International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Series J
Supplement 9
(04/2020)

SERIES J: CABLE NETWORKS AND TRANSMISSION OF TELEVISION, SOUND PROGRAMME AND OTHER MULTIMEDIA SIGNALS

Embedded common interface for exchangeable CA/DRM solutions; System validation

ITU-T J-series Recommendations – Supplement 9

Supplement 9 to ITU-T J-series Recommendations

Embedded common interface for exchangeable CA/DRM solutions; System validation

Summary

Supplement 9 to the ITU-T J-series Recommendations addresses scenarios for system validation for the embedded common interface (ECI) for exchangeable CA/DRM solutions and complements ECI-related ITU-T Recommendations covering the ECI ecosystem.

This ITU-T Supplement is a transposition of the ETSI standard [b-ETSI GS ECI 002] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J Suppl. 9	2020-04-23	9	11.1002/1000/14288

Keywords

CA/DRM, validation.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

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

Compliance with this publication is voluntary. However, the publication may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the publication 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 publication is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this publication 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 publication development process.

As of the date of approval of this publication, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this publication. 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/.

Table of Contents

Introduction

This ITU-T Supplement is a transposition of the ETSI standard [b-ETSI GS ECI 002] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI.

The **ECI** system combines security with interoperability to provide a flexible and future-proof content protection system. It is an open, standardized system, which allows CA/DRM vendors to implement a wide range of products and permits consumers to easily switch between vendors on **ECI** compliant **Customer Premises Equipment (CPEs)**.

An **ECI Ecosystem** as specified in [ITU-T J.1011], [ITU-T J.1012] together with further ECI-related ITU-T Recommendations addresses important attributes, such as trustworthiness, flexibility and scalability due to software-based implementation, exchangeability fostering a future-proof solution and enabling innovation. Further aspects are applicability to content distributed via different types of networks, including classical digital broadcasting, Internet protocol television (IPTV) and over the top (OTT) services.

In order to support an implementation of the ECI Ecosystem, a set of procedures and life-cycle oriented use cases with regard to the evaluation of the ECI system architecture and associated functionalities is necessary.

Supplement 9 to ITU-T J-series Recommendations

Embedded common interface for exchangeable CA/DRM solutions; System validation

1 Scope

Supplement 9 to ITU-T J-series Recommendations contains a set of life-cycle oriented use cases reflecting the usage of components of an **ECI** system from its installation via its usage for content-protected media up to its playout to an external device. For implementations of an **ECI Ecosystem** as described in [ITU-T J.1011] the evaluation of the system architecture is of high importance with respect to verifying the integrity and correctness of the features specified in the ECI-related ITU-T Recommendations. The requirements for such a system are given in [ITU-T J.1010].

The **ECI** system aims at the exchangeability of conditional access (CA) and digital rights management (DRM) systems in the user's end device by defining appropriate interfaces between such systems and the device. End-users are enabled to install security clients on their devices to ensure interoperability with the services and devices of their choice. The platform operator, in collaboration with the content provider, can select the most suitable technology for a chosen application and can offer the corresponding application to his customers for download. The following features are supported by an **ECI** system:

- Provisioning of a software container for a CA respectively DRM kernel, called an **ECI Client**,
- Implementation of multiple software containers in a device for the support of more than one protection scheme,
- Installation of **ECI Clients** is separated from the installation of other CPE software,
- Support for smartcard-less or smartcard-based protection systems,
- Support for the user to discover and download the appropriate kernel,
- Support for chip-set security, also known as Advanced Security,
- Applicable to classical digital broadcasting, IPTV and OTT services.

The fulfilment of these features is done via defined interfaces that are available for an **ECI Client**. The characteristics of these interfaces are described in clause 6.

Furthermore, several test cases are described in order to show the correctness and the completeness of the **ECI** architecture as described in [ITU-T J.1012], [ITU-T J.1013], [ITU-T J.1014], [ITU-T J.1015] and [b-ITU-T J-Suppl.8]. Test cases described in clauses 7 to 10 include the installation of **ECI Host** and **ECI Client**, the installation of a second **ECI Client** and the decryption of a protected content. Clause 11 shows the processing steps for a re-encryption of content whereas clause 12 describes the play-out of content to an external device. Besides these technically oriented tests cases, the handling of security aspects and the provisioning of trust within an **ECI Ecosystem** is described in clause 13.

2 References

[ITU-T J.1010] Recommendation ITU-T J.1010 (2016), *Embedded common interface for exchangeable CA/DRM solutions; Use cases and requirements*.

[ITU-T J.1011] Recommendation ITU-T J.1011 (2016), *Embedded common interface for exchangeable CA/DRM solutions; Architecture, definitions and overview*.

[ITU-T J.1012] Recommendation ITU-T J.1012 (2020), *Embedded common interface for exchangeable CA/DRM solutions; CA/DRM container, loader, interfaces, revocation.*

[ITU-T J.1013] Recommendation ITU-T J.1013 (2020), *Embedded common interface for exchangeable CA/DRM solutions; The virtual machine*.

[ITU-T J.1014] Recommendation ITU-T J.1014 (2020), *Embedded common interface for exchangeable CA/DRM solutions; Advanced security – ECI-specific functionalities*.

[ITU-T J.1015] Recommendation ITU-T J.1015 (2020), *Embedded common interface for exchangeable CA/DRM solutions; The advanced security system – Key ladder block.*

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Supplement

This Supplement defines the following terms:

3.2.1 advanced security system: Function of an **Embedded Common Interface** (**ECI)** compliant customer premises equipment (CPE), which provides enhanced security functions (hardware and software) for an **ECI Client**.

NOTE – The details are specified in [ITU-T J.1014].

3.2.2 advanced security (AS) slot: Resources of the **Advanced Security System** provided exclusively to an **ECI Client** by the **ECI Host**.

3.2.3 chipset-ID: Non-secret number that is used to identify a chipset in an **ECI Ecosystem**.

3.2.4 child, children: Entity (entities) referred to by a **Certificate** signed by a (common) **Father** in an **ECI Ecosystem**.

NOTE – **Father, children, brother** are referring to entities that manage **Certificates**: initialization data and software that is used to start the SoC of a **CPE**.

3.2.5 certificate: Data with a complementary secure digital signature that identifies an **Entity** in an **ECI Ecosystem**.

NOTE – The holder of the secret key of the signature attests to the correctness of the data – authenticates it – by signing it with its secret key. Its public key can be used to verify the data.

3.2.6 certificate chain: List of certificates in an **ECI Ecosystem** that authenticate each other including a Root **Revocation List**.

3.2.7 certificate processing subsystem: Subsystem of the **ECI Host** that provides certificate verification processing and providing additional robustness against tampering.

3.2.8 control word: Secret key used to encrypt and decrypt content in an **ECI Ecosystem**.

3.2.9 CPE manufacturer: Company that manufactures **ECI** compliant customer premises equipment (CPEs).

3.2.10 ECI (embedded CI): Architecture and the system specified in the ETSI ISG "Embedded CI", which allows the development and implementation of software-based swappable **ECI Clients** in customer premises equipment (CPE) and thus provides interoperability of CPE devices with respect to ECI.

3.2.11 ECI client (embedded CI client): Implementation of a conditional access/digital rights management (CA/DRM) client which is compliant with the Embedded CI specifications.

NOTE – It is the software module in a CPE which provides all means to receive, in a protected manner, and to control execution of a consumer's entitlements and rights concerning the content that is distributed by a content distributor or **Operator**. It also receives the conditions under which a right or an entitlement can be used by the consumer, and the keys to decrypt the various messages and content.

3.2.12 ECI client image: File with software as virtual machine (VM) code, and initialization data required by the **ECI Client Loader**.

3.2.13 ECI client loader: Software module part of the **ECI Host** which allows to download, verify, and install new **ECI Client** software in an ECI Container of the **ECI Host**.

3.2.14 ECI ecosystem: Real-world instantiation of a **Trust Environment** consisting of a **Trust Authority (TA)** and several platforms and **Embedded Common Interface (ECI)** compliant **customer premises equipment (CPE**s**)** in a commercial operation in the field.

3.2.15 ECI host: Hardware and software system of a customer premises equipment (CPE), which covers **Embedded Common Interface (ECI)** related functionalities and has interfaces to an **ECI Client**.

NOTE – The **ECI Host** is one part of the CPE firmware.

3.2.16 ECI host image: File with software and initialization data for an **Embedded Common Interface (ECI)** environment.

NOTE – It may also contain other software that does not cause interference with or permit undesirable observation of the **ECI Host**.

3.2.17 ECI host loader: Software module, which allows to download, verify, and install **ECI Host** software into a customer premises equipment (CPE).

NOTE – In a multi-stage loading configuration this term is used to refer to all security critical loading functions involved in loading the **ECI Host**.

3.2.18 ECI trust authority (TA): Organization governing all rules and regulations that apply to implementations of **ECI** and manages the interoperability and coexistence of conditional access (CA) and digital rights management (DRM) systems within the **ECI Ecosystem** with respect to the establishment of a chain of trust.

NOTE – The Trust Authority has to be a legal entity to be able to achieve legal claims. The Trust Authority needs to be impartial to all players in the downloadable conditional access/digital rights management (CA/DRM) ecosystem.

3.2.19 entity: Organization (e.g., manufacturer, **Operator** or vendor) or real-world item (e.g., **ECI Host**, **Platform Operation** or **ECI Client**) identified by an ID in a certificate.

3.2.20 export connection: Authenticated relation between an **ECI Client** that can decrypt content and a **Micro Server** that can re-encrypt content.

3.2.21 import connection: Approved connection from an **ECI Client** to a **Micro Server** that permits it to import decrypted content for subsequent re-encryption.

3.2.22 father: Signatory of the certificate of an **Entity** in an **ECI Ecosystem**.

NOTE – The ID of the **Entity** is defined by and is unique in the context of the Father.

3.2.23 key ladder: Function of the **Key Ladder Block** as defined in [ITU-T J.1015] for computing control words and associated control word usage information for application in the content decryption or re-encryption function of a customer premises equipment (CPE).

3.2.24 key ladder block: Robust secure mechanism in an **ECI Ecosystem** to compute decryption, encryption and authentication keys as defined in [ITU-T J.1015], both **Key Ladder** and **Authentication Mechanism**.

3.2.25 micro client: **ECI client** or non-**ECI** client that can decrypt content which was re-encrypted by a **Micro Server**.

3.2.26 micro server: **ECI Client** that can import decrypted content, re-encrypt this content, and authenticate a specific **ECI Client** or group of **ECI Client**s as the target for subsequent decryption.

3.2.27 micro DRM system: Content protection system that re-encrypts content on a customer premises equipment (CPE) with a **Micro Server** and that permits decoding of that re-encrypted content by authenticated **Micro Client**s.

NOTE – **Micro Server** and **Micro Client**s being provisioned by a Micro DRM **Operator**.

3.2.28 operator: Organization that provides **Platform Operation**s that is enlisted with the **Embedded Common Interface Trust Authority (ECI TA)** for signing the **ECI Ecosystem**.

NOTE – An **Operator** may operate multiple **Platform Operation**s.

3.2.29 platform operation: Specific instance of a technical service delivery operation having a single **ECI** identity with respect to security.

3.2.30 re-encryption session: Process controlled by a **Micro Server** of importing content from an **Import Connection**, re-encrypting it and producing the decryption information necessary for the authenticated target to subsequently decrypt it.

3.2.31 revocation list (RL): List of certificates in an **ECI Ecosystem** that have been revoked and therefore should no longer be used.

3.2.32 root: Public key or certificate containing a public key that serves as the basis for authenticating a chain of certificates in an **ECI Ecosystem**.

3.2.33 root certificate: Trusted certificate that is the single origin of a chain of certificates in an **ECI Ecosystem**.

3.2.34 security vendor: Company providing **ECI** security systems including **ECI Client**s for **Operators** of **ECI Platform Operation**s.

3.2.35 trust environment: Collection of rules and related process that constitutes the basis for an **ECI Ecosystem**.

4 Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

API Application Programming Interface

AS Advanced Security

BAT Bouquet Association Table

CA Conditional Access

CAT Conditional Access Table

CI Common Interface

CPE Customer Premises Equipment

DRM Digital Rights Management

DVB Digital Video Broadcasting

ECI Embedded Common Interface

ECM Entitlement Control Message

HTTP Hypertext Transfer Protocol

IP	Internet Protocol
IPTV	Internet Protocol Television
MPEG	Moving Picture Expert Group
NIT	Network Information Table
NV	Non Volatile
OTT	Over The Top (over the open Internet)
PVR	Personal Video Recorder
RL	Revocation List
SI	Service Information
SSU	System Software Update
TA	Trust Authority
URI	Usage Rights Information
URL	Uniform Resource Locator
VM	Virtual Machine

5 Conventions

The use of terms in bold and starting with capital characters in the present document shows that those terms are defined with an ECI specific meaning, which may deviate from the common use of those terms.

6 Characteristics of ECI interfaces

6.1 General remarks

Interfaces that are available for an **ECI Client** are classified in six groups of **Application Programming Interfaces** (APIs). These APIs provide functions and attributes that the **ECI Client** can benefit from. The classification is shown in Figure 6-1.

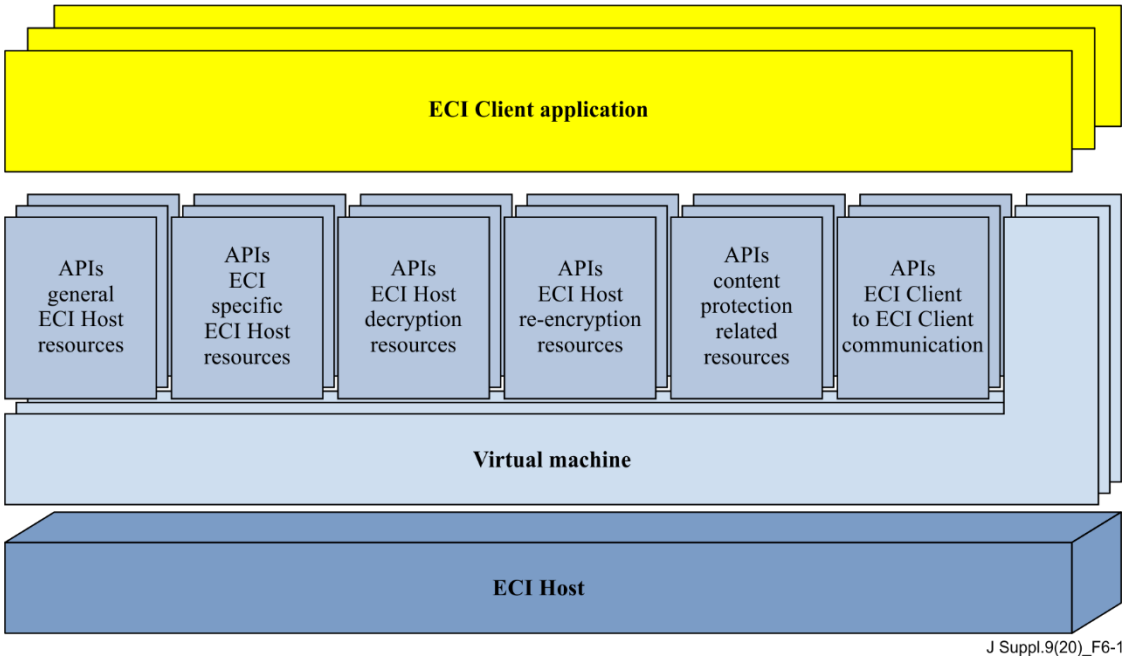

Figure 6-1 – API classification of the ECI architecture

6.2 General ECI Host resources

This API class provides the **ECI Client** with functionalities allowing the discovery of the available interface resources of the **ECI Host**. The messages that are exchanged between **ECI Client** and **ECI Host** are important to set up the features that will be possible for the **ECI Host** to offer to the end-user depending on the facilities of the device. This API class supports the communication with the user, can establish the IP connection to a device, allows the **ECI Client** to store data in the memory of the **ECI Host**, is responsible for the settings of time, data and languages and allows communication with the power management.

6.3 ECI specific ECI Host resources

This API class allows the **ECI Client** to gain access to the functionalities of the **Advanced Security System** of the **ECI Host** and it also handles the usage of a smart card reader. Taking into account that an **ECI Client** will very likely be active as part of a digital video broadcasting (DVB) environment, this API additionally allows to gain access to a data carousel according to the DVB standard.

6.4 ECI Host decryption resources

In the ECI architecture every media decryption is initiated and controlled by the **ECI Host**. This class supports the selection of an **ECI Client** following the content decryption requirements for the media to be decrypted. The **ECI** architecture supports two different types of media formats, the moving picture expert group (MPEG) Transport Stream and the ISOBMFF file format. The request of an **ECI Host** to open a descrambling session includes the check whether all the resources needed for accessing the content and the accompanying metadata are available at both sides, for the **ECI Host** as well as for the **ECI Client**. Only if this is guaranteed, the decryption session will start.

6.5 ECI re-encryption resources

Content that is going to leave the protected environment of the **ECI** architecture needs to possess the possibility to be protected again by an encryption scheme. This API class supports such a protection by provisioning functionalities for re-encryption, e.g., for further distribution or for storage. The **ECI Host** requests some information from the **ECI Client** about DRM servers that can deliver further information about re-encryption and this information is then used to set up appropriate sessions to

start such a re-encryption process. The re-encryption system is called a **Micro DRM System** and the **ECI Client** that initially decrypted the content can control to which **Micro DRM System** the export of content will take place.

6.6 Content protection related resources

This API class supports CA/DRM providers in setting up a content property system according to their needs. Access to usage rights information (URI) can be granted on several security levels. The URI is generated by the **ECI Client** and is used by the **ECI Host** to control applications possessing access to media content. This also includes support for blocking the presentation or processing of media in a selective way. Outgoing content can be water-marked by the **ECI Client** and parental control permits the **ECI Client** to authorize the consumer before displaying the content.

6.7 ECI Client Communication related resources

This API class supports the communication between **ECI Clients**. Those may communicate amongst themselves in order to provide additional functionality. **ECI Clients** can register their principal ability and willingness to support inter client communication through a discovery resource. After system initialization, they can read the identities of other **ECI Clients** including the established **Import/Export Connections.**

7 Installation of an ECI Host

In order to make an end-user's device **ECI** compatible, this device needs to be prepared in such a way that one or more **ECI Clients** can be installed. This part of the preparation is done with the help of an **ECI Host** which itself is installed via an **ECI Host Loader**.

This test case is characterized by the following terms:

- Prerequisite: **ECI Host** not yet installed on the device or update of an existing **ECI Host** necessary.
- Status after activity: **ECI Host** installed on device.

The steps for the installation of an **ECI Host** are shown in the flow diagram in Figure 7-1.

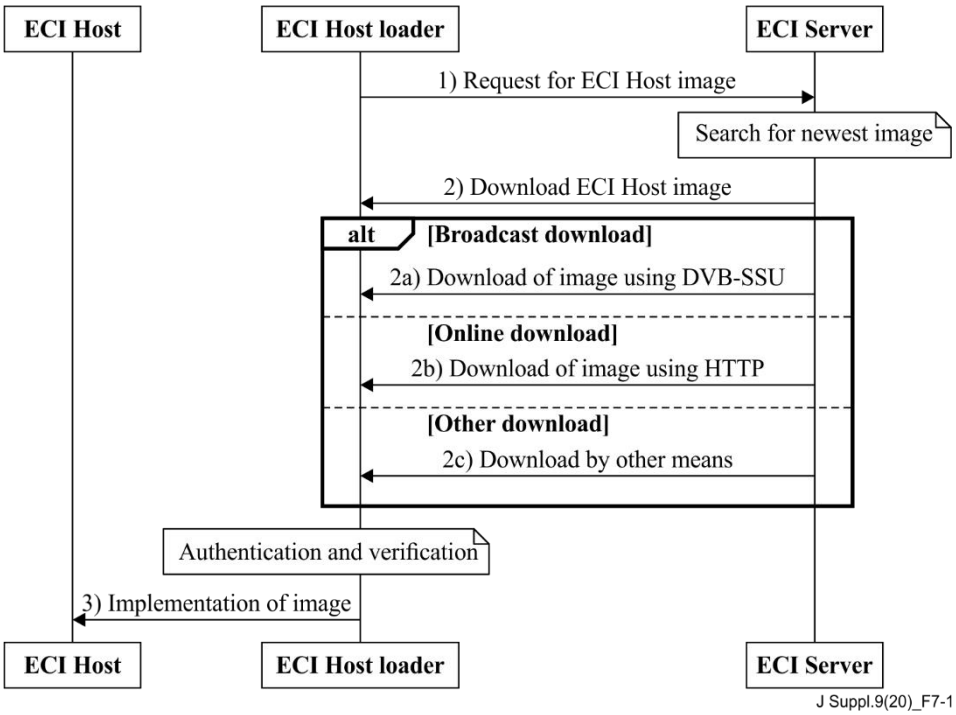

Figure 7-1 – Flow diagram for the installation of an ECI Host

The **ECI Host Loader** is created by the user's device at boot time. The way in which the location of the repository for the download of an **ECI Host Image** is detected is not specified in the specification and is left to implementation. In a broadcast scenario, it is very likely that a data carousel will be accessed whereas in an IP environment the URL of a server will be provisioned as follows:

- Step 1: The **ECI Host Loader** can clarify with the support of a **Revocation List** whether a new **ECI Host Image** for this type of CPE is available or not. For the very first installation of an **ECI Host** there will be no **Revocation List** on the CPE and the download of the appropriate **ECI Host Image** is initiated. The **ECI Host Loader** will skip step 1 and will jump to step 2a if the connected network is a unidirectional broadcast network. In this case the information about the location of the **ECI Host Image** is signalled as part of the DVB SI information as described in clause 6.4.2.3.1 of [ITU-T J.1012].
- Step 2: Depending on the type of information exchange possibilities between the **ECI Host Loader** and the **ECI Host** Repository there are three different ways for downloading an **ECI Host Image**:
  - Step 2a: In a broadcast environment with only a unidirectional connectivity being available with a downlink from the **ECI Host** Repository to the CPE, the DVB-SSU mechanism can be used for downloading **ECI Host Image** files. The information for the CPE, which needs an indication on which parts of the broadcast data stream relates to download, is provided by a collaboration of several DVB SI tables that need to be scanned. Details about the procedure and the structure of the descriptors used are described in clause 6.4.2 of [ITU-T J.1012].
  - Step 2b: With connectivity to the Internet the CPE has the possibility to use any kind of file transfer protocol for the download. There is no specific protocol defined but it is suggested to use HTTP 1.1. The bi-directional connectivity gives the CPE the chance to look for updates in regular periods. The API supporting the information exchange between the **ECI Host Loader** and the **ECI Host** Repository is described in clause 7.7.3.3 of [ITU-T J.1012].
  - Step 2c: The third possibility to gain access to an **ECI Host Image** is the one using any alternative delivery method which also includes the download from any storage device being directly connected to the CPE.
- Step 3: After successful download, the **ECI Host Image** is verified by checking certificates and information provided by **Revocation List**s. Processing rules for this verification are given in clause 5.4 of [ITU-T J.1012]. During this verification it is also checked whether the **ECI Host Image** is not invalidated by the minimum version number mentioned in the **Revocation List**. If a revocation is detected, it is up to the implementation of the **ECI Host** how to react to this situation or how to prevent it (e.g., by doing periodical checks for new **Revocation Lists**). The **ECI Host Image** is expected to be structured as a file with a header containing information about the manufacturer of the CPE. The **ECI Host Image** will be installed on and about the version number of the **ECI Host Image**. The **ECI Host Loader** also permits multi-stage loading. Details are given in clause 6.2.3 of [ITU-T J.1012]. If both checks are successful, the image can be implemented on the CPE, together with the necessary certificates. Details for checking the structure of the image files are not further specified but are left to the **ECI Host**.

8 Installation of an ECI Client

After the installation of the **ECI Host**, the device is ready to install an **ECI Client** consisting of a program image and accompanying certifications. This installation is done by using the already installed **ECI Host**.

This test case is characterized by the following terms:

- Prerequisite: **ECI Host** is installed on the device, **ECI Client** available by the service provider.
- Status after activity: **ECI Client** installed on device.

The steps for the installation of an **ECI Client** are shown in the flow diagram in Figure 8-1.

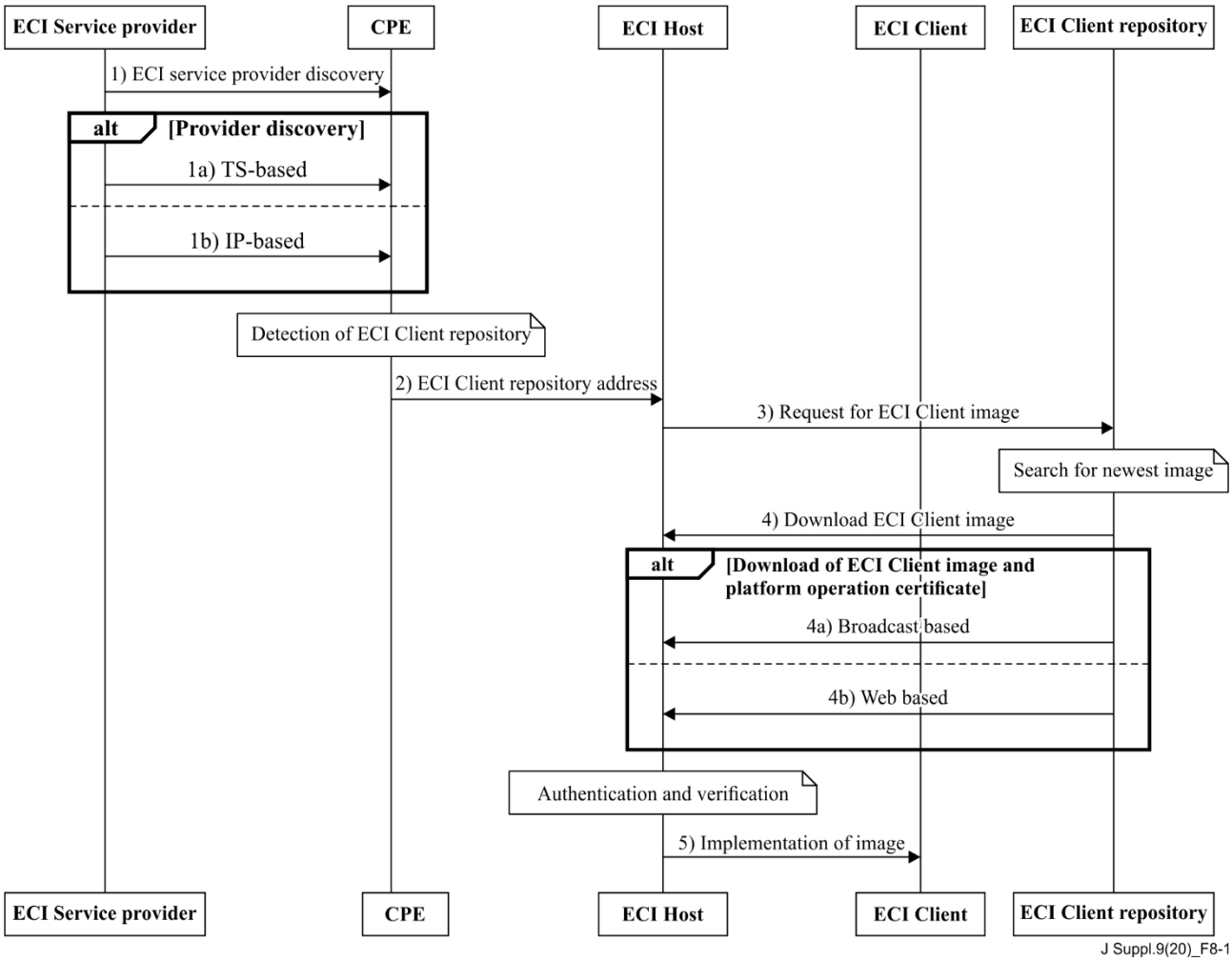

Figure 8-1 – Flow diagram for the installation of an ECI Client

- Step 1: The ECI architecture provides two possibilities for gaining information regarding the installation of the **ECI Client**:
  - Step 1a: Taking into account that a device normally does not contain an **ECI Client**, the CPE of the end-user needs to receive appropriate information that is made available by the ECI Service Provider. In a classical digital broadcasting environment, this information is conveyed by DVB tables, network information table (NIT) and bouquet association table (BAT) respectively and is distributed in MPEG Transport Streams as a private descriptor. Details about the structure of this descriptor and its usage is given in clause 7.2.2 of [ITU-T J.1012].
  - Step 1b: In an IP-based network connectivity to the information of an **ECI** Service Provider is done by using a URL (see clause 7.2.3 of [ITU-T J.1012]).
- Step 2: According to the information received from the **ECI** Service Provider, the CPE creates the address of the **ECI Client** Repository to contact and passes this information to the already installed **ECI Host** that takes care in the following steps about the download and the implementation of the **ECI Client**.

- Step 3: In a broadcast network, the **ECI Host** cannot directly contact the **ECI Client** Repository. The **ECI Host** evaluates the data cyclically transmitted as part of a DVB descriptor. By matching this data with the information for provider discovery received from the **ECI** Service Provider, the modules to be downloaded are identified. In a web-based environment with bidirectional connectivity query mechanisms of the HTTP protocol are used to find the appropriate data to be downloaded.
- Step 4: The **ECI** architecture provides either a broadcast based or web-based download of the **ECI Client** and of accompanying certificates to the CPE:
  - Step 4a: The broadcast-based download makes use of data carousels that are controlled by DVB structured service information. Such a data carousel is laid out for repeatedly delivering data to a consumer's end device in a continuous cycle. Considering the unidirectional connectivity in a broadcast environment, the end device may access the data stream at any time. For several types of content several carousel groups containing the data to be downloaded are defined in clause 7.7.2 of [ITU-T J.1012].
  - Step 4b: A web-based download of the **ECI Client** and accompanying data is also possible. In this case, the **ECI Client** Repository responds to the HTTP queries of the **ECI Host** and as a result of this communication the **ECI Client** and accompanying certificates are downloaded to the **ECI Host**. The **ECI** architecture provides a Web API to facilitate the conversation between the CPE and the **ECI Client** Repository. The requests supported by this API are described in clauses 7.7.3.4 and 7.7.3.5 of [ITU-T J.1012] supporting the detection of credentials and revocation data as well as the download of an **ECI Client Image**.

After having downloaded all relevant data files the **ECI Host** starts a verification process for the **ECI Client** and the platform the client is dedicated to. For this purpose, a certificate chain for the platform is started, the components of which are described in clause 7.5 of [ITU-T J.1012]. The **ECI Client** itself is also verified by a certification chain shown in clause 7.4 of [ITU-T J.1012]:

- Step 5: After successful verification, the **ECI Client Image** is implemented, credentials and image are stored in the non-volatile (NV) memory of the CPE. Updated versions of the **ECI Client** overwrite older ones.

9 Installation of a second ECI Client on the same device

After the installation of the **ECI Host**, the device is ready to install one or more **ECI Client**s. The processing steps for installing an **ECI Client** were described in clause 8. The same processing steps must be repeated for installing an additional **ECI Client**.

This test case is characterized by the following terms:

- Prerequisite: One **ECI Client** is already installed on the device; a second **ECI Client** is available for installation by a service provider.
- Status after activity: Second **ECI Client** installed on device.

The steps for the installation of a second **ECI Client** are shown in the flow diagram in Figure 9-1:

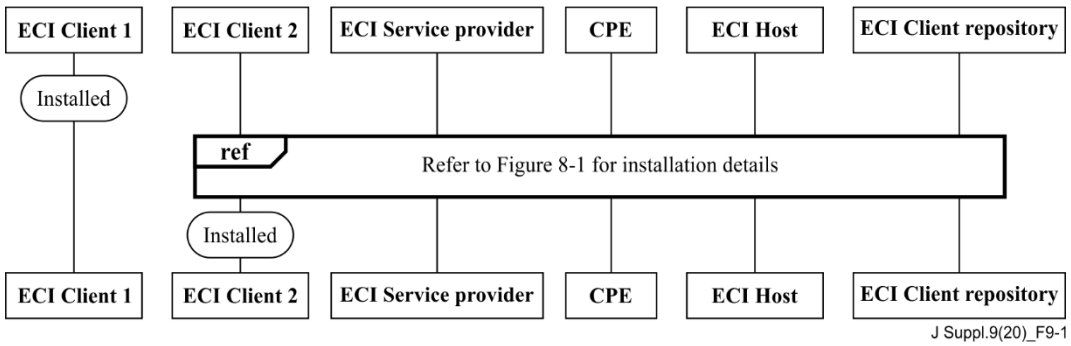

Figure 9-1 – Flow diagram for the installation of a second ECI Client

The installation of a second **ECI Client** does not differ in any way from the installation steps shown in Figure 8-1. The **ECI Host** that is responsible for the installation is capable to accommodate and to provide the runtime environment for as many **ECI Client**s as its resources can handle. The minimum number of **ECI Client**s to be supported is two (see clause 6.2.1 of [ITU-T J.1011]) and resources and processing power is laid out in an **ECI**-compatible device to allow the parallel usage of **ECI Client**s in order to enable simultaneous decryption and re-encryption of content streams being delivered by different service providers. The minimum number of **ECI Client**s is increased by two in case of personal video recorder (PVR) usage with additional **Micro Server** and **Micro Client** (see clause 9.7.1.2 of [ITU-T J.1012]).

ECI Clients are separated from each other in containers and are protected against unauthorized access from the outside world. A collaboration between **ECI Client**s, e.g., for decrypting and re-encrypting content is possible in a certified way in order to ensure that clients run with a high degree of integrity.

10 Decryption of protected content

The main scope of application of an **ECI**-compatible device will be the decryption of protected content. In this clause it is assumed that DVB mechanisms are used to transmit the information required for the descrambler. The content related information about the used Conditional Access system is provided in the system information of a DVB framework as part of the conditional access table (CAT) that needs to be accessed by the CPE.

This test case is characterized by the following terms:

- Prerequisite: At least one **ECI Client** is installed on the device; information about the CA/DRM system is available, encrypted media is accessible.
- Status after activity: Media is successfully decrypted.

The steps for the decryption of content are shown in the flow diagram in Figure 10-1:

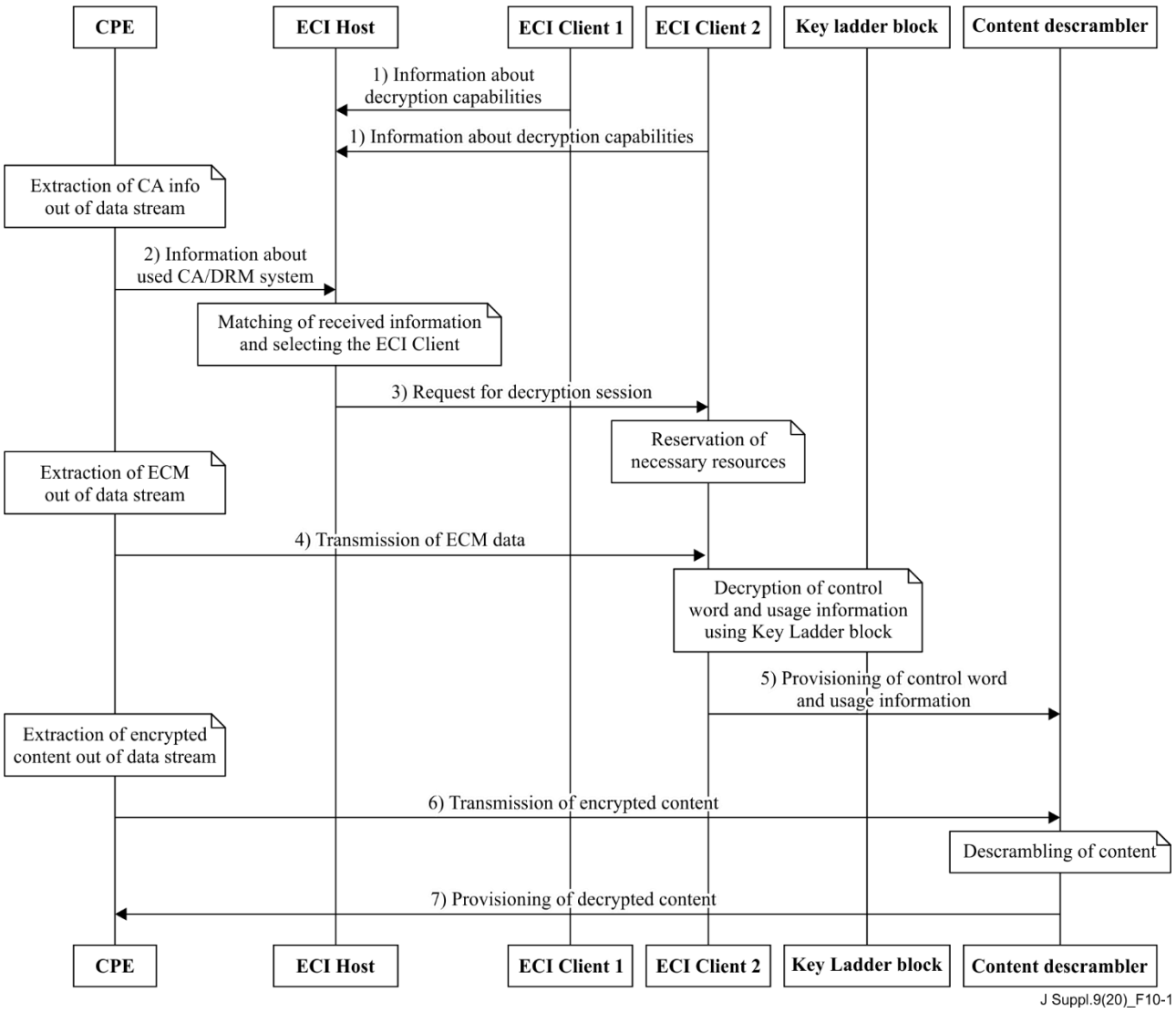

Figure 10-1 – Flow diagram for the decryption of content

In Figure 10-1 it is assumed that the CPE has access to the encrypted content as part of a DVB framework using MPEG Transport Streams for conveying a multiplex of encrypted content and CA/DRM relevant information. The service information of a DVB framework allows the CPE to extract from the data stream the different parts of information needed for decrypting the protected audio-visual content:

- Step 1: For preparing the decryption process each **ECI Client** informs the **ECI Host** about its capabilities for supporting the decryption of content. This message is sent to the **ECI Host** using the function that is described in clause 9.6.2.2.2 of [ITU-T J.1012].
- Step 2: The multiplexed data of an MPEG Transport Stream also carries information about the used CA/DRM system. The content of each packet of an MPEG Transport Stream is characterized by an identifier carried in the header of each packet. By searching for the fitting identifier, the CA information is extracted by the CPE from the data stream and is forwarded to the **ECI Host** for further processing. Acquisition rules for detecting the CA information is described in clause 9.6.2.3.3.2 of [ITU-T J.1012].
- Step 3: After having received information about the content to be decrypted and about the capabilities of the **ECI Client**s, it is the task of the **ECI Host** to select the most suitable **ECI Client** by matching the description of the CAT with the information about the **ECI Clients**. If more than one **ECI Client** is installed, rules for setting up a prioritization list are described

in clause 9.6.2.2.2 of [ITU-T J.1012]. As a result of such a selection process the **ECI Host** requests an **ECI Client** (in Figure 10-1 client no. 2 was selected) to set up a session for decrypting as described in clause 9.6.2.2.3 of [ITU-T J.1012]. It is then the task of the **ECI Client** to reserve all resources that are needed for performing a successful decryption.

- Step 4: The multiplexed data of an MPEG Transport Stream also carries information about the control words required for a proper descrambling of content. Under normal conditions this control word is changed for safety reasons several times per minute. It is encrypted for transmission as an entitlement control message (ECM). Those packets of the MPEG Transport Stream that contain ECMs are extracted from the data stream by the CPE and they are forwarded to the selected **ECI Client** that takes care of decryption in collaboration with a component of the ECI Advanced Security System known as **Key Ladder Block** (for a detailed description see clause 7 of [ITU-T J.1015]) that is implemented in the CPE's chipset. The **Key Ladder Block** establishes a hierarchy of trust and can also be used for the personalization of a device.
- Step 5: The **Key Ladder Block** creates out of the received ECM data the control word that is used for descrambling the content and it also signals usage rules for the operation of the content descrambler. This information is packed into a 64-bit word and the meaning of each bit is described in clause 7.3.1 of [ITU-T J.1015] for calculating the decryption key word using the **Key Ladder Block** as described in clause 8.2.4.7 of [ITU-T J.1014]. Both types of data are provided by the **ECI Client** to the content descrambler.
- Step 6: The multiplexed data of an MPEG Transport Stream carries as the main part of its load the audio-visual content to be decrypted for presentation. According to the program identifiers signalled as part of the service information of a DVB-compatible system, the suitable packets are filtered out of the data stream and they are forwarded for decryption to the content descrambler. The possibilities for starting and stopping the decryption of an MPEG Transport Stream are shown in clause 9.6.2.3.4 of [ITU-T J.1012].
- Step 7: The decrypted content is forwarded to the CPE for presentation. Whether for this provisioning a protection of whatever kind is needed or not depends on business rules and is not specified as part of the ECI framework.

11 Re-encryption of content

Taking into account that several **ECI Client**s can be installed in parallel on one CPE using different encryption mechanisms, it is reasonable to provide a component to protect content delivered by one **ECI Client** for further applications within the CPE. Such a component is called in **ECI** a **Micro DRM System**. It consists of a **Micro Server** that is responsible for the re-encryption of content delivered by an **ECI Client** and a **Micro Client** that is able to decrypt the content that is imported from the **Micro Server**. The re-encryption process does not necessarily be followed immediately by the decryption of this re-encrypted content. For an application like a re-encryption on the fly for immediate consumption of content re-encrypting and decrypting follow close to each other. For an application associated with storing content on a PVR, both steps will be done with a certain temporal distance.

This test case is characterized by the following terms:

- Prerequisite: **ECI Client** and **Micro Server** are installed on the CPE; information about the CA/DRM system is available, encrypted media is accessible in a file format.
- Status after activity: Media is successfully re-encrypted into another format.

The steps for the re-encryption of content are shown in the flow diagram in Figure 11-1:

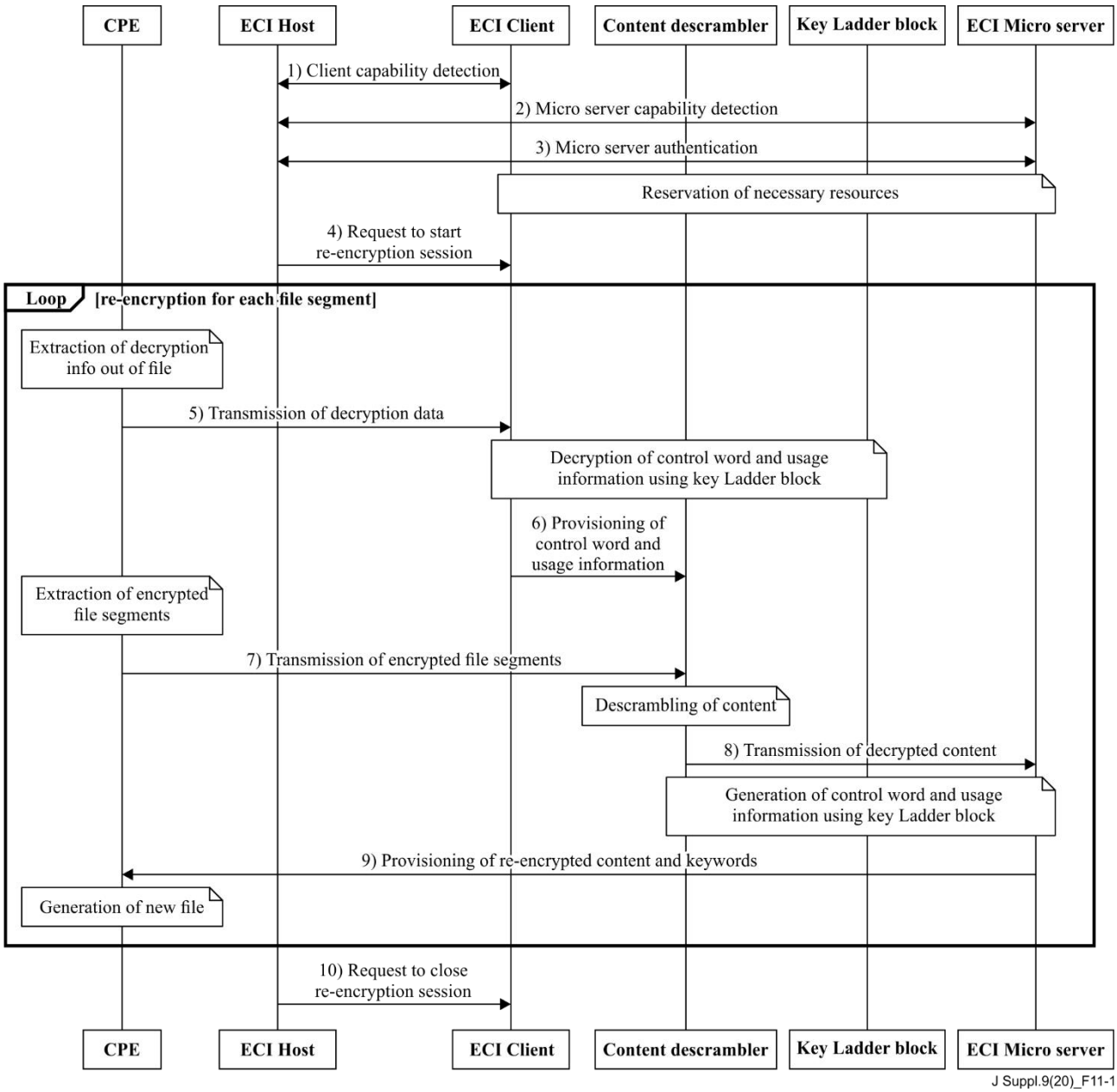

Figure 11-1 – Flow diagram for the re-encryption of content

In order to prepare the re-encryption of content the **ECI Host** has to know the technical capabilities of the installed **ECI Clients**:

- Step 1: The **ECI Client** informs the **ECI Host** about its decryption capabilities. This message follows the format as specified in clause 9.6.2.2.2 of [ITU-T J.1012]. In the following the **ECI Host** requests the **ECI Client** to return its list with possible **export connections**. Each entry in the returned list possesses a priority value. The higher the value, the higher is the chance that the **ECI Client** can successfully connect to the client characterized by this value. The capability exchange between **ECI Host** and **ECI Client** follows the format as specified in clause 9.7.2.3.2 of [ITU-T J.1012].
- Step 2: The **ECI Host** is contacting the **ECI Micro Server** in order to gain information about **Import Connections** the **ECI Micro Server** is willing to accept. An array of import nodes is given back to the **ECI Host** following the format as described in clause 9.7.2.4.2 of [ITU-T J.1012]. Using this array, the **ECI Host** can set up a connection between the exporting **ECI Client** and the importing **ECI Micro Server**.

- Step 3: The **ECI Host** also requests information from the **ECI Micro Server** about the targets for decrypting still encrypted content the **ECI Micro Server** can authenticate, considering that re-encrypted content will finally always be decrypted. This information is conveyed using the format as specified in clause 9.7.2.5.3 of [ITU-T J.1012].

Now that the **ECI Host** has collected all information needed for re-encryption, the necessary components can be reserved for the process to follow:

- Step 4: The **ECI Host** then requests the **ECI Client** to start a session for re-encrypting content under the control of the **ECI Micro Server**. This request follows the format as specified in clause 9.7.2.5.7 of [ITU-T J.1012].

In this test case it is assumed that the content to be re-encrypted is formatted as a file. Therefore, step 5 up to step 9 is repeated for each protected file segment until the end of file is reached:

- Step 5: Part of the metadata of such a file is information that is needed for a successful decryption of the content. This information is extracted from the file and is passed in this step to the **ECI Client** that takes care about decryption in collaboration with a component of the **ECI Advanced Security System** named **Key Ladder Block** (for a detailed description see clause 7 of [ITU-T J.1015]) that is implemented in the CPE's chipset allowing the establishment of a hierarchy of trust for the protected content.
- Step 6: The **Key Ladder Block** creates from the received decryption information the control word that is used for descrambling the content and it also signals usage rules for the operation of the content descrambler. This information is packed into a 64-bit word and the meaning of each bit is described in clause 7.3.1 of [ITU-T J.1015]. The necessary steps for calculating the decryption key word using the **Key Ladder Block** are described in clause 8.2.4.7 of [ITU-T J.1014]. Both types of data are provided by the **ECI Client** to the content descrambler.
- Step 7: The file carries as the main part of its load the audio-visual content to be re-encrypted. The appropriate data packets are filtered out of the file and they are forwarded for decryption to the content descrambler. The possibilities for starting and stopping the decryption of a file are shown in clause 9.6.2.4.4 of [ITU-T J.1012].
- Step 8: The decrypted content is forwarded to the **ECI Micro Server** for re-encryption. New control words are computed using the **Key Ladder Block** and the function described in clause 8.2.4.6 of [ITU-T J.1014].

Encrypted content and the appropriate keywords are combined in such a way as requested by the new encryption scheme:

- Step 9: This data is device-internally transmitted back to the CPE to generate a new file containing a new protection scheme.

When the complete file is re-encrypted, the necessary resources are released again, followed by:

- Step 10: The **ECI Host** closes the re-encryption session by sending a message to the **ECI Client** as described in clause 9.7.2.5.8 of [ITU-T J.1012].

12 Play-out to an external device

In an **ECI** system protected content cannot only be processed and presented on an **ECI**-compliant CPE but it can also be exported to an external device that is able to decrypt the content and to present it to the user. The preparation of content and the export are done by functionalities provided by a **Micro DRM System**, consisting of a **Micro Server** in the exporting device that is responsible for the re-encryption of content delivered by an **ECI Client** and a **Micro Client** that is able to decrypt the content that is imported from the **Micro Server**. For an application like a re-encryption on the fly for immediate consumption of content, re-encrypting and decrypting follow close to each other.

This test case is characterized by the following terms:

- Prerequisite: **ECI Client** and **Micro Server** are installed on the CPE; information about the CA/DRM system is available, encrypted media is accessible in a file format, **Micro Client** is installed on an external device.
- Status after activity: Media is successfully decoded on an external device.

The steps for the re-encryption of content are shown in the flow diagram in Figure 12-1:

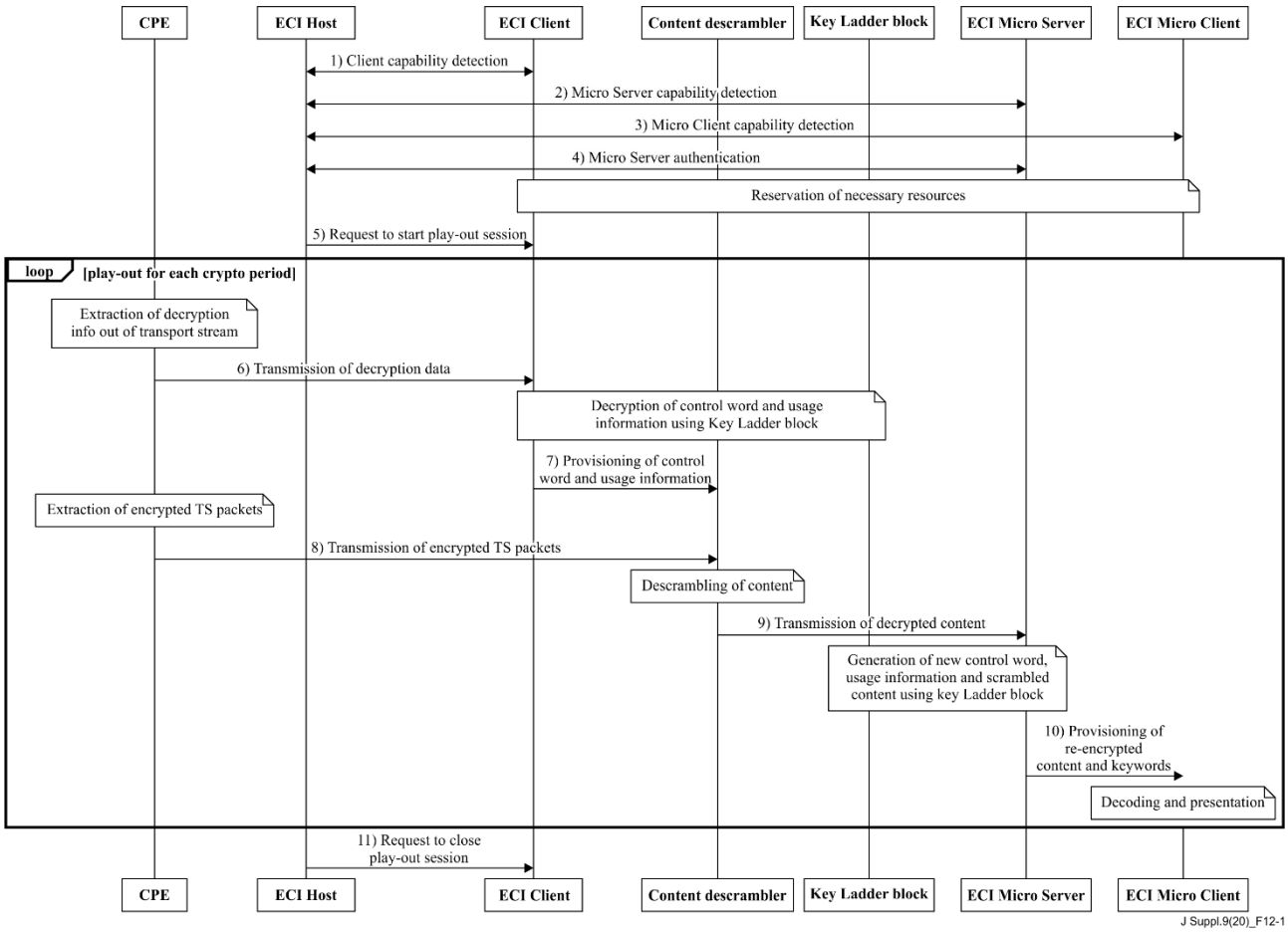

Figure 12-1 – Flow diagram for the play-out of content

In order to prepare the play-out of content to an external device the **ECI Host** has to know the technical capabilities of the installed **Micro Server** and the **ECI Clients**:

- Step 1: The **ECI Client** informs the **ECI Host** about its decryption capabilities. This message follows the format as specified in clause 9.6.2.2.2 of [ITU-T J.1012]. In the following the **ECI Host** requests the **ECI Client** to return its list with possible **Export Connections**. Each entry in the returned list possesses a priority value. The higher the value, the higher is the chance that the **ECI Client** can successfully connect to the client characterized by this value. The capability exchange between **ECI Host** and **ECI Client** follows the format as specified in clause 9.7.2.3.2 of [ITU-T J.1012].
- Step 2: The **ECI Host** contacts the **ECI Micro Server** to gain information about **Import Connections** the **ECI Micro Server** is willing to accept. An array of import nodes is given back to the **ECI Host** following the format as described in clause 9.7.2.4.2 of [ITU-T J.1012]. Using this array, the **ECI Host** can set up a connection between the exporting **ECI Client** and the importing **ECI Micro Server**.

- Step 3: The **ECI Host** needs to gather information about the capabilities of the **ECI Micro Client,** taking into account that re-encrypted content will finally be decrypted after play-out to an external device. How the existence of such a client on an external device can be detected, is up to the technical capabilities of the **ECI**-compatible device and the external device. **ECI** does not give any recommendations for this issue, for implementations it will very likely be the task of the **ECI Host** to locate any potential candidates. Using the message specified in clause 9.7.2.6.3 of [ITU-T J.1012] the **ECI Host** requests the **ECI Micro Client** to provide the encryption targets for which it can decrypt. For a successful connection, the identifier for the **ECI Micro Server** needs to be part of the list returned to the **ECI Host**.
- Step 4: With this information available, the **ECI Host** requests the **ECI Micro Server** to create a connection to the **ECI Micro Client** using the message as described in clause 9.7.2.5.4 of [ITU-T J.1012].

Now that the **ECI Host** has managed all information needed for re-encryption and play-out, the necessary components can be reserved for the process to follow:

- Step 5: The **ECI Host** then requests the **ECI Client** to start a play-out session under the control of the **ECI Micro Server**. This request follows the format as specified in clause 9.7.2.5.7 of [ITU-T J.1012].

In this test case it is assumed that the content to be played out is available as a Transport Stream:

Therefore, step 6 to step 10 is repeated for each crypto period until the session is closed. A detailed description of the elements and the syntax of a Transport Stream can be found in [b- ITU-T H.222.0].

Part of the data stream conveying encrypted content is also information that is needed for a successful decryption of the content. This information, consisting of encrypted control words, is carried in a DVB environment as an entitlement control message and it will only be decrypted when authorised to do so by the accompanying entitlement management message that contains details about usage rights. Both types of messages are transported in different Transport Stream packets and can therefore be easily extracted from the overall Transport Stream.

- Step 6: They are then passed to the **ECI Client** that takes care of decryption in collaboration with a component of the **ECI Advanced Security System** known as **Key Ladder Block** (for a detailed description see clause 7 of [ITU-T J.1015]) that is implemented in the CPE's chipset allowing the establishment of a hierarchy of trust for the protected content.
- Step 7: The **Key Ladder Block** creates from the received decryption information, the actual control word that is used for descrambling the content and it also signals usage rules for the operation of the content descrambler. This information is packed into a 64-bit word and the meaning of each bit is described in clause 7.3.1 of [ITU-T J.1015]. The necessary steps for calculating the decryption key word using the **Key Ladder Block** are described in clause 8.2.4.7 of [ITU-T J.1014]. Both types of data are provided by the **ECI Client** to the content descrambler.
- Step 8: The Transport Stream carries as the main part of its load the audio-visual content to be re-encrypted. The appropriate data packets are filtered out of the data stream and they are forwarded for decryption to the content descrambler. The possibilities for starting and stopping the decryption of a Transport Stream are described in clause 9.6.2.3.4 of [ITU-T J.1012].
- Step 9: The decrypted content is forwarded to the **ECI Micro Server** for re-encryption. New control words are computed using the **Key Ladder Block** and the function described in clause 8.2.4.6 of [ITU-T J.1014].

Encrypted content and the appropriate keywords are combined in such a way as requested by the new encryption scheme:

- Step 10: This data is exported to the **Micro Client** for further processing and final presentation following steps 4 to 7 of the decryption process already shown in Figure 10-1.
- Step 11: When there is no longer any need for a play-out, the necessary resources are released again and in this step the **ECI Host** closes the play-out session by sending a message to the **ECI Client** as described in clause 9.7.2.5.8 of [ITU-T J.1012].

13 Security aspects

13.1 Introduction

The essential task of an **ECI Client** is the decryption or encryption of the content on a **CPE** that is part of a trustful environment. Taking into account that **ECI Client**s from different providers can be installed on such a **CPE** and that each of these clients is allowed to follow its own security concept, a chain of trust has been established with the intention to ensure that only trusted components, hardware or software, are being used while retaining the wanted flexibility. An example for such a chain of trust is depicted in Figure 13-1.

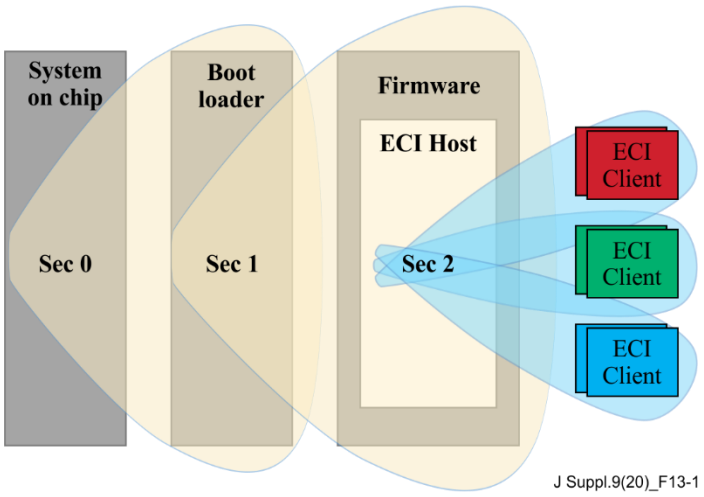

Figure 13-1 – Example for a chain of trust in an ECI environment

This chain is characterised by the linking of two adjacent components that make up a security section. Three of these sections are shown in Figure 13-1 named "Sec0", "Sec1" and "Sec2". By overlapping these sections trustworthiness is achieved because signatures and certificates passing through one section are accepted by the adjacent section as a source of trust for its own certification and signature process. Each of these sections can contain a mix of signatures, certificates, encryptions, and keys that are required by the responsible **Entity**. In this example the System on chip contains Sec0 and a boot strap that verifies, executes, and loads the Boot loader. The Boot loader contains Sec1 that verifies the CPE's Firmware. Part of the Firmware is the **ECI Host** that contains Sec2 which verifies any **ECI Client** to be loaded. Each **ECI Client** may have its own authority body with its own subjective view related to security for all the preceding sections.

In an **ECI** environment the **ECI TA** has the task of managing a list of CPEs and **ECI Host**s together with some information whether these components are described as being compromised or uncompromised. A **Revocation List** contains those certificates for components that have been revoked and therefore should no longer be used. It is the task of the **Certificate Processing Subsystem** to verify the status of the **ECI Host** to be installed on a CPE. The certificate chain for the **ECI Host** consists of **Root Certificate,** manufacturer **Certificate,** host **Certificate** and host image series **Certificate** as described in clause 6.2.2.1 of [ITU-T J.1012]. The general processing steps to follow are given in clause 10.1 of [ITU-T J.1014]. Specific rules can be found in clause 10.2 of

[ITU-T J.1014]. The basic principles and tasks for defining an **ECI** compatible **Trust Environment** intended for establishing an **ECI Ecosystem** are defined in [b-ITU-T J-Sup.8].

For the **ECI** environment a mechanism known as **Platform Operation** has been developed as a service delivery operation possessing a single **ECI** identity with respect to security. In order to permit an **ECI Client** to decrypt or encrypt content on a CPE, the **Platform Operation** has to trust the identification of the CPE and the identification of the chipset, the integrity of the **AS system** and the availability of an uncompromised **ECI Host** for the selected CPE. It is the decision of the **Platform Operation** to trust a CPE and to deliver services to it. The certificate chain for the **ECI Client** starts with the Vendor **Revocation List**, followed by Security Vendor **Certificate**, **ECI Client Revocation List** and finally the **ECI Client Image** file. A figure of this chain is given in clause 7.4.1 of [ITU-T J.1012]. The general processing steps to follow are given in clause 10.1 of [ITU-T J.1014]. Specific rules can be found in clause 10.3 of [ITU-T J.1014].

In order to ensure that an **ECI Host** cannot be easily compromised, the **Platform Operation** can enforce the usage of **ECI TA** provided revocation lists by specifying a minimum **Root Revocation List** that it wants to have applied or the **Platform Operation** stops its own operation on that CPE. This mechanism is called "service starvation". The **ECI Host** maintains in a **Revocation List** the latest versions of all items it has loaded and by inspecting this list the **ECI Host** is able to remove those items, which can cause service starvation. The structure of a **Revocation List** is given in clause 5.3 of [ITU-T J.1012]. The proper functioning of the **Platform Operation** relies on the **ECI TA** and its capability to provide a good certification regime for CPEs and to ensure that poorly or maliciously constructed **ECI Client**s do not have access to an **ECI** system or are removed from such a system to create a feasible environment for ensuring the integrity of the **ECI Host**.

13.2 General description of an ECI Certificate Chain

In order to allow the participation of several entities in an **ECI** value chain in a trustful way, the **ECI** ecosystem supports the implementation of a **Certificate Chain** and its processing in a CPE. A trustful collaboration between the entities is achieved by signing **Certificate**s as in the example shown in Figure 13-2.

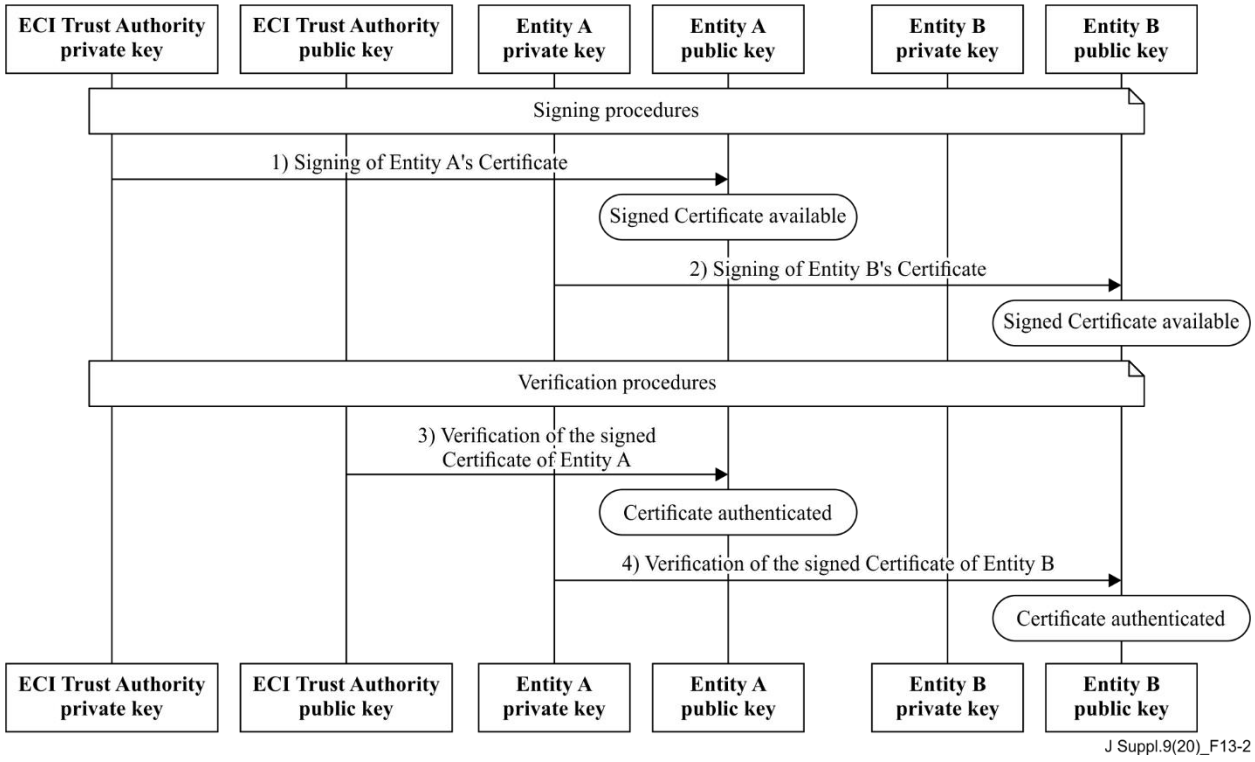

Figure 13-2 – Signing and verification of ECI Certificates

In this example three entities are shown, the **ECI Trust Authority**, **Entity** A and **Entity** B. Each of the entities creates its own pair of keys (a private and a public one) to sign and to verify **Certificate**s and/or **Revocation List**s**.** These three entities are related to each other in a hierarchical order in such a way that each **Entity** acts as a **Father** to the **Entity** to its right (called a **Child**) by signing with its private key the **Certificate** of its **Child**.

- Step 1: This is shown in Figure 13-2 for the collaboration between the **ECI Trust Authority** and **Entity** A.
- Step 2: The same is shown for the collaboration between **Entity** A and **Entity** B. In such a way, a chain of trust is created reaching from the **ECI Trust Authority** up to **Entity** B.
- Steps 3 and 4: Such signed **Certificate**s need to be verified before any usage. As shown in these steps of Figure 13-2 such an authentication of a signed **Certificate** happens by applying the public key of that **Entity** that also signed the **Certificate,** i.e., the public key of the **Father**. Chains of trust consisting of **Certificate**s and/or **Revocation List**s are widely used in an **ECI** ecosystem depending on the business relationship and the agreed responsibilities between the entities.

13.3 Trust provisioning for an ECI Host

In this clause the provisioning of trust for the implementation of an **ECI Host Image** is described, for example, by using the following three entities:

- **ECI Trust Authority**: this entity governs the rules and regulations for the implementation of an **ECI** system and is the owner of the root of trust for the verification of a trust chain.
- **CPE Manufacturer**: this entity develops and sells CPEs containing an implementation of an **ECI** system.
- **ECI Host** Provider: this entity provides an **ECI Host Image** to be implemented as part of a CPE.

The collaboration amongst the entities and the processing of certificates and revocation lists for signing purposes is shown in Figure 13-3. Each of these entities possesses its unique set of private and public keys for signing and validating **Certificate**s and **Revocation List**s.

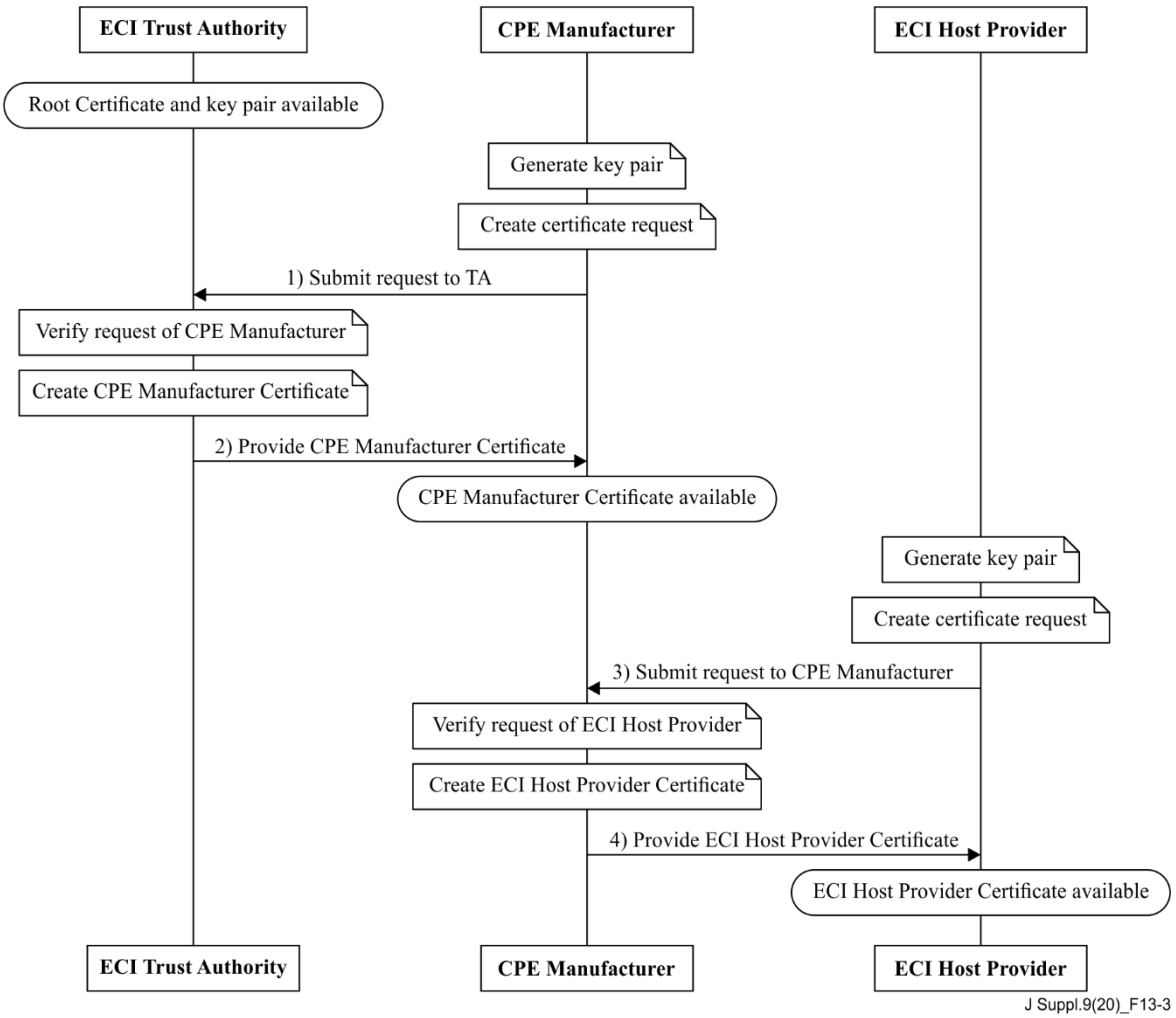

Figure 13-3 – Creation of certificates for an ECI Host image

In this example the **ECI Host** Provider is the **Entity** to provide the software image for an **ECI Host.** Following the example of a **Certificate Chain** as described in clause 5.4.1 of [ITU-T J.1012], each two adjacent entities collaborate for the production and the signing of a **Certificate**. In the selected example **Certificates** are created for the **CPE Manufacturer** and the **ECI Host** Provider. **Root Certificate** and the key pair for the **ECI TA** are already available:

- Step 1: For the collaboration with the **ECI Trust Authority** the **CPE Manufacturer** generates its pair of public and private keys which are used for the signing and the verification of **Certificate**s. The **CPE Manufacturer** creates a certificate request that is passed in this step to the **ECI Trust Authority**. It is a widely used method that such a request has already been signed by the **Entity** that asks for certification but **ECI** does not postulate such a procedure. The request is then verified by the **ECI TA** taking into account the existing bilateral agreements between the two entities. After successful verification, the **ECI TA** creates the **CPE Manufacturer Certificate** by signing the certificate using its private key. This signature is part of the **Certificate** and it uses the crypto graphical functions as defined in Annex A of [ITU-T J.1012]. A hash value is calculated for this **Certificate** and this value is encrypted by the private key of the **ECI TA.** The structure of an **ECI Certificate** is described in clause 5.2 of [ITU-T J.1012].
- Step 2: The **Certificate** is passed to the **CPE Manufacturer** for its usage.
- Step 3: For the provisioning of the **ECI Host** Provider **Certificate** the same processing steps are applied as described in steps 1 and 2 above. Collaboration takes place now between the **ECI Host** Provider and the **CPE Manufacturer.** The host provider's key pair is generated,

and a certificate request is created that is passed in this step to the **CPE Manufacturer**. If the request has been successfully verified, the **CPE Manufacturer** signs the certificate by its private key thus creating a valid **ECI Host** Provider **Certificate**.

- Step 4: This **Certificate**, containing information about the type of **Entity**, its identification number and its public key, is passed in this step to the **ECI Host** Provider for further usage. The **ECI Host Image** is signed by the private key of the **ECI Host** Provider.

In an **ECI** system **Certificate**s can be accompanied by **Revocation List**s that give information about the validity of **Certificate**s because it cannot always be ensured that at each point in time only one **Certificate** is circulated and installed. **Revocation List**s are linked to either one or even more **Certificate**s and they are published by those entities that also signed the accompanying **Certificate**s. The processing steps done by the entities introduced in this example are shown in Figure 13-4.

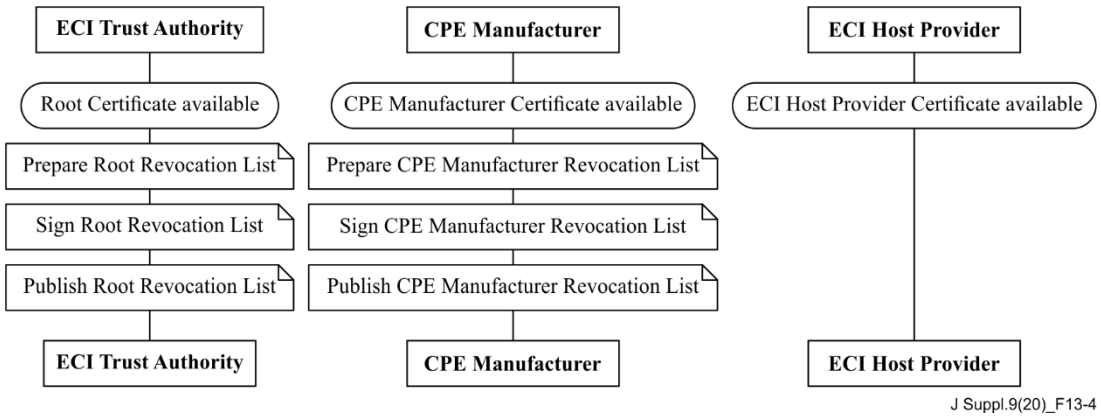

Figure 13-4 – Activities for the publication of Revocation Lists for an ECI Host

The two entities that were signing the appropriate **Certificate**s are also preparing the publishing of **Revocation List**s. These lists contain information about the trustfulness of certificates and their publishing entities but also about the validity of previously published **Revocation List**s by describing the minimum acceptable version number of these lists. Before the publication of the **Revocation List**s the entities sign them with their private keys using the crypto graphical functions as defined in Annex A of [ITU-T J.1012]. A chain of trust is established in such a way that a **Revocation List** also carries the minimum version of the accepted **Revocation List** of the **Child** of the **Entity** that is publishing this **Revocation List**. Two signed **Certificate**s and two signed **Revocation List**s are now existing and can be used for verification purposes during the installation of the **ECI Host Image**.

In this clause an example has been given for the creation of **Certificate**s and **Revocation List**s to ensure in a trustful way the installing of an **ECI Host Image**. For the following description of the verification process it is assumed that the entities that are involved in the deployment of an **ECI Host Image** agreed to store their signed documents in a common repository that is called **ECI** Trust Repository as shown in Figure 13-5.

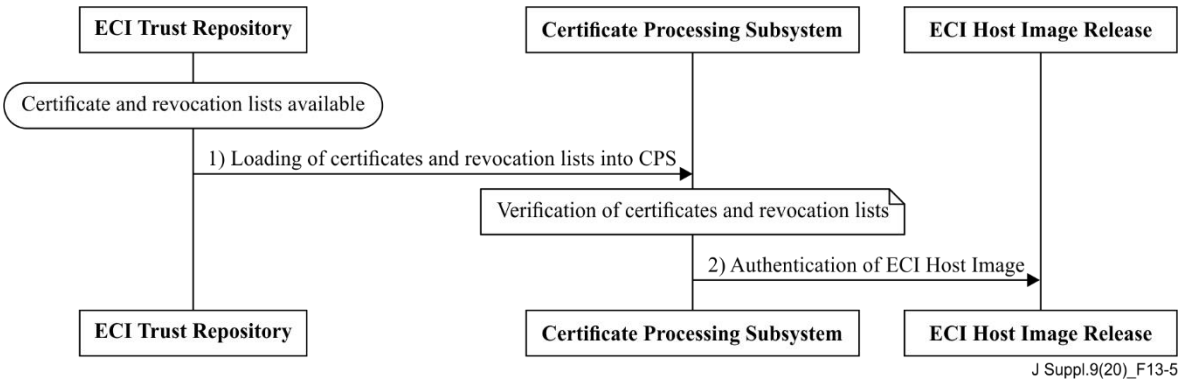

Figure 13-5 – Verification of certificates for an ECI Host image

In this example the repository holds the certificates and revocation lists of the three entities **ECI Trust Authority**, **CPE Manufacturer** and **ECI Host** Provider. These are the **Root Revocation List**, the **CPE Manufacturer Revocation List**, the **Root Certificate**, the **CPE Manufacturer Certificate** and the **ECI Host** Provider **Certificate**.

- Step 1: To start the verification process, all these documents are loaded into the **Certificate Processing Subsystem** of the CPE.

The verification steps only make use of the public keys of the involved entities. The public key of the **Root Certificate** is used to verify the **Root Revocation List** and the **CPE Manufacturer Certificate**. The public key of the **CPE Manufacturer Certificate** is used to verify the **CPE Manufacturer Revocation List** and the **ECI Host** Provider **Certificate**. The generic processing steps are described in clause 10.1 of [ITU-T J.1014]. Specific rules that hold for such a type of validation are given in clause 10.2 of [ITU-T J.1014].

In order to ensure that the **ECI Host Image** will only be installed on an authorized CPE, during this verification process some values described in Table 6.2.2.1-1 of [ITU-T J.1012] will be checked against the content of an identifier of the CPE that is protected by the **Advanced Security System** of the CPE. The data structure of this identifier, the **Chipset-ID,** is described in clause 6 of [ITU-T J.1015] and the authentication process in clause 8. The public key of the **ECI Host** Provider **Certificate** is then used to validate the **ECI Host Image**.

- Step 2: If this last validation step also succeeds, authentication is given for the implementation of the **ECI Host Image** as shown in step 2 in Figure 13-5.

13.4 Trust Provisioning for an ECI Client

After the installation of the **ECI Host** at least one **ECI Client** will be installed on the CPE. In this clause the provisioning of trust for the operation of such an **ECI Client** is described, for example, by using the following three entities:

- **ECI Trust Authority**: this entity governs the rules and regulations for the operation of an **ECI** system and is the owner of the root of trust for the verification of a trust chain.
- **Operator**: this entity provides one or more platforms for the delivery of services using one or more **ECI Client**s. An **Operator** may operate multiple platforms.
- **Platform Operation**: this entity represents a specific instance of a technical service platform containing a specific **ECI Client** having a single **ECI** identity.

The collaboration amongst the entities and the processing structure for certificates and revocation lists for signing purposes is the same as shown in Figure 13-3 but it involves different entities as shown in Figure 13-6. Each of these entities possesses its unique set of private and public keys for signing and validating **Certificate**s and **Revocation List**s.

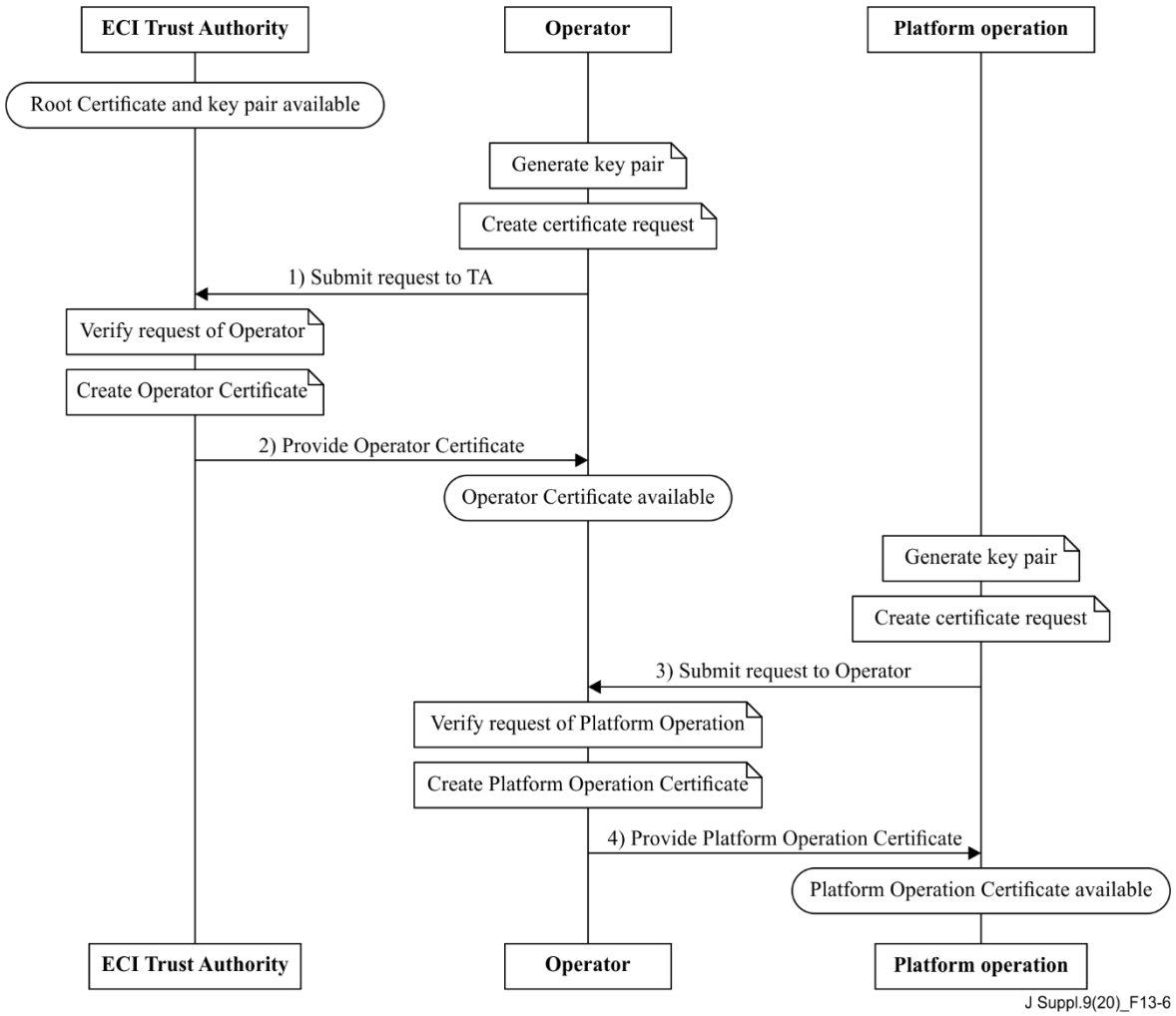

Figure 13-6 – Creation of certificates for an ECI Client

In this example the **Platform Operation** is the **Entity** that permits an installed **ECI Client** to run on a given platform. Following the explanation of a **Certificate Chain** as described in clause 5.4.1 of [ITU-T J.1012], each two adjacent entities collaborate for the production and the signing of a **Certificate**. In the selected example **Certificate**s are created for the **Operator** and the **Platform Operation**. **Root Certificate** and the key pair for the **ECI TA** are already available.

For the collaboration with the **ECI Trust Authority** the **Operator** generates its pair of public and private keys which are used for the signing and the verification of **Certificate**s.

- Step 1: The **Operator** creates a certificate request that is passed to the **ECI Trust Authority**.

It is a widely used method that such a request has already been signed by the **Entity** that asks for certification but **ECI** does not postulate such a procedure. The request is then verified by the **ECI TA** taking into account the existing bilateral agreements between the two entities. After successful verification, the **ECI TA** creates the **Operator Certificate** by signing the certificate using its private key. This signature is part of the **Certificate** and it uses the crypto graphical functions as defined in Annex A of [ITU-T J.1012]. A hash value is calculated for this **Certificate** and this value is encrypted by the private key of the **ECI TA.** The structure of an **ECI Certificate** is described in clause 5.2 of [ITU-T J.1012].

- Step 2: The **Certificate** is passed to the **Operator** for its usage.

For the provisioning of the **Operation Certificate** the same processing steps are applied as described in the previous paragraph. Collaboration takes place now between the **Platform Operation** and the **Operator**.

- Step 3: The **Platform Operation** key pair is generated, and a certificate request is created that is passed in this step to the **Operator**.

If the request has been successfully verified, the **Operator** signs the certificate by its private key thus creating a valid **Platform Operation Certificate**.

- Step 4: This **Certificate**, containing a numerical platform operation identifier, the **Platform Operation** public key and the version number of the **Certificate**, is passed in this step to the **Platform Operation** for further usage. The **ECI Client Image** is signed by the private key of the **ECI Platform Operation** to ensure the applicability of the **ECI Client** for a specified platform.

In an **ECI** system **Certificate**s can be accompanied by **Revocation List**s that give information about the validity of **Certificate**s because it cannot always be ensured that at each point in time only one **Certificate** is circulated and installed. **Revocation List**s are linked to either one or even more **Certificate**s and they are published by those entities that also signed the accompanying **Certificate**s. The processing steps done by the entities introduced in this example are shown in Figure 13-7.

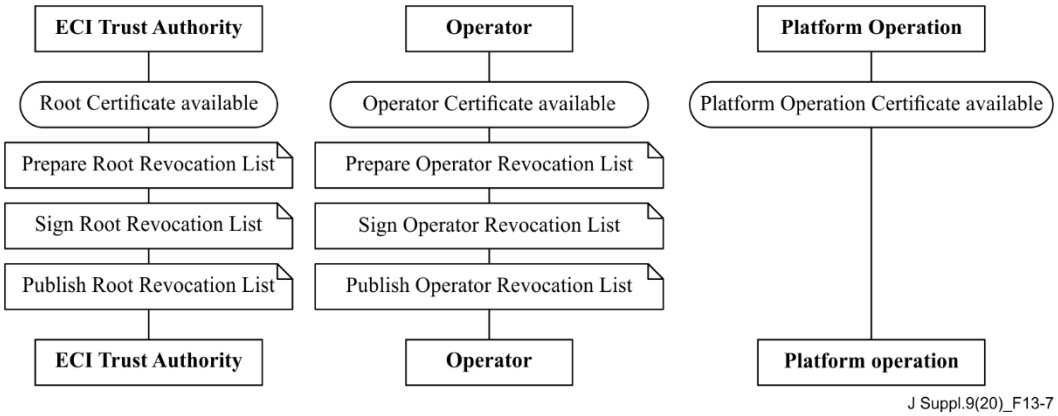

Figure 13-7 – Activities for the publication of Revocation Lists for an ECI Client

The two entities that were signing the appropriate **Certificate**s are also preparing the publishing of **Revocation List**s. These lists contain information about the trustfulness of certificates and their publishing entities but also about the validity of previously published **Revocation List**s by describing the minimum acceptable version number of these lists. Prior to the publication of the **Revocation List**s the entities sign them with their private keys using the crypto graphical functions as defined in Annex A of [ITU-T J.1012]. A chain of trust is established in such a way that a **Revocation List** also carries the minimum version of the accepted **Revocation List** of the **Child** of the **Entity** that is publishing this **Revocation List**. Two signed **Certificate**s and two signed **Revocation List**s are now existing and can be used for verification purposes during the authentication of the **ECI Client Image**.

In this clause an example has been given for the creation of **Certificate**s and **Revocation List**s to ensure in a trustful way the operation of an **ECI Client Image**. For the following description of the verification process it is assumed that the entities that are involved in the authentication of an **ECI Client Image** agreed to store their signed documents in a common repository that is called **ECI** Trust Repository as shown in Figure 13-8.

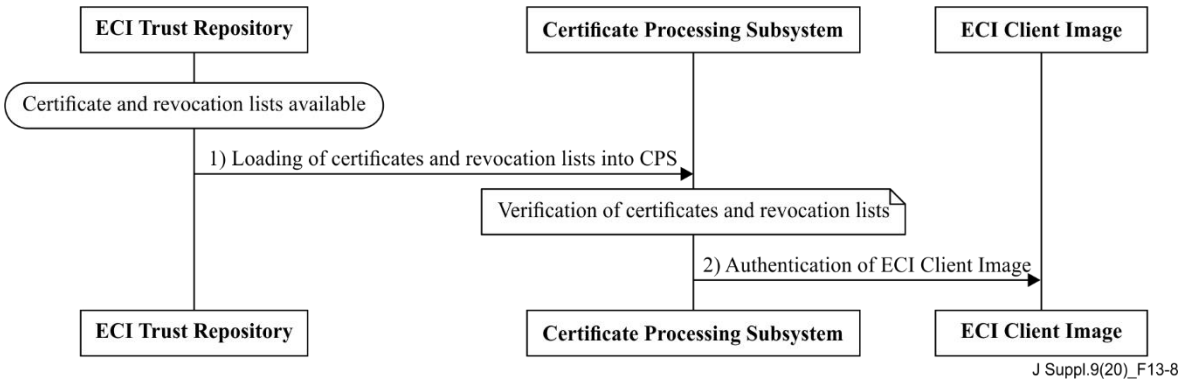

Figure 13-8 – Verification of certificates for the operation of an ECI Client

The repository holds the certificates and revocation lists of the three entities, i.e., **ECI Trust Authority**, **Operator** and **Platform Operation**. These are the **Root Revocation List**, the **Operator Revocation List**, the **Root Certificate**, the **Operator Certificate**, and the **Platform Operation Certificate**.

- Step 1: To start the verification process, all these documents are loaded into the **Certificate Processing Subsystem** of the CPE.

The verification steps only make use of the public keys of the involved entities. The public key of the **Root Certificate** is used to verify the **Root Revocation List** and the **Operator Certificate**. The public key of the **Operator Certificate** is used to verify the **Operator Revocation List** and the **Platform Operation Certificate**. The generic processing steps are described in clause 10.1 of [ITU-T J.1014]and the specific rules that hold for such a type of validation are given in clause 10.4.

In order to ensure that the **ECI Client** operates in a non-malicious way and will not get in conflicts with other **ECI Client**s that run in parallel on the same CPE, the actual status of the **ECI Client** and its capabilities are represented in the **Advanced Security System** of the CPE as an **AS Slot**. The structure of such a slot is defined in clause 8.2.2.1 of [ITU-T J.1014], the configuration for decryption purposes is described in clause 8.2.2.2 of [ITU-T J.1014], the configuration for the encryption of media data is given in clause 8.2.2.3 of [ITU-T J.1014]. The public key of the **Platform Operation Certificate** is finally used to validate the **ECI Client Image**.

- Step 2: If this last validation step also succeeds, authentication is given for the operation of the **ECI Client Image** as shown in step 2 of Figure 13-8.

Bibliography

[b-ITU-T H.222.0] Recommendation ITU-T H.222.0 (2018) | ISO/IEC 13818-1:2019, *Information technology – Generic coding of moving pictures and associated audio information: Systems*.

[b-ITU-T J-Suppl.8] ITU-T J-series Recommendations Supplement 8 (2020), *Embedded common interface for exchangeable CA/DRM solutions; Trust environment*.

[b-ETSI GS ECI 001-1] ETSI GS ECI 001-1 (V1.2.1): *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 1: Architecture, Definitions and Overview*.

[b-ETSI GS ECI 001-2] ETSI GS ECI 001-2 (V1.2.1): *"Embedded Common Interface (ECI) for exchangeable CA/DR solutions; Part 2: Use cases and requirements"*.

[b-ETSI GS ECI 002] ETSI GS ECI 002 (2018): *"Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; System validation"*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
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	Environment and ICTs, climate change, e-waste, energy efficiency; 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	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
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
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020