to be stored on the PVR storage with SCP<sub>2</sub>'s protection mechanism for later use, protection scheme for the content needs to be changed from SCP<sub>1</sub> to SCP<sub>2</sub> while storing. For the seamless conversion, it needs proprietary or standard protocol between SCP<sub>1</sub> and SCP<sub>2</sub>. Interoperable SCP framework is a recommended standard to provide interoperable conversion environment based on SCP interworking process between two different SCPs.

A comprehensive overview of interoperable SCP framework for IPTV TD is depicted in Figure 6-3.



**Figure 6-3 – Comprehensive overview of the interoperable SCP framework**

An open protocol that is specified in this Recommendation can be introduced for the interoperable SCP framework, or a legacy scheme may take on this role.

## 7 Interoperable SCP framework

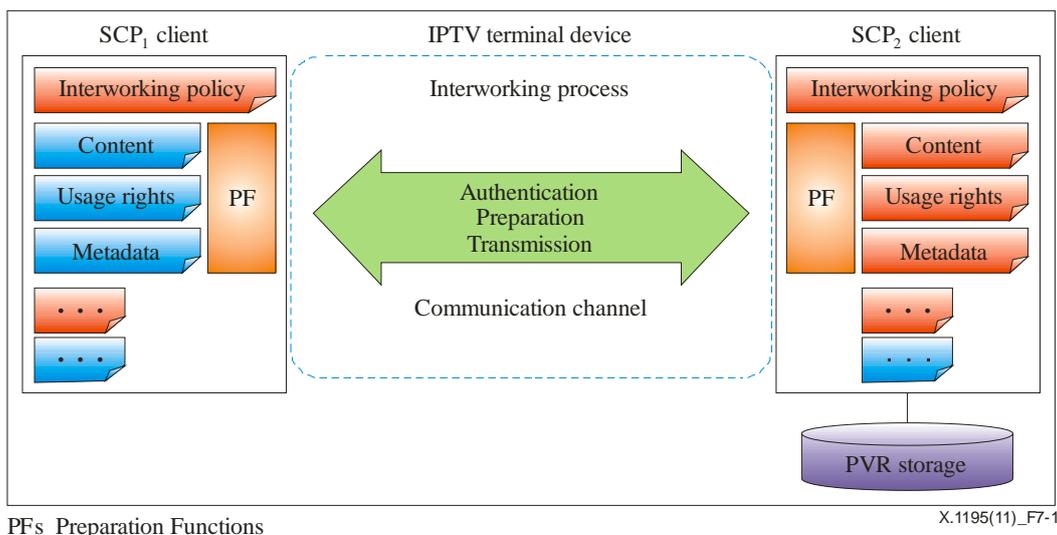
### 7.1 Interoperable SCP architecture

The interoperable SCP architecture consists of two SCP clients having would-be transferable content and related information and interworking process as in Figure 7-1.

Basically, the interworking process is a set of procedures wherein two SCP clients mutually authenticate, process preparation works for the transmission, and transmit interworking data (i.e., content, metadata, and usage rights) from SCP<sub>1</sub> to SCP<sub>2</sub> through a secure communication channel.

Rules and policies supervise the interworking process. Each SCP has its own rules and policies for the interchange of content. Rules and policies may be provided by the IPTV service provider, PVR provider or SCP providers, etc.

A detailed explanation will be described in the following subclauses.



**Figure 7-1 – Interoperable SCP architecture**

### 7.1.1 Interworking policy

For the interworking process, there can be rules, restrictions or policies from IPTV service provider or SCP provider such as 'blacklisted SCPs', 'viewing of the stored content is limited to 1 week', 'interworking is allowed only to the SCPs with certificates from specified authority', etc.

The interworking policy is differentiated from usage rights. Usage rights is a metadata which describes usage permission or the condition of the content itself. On the other hand, the interworking policy is the criteria used to determine whether an SCP<sub>1</sub> client starts the interworking or not against the SCP<sub>2</sub>.

### 7.1.2 Preparation functions

Preparation functions in the SCP client are behaviours to adapt or convert content, metadata and usage rights from SCP<sub>1</sub> client to SCP<sub>2</sub> client. Detailed functions of the preparation functions are described in clause 7.2.3.

### 7.1.3 SCP clients

SCP clients that take part in the interworking process are responsible for sending or receiving content, metadata and usage rights through the converting process, which is done by the preparation functions.

It has the interworking policy and preparation functions for the interworking process. SCP<sub>1</sub> client has the content, metadata and usage rights to be sent to the opponent SCP<sub>2</sub> client. SCP<sub>2</sub> client has also those ones received from SCP<sub>1</sub>.

### 7.1.4 Communication channel

The communication channel is a secure authenticated channel. It is for data transmission between SCP<sub>1</sub> client and SCP<sub>2</sub> one. The channel is required to be accessible only after passing the mutual authentication process between two SCP parties.

The communication channel is usually protected by a secure mechanism.

### 7.1.5 Interworking process

The interworking process comprises of three subprocesses which are authentication, preparation and transmission. Details of the interworking process are explained in clause 7.2.

## 7.2 Interworking processes

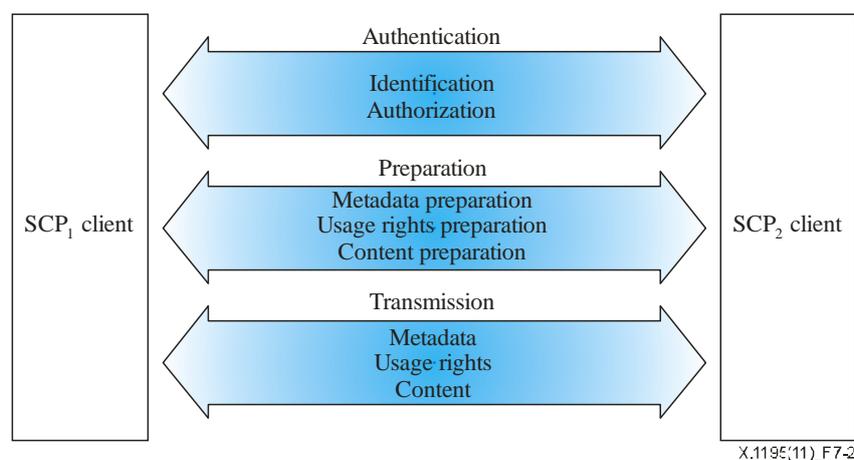


Figure 7-2 – Interworking process

### **7.2.1 Overview**

The interworking process comprises of three subprocesses which are authentication, preparation and transmission.

### **7.2.2 Authentication**

Each SCP party is required to judge whether the other party is a proper target for achieving SCP interworking. It is a preliminary step before data conversion and transmission, and is usually accompanied by the mutual authentication process.

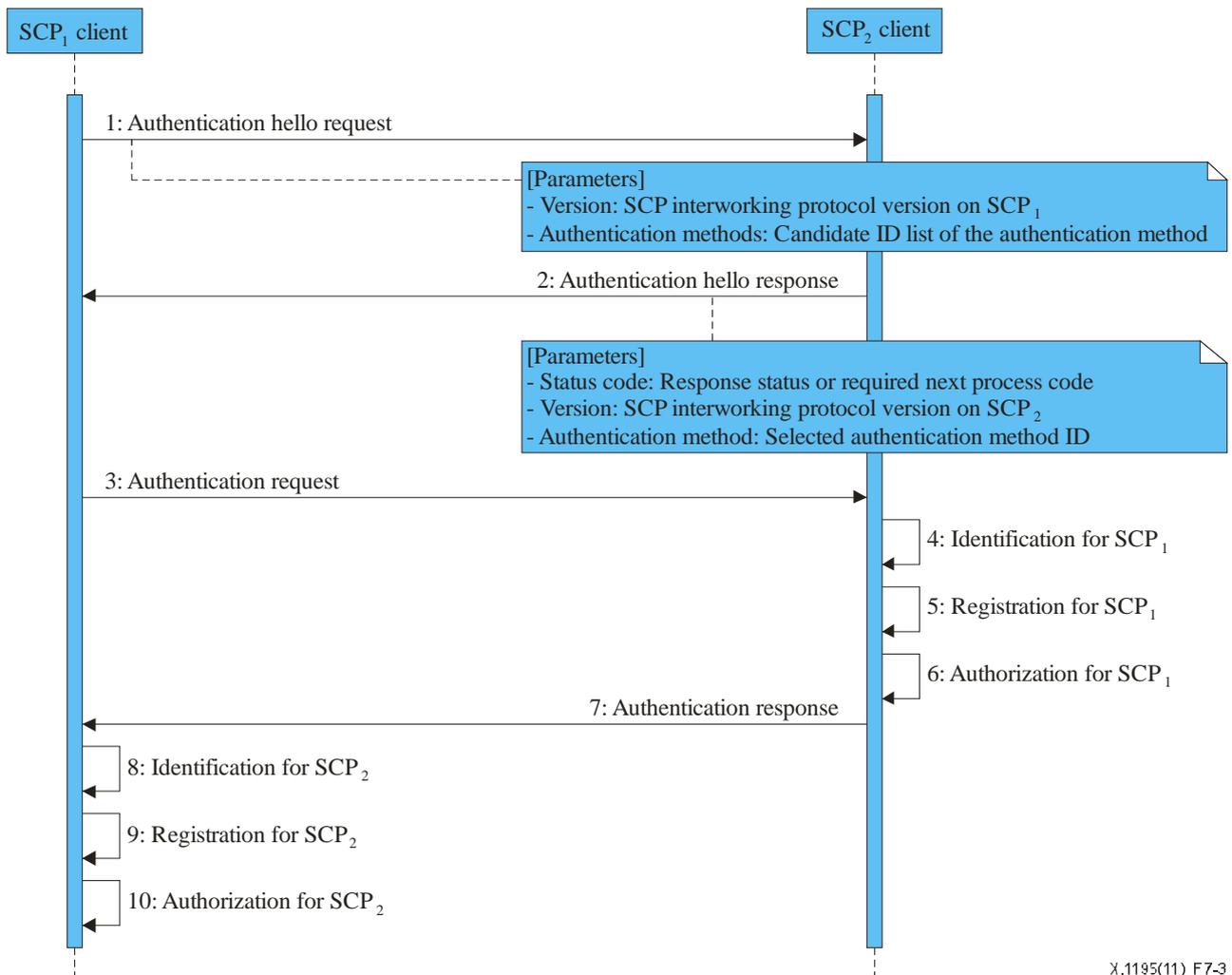
All SCP clients are required to be identified by the corresponding SCP with a unique identifier for later authorization process on the interworking process. This step includes verifying authenticity of the identifier by cryptographic mechanism.

To improve efficiency of the verifying process, some registration function is involved in this step.

All identified SCP clients are required to be authorized by the corresponding SCP with the identifier for further interworking process. Even though an SCP client is identified and verified properly, it can be rejected in the authorization process if the SCP client does not meet the requirements described in the interworking policy. The interworking policy can be managed in the form of a whitelist or blacklist.

#### **7.2.2.1 Authentication protocol**

The SCP interworking process always needs mutual authentication to confirm that both sender and receiver are secure parties. Authentication is recommended to use a well-known and proven technology for secure processing. ITU-T X.509 certificate technology is required to be used as an authentication method. It provides very useful infrastructure to exchange secure data and guarantee trustworthy validation of the participants' authenticity. Various authentication schemes based on ITU-T X.509 certificate such as SSL/TLS can be used as an authentication protocol. Figure 7-3 describes the authentication process of the interworking process.



X.1195(11)\_F7-3

**Figure 7-3 – Authentication protocol**

First, as indicated in the figure above, SCP<sub>1</sub> and SCP<sub>2</sub> clients agree together on an authentication method by exchanging Hello messages. Then the two parties carry out the mutual authentication process as agreed. The processing sequences are:

- 1) SCP<sub>1</sub> client sends an Authentication Hello Request message to SCP<sub>2</sub> to start protocol with the following parameters:
  - Version: SCP interworking protocol version number on SCP<sub>1</sub>.
  - Authentication methods : Candidate ID list of the authentication methods.
- 2) SCP<sub>2</sub> client sends an Authentication Hello Response message to SCP<sub>1</sub> client with the following parameters:
  - Status code: Return value.
  - Version: SCP interworking protocol version number on SCP<sub>2</sub>.
  - Authentication type: Selected authentication type ID.
- 3) SCP<sub>1</sub> client starts the authentication process with the agreed method.
- 4) Identification for SCP<sub>1</sub>: SCP<sub>2</sub> client verifies SCP<sub>1</sub>'s authenticity with its ID.

- 5) Registration for SCP<sub>1</sub>: SCP<sub>2</sub> client registers SCP<sub>1</sub>'s ID. It is used when SCP<sub>1</sub>'s ID is not registered. ID is an encoded subjectPublicKeyInfo component of the ITU-T X.509 digital certificate, which is digested with SHA256 and encoded in base64 format. Appendix VI shows an example how to get this value with the openssl tool.
- 6) Authorization for SCP<sub>1</sub>: SCP<sub>2</sub> client decides if it allows further processing with SCP<sub>1</sub> by its interworking policy.
- 7) SCP<sub>2</sub> client validates the authentication process with the agreed method.
- 8) Identification for SCP<sub>2</sub>: SCP<sub>1</sub> client verifies SCP<sub>2</sub>'s authenticity with its ID.
- 9) Registration for SCP<sub>2</sub>: SCP<sub>1</sub> client registers SCP<sub>2</sub>'s ID. It is used when SCP<sub>2</sub>'s ID is not registered. ID is an encoded subjectPublicKeyInfo component of the ITU-T X.509 digital certificate, which is digested with SHA256 and encoded in base64 format. Appendix VI shows an example how to get this value with the openssl tool.
- 10) Authorization for SCP<sub>2</sub>: SCP<sub>1</sub> client decides if it allows further processing with SCP<sub>2</sub> by its interworking policy.

SSL/TLS or proprietary authentication schemes are able to be used as a mutual authentication method (steps 3 to 7). During the authentication process, it is assumed that the ITU-T X.509 digital certificates of the SCP<sub>1</sub> and SCP<sub>2</sub> clients are sent to each other, based on the agreed authentication method. A proprietary authentication scheme based on the ITU-T X.509 certificate is introduced in Appendix IV.

### **7.2.3 Preparation**

Each SCP party needs to convert the content protection scheme, expression way of the metadata and usage rights before they are transmitted. It is a preliminary step before data transmission and is usually accompanied by exchanging some information on each SCP's preferences, e.g., usage rights or metadata format, bit rates and resolution, etc.

#### **7.2.3.1 Content preparation**

Content preparation is responsible for converting cryptographic algorithm for content protection. Several predefined standard encryption algorithms are able to be considered.

If the communication channel for data transmission is well protected, content can be transmitted with plain text.

Bit rates or resolution of the content can be changed after exchanging information between 2 SCPs in this step.

#### **7.2.3.2 Usage rights preparation**

Usage rights preparation is responsible for converting the expression way of usage rights. Some standard or preferred expression ways of usage rights known to both parties are used for converting.

The converted usage rights are required to maintain the same semantics defined in the original usage rights.

The standard expression ways are the predefined ones that every IPTV TD compliant SCP has to know. The preferred expression way can also be determined after exchanging information between 2 SCPs in this step.

IPTV rights metadata specified by the ITU-T standard may be used for the standard or preferred expression way for usage rights preparation.

### 7.2.3.3 Metadata preparation

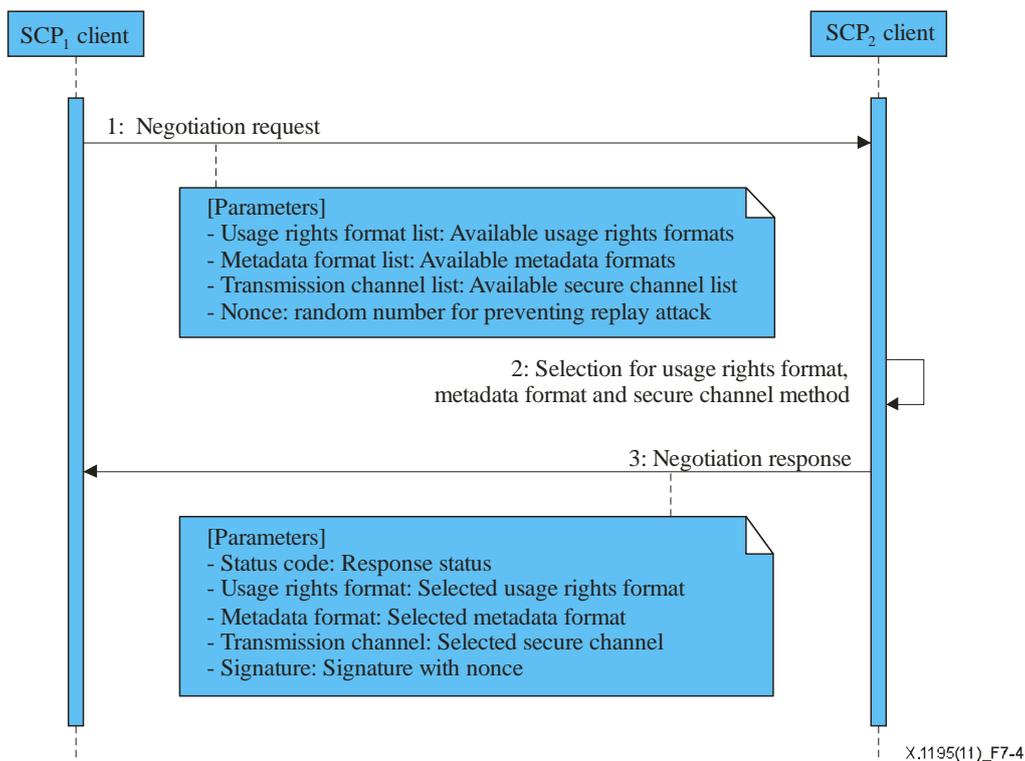
Metadata preparation is responsible for converting the expression way of metadata information. Some standard or preferred expression ways of metadata known to both parties are used for converting.

The converted metadata is required to maintain the same semantics defined in the original metadata.

IPTV metadata specified by ITU-T standard may be used for standard or preferred expression way for metadata preparation.

### 7.2.3.4 Negotiation protocol

Preparation requires a negotiation process to get a common exchangeable format and secure channel method. The preparation is required to be processed right after the authentication protocol within the same connection session.



**Figure 7-4 – Negotiation protocol**

As indicated in Figure 7-4, the SCP<sub>1</sub> client may perform the preparation process after exchanging a negotiation message with SCP<sub>2</sub>. The following are the processing sequences:

- 1) The SCP<sub>1</sub> client sends a NegotiationRequest message to SCP<sub>2</sub> to start the protocol with the following parameters:
  - Usage rights format list: List of available usage rights formats to which SCP<sub>1</sub> is able to send or convert.
  - Metadata format list: List of available metadata formats to which SCP<sub>1</sub> is able to send or convert.
  - Transmission channel list: Available secure channel list that SCP<sub>1</sub> can establish to transfer the data.
  - Nonce: A random number which is used for verifying the whole negotiation message at the next step.

- 2) SCP<sub>2</sub> decides the possible usage rights format, metadata format, and secure transmission channel among the delivered lists from SCP<sub>1</sub>.
- 3) The SCP<sub>2</sub> client sends a NegotiationResponse message to SCP<sub>1</sub> with the following parameters:
  - Status code: Return value.
  - Usage rights format: Selected usage rights format to which SCP<sub>2</sub> is able to receive or convert.
  - Metadata format: Selected metadata format to which SCP<sub>2</sub> is able to receive or convert.
  - Transmission channel: Selected secure channel that the SCP<sub>2</sub> can establish to receive the data.
  - Signature: Digital signature with nonce value.

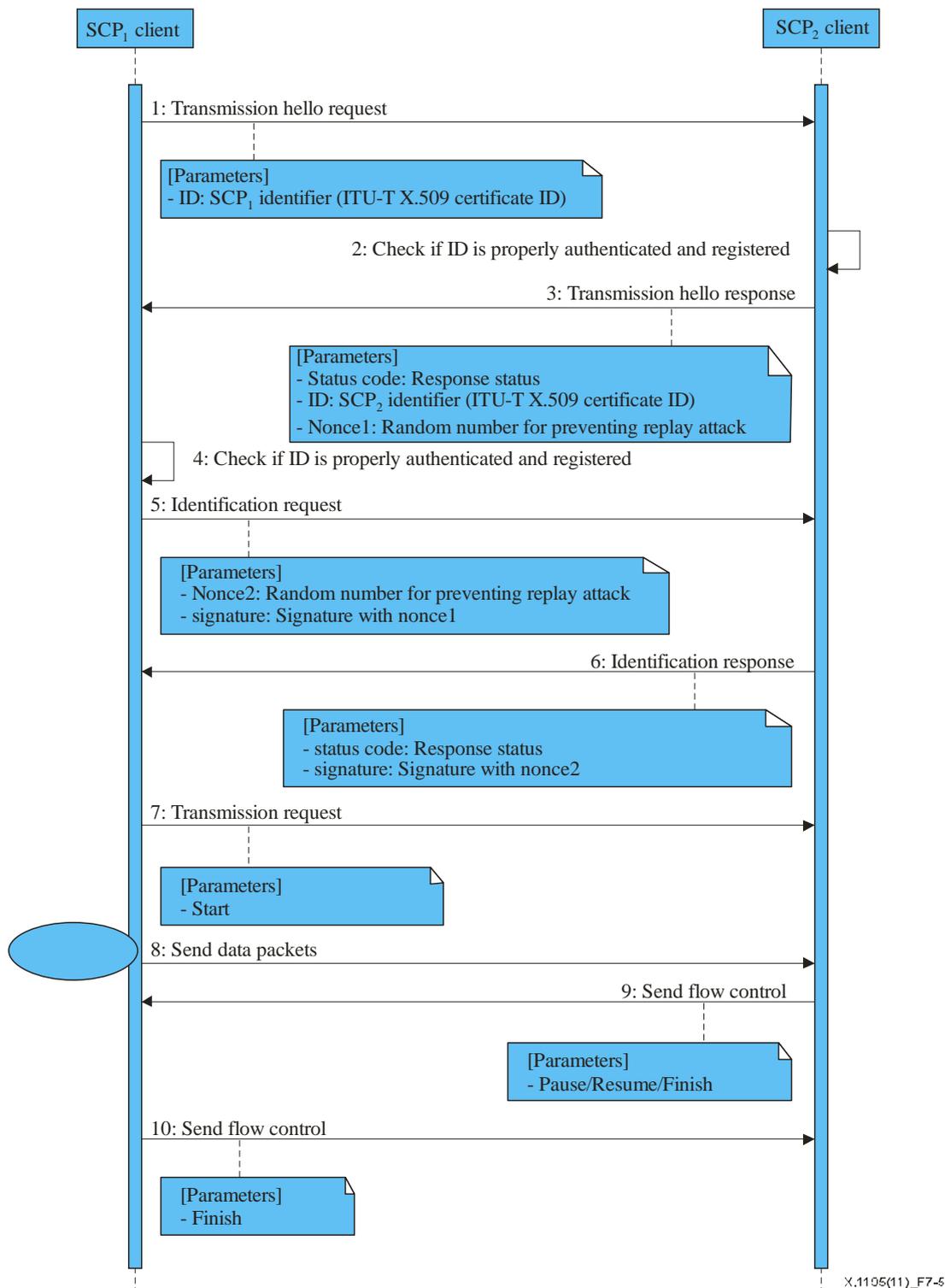
After this protocol, the SCP<sub>1</sub> prepares usage rights and metadata in the negotiated format.

## **7.2.4 Transmission**

After the preparation step, the content, metadata, and usage rights, which are in a converted form, are transmitted from the SCP<sub>1</sub> client to the SCP<sub>2</sub> client through a secure communication channel.

### **7.2.4.1 Transmission protocol**

Transmission requires a secure channel to exchange data. Before this protocol is started, a secure channel is required to be established between the SCP<sub>1</sub> client and the SCP<sub>2</sub> client based on the result of the negotiation process. This protocol is required to start with the identification process first through the secure connection channel because transmission protocol can be invoked directly without the authentication and negotiation process, based on previously registered and negotiated information.



**Figure 7-5 – Transmission protocol**

As indicated in Figure 7-5, the SCP<sub>1</sub> client may perform the data transmission process after re-validation for SCP<sub>2</sub>'s ID and vice versa. The ID re-validation process confirms that two parties of the open channel are already mutually authenticated. The following are the processing sequences:

- 1) SCP<sub>1</sub> Client sends Transmission Hello Request message to SCP<sub>2</sub> to start protocol with the following parameters:
  - ID: ITU-T X.509 certificate ID of SCP<sub>1</sub> client.

- 2) SCP<sub>2</sub> checks the SCP<sub>1</sub> ID to see if it is authenticated and negotiated. If it is not, SCP<sub>2</sub> requests authentication process to SCP<sub>1</sub> which is indicated in the status code. Otherwise, it simply needs the ID re-verification request process.
- 3) SCP<sub>2</sub> client sends a Transmission Hello Response message to SCP<sub>1</sub> client with the following parameters:
  - Status code: Return value. This value can indicate SCP<sub>1</sub>'s next action.
  - ID: ITU-T X.509 certificate ID of SCP<sub>2</sub> client.
  - Nonce1: A random number which is used for verifying the SCP<sub>1</sub>'s ID at the next step.
- 4) SCP<sub>1</sub> checks the SCP<sub>2</sub> ID to see if it is authenticated and negotiated. If it is not, SCP<sub>1</sub> disconnects the channel, and starts the authentication and preparation process again. Otherwise, it proceeds to the next step.
- 5) SCP<sub>1</sub> client sends an Identification Request message to SCP<sub>2</sub> client with the following parameters. This message is necessary to prove the authenticity of the previous SCP<sub>1</sub>'s ID.
  - Nonce2: A random number which is used for verifying the SCP<sub>1</sub>'s ID at the next step.
  - Signature: Digital signature with Nonce1 value.
- 6) SCP<sub>2</sub> client sends an Identification Response message to SCP<sub>2</sub> client with the following parameters. This message is necessary to prove the authenticity of the previous SCP<sub>2</sub>'s ID.
  - Status code: Return value.
  - Signature: Digital signature with Nonce2 value.
- 7) SCP<sub>1</sub> client sends a Transmission Request message to SCP<sub>2</sub> client to send the flow control to start data transmission.
- 8) SCP<sub>1</sub> client sends data packets until it finishes sending all data or receives the flow control data from SCP<sub>2</sub>. In this step, content, metadata and usage rights data are transferred as a predefined packet form.
- 9) SCP<sub>2</sub> client sends a Transmission Response message to SCP<sub>1</sub> client to send the flow control data to SCP<sub>1</sub> to manage the receiving buffer.
- 10) SCP<sub>1</sub> client sends a Transmission Request message to SCP<sub>2</sub> client to send the flow control to finish data transmission.

## 8 Interworking process protocol specification

### 8.1 Overview of the protocol message

The interworking process consists of three steps which are: authentication, preparation, and transmission protocols. Each protocol has several XML-based protocol messages.

**Table 8-1 – Interworking process protocol messages**

Protocol messages		Description
Authentication	AuthenticationHelloRequest	Request for starting authentication protocol
	AuthenticationHelloResponse	Response for AuthenticationHelloRequest
	AuthenticationRequest	Request for authentication process
	AuthenticationResponse	Response for AuthenticationRequest
Preparation	NegotiationRequest	Request for negotiation process for preparation
	NegotiationResponse	Response for NegotiationRequest

**Table 8-1 – Interworking process protocol messages**

Protocol messages		Description
Transmission	TransmissionHelloRequest	Request starting transmission protocol
	TransmissionHelloResponse	Response for TransmissionRequestHello
	IdentificationRequest	Request for identification of requester
	IdentificationResponse	Response for IdentificationRequest
	TransmissionRequest	Request for starting data transmission
	TransmissionResponse	Response for TransmissionRequest

## 8.2 Authentication protocol

### 8.2.1 AuthenticationHelloRequest

**Table 8-2 – AuthenticationHelloRequest parameters**

Parameter	Mandatory/Optional
protocolVersion	M
authenticationMethods	M
extensions	O

#### protocolVersion

The version number is used to let the Responder know which version of interworking process function reacts. It specifies the highest version which the sender can support. A higher version of the protocol is recommended to support backward compatibility for a lower version.

#### authenticationMethods

It is a list of authentication methods which can be used as authentication protocol in the next step. Authentication methods can be a well-known protocol name or a proprietary one. It is represented with multiple URIs which specifies identification (e.g., urn:ETRI:iwp:authentication:2010-1) or name (e.g., SSL/TLS) of the protocol.

#### extensions

It is used for extension purposes to exchange additional proprietary information.

### 8.2.2 AuthenticationHelloResponse

**Table 8-3 – AuthenticationHelloResponse parameters**

Parameter	Mandatory/Optional
@status	M
protocolVersion	M
selectedAuthenticationMethod	M
extensions	O

## status

This attribute value specifies the result of the AuthenticationHelloRequest message. If there is no error during the request message process, it would be 'Success'. Otherwise, it specifies one of the corresponding reasons described in clause B.3.

## protocolVersion

The version number is used to let the Requester know which version of the interworking process function works. It specifies a protocol version which the responder can support. The responder can choose the same or lower version than the one which the requester sent.

## selectedAuthenticationMethod

It specifies the selected authentication method which is one of the requested possible candidates from the requester and would be used as an authentication protocol in the next step. It is represented with a single URL which specifies identification (e.g., urn:ETRI:iwp:authentication:2010-1) or name (e.g., SSL/TLS) of the protocol.

## extensions

It is used for extension purposes to exchange additional proprietary information.

### 8.2.3 AuthenticationRequest

The AuthenticationRequest message can be replaced with any kind of a secure authentication method which is agreed between the Requester and the Responder at the AuthenticationHelloRequest/Response step. The AuthenticationRequest step can be divided into several sub-steps according to the selected authentication protocol.

### 8.2.4 AuthenticationResponse

The AuthenticationResponse message can be replaced with any kind of a secure authentication method which is agreed between the Requester and the Responder at the AuthenticationHelloRequest/Response step. The AuthenticationResponse step can be divided into several sub-steps according to the selected authentication protocol.

## 8.3 Negotiation protocol

### 8.3.1 NegotiationRequest

Table 8-4 – NegotiationRequest parameters

Parameter		Mandatory/Optional
items		M
	rightsFormats	M
	metadataFormats	M
	secureChannels	M
nonce		M
extensions		O

## items

The items element specifies the list of formats or methods which are required to be agreed for the adaptation between the Requester and Responder before data transmission. It describes three child elements: the rights formats, the metadata formats, and the secure channel list.

### items/rightsFormats

The rightsFormats sub-element specifies a list of content usage rights formats that the Requester can support, from which the Responder is required to choose a specific one as an interoperable rights format. It is represented with multiple URIs which specifies the ID (e.g., urn:odrl:oma:profile-1023) of the standard rights notation.

### items/metadataFormats

The metadataFormats sub-element specifies a list of content metadata formats that the Requester can support, from which the Responder is required to choose a specific one as an interoperable metadata format. It is represented with multiple URIs which specifies the ID (e.g., urn:mpeg:mpeg7:schema:2001:profile-1023) of the standard metadata notation.

### items/secureChannels

The secureChannel sub-element specifies a list of possible secure channels that the Requester can support, from which the Responder is required to choose a specific one as a sharable secure channel. It is represented with multiple URIs which specifies the ID (e.g., urn:etri:securechannel:2007:11) or the name (e.g., DTCP, HDCP or SSL/TLS) of the secure channel protocol.

### nonce

This specifies a disposable random number to prevent the replay attack while checking the message integrity. It follows the rule in clause B.5.

### extensions

These are used for extension purposes to exchange additional proprietary information.

## 8.3.2 NegotiationResponse

Table 8-5 – NegotiationResponse parameters

Parameter		Mandatory/Optional
@status		M
items		M
	selectedRightsFormat	M
	selectedMetadataFormat	M
	selectedSecureChannel	M
extensions		O
signature		M

### status

This attribute value specifies a result of the NegotiationRequest message. If there is no error during the request message process, it would be 'Success'. Otherwise, it specifies one of the corresponding reasons described in clause B.3.

### items

The items element specifies a selected format or method which is chosen from a candidate list proposed by the Requester. It describes three child elements: the chosen rights format, the metadata format, and the secure channel list.

### **items/selectedRightsFormat**

The selectedRightsFormats sub-element specifies a chosen content usage rights format that the Responder can support. It is represented with a URI which specifies the identification (e.g., urn:odrl:oma:profile-1023) of the standard rights notation.

### **items/selectedMetadataFormat**

The selectedMetadataFormats sub-element specifies a chosen content metadata format that the Responder can support. It is represented with a URI which specifies the ID (e.g., urn:mpeg:mpeg7:schema:2001: profile-1023) of the standard metadata notation.

### **items/selectedSecureChannel**

The selectedSecureChannel sub-element specifies a chosen secure channel that the Responder can support. It is represented with a URI which specifies the identification (e.g., urn:etri:securechannel:2007:11) or name (e.g., DTCP) of the secure channel protocol.

### **extensions**

It is used for extension purposes to exchange additional proprietary information.

### **signature**

It specifies a digital signature value of the concatenated messages which are the previous NegotiationRequest message and this NegotiationResponse one. The concatenated message is made through the canonicalization of each message and the concatenation of two messages without a blank. The signing method is an "Enveloped XML Signature". The algorithm of canonicalization and digital signature follows the rules in clause B.6.

## **8.4 Transmission protocol**

### **8.4.1 TransmissionHelloRequest**

**Table 8-6 – TransmissionHelloRequest parameters**

<b>Parameter</b>	<b>Mandatory/Optional</b>
entityId	M
extensions	O

### **entityId**

The entityId is used to let the Responder know which entity wants to communicate and retrieve the ID DB stored with the Responder. It is required to register at the Authentication step to check later whether it is an authenticated ID and already negotiated on its adaptation formats. It is digested with SHA256 and encoded with base64 string of the subjectPublicKeyInfo component in the ITU-T X.509 digital certificate which is DER type encoded. Appendix VI shows an example of how to get this value with the openssl tool.

### **extensions**

It is used for extension purposes to exchange additional proprietary information.

## 8.4.2 TransmissionHelloResponse

**Table 8-7 – TransmissionHelloResponse parameters**

Parameter	Mandatory/Optional
@status	M
entityId	M
nonce	M
extensions	O

### **status**

This attribute value specifies the result of the TransmissionHelloRequest message. If there is no error during the request message process, it would be 'Success'. Otherwise, it specifies one of the corresponding reasons which are described in clause B.3.

### **entityId**

The entityId is used to let the Requester know which entity wants to communicate and retrieve the ID DB stored with the Requester. It is required to register at the Authentication step and to check later to see whether it is an authenticated ID and already negotiated on its adaptation formats. It is SHA256 and encoded with base64 string of the subjectPublicKeyInfo component in the ITU-T X.509 digital certificate which is DER type encoded. Appendix VI shows an example how to get this value with the openssl tool.

### **nonce**

It specifies a disposable random number to prevent a replay attack while checking the message integrity. It follows the rule in clause B.5.

### **extensions**

It is used for extension purposes to exchange additional proprietary information.

## 8.4.3 IdentificationRequest

**Table 8-8 – IdentificationRequest parameters**

Parameter	Mandatory/Optional
nonce	M
extensions	O
signature	M

### **nonce**

It specifies a disposable random number to prevent a replay attack while checking the message integrity. It follows the rule in clause B.5.

### **extensions**

It is used for extension purposes to exchange additional proprietary information.

### **signature**

It specifies the digital signature value of this protocol message. The scope of the signing text is the IdentificationRequest element itself and the method of signing is "Enveloped XML Signature". The algorithm of canonicalization and digital signature follows the rules in clause B.6.

#### 8.4.4 IdentificationResponse

**Table 8-9 – IdentificationResponse parameters**

Parameter	Mandatory/Optional
@status	M
extensions	O
signature	M

##### **status**

This attribute value specifies the result of the IdentificationRequest message. If there is no error during the request message process, it would be 'Success'. Otherwise, it specifies one of the corresponding reasons which are described in clause B.3

##### **extensions**

These are used for extension purposes to exchange additional proprietary information.

##### **signature**

This specifies the digital signature value of the concatenated messages which are part of the previous IdentificationRequest message and this IdentificationResponse one. The concatenated messages are made by the canonicalization of each message and the concatenation of two messages without blank. The signing method is "Enveloped XML Signature". The algorithm of canonicalization and digital signature follows clause B.6 rules.

#### 8.4.5 TransmissionRequest

**Table 8-10 – TransmissionRequest parameters**

Parameter	Mandatory/Optional
@controlCommand	M
extensions	O

##### **controlCommand**

The transmissionRequest message is used to transmit control command to the Responder before sending content, metadata and usage rights packets. Control command has two types as follows:

- start : To start data (content, metadata, usage rights) transmission
- finish : To terminate data transmission

##### **extensions**

It is used for extension purposes to exchange additional proprietary information.

#### 8.4.6 TransmissionResponse

**Table 8-11 – TransmissionResponse parameters**

Parameter	Mandatory/Optional
@status	M
@controlCommand	M
extensions	O

## status

This attribute value specifies the result of the TransmissionRequest message. If there is no error during the request message process, it would be 'Success'. Otherwise, it specifies one of the corresponding reasons which are described in clause B.3.

## controlCommand

The TransmissionRequest message is used to transmit control command to the Responder before sending content, metadata and usage rights packets. The control command has three types as follows:

- pause: to stop temporary data transmission
- resume: to restart data transmission
- finish: to terminate data transmission.

## extensions

It is used for extension purposes to exchange additional proprietary information.

## 8.5 Transmission packet format

The interworking process uses a specific data transmission packet which has a binary format. The packet is used to convey metadata, usage rights and content during a transmission protocol.

ID (2 bytes)	Packet info (2 bytes)
Payload length (4 bytes)	
Footer Length (1 byte)	Reserved (7 bytes)
Payload	
Footer	

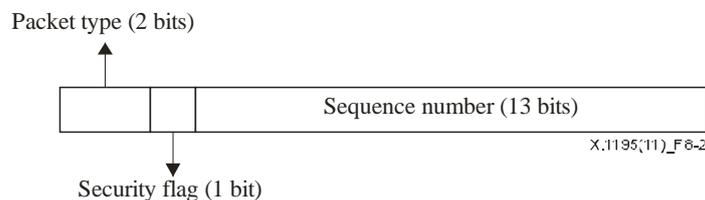
**Figure 8-1 – Transmission packet format**

### 8.5.1 ID (2 bytes)

It represents the ID of the interworking process packet. It is required to use the fixed value 0xFFAA.

### 8.5.2 Packet info (2 bytes)

It contains the information of the packet: packet type, security flag, and sequence number.

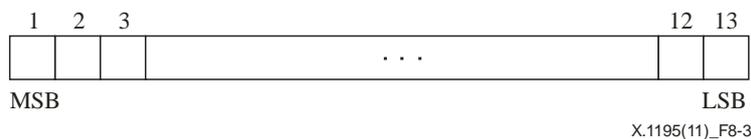


**Figure 8-2 – Structure of packet Info**

- Packet type (2 bits)
  - 00: metadata
  - 01: usage rights
  - 10: reserved
  - 11: content

- Security flag (1 bit)
  - 0: plain data
  - 1: ciphered data
- Sequence number (13 bits)

A large content can be transferred by a number of small pieces of packets for efficient error correction. It is a sequence number of the packet ranging from 0 to  $2^{13} - 1$ . The number starts from 0 and increases by 1. If the number of packets is over  $2^{13} - 1$ , the next sequence number becomes 0 again.



**Figure 8-3 – Structure of sequence number**

### 8.5.3 Payload length (4 bytes)

It is the size of the content in the packet to be transmitted which ranges from 0 to  $2^{32} - 1$ .

### 8.5.4 Footer length (1 byte)

It is the footer data size which ranges from 0 to 255.

### 8.5.5 Reserved (7 bytes)

Not used in this version.

### 8.5.6 Payload (payload length bytes)

This area is for the data itself to be transmitted and allocated as much as the payload length specified in clause 8.5.3.

### 8.5.7 Footer (footer length bytes)

This area is additional data for integrity checking such as hash value, and is allocated as much as the footer length specified in clause 8.5.4.

## Annex A

### Scenarios of SCP-B or SCP-IX deployed in IPTV TD

(This annex forms an integral part of this Recommendation.)

This clause describes three possible scenarios requiring SCP interchange between service security and content security.

#### A.1 Definitions of terms used in Figure A.1

- SCP-IN: Input port through which the IPTV content protected by the SCP comes in
- SCP-OUT: Output port through which the IPTV content protected by the SCP goes out
- SCP-B: SCP bridging (see clause I.2)
- SCP-IX: SCP interchange (see clause I.3).

#### A.2 Scenario 1: SCP with SCP-IX

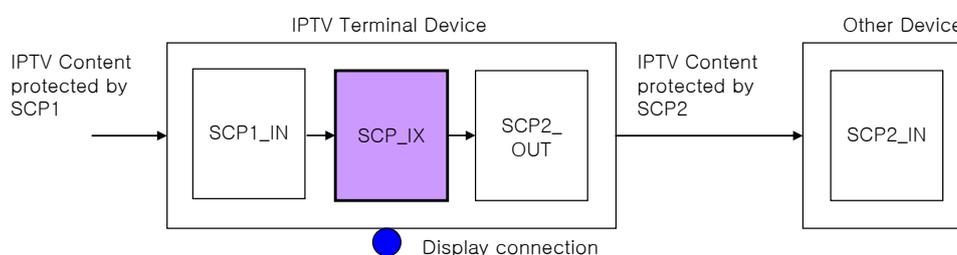


Figure A.1 – SCP with SCP-IX

IPTV TD (terminal device) in this case has SCP with SCP-IX to support interoperability between the IPTV TD without storage that adopts only specific service security, and the external device with storage having specific content protection only.

To support secure and flexible connectivity to any kind of external device adopting various content protection mechanisms, IPTV TD should have SCP-IX rather than case-to-case implementation for security connection between two devices.

#### A.3 Scenario 2: SCP with optional SCP-B and storage

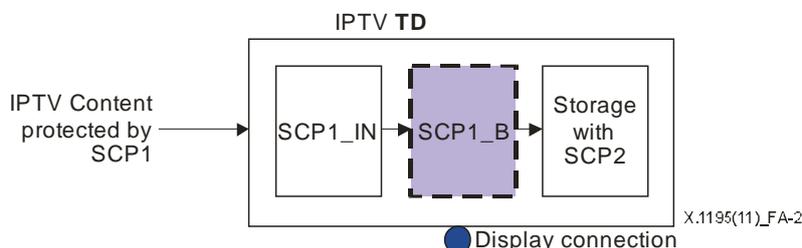
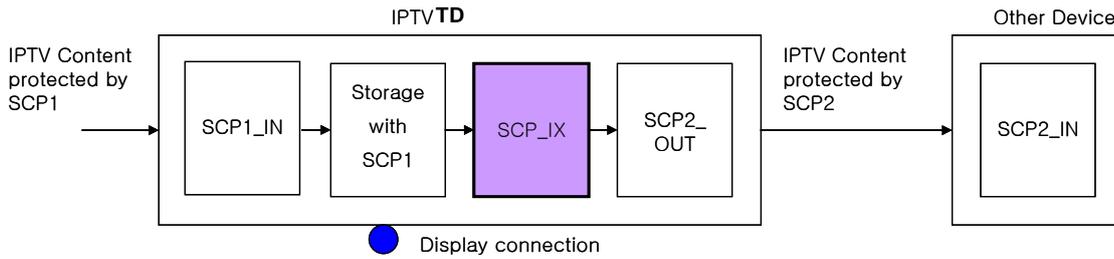


Figure A.2 – SCP with optional SCP-B and storage

IPTV TD in this case has SCP with SCP-B to support interoperability between service protection and content protection on a single device.

To support flexible connectivity to any kind of internal storage that adopts various content protection mechanisms, IPTV TD is recommended to have SCP-B rather than the case-to-case implementation for security connection between service protection and content protection.

#### A.4 Scenario 3: SCP with storage and SCP-IX



**Figure A.3 – SCP with storage and SCP-IX**

In this case, IPTV TD has SCP with storage and SCP-IX supporting interoperability between the internal content protection mechanism and the external one.

## Annex B

### Basic type and premises of the interworking process protocol

(This annex forms an integral part of this Recommendation.)

#### B.1 Request type

All interworking processes start with the Request-type inherited message.

```
<complexType name="Request" abstract="true"/>
```

#### B.2 Response type

All interworking processes respond with a Respond type inherited-message to request a message. The Respond message has a 'status' attribute to describe success or failure of the previous request message.

```
<complexType name="Response" abstract="true">  
  <attribute name="status" type="iwp:Status" use="required"/>  
</complexType>
```

#### B.3 Status type

Status type, as an attribute of the Respond element, is used to describe the success or reason for failure of the previous request message. If the value of status is not "Success", it means that an error or an exception has occurred while a message is being processed.

When the Response message arrives, the value of the status attribute is checked. If "Success" is not outputted, the interworking process is required to stop at that moment. Temporal security data such as nonce value or key value is recommended to be removed at once as well.

Table B.1 shows the possible error message strings that can be specified in the status attribute value of the Response element. This string can expand with proprietary error strings.

**Table B.1 – Status type examples**

Status value	Meaning
Abort	Request message was denied because of unknown error.
AuthenticationFail	Request message was denied because the requester was not authenticated.
InvalidCertificate	Authentication failed because certificate is not in ITU-T X.509 format.
InvalidCertificateChain	Authentication failed because certificate chain is not valid.
InvalidMessage	Request message was denied because it is an undefined one.
InvalidPacket	Transmission failed because the packet format is not valid.
MalformedRequest	Request message has semantic error.
NotNegotiated	Request message was denied because the request ID has no live negotiation record.
NotRegistered	Request message was denied because the request ID is not registered.
NotSupportedVersion	Responder cannot support requester's protocol version.

**Table B.1 – Status type examples**

Status value	Meaning
SignatureError	Process was stopped because of signature validation failure.
Success	Request message was successfully processed.

#### **B.4 Extension type**

The Extension type is optionally used to send an additional message or information that is not defined in this specification. If the value of attribute "critical" is not "true", the extension element could be ignored.

```
<complexType name="Extensions">
  <sequence maxOccurs="unbounded">
    <any namespace="##any" minOccurs="0"/>
  </sequence>
  <attribute name="critical" type="boolean"/>
</complexType>
```

#### **B.5 Nonce type**

Nonce is used for the prevention of the replay attack which may occur during an interworking processing. Nonce is a disposable random number which is required to have at least 16 bytes and a maximum of 512 bytes when it is represented with base64 encoded string.

```
<simpleType name="Nonce">
  <restriction base="base64Binary">
    <minLength value="16"/>
    <maxLength value="512"/>
  </restriction>
</simpleType>
```

#### **B.6 Canonicalization and digital signature**

This specification uses a digital signature technology based on PKI to verify the integrity and authenticity of the message. An input message for digital signature is required to be canonicalized before calculating a digital signature. By default, this specification uses the following algorithms for digital signing and canonicalization:

- Canonicalization: <http://www.w3.org/2001/10/xml-exc-c14n#>
- Signature: <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>
- Digest: <http://www.w3.org/2001/04/xmllenc#sha256>

## Annex C

### Interworking process protocol schema

(This annex forms an integral part of this Recommendation.)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:iwp="urn:itu-t:sg17:2010:04-iwp"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  targetNamespace="urn:itu-t:sg17:2010:04-iwp"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <import namespace="http://www.w3.org/2001/04/xmlenc#"
    schemaLocation="http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-
schema.xsd"/>
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
    schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-
core-schema.xsd"/>

  <!-- Elements -->
  <element name="authenticationHelloRequest" type="iwp:AuthenticationHelloRequest"/>
  <element name="authenticationHelloResponse" type="iwp:AuthenticationHelloResponse"/>
  <element name="authenticationRequest" type="iwp:AuthenticationRequest"/>
  <element name="authenticationResponse" type="iwp:AuthenticationResponse"/>
  <element name="negotiationRequest" type="iwp:NegotiationRequest"/>
  <element name="negotiationResponse" type="iwp:NegotiationResponse"/>
  <element name="transmissionHelloRequest" type="iwp:TransmissionHelloRequest"/>
  <element name="transmissionHelloResponse" type="iwp:TransmissionHelloResponse"/>
  <element name="identificationRequest" type="iwp:IdentificationRequest"/>
  <element name="identificationResponse" type="iwp:IdentificationResponse"/>
  <element name="transmissionRequest" type="iwp:TransmissionRequest"/>
  <element name="transmissionResponse" type="iwp:TransmissionResponse"/>

  <!-- Basic Types -->
  <complexType name="Request" abstract="true"/>

  <complexType name="Response" abstract="true">
    <attribute name="status" type="string" use="required"/>
  </complexType>

  <simpleType name="Version">
    <restriction base="string">
      <pattern value="\d{1,2}\.\d{1,9}"/>
    </restriction>
  </simpleType>

  <complexType name="idListType">
    <sequence>
      <element name="id" type="anyURI" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <simpleType name="Nonce">
    <restriction base="base64Binary">
      <minLength value="16"/>
      <maxLength value="512"/>
    </restriction>
  </simpleType>

  <complexType name="Extensions">
    <sequence maxOccurs="unbounded">
      <any namespace="##any" minOccurs="0"/>
    </sequence>
  </complexType>
</schema>
```

```

    <attribute name="critical" type="boolean"/>
</complexType>

<complexType name="Identifier">
  <sequence>
    <element name="keyIdentifier" type="base64Binary"/>
  </sequence>
</complexType>

<simpleType name="TransmissionRequestControlType">
  <restriction base="string">
    <enumeration value="start"/>
    <enumeration value="finish"/>
  </restriction>
</simpleType>

<simpleType name="TransmissionResponseControlType">
  <restriction base="string">
    <enumeration value="pause"/>
    <enumeration value="resume"/>
    <enumeration value="finish"/>
  </restriction>
</simpleType>

<!-- iwp-TransmissionHelloRequest-->
<complexType name="TransmissionHelloRequest">
  <complexContent>
    <extension base="iwp:Request">
      <sequence>
        <element name="entityId" type="iwp:Identifier"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- iwp-TransmissionHelloResponse -->
<complexType name="TransmissionHelloResponse">
  <complexContent>
    <extension base="iwp:Response">
      <sequence>
        <element name="entityId" type="iwp:Identifier"/>
        <element name="nonce" type="iwp:Nonce"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- iwp-AuthenticationHelloRequest -->
<complexType name="AuthenticationHelloRequest">
  <complexContent>
    <extension base="iwp:Request">
      <sequence>
        <element name="protocolVersion" type="iwp:Version"/>
        <element name="authenticationMethods" type="iwp:idListType"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- iwp-AuthenticationHelloResponse -->
<complexType name="AuthenticationHelloResponse">
  <complexContent>
    <extension base="iwp:Response">
      <sequence>
        <element name="protocolVersion" type="iwp:Version"/>
        <element name="selectedAuthenticationMethod" type="anyURI"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

    </extension>
  </complexContent>
</complexType>

<!-- iwp-AuthenticationRequest -->
<complexType name="AuthenticationRequest" abstract="true"/>
<complexType name="AuthenticationResponse" abstract="true"/>

<!-- iwp-NegotiationRequest -->
<complexType name="NegotiationRequest">
  <complexContent>
    <extension base="iwp:Request">
      <sequence>
        <element name="items" type="iwp:NegotiationRequestItemType"/>
        <element name="nonce" type="iwp:Nonce"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- iwp-NegotiationResponse -->
<complexType name="NegotiationResponse">
  <complexContent>
    <extension base="iwp:Response">
      <sequence>
        <element name="items" type="iwp:NegotiationResponseItemType"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
        <element name="signature" type="base64Binary"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="NegotiationRequestItemType">
  <sequence>
    <element name="rightsFormats" type="iwp:idListType"/>
    <element name="metadataFormats" type="iwp:idListType"/>
    <element name="secureChannels" type="iwp:idListType"/>
  </sequence>
</complexType>
<complexType name="NegotiationResponseItemType">
  <sequence>
    <element name="selectedRightsFormat" type="anyURI"/>
    <element name="selectedMetadataFormat" type="anyURI"/>
    <element name="selectedSecureChannel" type="anyURI"/>
  </sequence>
</complexType>

<!-- iwp-IdentificationRequest -->
<complexType name="IdentificationRequest">
  <complexContent>
    <extension base="iwp:Request">
      <sequence>
        <element name="nonce" type="iwp:Nonce"/>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
        <element name="signature" type="base64Binary"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- iwp-IdentificationResponse -->
<complexType name="IdentificationResponse">
  <complexContent>
    <extension base="iwp:Response">
      <sequence>
        <element name="extensions" type="iwp:Extensions" minOccurs="0"/>
        <element name="signature" type="base64Binary"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

    </complexContent>
  </complexType>

  <!-- iwp-TransmissionRequest -->
  <complexType name="TransmissionRequest">
    <complexContent>
      <extension base="iwp:Request">
        <attribute name="controlCommand" type="iwp:TransmissionRequestControlType"
use="required"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- iwp-TransmissionResponse -->
  <complexType name="TransmissionResponse">
    <complexContent>
      <extension base="iwp:Response">
        <attribute name="controlCommand" type="iwp:TransmissionResponseControlType"
use="required"/>
      </extension>
    </complexContent>
  </complexType>
</schema>

```

# Appendix I

## Interoperable SCP types

(This appendix does not form an integral part of this Recommendation.)

Interoperable SCP types are classified into at least three modes: SCP end-to-end (SCP-EE), SCP bridging (SCP-B), and SCP interchange (SCP-IX).

### I.1 SCP end-to-end (SCP-EE)

**SCP-EEB:** Using a single SCP, two or more devices exchange and access content according to the granted rights. This mode is required to be the simplest mode to be implemented since only a single SCP is used.

### I.2 SCP bridging (SCP-B)

**SCP-B:** On a single TD, two or more SCPs are deployed. Content acquired via one SCP system (e.g., from a network) can be accessed via another SCP residing on the same device according to the granted rights.

### I.3 SCP interchange (SCP-IX)

**SCP-IX:** This case is characterized by two or more devices with each device having one or more deployed SCPs. Content acquired by one device through one of its SCPs can be securely transferred to and accessed on another device through a different SCP according to the granted rights.

Figure I.1 illustrates a model of the cases described above.

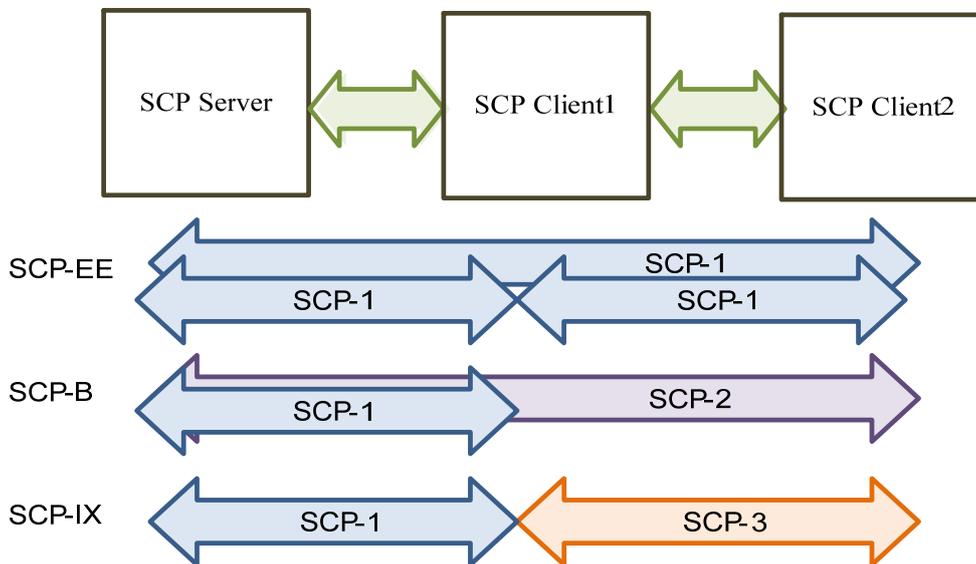


Figure I.1 – SCP interoperability mode

## Appendix II

### Security considerations

(This appendix does not form an integral part of this Recommendation.)

#### II.1 Channel security

When the content is transmitted from SCP<sub>1</sub> client to SCP<sub>2</sub> in an unencrypted form, the transmission channel must be protected by some means, or other protection schemes have to be provided.

A secure channel can be established based on PKI environment which needs a secure authentication before its establishment. The secure channel based on secure authentication is called secure authenticated channel (SAC). There are already proven and very well-known secure channel protocols between two devices such as SSL/TLS, IPSec, DTCP or HDCP etc.

#### II.2 Possible attacks

There are several possible network attacks during an authentication process. Once proprietary authentication scheme is adopted instead of proven one, it is required to consider if there is no possibility on the security hole in terms of the following possible attacks.

##### II.2.1 Masquerading attack

A masquerading attack is used to disguise an unauthorized device as a specific eligible one by sending forged authentication data to the target device which requires an authentication process. In order to avoid this kind of attack, the authentication process needs to have a re-checking step for the opponent's authenticity by requesting specific data which is only made with secure and private information of the opponent. The digital signature scheme based on PKI is an example of this solutions.

##### II.2.2 Man-in-the-middle attack

Man-in-the-middle attack is to intercept all data transmitted between two parties legally and transfer them to its original destination so that two parties cannot know that data are leaking. In order to avoid this kind of attack, an authentication process needs to have a checking step for the opponent's reliability from the specific authority which has a role to manage all the participants' identification. The trusted authority (TA) scheme based on PKI is an example of this solution.

##### II.2.3 Replay attack

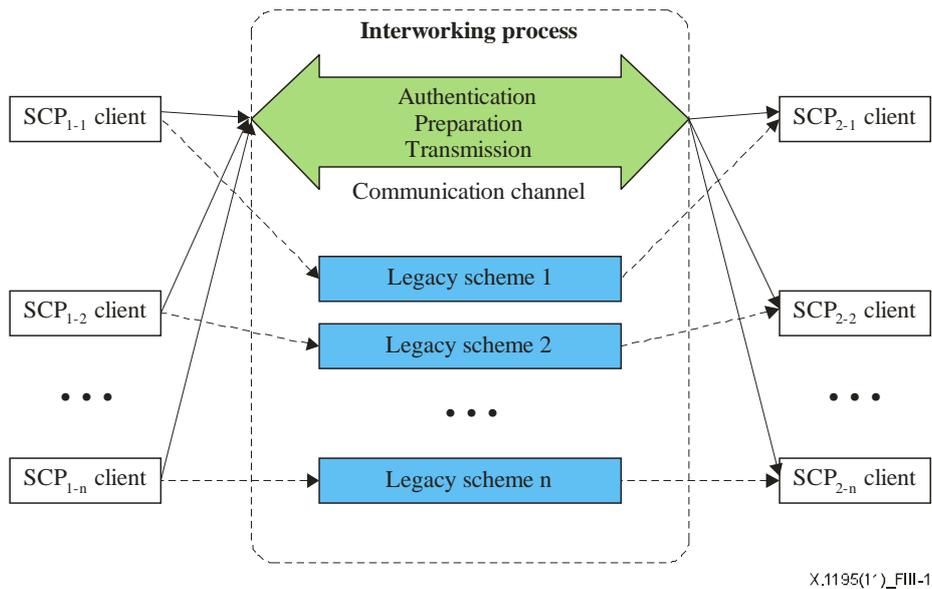
A replay attack is the reuse of confirming data which was captured during the previous authentication process between legal parties so as to pretend that a certain disguised device is a proper party. In order to avoid this kind of attack, the authentication process needs to add disposable random numbers into the message during re-checking the step for the opponent's authenticity. The nonce scheme is an example of this solution.

## Appendix III

### SCP interoperability and legacy scheme

(This appendix does not form an integral part of this Recommendation.)

Figure III.1 illustrates that various SCP<sub>1</sub>s (at the service security side) can interoperate with various SCP<sub>2</sub>s (at the content security side) through an open protocol that is specified in this Recommendation, or through legacy schemes. In this figure, SCP<sub>1</sub>s means service security systems such as NDS, Irdeto or Nagravision, etc. SCP<sub>2</sub>s means content security systems such as MS-DRM, OMA-DRM or fair play DRM, etc.

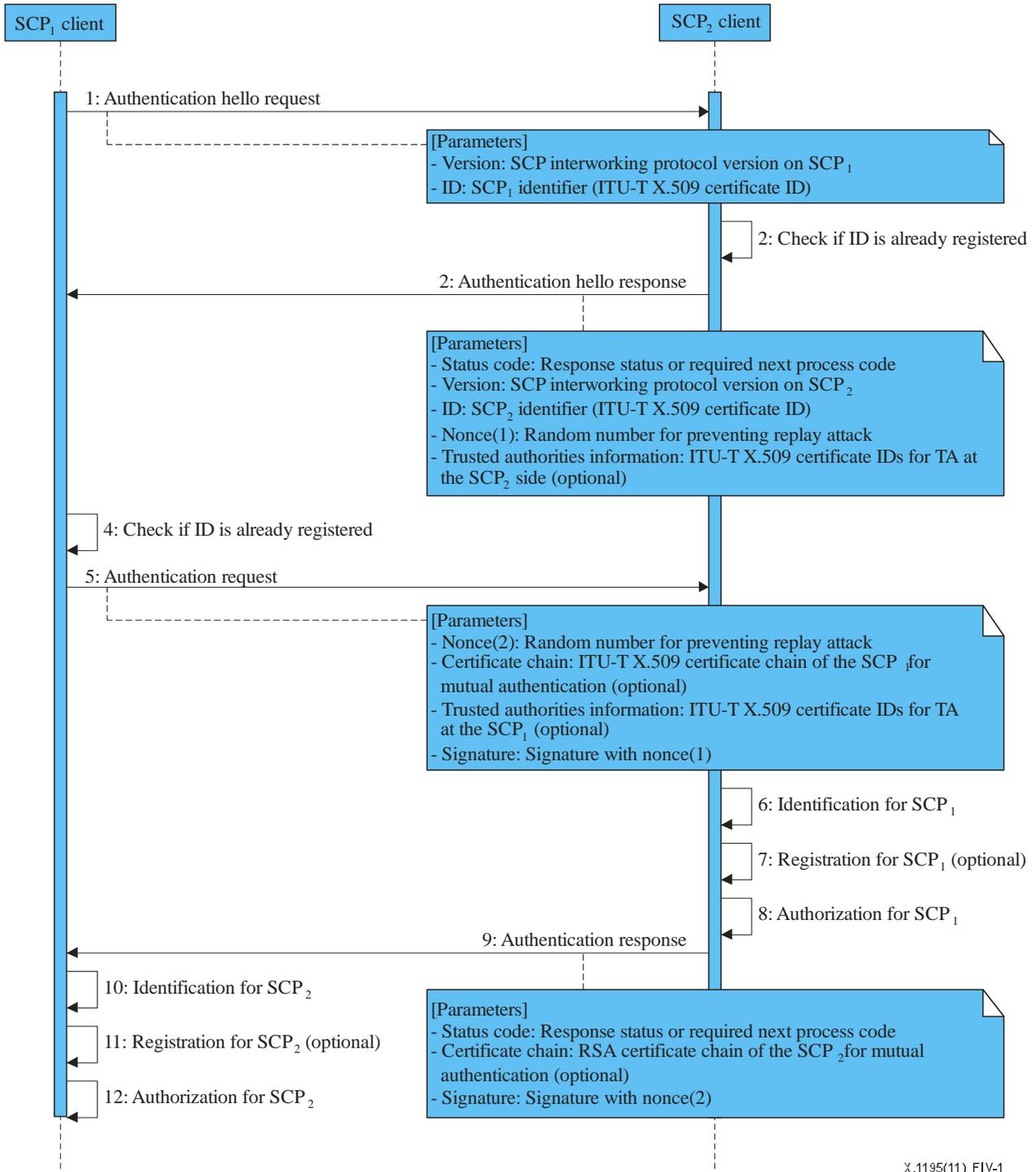


**Figure III.1 – SCP interoperability and legacy scheme**

## Appendix IV

### Proprietary authentication scheme

(This appendix does not form an integral part of this Recommendation.)



X.1195(11)\_FIV-1

Figure IV.1 – Authentication protocol

As indicated in the figure above, the SCP<sub>1</sub> client may perform an ITU-T X.509 certificate based authentication request followed by Hello message to SCP<sub>2</sub>. The processing sequences are:

- 1) SCP<sub>1</sub> client sends an Authentication Hello Request message to SCP<sub>2</sub> to start the protocol with the following parameters:
  - Version: SCP interworking protocol version number on SCP<sub>1</sub>.
  - ID: ITU-T X.509 certificate ID of SCP<sub>1</sub> Client.
- 2) SCP<sub>2</sub> checks if SCP<sub>1</sub> ID is already registered. If not, SCP<sub>2</sub> requests the registration process from SCP<sub>1</sub> providing the trusted authority information. Otherwise, it simply needs the ID verification request process.
- 3) SCP<sub>2</sub> client sends an Authentication Hello Response message to SCP<sub>1</sub> client with the following parameters:
  - Status code: Return value.
  - Version: SCP interworking protocol version number on SCP<sub>2</sub>.
  - ID: ITU-T X.509 certificate ID of SCP<sub>2</sub> client.
  - Nonce(1): A random number which is used for verifying the SCP<sub>1</sub>'s ID at the next step.
  - Trusted authorities information (optional): A list of CA certificate IDs used for giving information to SCP<sub>1</sub> that ITU-T X.509 certificates issued by the specified trusted authorities are only allowed. It is used when SCP<sub>1</sub>'s ID is not registered.
- 4) SCP<sub>1</sub> checks if SCP<sub>2</sub> ID is already registered. If not, SCP<sub>1</sub> requests the registration process from SCP<sub>2</sub> providing the trusted authority information. Otherwise, it simply needs the ID verification request process.
- 5) SCP<sub>1</sub> client sends an Authentication Request message to SCP<sub>2</sub> with the following parameters:
  - Nonce(2): A random number which is used for verifying the SCP<sub>2</sub>'s ID at the next step.
  - Certificate chain (optional): ITU-T X.509 certificate chain issued by a specific CA that the SCP<sub>2</sub> has indicated. It is used when SCP<sub>1</sub>'s ID is not registered.
  - Trusted authorities information (optional): A list of CA certificate IDs used for giving information to SCP<sub>1</sub> that ITU-T X.509 certificates issued by the specified trusted authorities are only allowed. It is used when SCP<sub>1</sub>'s ID is not registered.
  - Signature: Digital signature with Nonce(1) value.
- 6) Identification for SCP<sub>1</sub>: SCP<sub>2</sub> client verifies SCP<sub>1</sub>'s authenticity with its ID and signature.
- 7) Registration for SCP<sub>1</sub> (optional): Registration of the SCP<sub>1</sub>'s ID. It is used when SCP<sub>1</sub>'s ID is not registered.
- 8) Authorization for SCP<sub>1</sub>: SCP<sub>2</sub> client decides if it allows further processing with SCP<sub>1</sub> by its interworking policy.
- 9) SCP<sub>2</sub> client sends an Authentication Response message to SCP<sub>1</sub> with the following parameters:
  - Status code: Return value.
  - Certificate chain (optional): ITU-T X.509 certificate chain issued by a specific CA that the SCP<sub>1</sub> has indicated. It is used only when SCP<sub>1</sub>'s ID is not registered.
  - Signature: Digital signature with Nonce(2) value
- 10) Identification for SCP<sub>2</sub>: SCP<sub>1</sub> client verifies SCP<sub>2</sub>'s authenticity with its ID and signature.
- 11) Registration for SCP<sub>2</sub>: Registration of the SCP<sub>2</sub>'s ID. It is used when SCP<sub>2</sub>'s ID is not registered.

- 12) Authorization for SCP<sub>2</sub>: SCP<sub>1</sub> client decides if it allows further processing with SCP<sub>2</sub> by its interworking policy.

## Appendix V

### Sample interworking process protocol messages

(This appendix does not form an integral part of this Recommendation.)

#### V.1 AuthenticationHelloRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<authenticationHelloRequest xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd">
  <protocolVersion>1.0</protocolVersion>
  <authenticationMethods>
    <id>TLS</id>
    <id>urn:ETRI:iwp:authentication:2010-1</id>
  </authenticationMethods>
</authenticationHelloRequest>
```

#### V.2 AuthenticationHelloResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<authenticationHelloResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd"
  status="Success">
  <protocolVersion>1.0</protocolVersion>
  <selectedAuthenticationMethod>TLS</selectedAuthenticationMethod>
</authenticationHelloResponse>
```

#### V.3 NegotiationRequest

```
<?xml version="1.0" encoding="utf-8"?>
<negotiationRequest xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd">
  <items>
    <rightsFormats>
      <id>urn:mpeg:mpeg21:2006:01-REL-M2X-NS</id>
      <id>urn:odrl:oma:profile-1023</id>
      <id>urn:cas:cci:bits-presentation</id>
      <id>urn:etri:REL:2007:11</id>
    </rightsFormats>
    <metadataFormats>
      <id>urn:tva:metadata</id>
      <id>urn:mpeg:mpeg7:schema:2001:profile-1023</id>
      <id>urn:iwp:metadata:2007:11</id>
    </metadataFormats>
    <secureChannels>
      <id>DTCP</id>
      <id>SSL</id>
      <id>urn:etri:securechannel:2007:11</id>
    </secureChannels>
  </items>
  <nonce>b0837559f9c780da799214524dc2dde7228=</nonce>
</negotiationRequest>
```

## V.4 NegotiationResponse

```
<?xml version="1.0" encoding="utf-8"?>
<negotiationResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd"
  status="Success">
  <items>
    <selectedRightsFormat>urn:iwp:REL:2007:11</selectedRightsFormat>
    <selectedMetadataFormat>urn:iwp:metadata:2007:11</selectedMetadataFormat>
    <selectedSecureChannel>urn:iwp:securechannel:2007:11</selectedSecureChannel>
  </items>
  <signature> hKAU8EQNgNM90NX+mDCdEvfBOQaRfYU88nZfYy3RJibtE12uyZI1GowVA98YtQop
KZBQvr9NEyqT0e5JIpHhAI2F7n6bgFkiQcOOdot8wLGhBVy1rs/Tn/PaOqluS60
Dgn3IcnUbMtYL7hCwX/FVtwW6hnzYIjOC7b4XCRaFrP3boAy1+BmhBbBcy+7OhNA
z//2ABF1RGEAodmUBCuxq96ITMaZxhHivjjWe2Rm2KWZQBrKR4oe874tAUgDuGzP
y7O9HZMi4muF1NfcG0yIAR6euo/WtdZBHjonpH2f29+bA7/5PYYSIgpzfzmdWMPPr0
BALJ4M1vPooQ5MtIn3J40g==</signature>
</negotiationResponse>
```

## V.5 TransmissionHelloRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<transmissionHelloRequest xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd">
  <entityId>
    <keyIdentifier>cMJpQJeKkmturkRRt0PrUOGWXiE</keyIdentifier>
  </entityId>
</transmissionHelloRequest>
```

## V.6 TransmissionHelloResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<transmissionHelloResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd"
  status="Success">
  <entityId>
    <keyIdentifier>QrWVs4KBY+MNfeWOhdBKm3+BfaE</keyIdentifier>
  </entityId>
  <nonce>b0837559f9c780da799214524dc2dde7228</nonce>
</transmissionHelloResponse>
```

## V.7 IdentificationRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<transmissionHelloResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwpl.0.xsd"
  status="Success">
  <entityId>
    <keyIdentifier>QrWVs4KBY+MNfeWOhdBKm3+BfaE</keyIdentifier>
  </entityId>
  <nonce>b0837559f9c780da799214524dc2dde7228</nonce>
</transmissionHelloResponse >
```

## V.8 IdentificationResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<identificationResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwp1.0.xsd"
  status="Success">
  <signature>Pn/Ibd59ITxTHySjn7geIP4GS15qeaQv3bSITLNL5EnaSdZ+9zl4aAs0rbopUtid
37H2k+ijFF5iotwLg5r1oEHgYgkW4T+5snIGAAPQiLb08ssi/xwmqVzHhaVVPKCH
AKx4VB+ERMSVwMUKGk206ZUHkTxo8rW2jb0wI+Y4SmesFxoR77YtO13Eb3bfTPrN
0ALyqXyME/wBFpdmI4PDectjT8MyieaTkmmFDNPolqb1y2B2ZnAg5cIStm6DSm6F
3zJucbHhduBzSPSI0owwacijZxPeOIHqabs7ZW8FESZcVan/98Xn1xHSQHbJMpFv
boknszuX+qZ4Y1fPcoqw5Q==</signature>
</identificationResponse>
```

## V.9 TransmissionRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<transmissionRequest xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwp1.0.xsd"
  controlCommand="start"/>
```

## V.10 TransmissionResponse

```
<?xml version="1.0" encoding="UTF-8"?>
<transmissionResponse xmlns="urn:itu-t:sg17:2010:04-iwp"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:itu-t:sg17:2010:04-iwp iwp1.0.xsd"
  controlCommand="pause"
  status="Success"/>
```

## Appendix VI

### Calculation of ITU-T X.509 certificate

(This appendix does not form an integral part of this Recommendation.)

The subjectPublicKeyInfo component in the ITU-T X.509 digital certificate can be extracted easily with openssl function by the command below:

```
openssl x509 -pubkey -noout -in <certificate> | openssl base64 -d | dd bs=1 skip=24 2>dev>null
```

The calculation of ITU-T X.509 certificate needs to be digested and encoded to use the text type identifier by the whole command below:

```
openssl x509 -pubkey -noout -in <certificate> | openssl base64 -d | dd bs=1 skip=24 2>dev>null |  
openssl sha256 -binary | openssl base64
```

All ITU-T X.509 digital certificates used in this command are required to be encoded as DER format.

## Bibliography

- [b-IETF RFC 2246] IETF RFC 2246 (1999), *The TLS Protocol Version 1.0*.
- [b-IETF RFC 3339] IETF RFC 3339 (2002), *Date and Time on the Internet: Timestamps*.
- [b-IETF RFC 4122] IETF RFC 4122 (2005), *A Universally Unique Identifier (UUID) URN Namespace*.
- [b-ISO/IEC 21000-5] ISO/IEC 21000-5:2004/Amd 2:2007, *Information technology – Multimedia framework (MPEG-21) – Part 5: Rights Expression Language – Amendment 2: DAC (Dissemination and Capture Profile)*.
- [b-CTP] CTP v1.0 (2009), *D-Cinema System Specification Compliance Test Plan*, Digital Cinema Initiative.
- [b-FIPS 180-2] FIPS 180-2 (2008), *Specifications for the Secure Hash Standard*. National Institute of Standards and Technology.
- [b-OMA DRM] OMA DRM v2.0 (2005), *Extensions for Broadcast Support*, OMA-TS-DRM-XBS-V1\_0-20051209-D.
- [b-PKCS-1] PKCS-1 (2002), *RSA Cryptography Standard*, RSA Laboratories.
- [b-SMPTE 430-2] SMPTE 430-2 (2006), *D-Cinema Operations-Digital Certificate*, SMPTE Technology Committee DC28 on D-Cinema.
- [b-Stallings] Stallings, W. (2003), *Network Security Essentials*, Prentice Hall.
- [b-XML signature] *XML Signature Syntax and Processing* (2002), W3C Recommendation.





## SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
<b>Series X</b>	<b>Data networks, open system communications and security</b>
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems