

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1014

(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; Advanced Security –
ECI-specific functionalities**

Recommendation ITU-T J.1014

Recommendation ITU-T J.1014

Embedded common interface for exchangeable CA/DRM solutions; Advanced Security – ECI-specific functionalities

Summary

Recommendation ITU-T J.1014 is one part of a multi-part deliverable covering the ECI-specific functionalities of an **advanced security system** for the **embedded common interface (ECI)** 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 001-5-1] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI. Modifications have been introduced to clauses 1, 3.2, 6.1, 6.2, 6.3 and 8.2.3 as well as editorial corrections and the substitution of the term "content processing system" by "Secure Video Path".

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T J.1014	2020-04-23	9	11.1002/1000/13575

Keywords

CA, 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/>.

© ITU 2020

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

Table of Contents

		Page
1	Scope.....	1
2	References.....	1
3	Definitions	2
	3.1 Terms defined elsewhere	2
	3.2 Terms defined in this Recommendation.....	2
4	Abbreviations and acronyms	4
5	Conventions	5
6	Principles	6
	6.1 Overview	6
	6.2 System Robustness model	7
	6.3 Secure Video Path and Output Protection System Control	9
	6.4 Specification principles	10
7	Key Ladder Application and associated functions	11
	7.1 General	11
	7.2 AS System and client data authentication	11
	7.3 Asymmetrical Micro Server mode	11
	7.4 Interface to Secure Video Path	13
	7.5 AS Key Ladder Block input output definition	13
	7.6 ACF definition.....	15
8	Advanced Security Slot	16
	8.1 Advanced Security Slot introduction.....	16
	8.2 AS Slot Definition	16
9	Scrambling/descrambling and Content Export.....	41
	9.1 Basic functionality.....	41
	9.2 Scrambler and descrambler specifications	41
	9.3 Export control	42
	9.4 Output control.....	42
	9.5 Content Property comparison on coupled sessions	42
	9.6 Content Property propagation on Export.....	42
	9.7 Basic URI enforcement on Export	42
	9.8 Content Property application on industry standard outputs	43
	9.9 Control word synchronization	43
10	Certificate Processing Subsystem.....	44
	10.1 Basic processing rules for Certificate Chains.....	44
	10.2 Specific rules for Host Image Chains	45
	10.3 Specific rules for Client Image chains.....	46
	10.4 Specific rules for Platform Operation Certificates	46

	Page
10.5 Specific rules for export/import chains	46
10.6 CPS ECI Root Key initialization.....	48
11 Loader core	48
11.1 Introduction	48
11.2 Host loader rules.....	48
11.3 Client Loader rules	49
11.4 Revocation enforcement.....	49
11.5 Client image decryption	49
12 Timing requirements.....	50
12.1 Introduction	50
12.2 Administrative functions	50
12.3 Symmetrical cryptography functions	50
12.4 Asymmetrical cryptography functions	50
Annex A – Cryptography function definitions	51
A.1 Hash function.....	51
A.2 Asymmetrical cryptography	51
A.3 Random number generation	51
Appendix I – Sample Micro DRM System application	52
I.1 Introduction	52
I.2 Application scenario.....	52
I.3 Assumptions and notation	53
I.4 Micro Server pseudo code	54
I.5 Micro Client pseudo code.....	57
I.6 Micro DRM System cascading effect on ECM pre-delay.....	58
I.7 Content property change timing interface convention	58
Appendix II – Areas for further development.....	60
Bibliography.....	62

Introduction

This ITU-T Recommendation¹ is a transposition of the ETSI standard [b-ETSI GS ECI 001-5-1] and is a result of a collaboration between ITU-T SG9 and ETSI ISG ECI. Modifications have been introduced to clauses 1, 3.2, 6.1, 6.2, 6.3 and 8.2.3 as well as editorial corrections and the substitution of the term "content processing system" by "**Secure Video Path**"².

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.

Service and content protection realized by conditional access (CA) and digital rights management (DRM) are essential in the rapidly developing area of digital broadcast and broadband, including content, services, networks and customer premises equipment (CPE), to protect business models of content owners, network operators and PayTV operators. It is also essential for consumers that they are able to continue using the CPEs they bought for example after a move or a change of network provider or even utilize devices for services of different commercial video portals. This can be achieved by the implementation of interoperable CA and DRM mechanisms inside CPEs, based on an appropriate security architecture.

As part of a security architecture the present Recommendation defines a security processing system for the authentication and verification of protected media content and of software images to be processed inside an ECI-compliant CPE. The core of the security architecture is built by a **Key Ladder Block** that supports secure processing with secret keys, targeting of keys to specific chips and authentication of the origin of key material.

Clause 6 presents an overview of the system architecture, defines **Robustness** rules to fight attacks and describes the relation between the elements of the security architecture, **ECI Host** and **ECI Clients**.

Clause 7 describes the applications the **Key Ladder Block** can be used for, together with the associated functions.

For proper operations, the security processing system needs information about the state of each loaded **ECI Client**. This state information, as some of it needs to be secret, is handled with the help of an advanced security slot. The **ECI Host** assigns to each **ECI Client** such a slot that needs to be protected against malicious modifications. The definition of a slot and its configuration for several operations like decrypting or exporting content is described in clause 8.

In an **ECI-compliant** CPE content can be decrypted, it can be forwarded to standard outputs if permitted and it can be re-encrypted for export. The usage of an advanced security slot for these operations is specified in clause 9.

A **Certificate Processing Subsystem** that is realized as a special function of an advanced security slot is responsible for the authentication of items. Clause 10 specifies the rules that are applied for authentication.

The **ECI** system uses a loader mechanism that permits **ECI Clients** to securely verify the version of the **ECI Host** and **ECI Client** credentials that are loaded so as to detect any known security issue. The loader mechanism relies on **Robustness** principles that are described in clause 11.

Clause 12 contains timing constraints for the operations described in the present Recommendation.

¹ Several areas for further development have been identified in Appendix II.

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.1014

Embedded common interface for exchangeable CA/DRM solutions; Advanced Security – ECI-specific functionalities

1 Scope

The present Recommendation defines a robust security processing subsystem for **ECI** called the **Advanced Security System**. The **Advanced Security System** provides a secure basis for software elements to be authenticated and loaded, performs security computations and verifications, manages the encryption and decryption of content and the exchange of content with associated rights and obligations. The **Advanced Security System** applies the **ECI Key Ladder Block** [ITU-T J.1015]. Refer also to [b-ETSI GS ECI 001-5-2], to perform secure calculations.

2 References

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

- [ITU-T 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.1015] Recommendation ITU-T J.1015 (2020), *Embedded common interface for exchangeable CA/DRM solutions; The advanced security system – Key ladder block*.
- [ETSI ETR 289] ETSI ETR 289 (CSA1/2):1996, *Digital Video Broadcasting (DVB); Support for use of scrambling and Conditional Access (CA) within digital broadcasting systems*.
www.etsi.org/deliver/etsi_etr/200_299/289/01_60/etr_289e01p.pdf
- [ETSI TS 100 289] ETSI TS 100 289 (V1.2.1) (CSA3):2014 *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_100299/100289/01.02.01_60/ts_100289v010201p.pdf
- [ETSI TS 103 127] ETSI TS 103 127 (V1.1.1) (CISSA):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

- [ISO/IEC 9899] ISO/IEC 9899:2011, *Information technology – Programming languages – C*.
<https://www.iso.org/standard/57853.html>
- [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*.
<https://www.iso.org/standard/68042.html>
- [ISO/IEC 23009-4] ISO/IEC 23009-4:2013, *Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 4: Segment encryption and authentication*.
<https://www.iso.org/standard/62122.html>
- [NIST FIPS 180-4] NIST FIPS PUB 180-4:2015, *Secure Hash Standard (SHS)*.
https://ws680.nist.gov/publication/get_pdf.cfm?pub_id=910977
- [NIST 800-90Ar1] NIST Special Publication 800-90A Rev.1:2015, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*.
<https://csrc.nist.gov/publications/detail/sp/800-90a/rev-1/final>

3 Definitions

3.1 Terms defined elsewhere

None.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 Advanced Security System (AS System): A function of an **ECI** compliant **CPE**, which provides enhanced security functions (hardware and software) for an **ECI Client**.

3.2.2 AS Slot: Resources of the advanced security block provided exclusively to an **ECI Client** by the **ECI Host**.

3.2.3 AS-API: Application programming interface between the **ECI Client** and its **ECI Host** permitting the **ECI Client** to exchange information with and perform operations on its **AS Slot**.

3.2.4 Authentication Mechanism: Key Ladder Block function as defined in [ITU-T J.1015] that permits an **AS Slot** to provide secure key applications for purposes other than content decryption and encryption, like authentication.

3.2.5 Brother: Other **Child** of the same **Father**.

NOTE – **Father, Children, Brother** refer to **Entities** that manage **Certificates**.

3.2.6 Certificate: Data structure as defined in clause 5 of [ITU-T J.1012], refer also to [b-ETSI GS ECI 001-3], with a complementary secure digital signature that identifies an **Entity**.

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.

3.2.7 Certificate Chain: A list of **Certificates** that authenticate each other up to and including a **Root Revocation List**.

NOTE – Typically, in **ECI Certificates** are accompanied by a **Revocation List** that excludes **Certificates** that cannot be validated.

3.2.8 Certificate Processing Subsystem (CPS): Subsystem of the **ECI Host** that provides **Certificate** verification processing and providing additional **Robustness** against tampering.

3.2.9 Child, Children: Entity (Entities) referred to by a **Certificate** signed by a (common) **Father**.

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**.

3.2.10 Content Properties (CP): Properties of the content that provide information on rights and obligations associated with subsequent applications or transformations of the content, such as usage rights information, selective output control and parental control information.

3.2.11 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 and thus provides interoperability of CPE devices with respect to **ECI**.

3.2.12 ECI Client (Embedded CI Client): Implementation of a CA/DRM client that is compliant with the **ECI** specifications.

3.2.13 ECI Client Loader: Functionality of the **ECI Host** that uses the **AS System** to exclusively provide the function, verify and install a new **ECI Client** software image in an **ECI** container of the **ECI Host**.

3.2.14 ECI Ecosystem: Commercial operation consisting of a **TA** and several platforms and **ECI** compliant **CPEs** in the field.

3.2.15 ECI Host: Hardware and software system of a CPE, which covers **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 Loader: CPE bootloading functionality that uses the **AS System** to exclusively provide the function to verify and install **ECI Host** software into a 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**.

3.2.17 ECI Root Key: Public key providing the origin of authentication for **ECI** certified entities and **Certificates**.

3.2.18 Entity: Organization (e.g., manufacturer, **Operator** or **Security Vendor**) or real world item (e.g., **ECI Host**, **Platform Operation** or **ECI Client**) identified by a unique ID in an **ECI Ecosystem**.

3.2.19 Export Connection: Authenticated relation between an **AS Slot** decrypting content and an **AS Slot** subsequently re-encrypting the decrypted content indicating such re-encryption is permitted.

3.2.20 Father: Signatory of the **Certificate** of the **Child Entity**

NOTE – **Father**, **Children**, **Brother** refer to **entities** that manage **Certificates**.

3.2.21 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 CPE.

3.2.22 Key Ladder Block: Robust secure mechanism to compute decryption, encryption and authentication keys as defined in [ITU-T J.1015], both **Key Ladder** and **Authentication Mechanism**.

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

3.2.24 Micro DRM System: Content protection system that re-encrypts content on a CPE with a **Micro Server** and that permits decoding of that re-encrypted content by authenticated **Micro Clients**.

NOTE – **Micro Server** and **Micro Clients** being provisioned by a **Micro DRM System Operator**.

3.2.25 Micro Server: **ECI Client** that can import decrypted content, re-encrypt this content and authenticate a specific **ECI Client** or group of **ECI Clients** as the **Target** for subsequent decryption.

3.2.26 Operator: Organization that provides **Platform Operations** that is enlisted with the **ECI TA** for signing the **ECI Ecosystem**.

NOTE – An **Operator** may operate multiple **Platform Operations**.

3.2.27 Platform Operation (PO): Specific instance of a technical service delivery operation having a single **ECI** identity with respect to security.

3.2.28 Provisioning Server: Server, typically located in a secure back office location, that provisions keys and other secure information to facilitate an encryption or decryption function through an **AS Slot**.

3.2.29 Revocation: Status of exclusion of an **Entity** in accordance with its enumeration in a **Revocation List**.

3.2.30 Revocation List (RL): List of **Certificates** that have been revoked and therefore should no longer be used.

3.2.31 Robustness: Property of the implementation of a specified **ECI** secure function representing the effort and/or cost involved to compromise the security of the implemented secure function.

3.2.32 Root Certificate: Trusted certificate that is the origin of authentication for a chain of certificates in an **ECI Ecosystem**.

3.2.33 Secure Video Path: All CPE functions performing processing on content (and temporary storage required thereto) from and including Content Decryption through and including content re-encryption by means of a Micro Client or Output Protection System.

3.2.34 Security Vendor: Company providing **ECI** security systems including **ECI Clients** for **Operators of Platform Operations**.

3.2.35 Target: **Micro Client** or a group of **Micro Clients** for which content is re-encrypted by a **Micro Server**.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACF	Advanced Security Control Field
AD	Associated Data
AES	Advanced Encryption Standard
AK	Authentication Key
API	Application Programming Interface
ARK	Advanced Security Random Key
AS	Advanced Security
CA	Conditional Access
CBC	Cypher Block Chaining
CENC	Common Encryption
CISSA	Common IPTV Software-oriented Scrambling Algorithm
CP	Content Properties
CPE	Customer Premises Equipment
CPS	Certificate Processing Subsystem
CSA	Common Scrambling Algorithm
CTR	Counter Mode
CW	Control Word

DRM	Digital Rights Management
EAC	Export Authorization Certificate
EAOC	Export Authorization Operator Certificate
ECI	Embedded Common Interface
ECM	Entitlement Control Message
ECP	Enhanced Content Protection
EGC	Export Group Certificate
ERC	Export Revocation Certificate
ESC	Export System Certificate
LK	Link Key
MPEG	Moving Picture Experts Group
MSCSK	Micro Server Chipset Secret Key
PES	Packetized Elementary Stream
PO	Platform Operation
POC	Platform Operation Certificate
POPK	Platform Operation Public Key
REAOC	Revocation Export Authentication Operator Certificate
REE	Rich Execution Environment
RFU	Reserved for Future Use
RK	Random Key
RL	Revocation List
SPK	Sender Public Key
TA	Trust Authority
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TPEGC	Third Party Export Group Certificate
TS	MPEG 2 Transport Stream
URI	Usage Rights Information
XT	eXTension field

5 Conventions

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

6.1 Overview

The present Recommendation belongs to a series of ITU-T Recommendations, based on the **ECI** architecture [ITU-T J.1011], refer also to [b-ETSI GS ECI 001-1] and **ECI** basic requirements [ITU-T J.1010], refer also to [b-ETSI GS ECI 001-2].

Figure 6-1 presents the main principles of the **Advanced Security System**. The core of the **Advanced Security System** is formed by the **Key Ladder Block** as defined in [ITU-T J.1015], allowing secure processing with secret keys, targeting of keys to specific chips and authentication of the origin of key material.

The basis for loading images is embodied in the loader core. It uses the **Certificate Processing Subsystem** to verify the **ECI** status of **ECI Host** images, **ECI Client** images and **Platform Operation** (PO) credentials using a recent **ECI Root Key** and **ECI root Revocation List**. The version numbers of the **ECI Root Key** and **ECI root Revocation List** used by the **ECI Host** and other **ECI Clients** can be checked by **ECI Clients** that are loaded. These can refuse to descramble content on detecting unacceptable versions in accordance with the **ECI Revocation** enforcement principle. Encrypted **ECI Client** images are decrypted upon loading.

Each **ECI Client** uses an advanced security slot. The **AS Slot** is identified by the **Platform Operation** Public Key of the **ECI Client**. The **ECI Host** ensures that **ECI Client** interactions through the **AS-API** are directed to the **AS Slot** allocated to that **ECI Client**. Each **AS Slot** is described by a slot state and a session state per encryption/decryption operation. The **AS Slot** can be used either for decryption purposes or for encryption purposes. The **AS Slot** session state also includes a configuration (config) defining the details of the operation and how the session state should be authenticated. The **ECI Client** provides the configuration information and input for other state information. The **Key Ladder Block** is used to authenticate sender public key (SPK), **Platform Operation** public key (POPK) and the configuration information. The **AS Slot** can supply random numbers to selected **Key Ladder Block** inputs so as to generate random keys or to use as a nonce to ensure freshly computed **Key Ladder Block** inputs. This mechanism can be used to prevent replay of encrypted content and to ensure always online provisioning of an **AS Slot** by a **Provisioning Server**.

When decrypting content, the **Content Properties** can be authenticated along with computing the control words, thus creating a strong link with the decrypted content. **Content Properties** are forwarded with the content to any standard output to ensure the proper setting of protection mechanisms for such an output. These properties are compared to those with which the content is re-encrypted on an **Export Connection**. An **Export Connection** is permitted only through the appropriate export/import **Certificate Chains**. These are verified by the **Certificate Processing Subsystem** on behalf of the **AS Slot**. Standard outputs can be disabled through the output control mechanism.

Computed control words can be synchronized to MPEG transport stream formatted content using the alternating bit protocol. For this purpose, the **Secure Video Path** uses a double buffering mechanism with a current/next control word for stream processing.

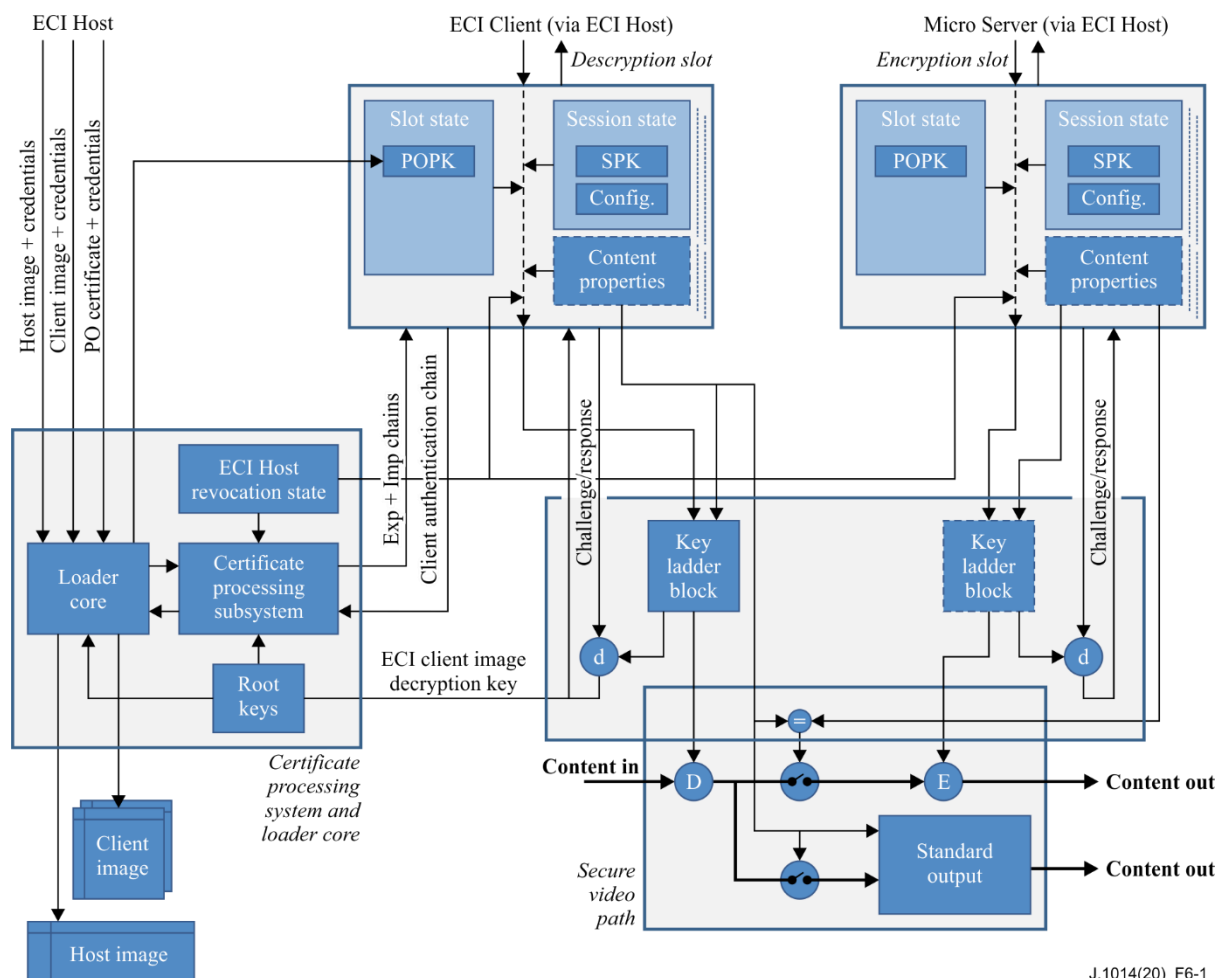


Figure 6-1 – Block diagram of Advanced Security System

6.2 System Robustness model

The **AS System** requires a robust implementation. **Robustness** is typically measured in terms of the effort and/or cost required to circumvent any security measures: i.e., observe secret values or manipulate state or values in a secure system.

The present Recommendation does not define a specific **Robustness** regime for various **ECI** functions. Nevertheless, the **ECI Robustness** architecture is based on the premise that some functions are more robust than others. This is illustrated in Figure 6-2.

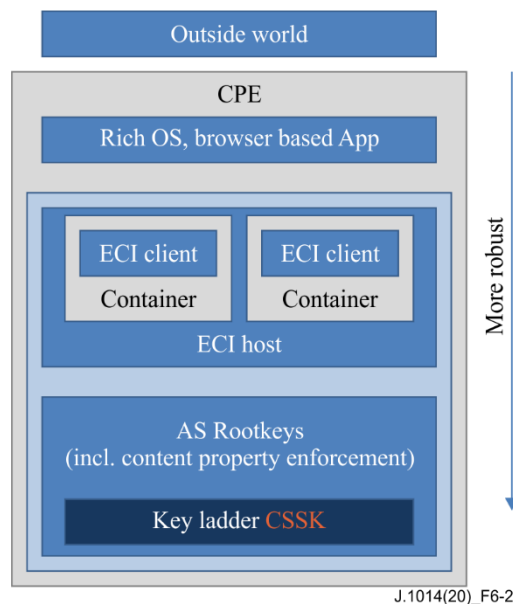


Figure 6-2 – System Robustness premise for ECI

The least robust environment is the outside world where any threat may exist. Data passing through that environment should be protected against manipulation and unauthorized inspection using authentication and encryption techniques. The rich operating system (typically including a browser) can be somewhat robust against manipulation and invasion, but is typically not able to withstand user-assisted or aggressive external hacking attacks. The **ECI Clients** and **ECI Host** security sensitive functions operate in an environment that is well protected from such attacks. In case the **ECI Host** is compromised **ECI Clients** are also compromised. In addition to resilience towards outside attacks the **ECI Clients** are sandboxed using the **ECI** virtual machine [ITU-T J.1013], refer also to [b-ETSI GS ECI 001-4]: i.e., they can neither access information in the **ECI Host** nor can they access any other **ECI Client**, apart from through defined **ECI** API interfaces. The **ECI Host** also ensures the **ECI Clients** have access to the **Advanced Security System** and the **Key Ladder Block**. At the core of the **Key Ladder Block** is the **Chip Set Secret Key** with allows each **ECI CPE** to be addressed uniquely. Typically, the **Key Ladder Block** and major parts of the **Advanced Security System** are implemented in hardware and/or highly robust firmware.

While in practice, robustness of security components falls on a continuum, architecturally they can be divided into a hierarchy. In the context of ECI, these can be described as follows:

0. The outside world (outside of the device) has robustness level 0. Therefore, the **ECI Client** needs a secure protocol to communicate with the headend. This is outside the scope of the **ECI** Recommendation.
1. There are the OS, drivers, applications and the browser of the device which are all at robustness level 1. The environment in which this code runs is often referred to as a Rich Execution Environment (REE) and consists of most application and operating system software and drivers. Because of the large size and functionality of this software, it often needs to be patched frequently to remove vulnerabilities. The **ECI Host** and **ECI Client** may use services provided by the REE, such as TLS networking, but cannot rely on it for any security, whether encryption or endpoint authentication.
2. There is the execution environment of functions of **ECI Host**, the VM and the **ECI Client**, which all have to run in a secured execution environment, often on the same processor. At robustness level 2, these environments only execute authenticated code, have hardware enforced memory and device isolation from the REE, including its kernel and drivers. These are sometimes referred to as Trusted Execution Environments (TEE). This level could also be implemented on a dedicated processor or a security processor with memory isolation.

3. The **Secure Video Path** requires a level of security above that of the **ECI Host** and **ECI Client**, robustness level 3. It can be implemented in a combination of secure hardware and secure firmware.
4. The most secure and robust are the **Key Ladder**, Boot Loader and Certificate processing and revocation subsystems, operating at robustness level 4. These are ideally integrated in secure hardware but due to the nature of the **ECI Recommendation**, portions may require firmware running on a dedicated security processor. Specific measures to preserve the secrecy of advanced security and key ladder secrets and computations therewith are required.

6.3 Secure Video Path and Output Protection System Control

Once content is decrypted, the **Secure Video Path** protects it from illegitimate use. The **Secure Video Path** may perform various video (and audio) operations on unencrypted content: demultiplexing, decoding, scaling, re-encoding, watermark insertion, etc. All operations and their constituent parts (including transitory memories involved and busses used to transfer content between different subsystems of the CPE) and control interfaces (content properties and possible proprietary or implementation specific interfaces) controlling content property critical processing steps and output protection settings of the content are deemed to be part of **Secure Video Path**.

The content properties [ITU-T J.1012] define which operations are permitted and which outputs, output protection systems and settings thereof are adequate to protect the content. The **Advanced Security System** provides a cryptographic mechanism called Content Protection Authentication (see clause 8.2.3) to implicitly authenticate content properties using the key ladder block when computing the control word. The content protection system can select the setup of the Content Property Authentication. **Content Properties** not implicitly authenticated by the Content Property Authentication function have the robustness level of the **ECI Client** and **ECI Host**.

It should be noted that complementary specifications are required to provide adequate detail on the implementation robustness of the **Secure Video Path** and the output protection systems.

Figure 6-3 provides details of the logic for securing the delivery of the **Content Properties**. First the **ECI Client** sets the session **Content Properties** for the next to be decoded section of content. Subsequently the **ECI Client** provides the inputs to compute the control word including Field1 (as part of the **elk** input vector: see clauses 7.1, 8.2.3, 8.2.4.7). Content property fields of Field1 are selected by the first two bytes of Field1 in the Content Property Authentication function and are subsequently implicitly authenticated by using this value as a symmetrical key value input to the key ladder calculation of CW. An improper symmetrical key value input will lead to a wrong CW, thus the content cannot be decrypted. Field1 values are then compared by the SVP to the corresponding content properties set by the ECI Client. In case of any discrepancy no output is possible. The output of a specific protection system may also be blocked by the corresponding output control field value.

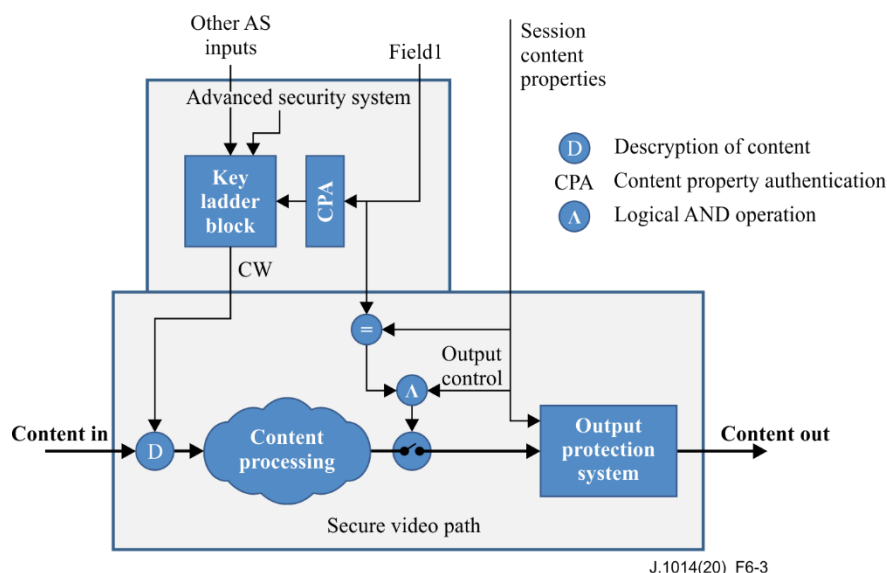


Figure 6-3 – Secure Video Path and Content Property Authentication

6.4 Specification principles

6.4.1 Implementation Freedom

The present Recommendation defines states and functions that operate on the **AS System** which results in a new state. The specific representation of the state of an implementation is not defined by the present Recommendation and may be completely defined by the implementation as long as the behaviour of the implementation can be reconstructed to states and state transition sequences using the transition functions as defined in the present Recommendation.

NOTE – In many cases the **Key Ladder** function as defined in [ITU-T J.1015] is a substantial part of the state transition function.

For example, an AS implementation may have a fast certificate processing subsystem (CPS) that can re-authenticate POPK from the **Platform Operation Certificate Chain** for every application of the **Key Ladder Block**. The **AS Slot** in this case does not have to store POPK as an authenticated value in a tamper proof way. Similarly, some implementations may decide to compute LK_1 (the top level symmetrical key in the **Key Ladder**) once using two asymmetrical cryptography operations for an **AS Slot**, while others, that can perform the asymmetrical cryptography operations sufficiently fast, may recompute LK_1 from its original inputs for every **Key Ladder** application.

6.4.2 Specification style and relation to AS-API

There is no direct API between the **ECI Client** and the **Advanced Security System**. The **ECI Host** acts as a conduit. Nevertheless, the definitions of the operations on an **AS Slot** map directly to the messages of the **AS API** as defined in [ITU-T J.1012], with exception of the slotId parameter, which is not required for **ECI Client** AS API messages. The **ECI Host** provides the slotId parameter to the **AS system**.

The transactions by the **ECI Host** (on behalf of an **ECI Client**) on an **AS Slot** are defined as C-functions declarations. These describe an atomic transaction on the state of the **AS Slot**. This can result in a new slot state. The specific representation of function parameters is not of direct consequence to functionality specified in the present Recommendation, except when cryptography functions are concerned. However, the representation is significant for the definition of the **AS API** in [ITU-T J.1012].

7 Key Ladder Application and associated functions

7.1 General

The **Key Ladder** and **Authentication Mechanism** defined in [ITU-T J.1015] play a central role in all robustly implemented secret key computations in an **ECI CPE**. The **Advanced Security System** shall apply these functions as defined in [ITU-T J.1015] with inputs and outputs as defined in the present Recommendation. All inputs to the **Key Ladder Block** shall be controlled by the **AS System**; any observation or manipulation shall not be possible in accordance with applicable AS **Robustness** rules and the **Key Ladder Block** specification [ITU-T J.1015].

7.2 AS System and client data authentication

The **Advanced Security System** can be provided with data from an **ECI Client**. The **AS System** provides a means to verify the authenticity of this data using the AD input of the **Key Ladder Block**.

The **AS System** computes the AD input of the **Key Ladder Block** as the hash of additional data to be authenticated in conjunction with a CW or AK computation. Following the bit-string notation of [ITU-T J.1015] AD shall be computed as:

$$AD = hash(ACF \parallel lm \parallel ARK \parallel P_1 \parallel \dots \parallel P_m \parallel C_1 \parallel \dots \parallel C_m \parallel XT)$$

Function *hash* is defined in clause A.1. *lm* is an 8-bit input containing the binary representation of *m*. The value of *m* corresponds to the value of *m* in the **Key Ladder Block** definition. The length of each P_i is 2 048 bits for the purpose of carrying public keys (POPK values). The length of C_i is defined as equal to the length of the SessionConfig structure in clause 8.2.2.5, the length of *XT* is 256 and serves a general purpose extension mechanism. It shall be set to 0. *ARK* is a 128-bit number, intended to represent a random value or all 0's in case no random input is needed. *ACF* is a control value defining the mode of operation.

7.3 Asymmetrical Micro Server mode

The **AS System** applies the **Authentication Mechanism** for the purpose of loading a **Micro Server** sender secret key into an **AS Slot** for the purpose of asymmetrical authentication between **Micro Servers** and **Micro Clients**.

Figure 7-1 shows the basic principle of the overall computation of asymmetrical authentication mode.

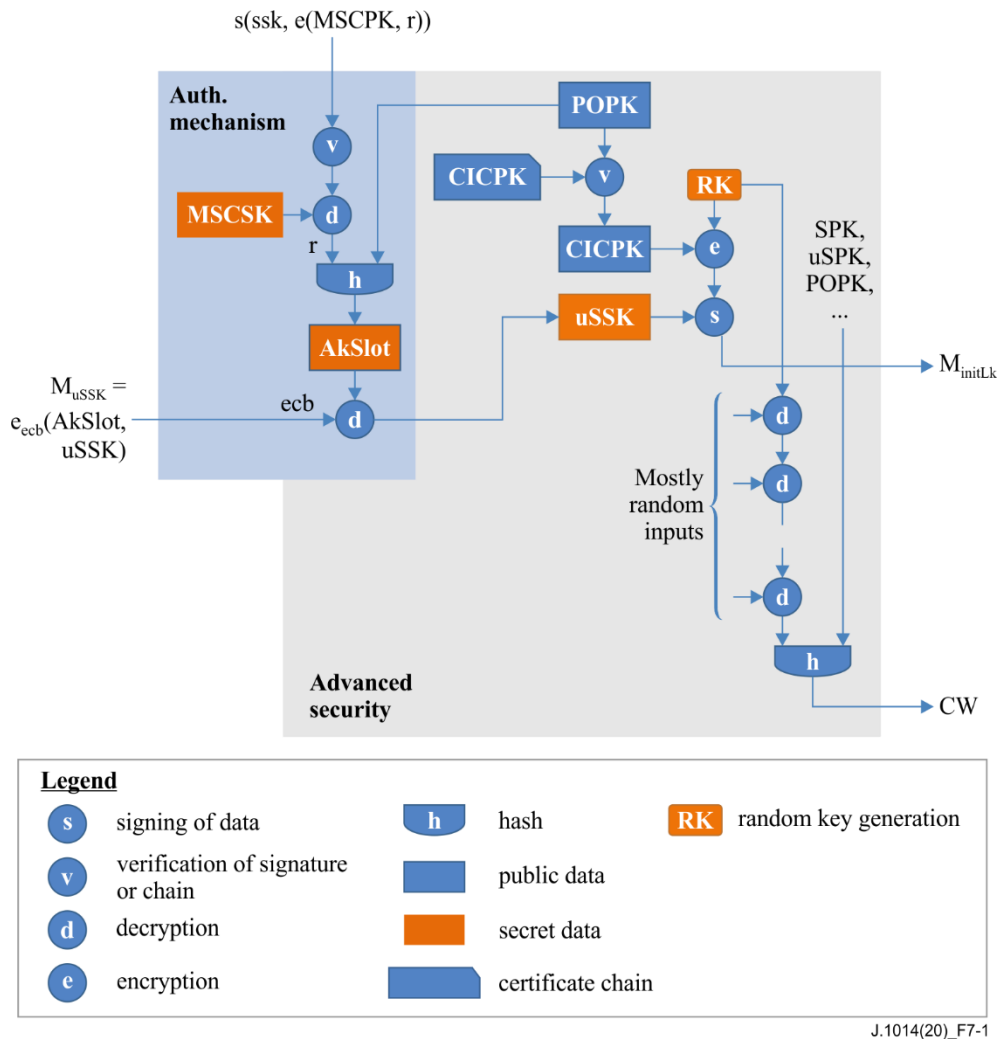


Figure 7-1 – Computation of the asymmetrical Micro Server mode

The **Authentication Mechanism** is used to compute the **AS Slot** authentication key **AkSlot**, using a.o. the ChipSet Key here called the **Micro Server** chipset secret key **MSCSK**. **AkSlot** is used to load the **Micro Server** Secret Key **uSSK**. The **Certificate Processing Subsystem** is used to authenticate the Chipset Public Key **CICPK** of the **Target Micro Client** using **POPCK** as a root and a **Certificate Chain** containing **CICPK** in the last **Certificate** of the chain. A random key **RK** is generated and **CICPK** and **uSSK** are used to generate the **Micro Client Key Ladder** initialization message M_{initLk} . The random key is also used as the top level symmetrical key of a **Key Ladder** with the same structure and hash function as defined in clause 7.1 of [ITU-T J.1015]. The computed control word **CW** is used for encryption by the **Micro Server**. The regular **Key Ladder** can be used in a **Micro Client** to compute **CW** for the purpose of decryption.

Clause 8.2.4.10 defines the specifics of the **uSSK** computation (the **ldUssk** function).

The computation of M_{initLk} shall be as defined in the computation scheme below, using the specification conventions as presented in clause 7.1 of [ITU-T J.1015]:

- $Mkey = cl\text{-}chipset\text{-}ID \parallel E(CICPK, LK)$
- $M_{initLk} = (Mkey \parallel S(uSSK, Mkey))$
- with $\parallel$ the bitwise concatenation function, **cl-chipset-ID** the chipset-ID of the client CPE and with **E()** the asymmetrical encryption function and **S()** the asymmetrical signature function as defined in clause 7.2 of [ITU-T J.1015].

On loading uSSK the **AS Slot** shall generate a new RK in accordance with clause A.3.

The computation of CW from LK and its inputs shall be identical to the **Key Ladder** mechanism defined in clause 7.2 of [ITU-T J.1015] with the same inputs as defined there and the same output (CW, CW-URI) but replacing the computation of LK1 by the RK random session key generated by the **Micro Server AS Slot**.

7.4 Interface to Secure Video Path

The **Key Ladder** including the asymmetrical **Micro Server** mode extension can compute control words with complementary CW-URI. These shall be passed securely to a decryption or encryption resource in the content processing subsystem which may (temporarily) store the CW and CW-URI information in association with key synchronization information. For transport stream applications the key synchronization information consists of the current/next bit, thus specifying two storage locations for CW values. For file-based applications a single CW location is available for the encryption and decryption resource.

Other information accompanying control words from the **AS Slot** to the **Secure Video Path**:

- Application of CW as encryption or decryption control word.

NOTE – Together with CW-URI this is a permission to apply CW.

- Export authentication information.
- Content properties.
- Odd/even property of the control word for TS mode descrambling.

7.5 AS Key Ladder Block input output definition

This clause defines the mapping of C-style variables and structures as a representation for the inputs to the **Key Ladder** and **Authentication Mechanism** and the extensions thereof as defined in clause 7.2 and clause 7.3. The symbols for the input names are used in the rest of the present Recommendation to define the various applications.

The mapping of C-structures to an octet sequence is defined with the following (little endian based) rules:

- Bit-fields of structures are mapped first field first starting with the lowest bit (0) of the first octet.
- Structures of a length other than a multiple of 8-bit that are padded at the end to the next multiple of 8-bits with reserved bits that shall be set to zero to the next multiple of 8-bit.
- 16-bit, 32-bit and 64-bit entities are mapped in little endian order (least significant byte first).
- Arrays are mapped in increasing index order.

Octet sequences are mapped to bit strings using the OS2BSP function as defined in [ITU-T J.1015].

NOTE – The above rules ensure that the bit numbering order as used for integer values represented by c-variables is equal to that used in [ITU-T J.1015] for the corresponding inputs to the **Key Ladder Block**.

The naming convention for the variables is listed in Table 7-1.

Table 7-1 – C-variable naming convention for Key Ladder interface

Key Ladder Block input or output	Bits	C-variable naming convention
CW-URI	64	ulong cwUri ;
AD	256	uchar ad [32]; /* not used directly, see clause 7.2 */
SPK-URI	64	ulong spkUri ;
SPK _i , i=1..16	2 048 × 16	typedef uchar PubKey[256]; PubKey spk [16]; /* (spk[i-1] == SPK _i) */
m		uchar nSpk ;
input to V	64+ 2 048 + 2 048	typedef struct InputV{ ulong chipsetId; uchar elk1[256]; uchar signature[256]; } InputV; InputV inputV ;
E(LK _i ,LK _{i+1}), i=1..24; LK _{t+1} =r	256 × 24	typedef uchar SymKey[32]; Symkey elk [i]; /* (elk[i-1] == E(LK _i ,LK _{i+1})) */ /* C-input == E(LK _{t-1} ,LK _t), i.e. the one but last input */
t		uchar nElk ;
T_b		value set to 0
Chipset-ID	64	ulong chipsetId ;
Challenge	128	uchar challenge [16];
Response	128	uchar response [16];
<i>The inputs and outputs identified in the present Recommendation are defined below.</i>		
ACF	128-8	uchar acf [15]; /* operation mode */
ARK	128	uchar ark [16];
P _i	2 048 × 32	PubKey pk [32]; /* first m values are applied */
C _i	sizeof(SessionConfig) × 32	SessionConfig config [?]; /* SessionConfig is defined in clause 8.2.2.6 */
XT	256	uchar XT [32]; /* value always set to 0 */
M _{initLk}	64+ 2 048 + 2 048	InputV mInitLk ;

The following c-functions are defined using the above input variables to produce results.

SymKey **blockV_blockC_KeyLadder**(InputV inputV, SymKey **spk**)

Semantics:

This function computes the function of block V and block C in the **Key Ladder** to produce lk1.

In case of the asymmetrical server the following function computes the initialization message for the **Target Micro Client** as defined in clause 7.3:

InputV **asymInitLk1**(SymKey **lk1**, PrivKey **ussk**, PubKey **spk**);

Semantics:

This function computes the initialization message initLk1 in accordance to clause 7.3.

The remaining functions of the **Key Ladder** are performed by:

keyLadder(SymKey **lk1**, ulong **cwUri**, uchar **acf**[15], uchar **ark**[16],
 PubKey **popk**[16], SSCnfg **clCnf**[16], uchar **XT**[32], ulong **spkUri**, uint **nSpk**, PubKey **spk**[16],
 uchar **nElk**, SymKey **elk**[32])

Semantics:

This function computes the remainder of the **Key Ladder** using lk1 as the result of block D in the **Key Ladder** to produce a CW result for the **Secure Video Path**.