International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1015

(04/2020)

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

Conditional access and protection – Exchangeable embedded conditional access and digital rights management solutions

Embedded common interface for exchangeable CA/DRM solutions: The advanced security system – Key ladder block

Recommendation ITU-T J.1015

Recommendation ITU-T J.1015

Embedded common interface for exchangeable CA/DRM solutions: The advanced security system – Key ladder block

Summary

Recommendation ITU-T J.1015 is part of a series covering the advanced security system key ladder block for the embedded common interface for exchangeable conditional access/digital rights management (CA/DRM) solutions specification.

This ITU-T Recommendation is a transposition of ETSI standard [b-ETSI GS ECI 001-5-2] and is a result of collaboration between ITU-T SG9 and ETSI ISG ECI.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.1015	2020-04-23	9	11.1002/1000/13576

Keywords

Conditional access, CA, digital rights management, DRM, swapping.

* 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 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 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/.

Table of Contents

Introduction

This ITU-T Recommendation[1] is a transposition of ETSI standard [b-ETSI GS ECI 001-5-2] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI.

The objective of this Recommendation is to facilitate interoperability and competition in electronic communications services and, in particular, in the market for broadcast and audio-visual devices. However other technologies are available and may also be appropriate and beneficial depending on the circumstances in Member States.

A content provider encrypts their digital content and uses a **content protection system**[2] in order to protect the content against unauthorized access. A consumer uses a **content receiver** to access protected content. To this end, the **content receiver** contains a chipset that implements one or more content decryption operations. A cryptographic key establishment protocol is used to secure the transport of content decryption keys from the **content protection system** to the chipset. The steps of the protocol that are implemented within the chipset are referred to as a key ladder in this Recommendation, which specifies a key ladder for the key establishment protocol presented in [b-ROEL].

The key ladder and the protocol may also be used to secure the transport of content encryption keys to the chipset. Such keys are required for use cases in which the chipset re-encrypts content. The chipset may implement one or more content encryption operations for this purpose. Personal video recording and exporting protected content to a different **content protection system** are typical examples of content re-encryption use cases. Content decryption keys and content encryption keys are both referred to as **control word**s (CWs) throughout this Recommendation.

This Recommendation also specifies an authentication mechanism. This mechanism is closely related to the key ladder and may be used for entity authentication; in other words, this mechanism may be used to authenticate the chipset.

The key ladder and authentication mechanism specified in this Recommendation are agnostic to both the **content protection system** and the **content provider**. This enables a **content provider** to use any compliant **content protection system**, and it enables a consumer to use the **content receiver** for accessing content of any **content provider** that uses a compliant **content protection system**.

A **certification authority** manages a public-key certificate of each chipset in the mechanisms specified in this Recommendation. In particular, the **certification authority** distributes such certificates and certificate revocation information to **content provider**s who wish to make use of the key ladder and/or the authentication mechanism. Next, **content provider**s use the certificates and certificate revocation information as input to their compliant **content protection system**; as detailed in clause 7; knowledge of the public key in the certificate of a chipset enables the **content protection system** to generate suitable input messages for the key ladder and authentication mechanism of the chipset.

---

[1] Several areas for further development have been identified in Appendix I.

[2] The use of boldface in the text of this Recommendation indicates terms with definitions specific to the context of the embedded common interface that may differ from common use.

Recommendation ITU-T J.1015

Embedded common interface for exchangeable CA/DRM solutions: The advanced security system – Key ladder block

1 Scope

This Recommendation specifies a key ladder block for implementation in a chipset of a **content receiver**. The key ladder block comprises a key ladder to secure the transport of **control word**s (CWs) to the chipset and an authentication mechanism. This Recommendation also specifies aspects of the personalization of a compliant chipset.

This Recommendation is intended for use by chipset manufacturers.

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.

[ISO/IEC 9797-1] ISO/IEC 9797-1:2011, *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher*.

[IETF RFC 8017] IETF RFC 8017 (2016), *PKCS #1: RSA cryptography specifications version 2.2*.

[NIST FIPS 180-4] NIST FIPS PUB 180-4 (2015), *Secure Hash Standard (SHS)*.

[NIST FIPS 197] NIST FIPS PUB 197 (2001), *Specification for the Advanced Encryption Standard (AES)*.

[NIST SP 800-107] NIST SP 800-107 Revision 1 (2012), *Recommendation for applications using approved hash algorithms*.

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 certification authority: Party that is responsible for managing public-key certificates in an embedded common interface **(ECI) ecosystem**. A certification authority is trusted by all other parties in the system to perform operations associated with certificates.

3.2.2 chipset-ID: Non-secret number that is used to identify a chipset within an **ECI ecosystem**.

3.2.3 content protection system: System in an **ECI ecosystem** that employs cryptographic techniques to manage access to content and services. The term may be interchanged frequently with

the alternate Service Protection system. Typical systems of this sort are either conditional access (CA) systems or digital rights management (DRM) systems.

3.2.4 content provider: Party that distributes digital content to a **content receiver** in an **ECI ecosystem**.

3.2.5 content receiver: Device that is used to access digital content within an **ECI ecosystem**. A **content receiver** contains a chipset with a **content descrambler**.

3.2.6 content descrambler: Component in the chipset of an **ECI ecosystem** that is capable of decrypting content. A content descrambler may also be capable of encrypting content (for the purpose of content re-encryption). In this Recommendation content encryption/decryption uses a **symmetric encryption scheme**. For MPEG-2 content, content encryption and decryption are also referred to as scrambling and descrambling, respectively.

3.2.7 control word: Secret key used to encrypt and decrypt content within an **ECI ecosystem**. In digital rights management systems, a control word is typically referred to as a content key.

3.2.8 cryptographic hash function: Unkeyed cryptographic function in an **ECI ecosystem** that takes data of arbitrary size, referred to as the message, as input and produces an output data block of fixed size, referred to as the message digest. Assumed properties of the **cryptographic hash function** in this Recommendation are that the **cryptographic hash function** behaves as a random function and is second preimage resistant.

3.2.9 digital signature scheme: Keyed asymmetric cryptographic scheme that is used to protect the authenticity of data in an **ECI ecosystem**. A **digital signature scheme** consists of a key generation algorithm, a signature generation operation and a signature verification operation. Keys are generated as (secret/private key, public key) pairs. The data is signed using a secret/private key and the corresponding public key is used to verify the signature. The **digital signature scheme** specified in this Recommendation is used to protect the authenticity of messages as defined in [b-ROEL]; in particular, the scheme is not used to provide non-repudiation or source authentication in this Recommendation.

3.2.10 ECI ecosystem: A commercial operation consisting of a trust authority and several platforms and **ECI** – compliant customer premises equipment in the field.

3.2.11 message authentication code algorithm: Keyed symmetric cryptographic algorithm that is used to protect the authenticity of data in an **ECI ecosystem**. A **message authentication code algorithm** takes a message and a secret key as inputs, and produces an output data block referred to as the MAC. The **message authentication code algorithm** as specified in this Recommendation is used to cryptographically bind a ciphertext message to its associated data; in particular, the algorithm is not used to provide source authentication in this Recommendation.

3.2.12 public-key encryption scheme: Keyed asymmetric cryptographic scheme that is used to protect the confidentiality of data in an **ECI ecosystem**. A **public-key encryption scheme** consists of a key generation algorithm, an encryption operation and a decryption operation. Keys are generated as (public key, secret/private key) pairs. Data is encrypted using a public key and the data is recovered from the ciphertext using the corresponding secret/private key.

3.2.13 symmetric encryption scheme: Keyed symmetric cryptographic scheme that is used to protect the confidentiality of data in an **ECI ecosystem**. A **symmetric encryption scheme** consists of a key generation algorithm, an encryption operation and a decryption operation. The encryption and decryption operations of a **symmetric encryption scheme** use the same secret key as input.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AES Advanced Encryption Standard

AD1	Associated Data 1
AD2	Associated Data 2
AK	Authentication Key
CA	Conditional Access
CID	Company Identifier
CISSA	Common IPTV Software-oriented Scrambling Algorithm
CPU	Central Processing Unit
CSA	Common Scrambling Algorithm
CPK	Chipset Public Key
CSK	Chipset Secret/private Key
CW	Control Word
DRM	Digital Rights Management
DVB	Digital Video Broadcasting
ECB	Electronic Code Book
ECI	Embedded Common Interface
ECP	Enhanced Content Protection
ID	Identifier
LK	Link Key
MAC	Message Authentication Code
MK	MAC Key
OUI	Organizationally Unique Identifier
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SIM	Signed Input Message
SPK	Sender Public Key
SSK	Sender Secret/private Key
T	Tag
URI	Usage Rules Information

5 Conventions

The use of boldface in the text of this Recommendation indicates terms with definitions specific to the context of the embedded common interface that may differ from those in common use.

6 Chipset-ID and chipset master key pair

This clause specifies aspects of the personalization of a compliant chipset. Each compliant chipset is associated with a bit string identifier (ID) for the chipset, referred to as the **chipset-ID**, and a chipset master key pair.

A globally unique 64-bit **chipset-ID** shall be allocated to every compliant chipset. If the bits of **chipset-ID** are numbered 0 to 63 from left to right and if the ith bit ($0 \leq i \leq 63$) is denoted by b_i, then (b_0, b_1, b_2, b_3) shall contain a registration authority ID. Each value of the registration authority ID shall be associated with at most one registration authority. The **chipset-ID** shall also contain a chipset manufacturer ID. The value of the registration authority ID and the chipset manufacturer ID of a compliant chipset's **chipset-ID** shall uniquely identify the chipset manufacturer that produced the

chipset. In addition, the registration authority identified by the value of (b_0, b_1, b_2, b_3) shall administer the assignment of chipset manufacturer IDs that can be used in combination with this value.

If (b_0, b_1, b_2, b_3) = (0, 0, 0, 0), then the IEEE Registration Authority (https://standards.ieee.org/faqs/regauth.html) shall be the registration authority and the 24-bit OUI or the 24-bit CID shall be used to identify chipset manufacturers. In addition, if (b_0, b_1, b_2, b_3) = (0, 0, 0, 0), then (b_4, b_5 … b_{27}) shall contain the OUI/CID, b_4 being the most significant bit of octet 0 of the OUI/CID (see also [b-IEEE-OUI]) and b_{27} being the least significant bit of octet 2 of the OUI/CID.

All other values of the registration authority ID are reserved for future use.

The chipset master key pair is associated with a public-key encryption scheme, and consists of a chipset secret/private key (CSK) and a chipset public key (CPK). As detailed in clauses 7 and 8, (CSK, CPK) is the master key pair of both the key ladder and the authentication mechanism. The **public-key encryption scheme** and the representations of CSK and CPK are specified in clause 10.2.

A compliant chipset should generate its own key pair (CSK, CPK) to prevent disclosure of the value of CSK to any party in the system. In this case, only **chipset-ID** needs to be distributed to the chipset during its personalization, and the authenticity of **chipset-ID** shall be protected during this distribution. If the chipset does not generate its own key pair, then the authenticity of the triple (**chipset-ID**, CSK, CPK) and the confidentiality of CSK shall be protected during the distribution of (**chipset-ID**, CSK, CPK) to the chipset.

The random number used as input to the (CSK, CPK) key pair generation algorithm shall have at least 128 bits of entropy.

A compliant chipset shall permanently store its triple (**chipset-ID**, CSK, CPK), and a compliant chipset shall implement measures to protect the confidentiality of the stored CSK and the integrity of the stored **chipset-ID**, CSK and CPK.

The central processing unit (CPU) of the **content receiver** shall have read access to the stored **chipset-ID** and the stored CPK. This enables a **certification authority** to use the exported information as input to create a public-key certificate for the chipset. In addition, the exported **chipset-ID** enables a **content provider** to identify the chipset and its certificate.

A **certification authority** shall maintain the pair (**chipset-ID**, CPK) for every chipset it manages. This Recommendation does not exclude the presence of more than one **certification authority** in the system. The authenticity of the pair (**chipset-ID**, CPK) shall be protected during its distribution to the associated **certification authority** or authorities.

7 Key ladder

7.1 Overview

This clause presents the functional design of the key ladder. The block in the chipset that implements the key ladder is referred to as the key ladder block throughout this Recommendation. The key ladder is depicted in Figure 7-1. As specified in clause 6 and shown in Figure 7-1, the chipset is personalized with a **chipset-ID** and with a CSK.

One of the outputs of the key ladder block is a **control word** denoted by CW, which is used for either content decryption or content encryption. A second output of the key ladder block is a bit string composed of the control word-usage rules information (CW-URI), which determines the URI for a CW (see clause 7.3.1 for the specification of the CW-URI and the associated usage rule). CW and CW-URI are inputs to the **content descrambler** (which is not depicted in Figure 7-1).

- The key ladder block and the **content descrambler** shall be implemented in a single silicon chip.

- If the **content descrambler** offers an interface to a processor in the **content receiver** that allows the processor to pass plaintext CWs to the **content descrambler** (i.e., by-passing the key ladder block), then it shall be possible to permanently disable this functionality.
- If the key ladder computes a CW, then only the key ladder block and the **content descrambler** shall have access to this CW.
- The authenticity of the pair (CW, CW-URI) and the confidentiality of CW shall be protected during their distribution from the key ladder block to the **content descrambler**.

The key ladder block shall interface with a processor of the content receiver. For example, the processor may be a security processor or the CPU of the **content receiver**. As specified in clause 6 and as shown in Figure 7-1, this processor has read access to the **chipset-ID**. This enables a **content provider** to identify the chipset and obtain the corresponding public-key certificate containing the CPK from the **certification authority**. The value of CPK needs to be known to compute one of the input messages to the key ladder, as described in clause 7.2.

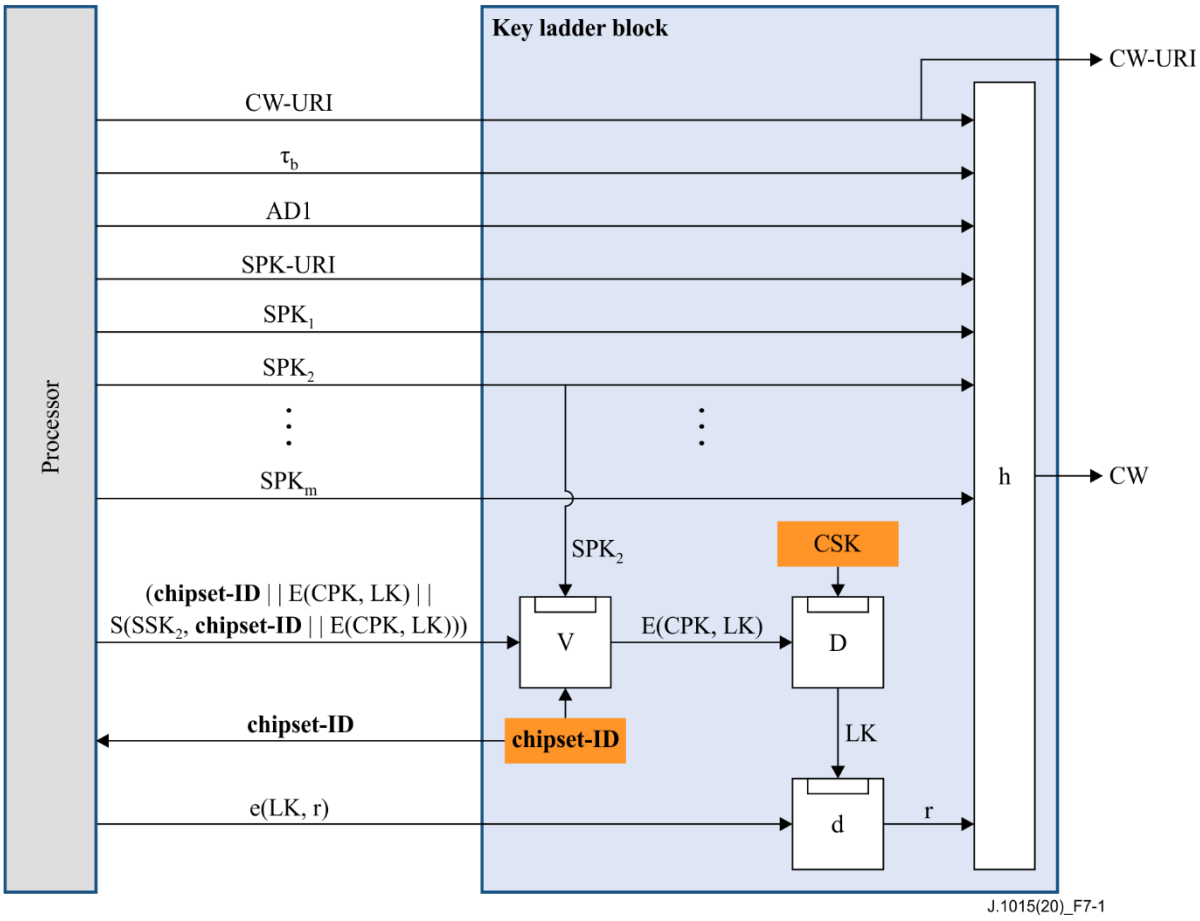

Figure 7-1 – Key ladder

As specified in clause 6, the pair (CSK, CPK) is associated with a **public-key encryption scheme**. The corresponding encryption and decryption operations are denoted by E and D, respectively. E and D are keyed cryptographic operations and each of these operations has two inputs: a key input and a message input. This Recommendation assumes that the first input of a keyed cryptographic operation or algorithm is the key.

Example: The encryption of a message *M* using E and CPK is written as E(CPK, *M*).

The mechanisms specified in this Recommendation also use a **digital signature scheme**; S and V denote the signature generation operation and the signature verification operation, respectively. A key pair of the **digital signature scheme** is associated with a sender and consists of a sender

secret/private key (SSK) and a sender public key (SPK). This Recommendation assumes that a sender is a **content protection system**. As shown in Figure 7-1, a number of different SPKs, denoted by SPK_1, SPK_2 ... SPK_m ($m \geq 1$) are input to the key ladder block. The key ladder block shall support key ladders for all values of m such that $1 \leq m \leq 16$. In practice, each key pair (SSK_i, SPK_i) with $1 \leq i \leq m$ will typically be associated with one **content protection system**; however, such a key pair may also be shared between multiple systems.

Associated with SPK_1, SPK_2 ... SPK_m is the SPK-URI. This input to the key ladder block determines the URI for SPK_1, SPK_2 ... SPK_m. The SPK-URI and the associated usage rules are specified in clause 7.3.2. As shown in Figure 7-1, one of the SPKs and the verification operation V are used to verify the signature of the input message (**chipset-ID** || E(CPK, LK) || S(SSK_i, **chipset-ID** || E(CPK, LK))). This signed input message (SIM) is also denoted by SIM_{KL} in the following text. Figure 7-1 assumes that $i = 2$ and that SPK-URI and the usage rules allow SPK_2 to be used for the verification of the signature.

The key ladder also implements a **symmetric encryption scheme**. The encryption and decryption operations of this scheme are denoted by e and d, respectively. The key ladder uses a link key (LK) as the key of this scheme and a random number r (or another LK as detailed in clause 7.4) as the message. The random number r is represented as a bit string and its length shall be 128 bits.

Finally, the key ladder block implements a function h. This function is based on a **cryptographic hash function** (see clause 10.4 for the specification of h).

As shown in Figure 7-1, the other two inputs to the key ladder block are τ_b and associated data 1 (AD1). Both these inputs are represented as bit strings:

- The length of τ_b shall be 8 bits. The use of τ_b is specified in clause 7.5.
- The length of AD1 shall be 256 bits. The specification of the contents of AD1 lies outside the scope of this Recommendation, which assumes that the key ladder block does not process AD1 other than providing it as input to h. The key ladder block may pass AD1, or part of AD1, together with CW-URI and CW to the **content descrambler**.

This Recommendation assumes that symmetric keys, ciphertext messages and signatures are represented as bit strings. Their lengths are specified in clause 10, in which the representation of asymmetric keys is also specified.

7.2 Key ladder computations

The sender associated with key pair (SSK_i, SPK_i) for some $1 \leq i \leq \underline{m}$ can create the signed input message SIM_{KL}, i.e.:

(**chipset-ID** || E(CPK, LK) || S(SSK_i, **chipset-ID** || E(CPK, LK)))

using the following steps.

Compute SIM_{KL} (sender)

1) Generate an LK.
2) Compute the ciphertext E(CPK, LK).
3) Concatenate **chipset-ID** and E(CPK, LK); the resulting bit string is denoted by (**chipset-ID** || E(CPK, LK)).
4) Sign the bit string (**chipset-ID** || E(CPK, LK)) using SSK_i; the signature is denoted by S(SSK_i, **chipset-ID** || E(CPK, LK)).
5) Append this signature to the bit string (**chipset-ID** || E(CPK, LK)).

After receiving SIM_{KL} and sender public key SPK_i, the key ladder block shall perform the following steps to retrieve LK (in Figure 7-1, it is assumed that the first three steps are performed by V).

Compute LK (key ladder block)

1) Verify whether the received **chipset-ID** equals the stored **chipset-ID**. If these two values are not equal, then the key ladder block shall abort the computations.
2) Check whether SPK-URI and the usage rules as specified in clause 7.3.2 allow V to use SPK_i to verify the signature. If this is not allowed, then the key ladder block shall abort the computations.
3) Use the received SIM_{KL} and SPK_i to verify the signature. If the signature is invalid, then the key ladder block shall abort the computations.
4) Compute LK = D(CSK, E(CPK, LK)).

Next, the key ladder block shall use LK to process the input message e(LK, *r*) (see also Figure 7-1). The sender can create this message using the following steps.

Compute e(LK, *r*) (sender)

1) Generate a random bit string *r*.
2) Compute e(LK, *r*).

After receiving e(LK, *r*) and after computing LK, the key ladder block shall compute *r* using the following step (see also Figure 7-1).

Compute *r* (key ladder block)

1) *r* = d(LK, e(LK, *r*)).

Next, the key ladder block shall use the function h to compute the CW. As shown in Figure 7-1, the inputs to h are CW-URI, τ_b, AD1, SPK-URI, SPK_1, SPK_2 ... SPK_m, and *r*. The implementation of the key ladder shall ensure that the public key that was used to verify the authenticity of SIM_{KL} (or more precisely, the SIM_{KL} containing the LK associated with the random number *r*) is provided as one of the SPK-inputs to h when CW is derived from *r*. Next, the key ladder block shall pass CW-URI and CW to the **content descrambler**.

7.3 Usage rules information

7.3.1 Control word-usage rules information

The length of CW-URI shall be 64 bits and these bits are numbered 0 to 63 from left to right. The value of CW-URI determines the allowed usage of the CW (see also Table 7-1) according to the following usage rule: if the value of a bit is 1, then the specified use is allowed, otherwise this use is not allowed. The **content descrambler** shall enforce this usage rule. The **content descrambler** shall ignore:

i) the value of the bits that are reserved for future use; and
ii) the value of bits that correspond to implementations that the **content descrambler** does not support.

Table 7-1 – Listing of control word-usage rules information

Bit number	Description
0	Encrypt
1	Decrypt
2 … 7	Reserved for future use
8	Digital video broadcasting (DVB) common scrambling algorithm 2 (CSA2) [b-ETSI TS 100 289]
9	DVB CSA3 [b-ETSI TS 100 289]

Table 7-1 – Listing of control word-usage rules information

Bit number	Description
10 … 15	Reserved for future use
16	DVB common IPTV software-oriented scrambling algorithm (CISSA) version 1 [b-ETSI TS 103 127]
17 … 23	Reserved for future use
24	Advanced Television Systems Committee common scrambling/descrambling algorithm [b-ATSC A/70-1]
25 … 31	Reserved for future use
32	Common encryption scheme – cenc protection scheme [b-ISO/IEC 23001-7]
33	Common encryption scheme – cbc1 protection scheme [b-ISO/IEC 23001-7]
34	Common encryption scheme – cens protection scheme [b-ISO/IEC 23001-7]
35	Common encryption scheme – cbcs protection scheme [b-ISO/IEC 23001-7]
36 … 39	Reserved for future use
40	ChinaDRM – cipher block chaining mode [b-GY/T 277-2014]
41	ChinaDRM – counter mode [b-GY/T 277-2014]
42 … 63	Reserved for future use

7.3.2 Sender public key-usage rules information

The length of the SPK-URI shall be 64 bits, each numbered 0 to 63 from left to right. The SPK-URI is listed in Table 7-2. In particular, if SPK_1, SPK_2 ... and SPK_m are provided as inputs to the key ladder together with the SPK-URI (as depicted in Table 7-2), then bits 0, 1 … $m - 1$ and bits 16, 17 ... $16 + m - 1$ of the SPK-URI are used to determine two subsets of {SPK_1, SPK_2 ... SPK_m}: SPK_i ($i = 1, 2 \ldots m$) is an element of the first subset if and only if the value of bit $i - 1$ equals one, and SPK_i is an element of the second subset if and only if the value of bit $i + 15$ equals one. In the following text, these two subsets are denoted by S_1 and S_2, respectively. Using this notation, the key ladder shall apply the following usage rule:

SPK usage rule 1: in the key ladder, V is allowed to use SPK_i to verify the signature of SIM_{KL}, if and only if $SPK_i \in S_1$.

SPK usage rule 1 shall be enforced by the key ladder block.

The set S_2 is used in the case of content re-encryption. If content is re-encrypted, then two sets of inputs as depicted in Figure 7-1 are required: one associated with the CW used for content decryption and one associated with the CW used for content encryption. In this case, the following usage rule connects these two sets of inputs:

SPK usage rule 2: in the key ladder, V is allowed to use SPK_i to verify the signature of SIM_{KL} of the CW used for content encryption, if and only if $SPK_i \in S_2$ associated with the CW used for content decryption.

If the **content descrambler** supports content re-encryption, then the key ladder block should enforce SPK usage rule 2. If the key ladder block does not enforce this usage rule, then another component in the chipset shall enforce SPK usage rule 2 and the implementation shall ensure that the value of the SPK-URI that is input to the function h is equal to the value of SPK-URI that is used to enforce SPK usage rule 2.

Table 7-2 – Definition of the sender public key -usage rules information

Bit number	Description
0	spk1_in_set1
1	spk2_in_set1
2	spk3_in_set1
…	…
15	spk16_in_set1
16	spk1_in_set2
17	spk2_in_set2
18	spk3_in_set2
…	…
31	spk16_in_set2
32 … 63	Reserved for future use

7.4 Additional key layers

7.4.1 Overview

The specification in clauses 7.1 and 7.2 assumes that one key layer of the key ladder is associated with LKs. The key ladder shall also support additional key layers for LKs, as depicted in Figure 7-2 (in which only the building blocks of the key ladder that are relevant to the discussion in this clause are depicted).

As in Figure 7-2, let t denote the number of LKs in the key ladder. The key ladder block shall support key ladders for all values of t with $1 \leq t \leq 24$. Note that $t = 1$ in the scheme specified in clauses 7.1 and 7.2.

7.4.2 Key ladder computations

The sender can generate the t input messages to the key ladder block using the following steps.

Compute $e(LK_1, LK_2)$, $e(LK_2, LK_3)$, … , $e(LK_{t-1}, LK_t)$, and $e(LK_t, r)$ (sender)

1) Generate LK_i for $i = 1, 2 \ldots t$ and a random bit string r.
2) Compute $e(LK_{i-1}, LK_i)$ for $i = 2, 3 \ldots t$.
3) Compute $e(LK_t, r)$.

After receiving $e(LK_1, LK_2)$, $e(LK_2, LK_3)$ … $e(LK_{t-1}, LK_t)$, and $e(LK_t, r)$, the key ladder block shall compute r using the following steps (see also Figure 7-2).

Compute r (key ladder block)

1) Compute $LK_i = d(LK_{i-1}, e(LK_{i-1}, LK_i))$ for $i = 2, 3 \ldots t$.
2) Compute $r = d(LK_t, e(LK_t, r))$.

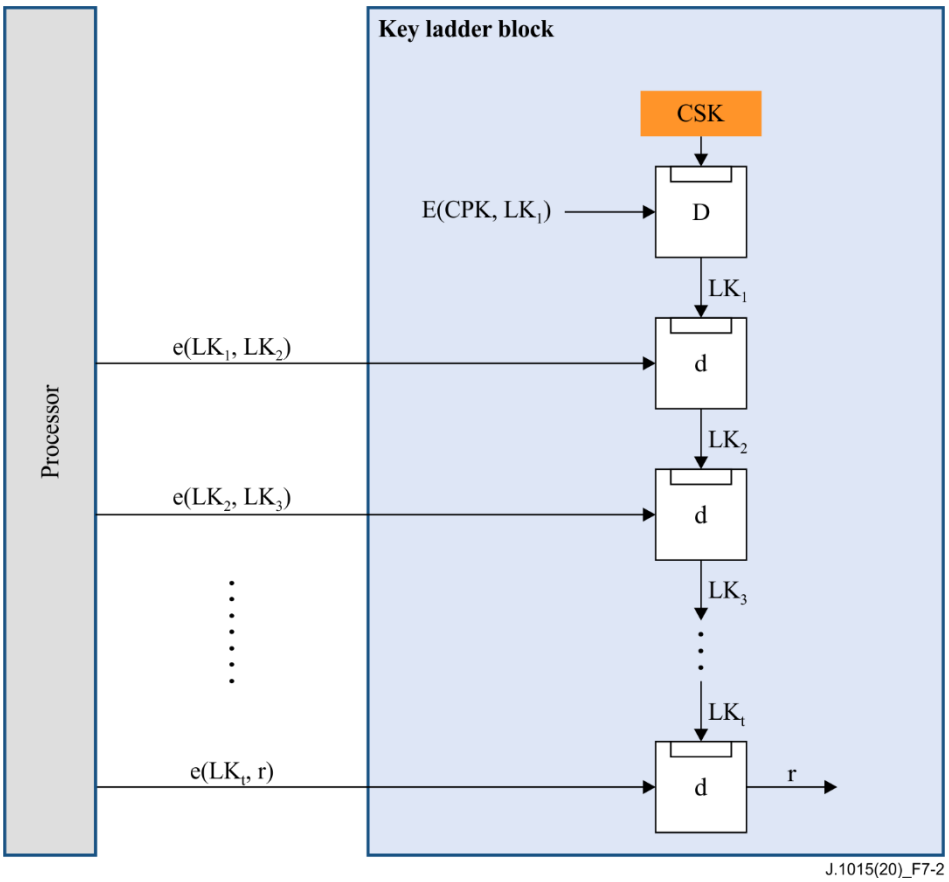

Figure 7-2 – Additional key layers

7.5 Associated data 2

The sender can optionally send associated data 2 (AD2), together with e(LK_t, r) to the key ladder. The value of the bit string τ_b (as introduced in clause 7.1) signals the presence of AD2. As specified in clause 7.1, the length of τ_b is 8 bits. In the following text, τ denotes the integer representation of τ_b in which the least significant bit of τ corresponds to the rightmost bit of τ_b. The key ladder shall support all values of $\tau \in \{0, 64, 96, 128\}$. If $\tau = 0$, then AD2 shall not be present, and the scheme as presented in clauses 7.1 to 7.4 shall be used. The following description assumes that $\tau \neq 0$. In this case, AD2 shall be present. The key ladder block shall verify whether the received $\tau \in \{64, 96, 128\}$; if this does not hold true, then the key ladder block shall abort the computations.

This Recommendation assumes that AD2 is represented as a bit string. Let Len denote the bit length of AD2. The key ladder block shall support all values of Len with $1 \leq \text{Len} \leq 256$.

Let Len_b denote the binary representation of the integer Len − 1 in which the rightmost bit of Len_b corresponds to the least significant bit of Len − 1. The length of Len_b shall be 8 bits.

The bit string AD2 shall be cryptographically bound to the corresponding ciphertext e(LK_t, r) during its transport to the key ladder. To this end, the sender and the key ladder block shall use a **message authentication code algorithm**, denoted by mac. The inputs to this algorithm are the secret MAC key (MK), the message Len_b || e(LK_t, r) || AD2, and τ. The output of a mac is a tag (T) with a length of τ bits. The MAC algorithm is specified in clause 10.5.

Let 00…0 denote the all-zero bit string of 128 bits. The sender can use the following steps to generate the input message Len_b || e(LK_t, r) || AD2 || T to the key ladder (as depicted in Figure 7-3).

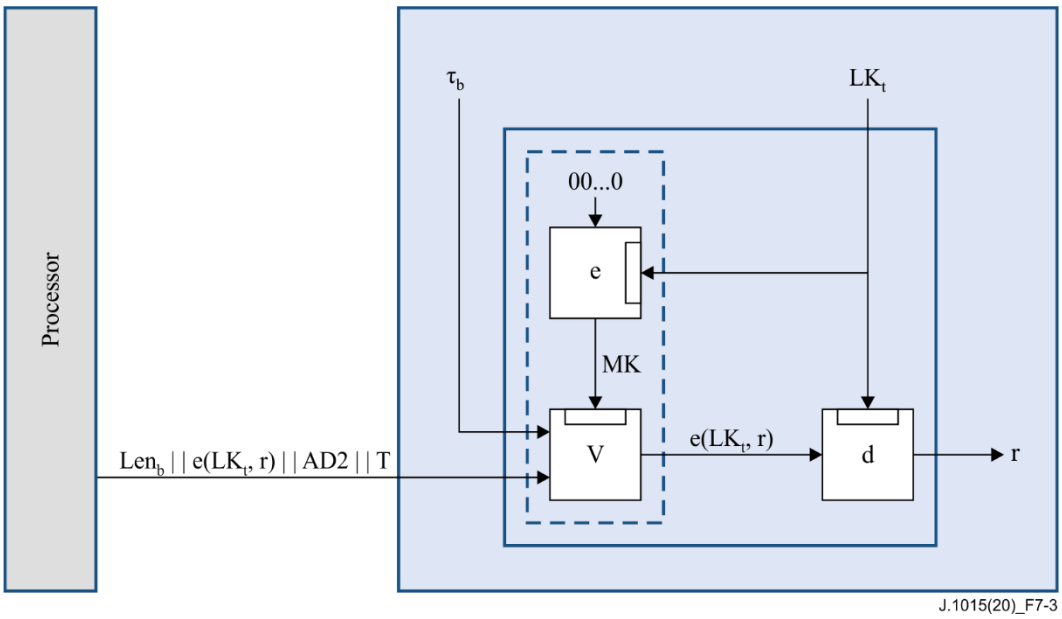

Figure 7-3 – Associated data 2

Compute Len_b || e(LK_t, r) || AD2 || T (sender)

1) Compute MK = e(LK_t, 00…0).
2) Prepend Len_b to the bit string e(LK_t, r) || AD2.
3) Compute T = mac(MK, Len_b || e(LK_t, r) || AD2, τ).
4) Append T to the bit string Len_b || e(LK_t, r) || AD2.

The message Len_b || e(LK_t, r) || AD2 || T replaces the message e(LK_t, r) that is used if $\tau = 0$. After receiving Len_b || e(LK_t, r) || AD2 || T, the key ladder block shall compute e(LK_t, r) using the following steps (Steps 2-4 are denoted by V in Figure 7-3).

Compute e(LK_t, r) (key ladder block)

1) Compute MK = e(LK_t, 00…0).
2) Compute T = mac(MK, Len_b || e(LK_t, r) || AD2, τ).
3) Verify whether the received T equals the computed T. If these two values are not equal, then the key ladder block shall abort the computations.
4) Retrieve e(LK_t, r) from the received message.

The specification of the contents of AD2 lies outside the scope of this Recommendation, which assumes that the key ladder block does not process AD2 other than using it as input to the MAC algorithm. The key ladder block may pass AD2, or part of AD2, together with CW-URI and CW to the **content descrambler**.

If $\tau \neq 0$, then the implementation of the key ladder shall ensure that the value of τ_b that was provided together with Len_b || e(LK_t, r) || AD2 as input to the MAC algorithm is also provided as input to h when CW is derived from r.

Only the key ladder block shall have access to a plaintext CSK, LK_i ($1 \leq i \leq t$), MK, and r.

8 Authentication mechanism

8.1 Overview

This clause presents the functional design of the authentication mechanism in the key ladder block. The authentication mechanism is depicted in Figure 8-1. This mechanism is closely related to the key

ladder as described in clause 7; in particular, it uses the same operations V, D and function h, and it uses the same **chipset-ID** and CSK. However, the authentication mechanism does not make use of LKs and CW-URI, SPK-URI and τ_b are not inputs to the authentication mechanism and the function h. Moreover, the output of h is not a CW used for content decryption or content encryption (in particular, the authentication mechanism does not interface directly with the **content descrambler**), but an authentication key (AK), which is a key of the same **symmetric encryption scheme** as used in the key ladder. As shown in Figure 8-1, only its decryption operation d is used in the implementation of the authentication mechanism.

The input message (**chipset-ID** || E(CPK, *r*) || S(SSK$_i$, **chipset-ID** || E(CPK, *r*))) is also denoted by SIM$_{AUTH}$ in the following text.

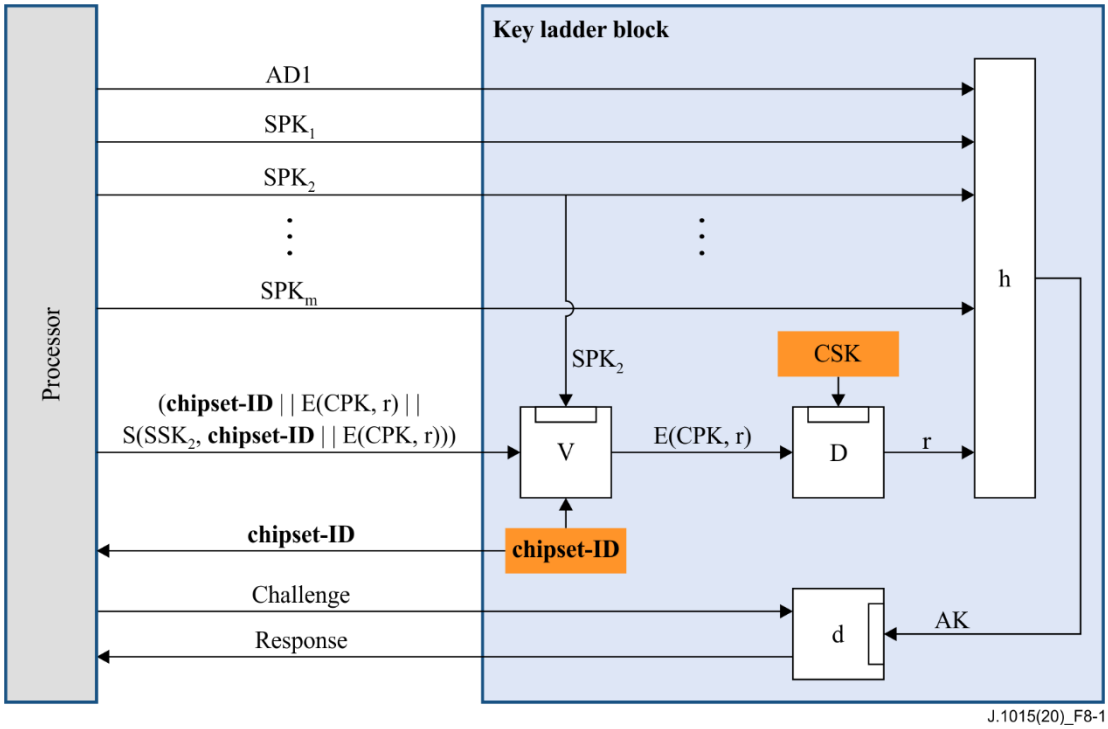

Figure 8-1 – Authentication mechanism

8.2 Authentication mechanism computations

The sender associated with key pair (SSK$_i$,SPK$_i$) for some $1 \le i \le$ m can create SIM$_{AUTH}$, i.e.,

(**chipset-ID** || E(CPK, *r*) || S(SSK$_i$, **chipset-ID** || E(CPK, *r*))),

using the following steps.

Compute SIM$_{AUTH}$ (sender)

1) Generate a random bit string *r*.
2) Compute the ciphertext E(CPK, *r*).
3) Concatenate **chipset-ID** and E(CPK, *r*); the resulting bit string is denoted by (**chipset-ID** || E(CPK, *r*)).
4) Sign the message (**chipset-ID** || E(CPK, *r*)) using SSK$_i$; the signature is denoted by S(SSK$_i$, **chipset-ID** || E(CPK, *r*)).
5) Append this signature to the bit string (**chipset-ID** || E(CPK, *r*)).

After receiving SIM$_{AUTH}$ and SPK$_i$, the key ladder block shall perform the following steps to retrieve *r* (in Figure 8-1, the first two steps are denoted by V).

Compute *r* (key ladder block)

1) Verify whether the received **chipset-ID** equals the stored **chipset-ID**. If these two values are not equal, then the key ladder block shall abort the computations.
2) Use the received SIM_{AUTH} and SPK_i to verify the signature. If the signature is invalid, then the key ladder block shall abort the computations.
3) Compute r = D(CSK, E(CPK, r)).

In the case of the authentication mechanism, V is allowed to use any SPK_i with $1 \leq i \leq m$ to verify the signature of SIM_{AUTH} (in Figure 8-1, it is assumed that $i = 2$).

Next, AD1, SPK_1, SPK_2 ... SPK_m and r shall be provided as inputs to the function h. The output of h is the AK. The implementation of the authentication mechanism shall ensure that the public key that was used to verify the authenticity of SIM_{AUTH} (or more precisely, the SIM_{AUTH} containing the random number r) is provided as one of the SPK-inputs to h when AK is derived from r.

Next, the sender can create an input message referred to as a challenge. After receiving a challenge from the processor, the authentication mechanism shall compute a response using the following step.

Compute Response (key ladder block)

1) Compute Response = d(AK, Challenge).

As depicted in Figure 8-1, the authentication mechanism shall return "Response" to the processor.

Only the key ladder block shall have access to a plaintext AK.

9 Data conversion primitives

9.1 BS2OSP

The bit string to octet string primitive BS2OSP is used to determine the **public-key encryption scheme** in clause 10.2 and the **digital signature scheme** in clause 10.3.

Function: BS2OSP(x)

Input: x a bit string of length $8j$ ($j \geq 1$).

Output: X an octet string of length j.

Steps:

1) Let x_i denote a bit for $0 \leq i \leq 8j - 1$, and let $x = x_0 x_1 \ldots x_{8j-1}$.
2) Let X_i be the octet determined by $X_i = x_{8i+1} \ldots x_{8i+7}$ for $0 \leq i \leq j - 1$.
3) Output the octet string $X = X_0 X_1 \ldots X_{j-1}$.

9.2 OS2BSP

The octet string to bit string primitive OS2BSP is used to determine the I2BSP primitive in clause 9.3 and the **public-key encryption scheme** in clause 10.2.

Function: OS2BSP(X)

Input: X an octet string of length j ($j \geq 1$).

Output: x a bit string of length $8j$.

Steps:

1) Let X_i denote an octet for $0 \leq i \leq j - 1$ and let $X = X_0 X_1 \ldots X_{j-1}$.
2) Let x_i denote a bit for $0 \leq i \leq 8j - 1$ and let these bits be determined by $x_i = x_{8i+1} \ldots x_{8i+7}$ for $0 \leq i \leq j - 1$.

3) Output the bit string $x = x_0 x_1 \ldots x_{8j-1}$.

9.3 I2BSP

The integer to bit string primitive I2BSP is used to determine the function h in clause 10.4.

Function: I2BSP(x)

Input: x a 2 048-bit integer.

Output: a bit string of length 2 048.

Step: If the integer to octet string data conversion primitive I2OSP(x, xLen) is specified as in [IETF RFC 8017], then I2BSP(x) = OS2BSP(I2OSP(x, 256)).

10 Cryptographic operations

10.1 Symmetric encryption scheme

The **symmetric encryption scheme** used in the key ladder block shall be Advanced Encryption Standard-128 (AES-128) [NIST FIPS 197] in the electronic code book (ECB) mode of operation.

10.2 Public-key encryption scheme

The **public-key encryption scheme** shall be RSAES-PKCS1-v1_5 [IETF RFC 8017]. This scheme consists of an encryption operation and a decryption operation. Recall from clauses 7 and 8 that only the decryption operation is implemented in the key ladder block, and that the sender's system implements the encryption operation.

The CPK shall be equal to the Rivest Shamir Adleman (RSA) public key of the recipient, (*n*, *e*) in [IETF RFC 8017]. In this Recommendation the bit length of *n* shall be 2 048, i.e., the octet length of *n*, denoted by *k* in [IETF RFC 8017], equals 256. The CSK shall be equal to the recipient's RSA private key *K* in [IETF RFC 8017]. The representation of CSK shall be equal to one of the representations of *K* specified in [IETF RFC 8017].

The message to be encrypted is denoted by *M* in [IETF RFC 8017] and represented as an octet string. In this Recommendation, *M* is determined by *M* = BS2OSP(LK) in the case of the key ladder, and *M* = BS2OSP(*r*) in case of the authentication mechanism. As a result, the octet length of *M* equals 16 in this Recommendation.

CPK and *M* are input to the encryption operation:

RSAES-PKCS1-v1_5-Encrypt(CPK, *M*).

The output of this operation is a ciphertext, denoted by *C* and represented as an octet string in [IETF RFC 8017]. The octet length of *C* equals 256 in this Recommendation. *C* equals BS2OSP(E(CPK, LK)) in the case of the key ladder and BS2OSP(E(CPK, *r*)) in the case of the authentication mechanism.

CSK and *C* are input to the decryption operation:

RSAES-PKCS-v1_5-Decrypt(CSK, *C*).

This operation is implemented in the key ladder block. The output of this operation is the message *M*, represented as an octet string. Next, the key ladder block shall compute LK = OS2BSP(*M*) in the case of the key ladder and *r* = OS2BSP(*M*) in the case of the authentication mechanism.

10.3 Digital signature scheme

The **digital signature scheme** shall be RSASSA-PKCS1-v1_5 [IETF RFC 8017]. This scheme consists of a signature generation operation and a signature verification operation. Recall from

clauses 7 and 8 that only the signature verification operation is implemented in the key ladder block, and that the sender's system implements the signature generation operation.

The RSA public key of the signer is denoted by (*n*, *e*) and represented as a pair of integers in [IETF RFC 8017]. In this Recommendation, the length of the modulus *n* of the RSA public key of the signer shall be 2 048 bits, i.e., the octet length of *n*, denoted by *k* in [IETF RFC 8017], equals 256. Further, the value of the public exponent *e* shall be equal to $2^{16} + 1$ in this Recommendation. A compliant chipset shall permanently store this value and the chipset shall implement measures to protect its integrity. The SPK shall be equal to the modulus *n*, implying that the length of SPK equals 2 048 bits and that SPK is represented as an integer. The SSK shall be equal to the RSA private key of the signer, *K* in [IETF RFC 8017].

The message to be signed is denoted by *M* and represented as an octet string in [IETF RFC 8017]. In this Recommendation, *M* is determined by:

$$M = \text{BS2OSP}(\textbf{chipset-ID} \,||\, \text{E}(\text{CPK}, \text{LK}))$$

in the case of the key ladder; and

$$M = \text{BS2OSP}(\textbf{chipset-ID} \,||\, \text{E}(\text{CPK}, r))$$

in the case of the authentication mechanism. As a result, the octet length of *M* is equal to 8 + 256 = 264 in both cases.

SSK and *M* are input to the signature generation operation:

$$\text{RSASSA-PKCS1-v1\_5-Sign}(\text{SSK}, M).$$

The output of the signature generation operation is a signature, denoted by *S* and represented as an octet string in [IETF RFC 8017]. The octet length of *S* equals 256 in this Recommendation. *S* equals $\text{BS2OSP}(\text{S}(\text{SSK}_i, \textbf{chipset-ID} \,||\, \text{E}(\text{CPK}, \text{LK})))$ in the case of the key ladder and $\text{BS2OSP}(\text{S}(\text{SSK}_i, \textbf{chipset-ID} \,||\, \text{E}(\text{CPK}, r)))$ in the case of the authentication mechanism.

$(\text{SPK}, 2^{16} + 1)$, *M* and *S* are input to the signature verification operation:

$$\text{RSAES-PKCS-v1\_5-Verify}((\text{SPK}, 2^{16} + 1), M, S).$$

This operation is implemented in the key ladder block. The output of this operation is either "valid signature" or "invalid signature".

10.4 Function h

This clause specifies the function h. Recall from clauses 7 and 8 that the inputs to h are:

- CW-URI, τ_b, AD1, SPK-URI, SPK_1, SPK_2 ... SPK_m, and *r* in the case of the key ladder;
- AD1, SPK_1, SPK_2 ... SPK_m, and *r* in the case of the authentication mechanism.

If the key ladder block does not receive one of these two sets of inputs or if the length of an input is not as specified in this Recommendation, then the key ladder block shall abort the computations. Otherwise, h shall first apply the I2BSP data conversion primitive to each of its SPK inputs. Next, h shall concatenate the bit stings representing its inputs as follows to obtain the message *M*. In the case of the key ladder:

$$M = r \,||\, \text{CW-URI} \,||\, \tau_b \,||\, \text{AD1} \,||\, \text{SPK-URI} \,||\, \text{I2BSP}(\text{SPK}_1) \,||\, \text{I2BSP}(\text{SPK}_2) \,||\, \ldots \,||\, \text{I2BSP}(\text{SPK}_m),$$

and in the case of the authentication mechanism:

$$M = r \,||\, \text{AD1} \,||\, \text{I2BSP}(\text{SPK}_1) \,||\, \text{I2BSP}(\text{SPK}_2) \,||\, \ldots \,||\, \text{I2BSP}(\text{SPK}_m).$$

Recall that the lengths of *r*, CW-URI, τ_b, AD1, SPK-URI and $\text{I2BSP}(\text{SPK}_i)$ are 128 bits, 64 bits, 8 bits, 256 bits, 64 bits and 2 048 bits, respectively. As a result, the bit length of *M*, denoted by *l* in [NIST FIPS 180-4], equals $520 + 2\,048m$, where *m* is the number of $\text{I2BSP}(\text{SPK}_m)$ elements in

message block *M*, in the case of the key ladder and 384 + 2 048*m* in the case of the authentication mechanism.

Next, h shall compute the secure hash algorithm-256(*M*) [SHA-256(*M*)] as specified in [NIST FIPS 180-4].

In the case of the authentication mechanism, h shall truncate this 256-bit message digest to 128 bits, and it shall output this truncated message digest. In case of the key ladder, the key ladder block shall pass the 256-bit output of SHA-256 to the **content descrambler**, and if the length of CW is *N* bits, then the **content descrambler** shall truncate this output to *N* bits. In both cases, the truncation method as specified in [NIST SP 800-107] shall be used.

10.5 Message authentication code algorithm

This clause specifies the MAC algorithm introduced in clause 7.5. This **message authentication code algorithm** is specified as in [ISO/IEC 9797-1] with the following selections.

- The block cipher shall be AES-128.
- Padding of the message shall be done with padding method 3. This method needs the bit length of the unpadded message $e(LK_t, r) \| A$. This bit length equals Len + 129.
- MAC algorithm 1 shall be used to compute the MAC from the message and the secret key.
- The length of the MAC shall be τ bits.

Appendix I

Areas for further development

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

It has been identified that this Recommendation needs further development and validation for it to meet the requirements set out in [b- ITU-T J.1010], and that [b-ITU-T J.1010] needs to be updated to reflect the requirements of the MovieLabs Enhanced Content Protection (ECP) specification [b-ECP]. Recommendations [b-ITU-T J.1011], [b-ITU-T J.1012], [b-ITU-T J.1013], [b-ITU-T J.1014], ITU-T J.1015 and [b-ITU-T J.1015.1] should in the future be updated to reflect those updates to [b-ITU-T J.1010].

A number of ITU Member States, as well as stakeholders from a variety of industries – including manufacturers of devices and electronic components, owners and licensees of copyrighted content, providers of over-the-top (OTT) and linear television services, and providers of conditional access system (CAS) and digital rights management (DRM) solutions – based all around the world have expressed concern that the Embedded Common Interface (ECI) does not fully meet the requirements of ECP, nor wider industry content protection requirements.

More specifically, their concerns were raised in contributions to the ITU-T Study Group 9 (SG9) meeting (16-23 April 2020). Contributions from Israel, Australia, ITU-T Sector Member Samsung, and SG9 Associates Sky Group and MovieLabs proposed that a number of changes be included in the ECI Recommendations, but agreement on them was not reached. These items are inventoried in [b-SG9 Report 17 Ann.1].

They include proposals to:

1) Simplify the ECI system by reducing its scope;
2) Remove DRM;
3) Remove the re-encryption of content;
4) Remove software management;
5) Add APIs for secure storage and cryptographic operations;
6) Allow vendor-specific key ladders;
7) Use J.1207 TEE requirements;
8) Include TEE implementation for VM;
9) Upgrade the strength of the cryptographic algorithms, e.g., using SHA-384;
10) Use standard certificates, like ITU-T X.509;
11) Reconsider communications between clients;
12) Perform additional liaisons with ETSI;
13) Perform additional peer-review;
14) Explore alternatives to the Trust Authority model;
15) Define further the technical aspects of ECI compliance and robustness rules;
16) Add requirements for diversity, e.g., address space randomization;
17) Add requirements on runtime integrity checking.

These proposals reflect that content protection and the threats of its compromise are continuously evolving. ECI was originally conceived nearly a decade before approval of this ITU-T Recommendation. Systems like ECI need to be assessed on a regular basis against the current state-of-the-art in both attack techniques and industry protection requirements.

Other mechanisms exist to enable interoperability. In particular for the DRM use case, most internet video services have deployed other solutions to provide interoperability and to address their needs.

Further clarity is important as many Member States regard ITU standards as influential sources of guidance for the development of their markets and industries. The list of concerns ensures ECI's implementation in their domestic markets can involve a full appreciation of implications of this ITU-T Recommendation and ensure that the issues are considered when legislation, regulation or market need requiring consumer digital television equipment to be interoperable are being considered. It also ensures that technology equipment manufacturers, who may prefer to use a unique set of requirements or other standards to design the products, can consider these issues in developing products for different markets.

Bibliography

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

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

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

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

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

[b-ITU-T J.1015.1] Recommendation ITU-T J.1015.1 (2020), *Embedded common interface for exchangeable CA/DRM solutions; The advanced security system - Key ladder block: Authentication of control word-usage rules information and associated data 1*.

[b-SG9 Report 17 Ann.1] ITU-T SG9 meeting report, SG9-R17-Annex 1 (2020), Annex 1 to Report 17 of the SG9 fully virtual meeting held 16-23 April 2020. https://www.itu.int/md/T17-SG09-R-0017/en

[b-ISO/IEC 23001-7] ISO/IEC 23001-7:2016: *Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files*.

[b-ATSC A/70-1] ATSC Standard A/70 Part 1:2010, *Conditional access system for terrestrial broadcast*. https://www.atsc.org/standard/a70-part-12010-conditional-access-system-for-terrestrial-broadcast/

[b-ETSI GS ECI 001-5-2] ETSI GS ECI 001-5-2 V1.1.1 (2017-07), *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 5: The Advanced Security System; Sub-part 2: Key Ladder Block*. https://www.etsi.org/deliver/etsi_gs/ECI/001_099/0010502/01.01.01_60/gs_ECI0010502v010101p.pdf

[b-ETSI TS 100 289] ETSI TS 100 289 V1.1.1 (2011), *Digital video broadcasting (DVB); Support for use of the DVB scrambling algorithm version 3 within digital broadcasting systems*: https://www.etsi.org/deliver/etsi_ts/100200_100299/100289/01.01.01_60/ts_100289v010101p.pdf

[b-ETSI TS 103 127] ETSI TS 103 127 V1.1.1 (2013), *Digital video broadcasting (DVB); Content scrambling algorithms for DVB-IPTV services using MPEG2 transport streams*: https://www.etsi.org/deliver/etsi_ts/103100_103199/103127/01.01.01_60/ts_103127v010101p.pdf

[b-GY/T 277-2014] ChinaDRM Lab, GY/T 277-2014: 互联网电视数字版权管理技术规范 [Technical specification for digital rights management for internet television].

[b-IEEE-OUI] IEEE Standards Association (2017), *Guidelines for use of extended unique identifier (EUI), organizationally unique identifier (OUI) and company ID (CID)*.
https://standards.ieee.org/develop/regauth/tut/eui.pdf

[b-ROEL] Roelse, P. (2014). *A new key establishment protocol and its application in pay-TV systems*.
https://arxiv.org/pdf/1308.4371.pdf

[b-ECP] MovieLabs Specification for Enhanced Content Protection – Version 1.2
https://movielabs.com/ngvideo/MovieLabs_ECP_Spec_v1.2.pdf

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