الاتحـــــــــــاد الدولـــــــــــي للاتصـــــــــــالات

ITU-T

قطاع تقييس الاتصالات
في الاتحاد الدولي للاتصالات

J.1014
(2020/04)

السلسلة J: الشبكات الكبلية وإرسال إشارات تلفزيونية وبرامج صوتية وإشارات أخرى متعددة الوسائط

النفاذ المشروط والحماية – الحلول المدمجة القابلة للمبادلة للنفاذ المشروط وإدارة الحقوق الرقمية

السطح البيني المشترك المدمج من أجل الحلول CA/DRM القابلة للمبادلة؛ الأمن المعزز – وظائف محددة بشأن السطح البيني المشترك المدمج

التوصية ITU-T J.1014

الاتحاد الدولي للاتصالات ITU

التوصية ITU-T J.1014

السطح البيني المشترك المدمج من أجل الحلول CA/DRM القابلة للمبادلة؛ الأمن المعزز – وظائف محددة بشأن السطح البيني المشترك المدمج

ملخص

التوصية ITU-T J.1014 هي جزء من مجموعة تضم عدة وثائق وتتناول خواص وظيفية محددة بشأن السطح البيني المشترك المدمج **لنظام الأمن المعزز** فيما يخص **السطح البيني المشترك المدمج (ECI)** من أجل توصيف حلول النفاذ المشروط/إدارة الحقوق الرقمية (CA/DRM) القابلة للمبادلة.

وهذه التوصية الصادرة عن قطاع تقييس الاتصالات هي نقل للمعيار [b-ETSI GS ECI 001-5-1] الصادر عن المعهد الأوروبي لمعايير الاتصالات (ETSI) ونتاج للتعاون بين لجنة الدراسات 9 لقطاع تقييس الاتصالات والفريق ETSI ISG ECI. وقد أدخلت تعديلات على الفقرات 1 و2.3 و1.6 و2.6 و3.6 و3.2.8 بالإضافة إلى التصحيحات الصياغية والاستعاضة عن مصطلح "نظام معالجة المحتوى" بمصطلح "مسار فيديوي آمن".

التسلسل التاريخي

الطبعة	التوصية	تاريخ الموافقة	لجنة الدراسات	معرف الهوية الفريد*
1.0	ITU-T J.1014	2020-04-23	9	11.1002/1000/13875

مصطلحات أساسية

النفاذ المشروط (CA)، إدارة الحقوق الرقمية (DRM)، المبادلة.

* للنفاذ إلى توصية، ترجى كتابة العنوان http://handle.itu.int/ في حقل العنوان في متصفح الويب لديكم، متبوعاً بمعرف التوصية الفريد. ومثال ذلك، http://handle.itu.int/11.1002/1000/11830-en.

تمهيد

الاتحاد الدولي للاتصالات وكالة متخصصة للأمم المتحدة في ميدان الاتصالات وتكنولوجيات المعلومات والاتصالات (ICT). وقطاع تقييس الاتصالات (ITU-T) هو هيئة دائمة في الاتحاد الدولي للاتصالات. وهو مسؤول عن دراسة المسائل التقنية والمسائل المتعلقة بالتشغيل والتعريفة، وإصدار التوصيات بشأنها بغرض تقييس الاتصالات على الصعيد العالمي.

وتحدد الجمعية العالمية لتقييس الاتصالات (WTSA) التي تجتمع مرة كل أربع سنوات المواضيع التي يجب أن تدرسها لجان الدراسات التابعة لقطاع تقييس الاتصالات وأن تُصدر توصيات بشأنها.

وتتم الموافقة على هذه التوصيات وفقاً للإجراء الموضح في القرار 1 الصادر عن الجمعية العالمية لتقييس الاتصالات.

وفي بعض مجالات تكنولوجيا المعلومات التي تقع ضمن اختصاص قطاع تقييس الاتصالات، تُعد المعايير اللازمة على أساس التعاون مع المنظمة الدولية للتوحيد القياسي (ISO) واللجنة الكهرتقنية الدولية (IEC).

ملاحظة

تستخدم كلمة "الإدارة" في هذه التوصية لتدل بصورة موجزة سواء على إدارة اتصالات أو على وكالة تشغيل معترف بها.

والتقيد بهذه التوصية اختياري. غير أنها قد تضم بعض الأحكام الإلزامية (بهدف تأمين قابلية التشغيل البيني والتطبيق مثلاً). ويعتبر التقيّد بهذه التوصية حاصلاً عندما يتم التقيّد بجميع هذه الأحكام الإلزامية. ويستخدم فعل "يلزم" وصيغ ملزمة أخرى مثل فعل "يجب" وصيغها النافية للتعبير عن متطلبات معينة، ولا يعني استعمال هذه الصيغ أن التقيّد بهذه التوصية إلزامي.

حقوق الملكية الفكرية

يسترعي الاتحاد الانتباه إلى أن تطبيق هذه التوصية أو تنفيذها قد يستلزم استعمال حق من حقوق الملكية الفكرية. ولا يتخذ الاتحاد أي موقف من القرائن المتعلقة بحقوق الملكية الفكرية أو صلاحيتها أو نطاق تطبيقها سواء طالب بها عضو من أعضاء الاتحاد أو طرف آخر لا تشمله عملية إعداد التوصيات.

وعند الموافقة على هذه التوصية، كان الاتحاد قد تلقى إخطاراً بملكية فكرية تحميها براءات الاختراع يمكن المطالبة بها لتنفيذ هذه التوصية. ومع ذلك، ونظراً إلى أن هذه المعلومات قد لا تكون هي الأحدث، يوصى المسؤولون عن تنفيذ هذه التوصية بالاطلاع على قاعدة البيانات الخاصة ببراءات الاختراع في مكتب تقييس الاتصالات (TSB) في الموقع http://www.itu.int/ITU-T/ipr/.

جدول المحتويات

الصفحة

مقدمة

هذه التوصية[1] الصادرة عن قطاع تقييس الاتصالات هي نقل للمعيار [b-ETSI GS ECI 001-5-1] الصادر عن المعهد الأوروبي لمعايير الاتصالات (ETSI) ونتاج للتعاون بين لجنة الدراسات 9 لقطاع تقييس الاتصالات والفريق ETSI ISG ECI. وقد أدخلت تعديلات على الفقرات 1 و2.3 و1.6 و2.6 و3.6 و3.2.8 بالإضافة إلى التصحيحات الصياغية والاستعاضة عن مصطلح "نظام معالجة المحتوى" بمصطلح "**مسار فيديوي آمن**"[2].

والهدف من هذه التوصية هو تيسير قابلية التشغيل البيني والمنافسة في خدمات الاتصالات الإلكترونية، وبشكل خاص، في سوق أجهزة الإذاعة والأجهزة السمعية المرئية. ولكن ثمة تكنولوجيات أخرى متاحة يمكن أن تكون مناسبة ومفيدة أيضاً حسب الظروف السائدة في الدول الأعضاء.

وتعتبر كلّ من حماية الخدمة والمحتوى المتحققة بالنفاذ المشروط (CA) وإدارة الحقوق الرقمية (DRM) ضرورية في مجال الإذاعة الرقمية والنطاق العريض الذي يتسم بالتطور السريع، بما في ذلك المحتوى والخدمات والشبكات ومعدّات منشآت العميل (CPE)، وذلك لحماية نماذج الأعمال لمالكي المحتوى ومشغّلي الشبكات ومشغلي التلفزيون غير المجاني (PayTV). ومن مصلحة المستهلك أن يكون قادراً على مواصلة استعمال أجهزة منشآت العميل التي اشتراها، بعد الانتقال أو تغيير مقدّم الخدمة أو حتى عند استعماله أجهزة لخدمات خاصة ببوابات فيديوية تجارية مختلفة، على سبيل المثال. ويمكن تحقيق ذلك بتنفيذ آليات للنفاذ المشروط (CA) وإدارة الحقوق الرقمية (DRM) قابلة للتشغيل البيني داخل معدّات منشآت العميل استناداً إلى معمارية أمنية مناسبة.

وكجزء من معمارية أمنية، تحدّد هذه التوصية نظام معالجة أمنية للاستيقان والتحقق من محتوى الوسائط المحمي ومن صور البرمجيات التي يتعين معالجتها داخل معدات منشآت العميل المطابقة للسطح البيني ECI. ويُبنى الجزء الأساسي من المعمارية الأمنية بواسطة **مجموعة سلّم المفاتيح** التي توفر معالجة آمنة عن طريق مفاتيح سرية، واستهداف مفاتيح لرقائق إلكترونية محددة والاستيقان من أصل مواد المفاتيح.

وتعرض الفقرة 6 لمحة عامة عن معمارية النظام، وتحدد قواعد **المتانة** لمكافحة الهجمات وتصف العلاقة بين عناصر المعمارية الأمنية و**مضيف السطح البيني ECI (ECI Host)** و**وسطاء السطح البيني ECI (ECI Clients)**.

وتصف الفقرة 7 التطبيقات التي يمكن أن تستعمل من أجلها **مجموعة سلّم المفاتيح**، إلى جانب الوظائف المرتبطة بها.

وللتشغيل الصحيح، يحتاج نظام المعالجة الأمنية إلى معلومات عن حالة كل **وسيط ECI** جرى تحميله. وتتم مداولة هذه المعلومات عن الحالة، التي يتعين أن يكون بعضها سرياً، بمساعدة فتحة الأمن المعزّز. فيخصص **مضيف ECI** لكل **وسيط ECI** هذه الفتحة التي يجب أن تكون محمية من التعديلات الضارة. ويرد في الفقرة 8 تعريف للفتحة وتشكيلتها بالنسبة لعمليات متعددة مثل فك التجفير أو تصدير المحتوى.

وفي المعدات CPE المطابقة **للسطح البيني ECI** يمكن فك تجفير المحتوى وإحالته إلى نواتج معيارية إذا كان ذلك مسموحاً كما يمكن إعادة تجفيره بهدف تصديره. ويرد في الفقرة 9 وصف لاستخدام فتحة الأمن المعزّز في هذه العمليات.

ويعتبر **النظام الفرعي لمعالجة الشهادة** الذي يتم تحقيقه على شكل وظيفة خاصة لفتحة الأمن المعزز مسؤولاً عن الاستيقان من البنود. وتحدد الفقرة 10 القواعد التي تطبق في الاستيقان.

ويستخدم النظام **ECI** آلية تحميل تسمح **لوسطاء ECI** بالتحقق بشكل آمن من نسخة أوراق اعتماد **المضيف ECI** و**الوسيط ECI** التي جرى تحميلها بهدف كشف أي مشكلة أمنية معروفة. وتعتمد آلية التحميل على مبادئ **المتانة** الواردة في الفقرة 11.

وتتضمن الفقرة 12 قيود التوقيت الخاصة بالعمليات الواردة في التوصية الحالية.

---

1 حُددت عدة مجالات لمزيد من التطوير في التذييل II.

2 يشير استخدام الخط الداكن في نص هذه التوصية إلى مصطلحات ذات تعاريف خاصة بسياق السطح البيني المشترك المدمج الذي يمكن أن يختلف عن الاستعمال الشائع.

السطح البيني المشترك المدمج من أجل الحلول CA/DRM القابلة للمبادلة؛ الأمن المعزز – وظائف محددة بشأن السطح البيني المشترك المدمج

1 مجال التطبيق

تحدد هذه التوصية نظاماً فرعياً متيناً للمعالجة الأمنية للسطح البيني **ECI** يعرف باسم **نظام الأمن المعزّز**. ويوفر **نظام الأمن المعزّز** أساساً مضموناً للاستيقان من عناصر البرمجيات وتحميلها، ويجري حسابات وعمليات تحقق أمنية، ويدير عمليات تجفير وفك تجفير المحتوى وتبادله مع الحقوق والالتزامات المرتبطة به. ويطبق **نظام الأمن المعزّز مجموعة سلم المفاتيح** التابعة **للسطح البيني ECI** [ITU-T J.1015]. راجع أيضاً [b-ETSI GS ECI 001-5-2]، لإجراء حسابات آمنة.

2 المراجع

يشتمل ما يلي من توصيات قطاع تقييس الاتصالات والمراجع الأخرى على أحكام تشكّل، من خلال الإشارة إليها في هذا النص، أحكاماً في هذه التوصية. وكانت الطبعات المشار إليها صالحة وقت نشر هذه التوصية. ولما كانت جميع التوصيات والمراجع الأخرى تخضع للمراجعة يرجى من جميع المستخدمين لهذه التوصية السعي إلى تطبيق أحدث طبعة للتوصيات والمراجع الواردة أدناه. وتنشر بانتظام قائمة توصيات قطاع تقييس الاتصالات سارية الصلاحية. والإشارة إلى أي وثيقة في هذه التوصية لا يضفي على الوثيقة، بحد ذاتها، صفة توصية.

[ITU-T J.1010] التوصية ITU-T J.1010 (2016)، *السطح البيني المشترك المدمج (ECI) من أجل الحلول CA/DRM القابلة للمبادلة؛ حالات ومتطلبات الاستعمال*.

[ITU-T J.1011] التوصية ITU-T J.1011 (2016)، *السطح البيني المشترك المدمج (ECI) من أجل الحلول CA/DRM القابلة للمبادلة؛ المعمارية والتعاريف ولمحة عامة*.

[ITU-T J.1012] التوصية ITU-T J.1012 (2020)، *السطح البيني المشترك المدمج (ECI) من أجل الحلول CA/DRM القابلة للمبادلة؛ الحاوية، وأداة التحميل، والسطوح البينية، والإبطال فيما يخص الحلول CA/DRM*.

[ITU-T J.1013] التوصية ITU-T J.1013 (2020)، *السطح البيني المشترك المدمج (ECI) من أجل الحلول CA/DRM القابلة للمبادلة؛ الآلة الافتراضية*.

[ITU-T J.1015] التوصية ITU-T J.1015 (2020)، *السطح البيني المشترك المدمج (ECI) من أجل الحلول CA/DRM القابلة للمبادلة؛ نظام الأمن المعزّز – مجموعة سلّم المفاتيح*.

[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 التعاريف

1.3 المصطلحات المعرّفة في وثائق أخرى

لا توجد.

2.3 المصطلحات المعرّفة في هذه التوصية

تعرف هذه التوصية المصطلحات التالية:

1.2.3 نظام الأمن المعزّز (AS System): وظيفة **لمعدّات منشآت العميل** المطابقة **للسطح البيني ECI**، توفر وظائف معززة للأمن (العتاد والبرمجيات) **لوسيط ECI**.

2.2.3 فتحة الأمن المعزّز (AS Slot): مصدر معلومات لمجموعة الأمن المعزّز يقدمها **المضيف ECI** حصراً إلى **وسيط ECI**.

3.2.3 السطح البيني لبرمجة التطبيقات في الأمن المعزّز (AS-API): سطح بيني لبرمجة التطبيقات بين **وسيط ECI** و**مضيف ECI** المتعلق به يسمح **للوسيط ECI** بتبادل المعلومات معه وتأدية العمليات على **فتحة الأمن المعزّز** الخاصة به.

4.2.3 آلية الاستيقان (Authentication Mechanism): إحدى وظائف **مجموعة سلم المفاتيح** كما هي معرّفة في التوصية [ITU-T J.1015] تسمح **لفتحة الأمن المعزّز** بتوفير تطبيقات رئيسية مأمونة لأغراض غير تجفير وفك تجفير المحتوى، من قبيل الاستيقان.

5.2.3 شقيق (Brother): **خلَف** آخر لنفس **السلَف**.

ملاحظة – يشار إلى **السلَف** و**التابع** و**الشقيق** بأنهم **كيانات** تدير **الشهادات**.

6.2.3 شهادة (Certificate): هيكل بيانات كما هو معرّف في الفقرة 5 من التوصية [ITU-T J.1012]، راجع أيضاً [b-ETSI GS ECI 001-3]، يتسم بتوقيع رقمي آمن تكميلي يحدد هوية **كيان**.

ملاحظة – يشهد حامل المفتاح السري للتوقيع على صحة البيانات – ويستيقن منها – بالتوقيع عليها بمفتاحه السري. ويمكن استعمال مفتاحه العمومي للتحقق من البيانات.

7.2.3 سلسلة الشهادات (Certificate Chain): قائمة بالشهادات التي تستيقن إحداها من الأخرى حتى **قائمة الإبطال الرئيسية** ضمناً.

ملاحظة – تكون **الشهادات** في **السطح البيني ECI** مصحوبة **بقائمة إبطال** تستثني **الشهادات** التي لا يمكن التحقق من صلاحيتها.

8.2.3 النظام الفرعي لمعالجة الشهادات (CPS): نظام فرعي في **المضيف ECI** يوفر معالجة للتحقق من **الشهادات** و**متانة** إضافية ضد التلاعب.

9.2.3 تابع، أتباع (Child, Children): كيان (**كيانات**) يشار إليه (إليها) **بشهادة** موقعة من **سلَفٍ** (مشترك).

ملاحظة – يشار إلى **السلَف والأتباع والشقيق** بكيانات تدير **الشهادات**: بيانات وبرمجيات استهلال تستعمل لبدء عمل نظام على رُقاقة (SoC) في معدات منشآت العميل (**CPE**).

10.2.3 خصائص المحتوى (Content Properties): خصائص تقدم معلومات بشأن الحقوق والالتزامات مع التطبيقات أو التحويلات اللاحقة للمحتوى، مثل المعلومات المتعلقة بحقوق الاستعمال والتحكم الانتقائي بالنواتج والمعلومات المتصلة بتحكم الأهل.

11.2.3 السطح البيني المشترك المدمج (ECI): المعمارية والنظام الموصفان في المعيار "السطح البيني المشترك المدمج" لفريق المواصفات الصناعية (ISG) التابع للمعهد الأوروبي لمعايير الاتصالات (ETSI)، واللذان يسمحان بتطوير وتنفيذ **وسطاء ECI** قابلين للتبادل وقائمين على البرمجيات في معدّات منشآت العميل، وبالتالي توفير قابلية التشغيل البيني لهذه المعدّات بالنسبة **للسطح البيني ECI**.

12.2.3 وسيط السطح البيني ECI (Embedded CI Client): تنفيذ وسيط CA/DRM يمتثل لمواصفات السطح البيني **ECI**.

13.2.3 أداة تحميل الوسيط ECI (ECI Client Loader): إحدى وظائف **المضيف ECI** التي تستخدم **نظام الأمن المعزّز** لتوفير الوظيفة بشكل حصري، والتحقق من صورة جديدة لبرمجيات **الوسيط ECI** وتثبيتها في حاوية **السطح البيني ECI** التابعة **للمضيف ECI**.

14.2.3 النظام الإيكولوجي للسطح البيني ECI (ECI Ecosystem): عملية تجارية مكوّنة من **سلطة استوثاق (TA)** وعدة منصات و**معدات CPE** في الميدان مطابقة **للسطح البيني ECI**.

15.2.3 مضيف السطح البيني ECI: نظام من عتاد وبرمجيات لإحدى معدّات منشآت العميل، يغطي الوظائف المتعلقة **بالسطح البيني ECI** ولديه سطوح بينية مع **وسيط ECI**.

ملاحظة – **المضيف ECI** هو جزء من البرمجيات الثابتة لمعدّات منشآت العميل.

16.2.3 أداة تحميل المضيف ECI: وظيفة من وظائف استنهاض معدّات منشآت العميل تَستخدم **نظام الأمن المعزز** لتوفير وظيفة التحقق من برمجيات **المضيف ECI** في معدات CPE وتثبيتها فيها.

ملاحظة – هذا المصطلح يستعمل في تشكيلة التحميل متعدد المراحل للإشارة إلى جميع وظائف التحميل الحساسية من حيث الأمن المتضمنة في عملية تحميل **المضيف ECI**.

17.2.3 مفتاح جذري للسطح البيني ECI (ECI Root Key): مفتاح عمومي يوفر مصدر الاستيقان من الكيانات و**الشهادات** المعتمدة من **السطح البيني ECI**.

18.2.3 كيان (Entity): منظمة (مصنِّع أو **مشغِّل** أو **بائع خدمة أمنية** مثلاً) أو عنصر من عناصر العالم الحقيقي (مثل **مضيف ECI** أو **عملية المنصة** أو **وسيط ECI**) يعرَّف بواسطة معرّف هوية فريد في **النظام الإيكولوجي للسطح البيني ECI**.

19.2.3 وصلة التصدير (Export Connection): علاقة مستيقن منها بين **فتحة أمن معزّز** تقوم بفكّ تجفير المحتوى و**فتحة أمن معزز** تعيد لاحقاً تجفير المحتوى الذي أزيل تجفيره وتبين أن إعادة التجفير هذه مسموحة.

20.2.3 سلَف (Father): الموقّع على **الشهادة** الخاصة **بالكيان التابع**.

ملاحظة – يشار إلى **السلَف** و**التابع** و**الشقيق** بأنهم **كيانات** تدير **الشهادات**.

21.2.3 سلم المفاتيح (Key Ladder): وظيفة من وظائف **مجموعة سلم المفاتيح** كما هي معرّفة في التوصية [ITU-T J.1015] تقوم بحساب كلمات التحكم والمعلومات المرتبطة بها المتعلقة باستعمال كلمات التحكم لتطبيقها في وظيفة فك تجفير المحتوى أو إعادة تجفيره في معدات منشآت العميل.

22.2.3 مجموعة سلم المفاتيح (Key Ladder Block): آلية آمنة ومتينة لحساب مفاتيح فك التجفير والتجفير والاستيقان كما هي معرفة في التوصية [ITU-T J.1015]، وهي عبارة عن **سلم مفاتيح** و**آلية استيقان**.

23.2.3 وسيط صغير (Micro Client): **وسيط ECI** أو وسيط غير مطابق **للسطح البيني ECI** يمكنه فك تجفير المحتوى الذي قام **مخدّم صغير** بإعادة تجفيره.

24.2.3 نظام صغير لإدارة الحقوق الرقمية (Micro DRM System): نظام لحماية المحتوى يعيد تجفير المحتوى في معدات منشآت العميل بواسطة **مخدّم صغير** ويسمح **لوسطاء صغار** مستيقن منهم بتشفير ذلك المحتوى الذي أزيل تجفيره.

ملاحظة – يقوم **مشغّل النظام الصغير لإدارة الحقوق الرقمية** بتوفير الخدمة إلى **المخدّم الصغير والوسطاء الصغار**.

25.2.3 مخدّم صغير (Micro Server): **الوسيط ECI** الذي يمكنه استيراد المحتوى الذي أزيل تجفيره وإعادة تجفيره والاستيقان من **وسيط ECI** محدد أو من مجموعة من **وسطاء ECI** باعتبارهم **الهدف** لعملية فك تجفير لاحقة.

26.2.3 مشغّل (Operator): منظمة توفر **عمليات المنصة** المدرجة مع **سلطة استوثاق (TA) السطح البيني ECI** لتوقيع **النظام الإيكولوجي للسطح البيني ECI**.

ملاحظة – يمكن أن يقوم **المشغّل** بتشغيل عدة **عمليات للمنصة**.

27.2.3 عملية المنصة ((Platform Operation (PO): حالة محددة لعملية تقديم خدمة تقنية يكون لها هوية واحدة **للسطح البيني ECI** بالنسبة للأمن.

28.2.3 مخدّم التجهيز (Provisioning Server): مخدّم يقع عادة في مكتب خلفي آمن يجهز المفاتيح ومعلومات آمنة أخرى لتسهيل وظيفة التجفير أو فك التجفير عبر **فتحة الأمن المعزّز**.

29.2.3 إبطال (Revocation): حالة استبعاد **كيان** طبقاً لتعداده في **قائمة الإبطال**.

30.2.3 قائمة الإبطال ((Revocation List (RL): قائمة **الشهادات** التي أبطلت وبالتالي ينبغي عدم استعمالها بعد الآن.

31.2.3 المتانة (Robustness): خاصية تنفيذ وظيفة أمنية **للسطح البيني ECI** تمثل الجهود و/أو التكلفة التي ينطوي عليها إلحاق الضرر بأمن الوظيفة الأمنية المنفذة.

32.2.3 شهادة جذرية (Root Certificate): شهادة موثوقة تعتبر مصدر الاستيقان من سلسلة شهادات **النظام الإيكولوجي للسطح البيني ECI**.

33.2.3 مسار فيديوي آمن (Secure Video Path): جميع وظائف معدات منشآت العميل (CPE) التي تقوم بمعالجة المحتوى (والتخزين المؤقت المطلوب له) من، وبما يشمل، فك تجفير المحتوى من خلال تضمين إعادة تجفير المحتوى عن طريق وسيط صغير أو نظام حماية الخرج.

34.2.3 بائع خدمة أمنية (Security Vendor): شركة تقدم أنظمة أمنية **للسطح البيني ECI** بما في ذلك **وسطاء ECI لمشغلي عمليات المنصة**.

35.2.3 الهدف (Target): **وسيط صغير** أو مجموعة **وسطاء صغار** يعاد من أجلهم تجفير المحتوى بواسطة **مخدم صغير**.

4 الاختصارات والأسماء المختصرة

تستعمل هذه التوصية الاختصارات والأسماء المختصرة التالية:

ACF حقل المراقبة الأمنية المعزّزة *(Advanced Security Control Field)*

AD البيانات المرتبطة *(Associated Data)*

AES معيار التجفير المعزّز *(Advanced Encryption Standard)*

AK مفتاح الاستيقان *(Authentication Key)*

API السطح البيني لبرمجة التطبيق *(Application Programming Interface)*

ARK مفتاح عشوائي للأمن المعزّز *(Advanced Security Random Key)*

AS الأمن المعزّز *(Advanced Security)*

CA النفاذ المشروط *(Conditional Access)*

CBC سَلسَلة وحدات التجفير *(Cypher Block Chaining)*

CENC التجفير العام *(Common Encryption)*

CISSA خوارزمية التخليط العامة القائمة على البرمجيات في تلفزيون بروتوكول الإنترنت *(Common IPTV Software-oriented Scrambling Algorithm)*

CP خصائص المحتوى *(Content Properties)*

CPE معدات منشآت العميل *(Customer Premises Equipment)*

CPS النظام الفرعي لمعالجة الشهادات *(Certificate Processing Subsystem)*

CSA خوارزمية التخليط العامة *(Common Scrambling Algorithm)*

CTR أسلوب العدّاد *(Counter Mode)*

CW كلمة تحكم *(Control Word)*

DRM إدارة الحقوق الرقمية *(Digital Rights Management)*

EAC شهادة ترخيص التصدير *(Export Authorization Certificate)*

EAOC **شهادة مشغل** تراخيص التصدير *(Export Authorization **Operator Certificate**)*

ECI السطح البيني المشترك المدمج *(Embedded Common Interface)*

ECM رسالة مراقبة الأحقية *(Entitlement Control Message)*

ECP حماية محسنة للمحتوى *(Enhanced Content Protection)*

EGC شهادة مجموعة التصدير *(Export Group Certificate)*

ERC شهادة **إبطال** التصدير *(Export **Revocation** Certificate)*

ESC شهادة نظام التصدير *(Export System Certificate)*

LK مفتاح الوصلة *(Link Key)*

MPEG فريق خبراء الصور المتحركة *(Moving Picture Experts Group)*

MSCSK المفتاح السري لشريحة المخدّم الصغير *(Micro Server Chipset Secret Key)*

PES التدفق الأولي بأسلوب الرزم *(Packetized Elementary Stream)*

PO **عملية** المنصة *(Platform **Operation**)*

POC شهادة **عملية المنصة** *(**Platform Operation** Certificate)*

POPK المفتاح العمومي **لعملية المنصة** *(**Platform Operation** Public Key)*

REAOC شهادة مشغل الاستيقان من إبطال التصدير *(**Revocation** Export Authentication **Operator** Certificate)*

REE بيئة تنفيذ غنية *(Rich Execution Environment)*

RFU محجوز لاستعمال لاحق *(Reserved for Future Use)*

RK مفتاح عشوائي *(Random Key)*

RL **قائمة الإبطال *(Revocation List)***

SPK المفتاح العمومي للمرسِل *(Sender Public Key)*

TA سلطة الاستوثاق *(Trust Authority)*

TEE بيئة تنفيذ موثوقة *(Trusted Execution Environment)*

TLS أمن طبقة النقل *(Transport Layer Security)*

TPEGC شهادة مجموعة التصدير الخاصة بطرف ثالث *(Third Party Export Group Certificate)*

TS تدفق النقل MPEG-2 *(MPEG 2 Transport Stream)*

URI المعلومات المتعلقة بحقوق الاستعمال *(Usage Rights Information)*

XT حقل التمديد *(eXTension field)*

5 الاصطلاحات

يبيّن استخدام المصطلحات المكتوبة بحروف داكنة أو كبيرة في التوصية الحالية أن المعنى المحدد لهذه المصطلحات خاص بالسطح البيني المشترك المدمج (ECI) وقد يحيد عن الاستخدام الشائع لهذه المصطلحات.

6 المبادئ

1.6 نظرة عامة

تنتمي هذه التوصية إلى سلسلة توصيات قطاع تقييس الاتصالات التي تستند إلى معمارية **السطح البيني ECI** [ITU-T J.1011]، راجع أيضاً [b-ETSI GS ECI 001-1]، والمتطلبات الأساسية **للسطح البيني ECI**، راجع أيضاً [b-ETSI GS ECI 001-2].

ويعرض الشكل 6-1 المبادئ الرئيسية **لنظام الأمن المعزّز**. ويتألف الجزء الأساسي من **نظام الأمن المعزّز** من **مجموعة سلم المفاتيح** كما هي معرّفة في التوصية [ITU-T J.1015]، ما يسمح بمعالجة آمنة عن طريق مفاتيح سرية، واستهداف مفاتيح شرائح إلكترونية محددة، والاستيقان من أصل مواد المفاتيح.

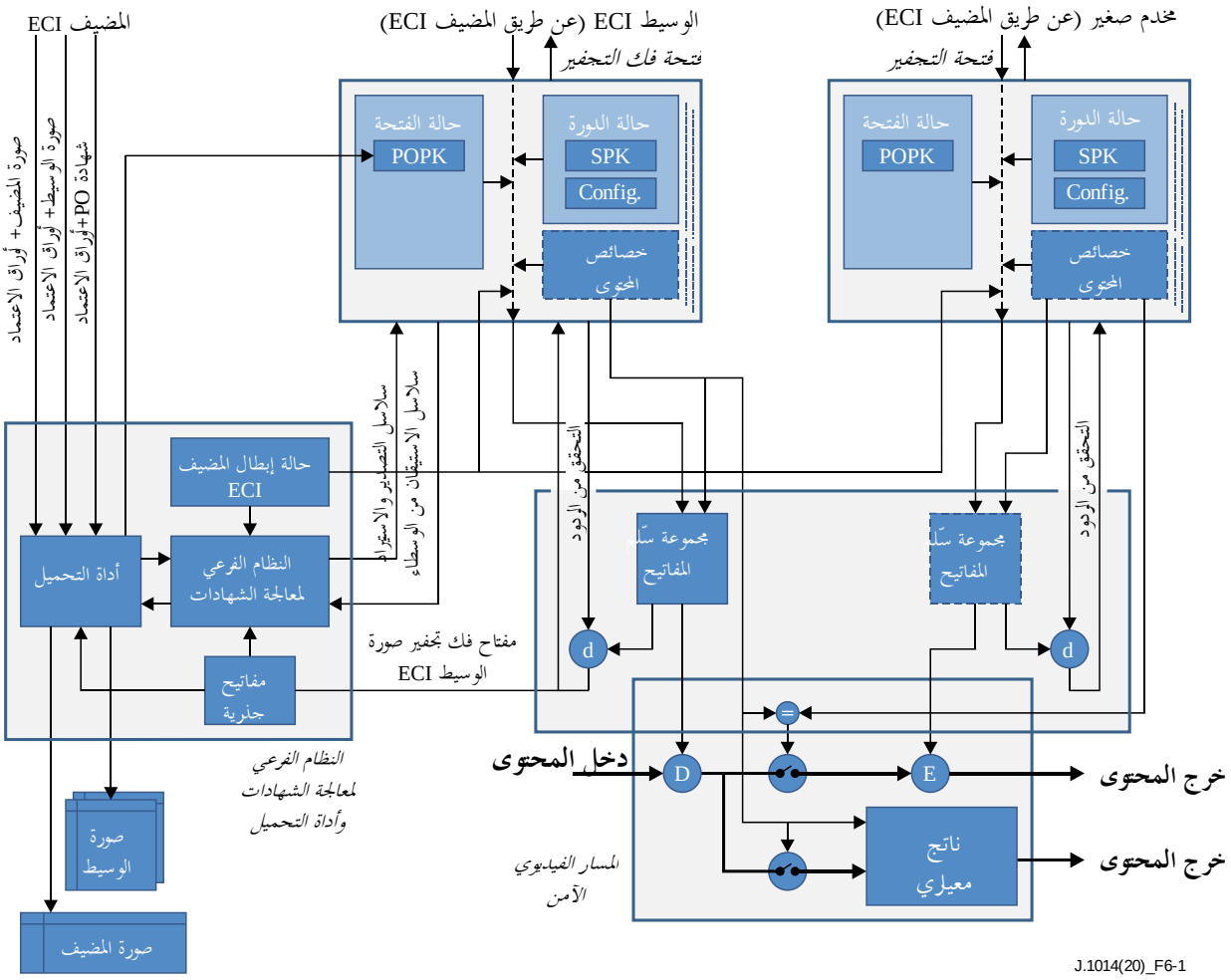
ويتجسد الأساس لتحميل الصور في **أداة التحميل**. وهي تستخدم **النظام الفرعي لمعالجة الشهادات** للتحقق من حالة **السطح البيني ECI** لصور **المضيف ECI** وصور **وسطاء ECI** وأوراق اعتماد **عمليات المنصة** باستخدام **مفتاح جذري** حديث **للسطح البيني ECI** و**قائمة إبطال** جذرية **للسطح البيني ECI**. ويمكن التحقق من أرقام صيغ **المفتاح الجذري للسطح البيني ECI** و**قائمة الإبطال** الجذرية **للسطح البيني ECI** التي يستعملها **المضيف ECI** و**وسطاء ECI** الآخرين بواسطة **وسطاء ECI** الذين تم تحميلهم. وبإمكان هؤلاء رفض إزالة تخليط المحتوى عند اكتشاف صيغ غير مقبولة طبقاً لمبدأ إنفاذ **الإبطال** في **السطح البيني ECI**. ويتم فك تجفير صور **وسطاء ECI** المجفّرة فور تحميلها.

ويستخدم كل **وسيط ECI** فتحة أمن معزز. ويحدد المفتاح العمومي **لعمليات المنصة** في **الوسيط ECI** هوية **الفتحة AS**. ويضمن **المضيف ECI** أن تكون تفاعلات **الوسيط ECI** من خلال السطح البيني **AS-API** موجهة إلى **الفتحة AS** المخصصة لذلك **الوسيط ECI**. وتوصف كل **فتحة AS** بحالة الفتحة وحالة الدورة لكل عملية تجفير/فك تجفير. ويمكن استعمال **الفتحة AS** لأغراض فك التجفير أو لأغراض التجفير. وتشمل حالة دورة **الفتحة AS** تشكيلة (Config) تحدد تفاصيل العملية وكيف ينبغي الاستيقان من حالة الدورة. ويوفر **الوسيط ECI** المعلومات المتعلقة بالتشكيلة ودخلاً لمعلومات أخرى عن الحالة. وتستعمل **مجموعة سلّم المفاتيح** للاستيقان من المفتاح العمومي للمرسِل (SPK) والمفتاح العمومي **لعملية المنصة** (POPK) والمعلومات المتعلقة

بالتشكيلة. ويمكن **للفتحة AS** أن توفر أعداداً عشوائية لمدخلات مختارة **لمجموعة سلم المفاتيح** بحيث تنتج مفاتيح عشوائية أو أن تستخدم كقيمة ظرفية لضمان مدخلات محسوبة حديثاً **لمجموعة سلم المفاتيح**. ويمكن استخدام هذه الآلية لمنع إعادة عرض المحتوى المجفّر ولضمان التجهيز الدائم **لفتحة أمن معزز** بواسطة **مخدّم التجهيز**.

وعند فك تجفير المحتوى يمكن الاستيقان من **خصائص المحتوى** إلى جانب حساب كلمات التحكم، ما يحقق بالتالي ارتباطاً وثيقاً مع المحتوى الذي أزيل تجفيره. وتحال **خصائص المحتوى** مع المحتوى إلى أي ناتج معياري لضمان الوضعية المناسبة لآليات التحكم المتعلقة بهذا الناتج. وتقارن هذه الخصائص على **وصلة التصدير** بتلك التي من أجلها أعيد تجفير المحتوى. ولا تتاح **وصلة التصدير** إلا من خلال **سلاسل شهادات** التصدير/الاستيراد المناسبة. ويتم التحقق منها بواسطة **النظام الفرعي لمعالجة الشهادات** بالنيابة عن **الفتحة AS**. ويمكن عدم تفعيل النواتج المعيارية من خلال آلية التحكم في الخرج.

ويمكن مزامنة كلمات التحكم المحسوبة مع المحتوى المنسّق لتدفق النقل MPEG باستخدام بروتوكول تناوب البتّات. ولهذا الغرض يستخدم **مسار فيديوي آمن** آلية تخزين احتياطي مضاعف مع كلمة تحكم حالية/تالية لمعالجة التدفق.



الشكل 6-1 – المخطط الوظيفي لنظام الأمن المعزز

2.6 نموذج متانة النظام

يحتاج **نظام الأمن المعزز** إلى تنفيذ متين. وتقاس **المتانة** عادة بدلالة الجهد و/أو التكلفة اللازمة لتفادي أي تدابير أمنية: أي رصد القيم السرية أو التلاعب بالحالة أو القيم في نظام أمني.

ولا تعرّف هذه التوصية أي نظام محدد **للمتانة** فيما يتعلق بوظائف السطح البيني **ECI** المتنوعة. ومع ذلك، تستند معمارية **متانة السطح البيني ECI** إلى فرضية أن بعض الوظائف هي أكثر متانة من غيرها. ويوضِح ذلك الشكل 6-2.

الشكل 6-2 – فرضية متانة النظام في السطح البيني ECI

ويمثل العالم الخارجي البيئة الأقل متانة التي يمكن لأي تهديد أن يوجد فيها. وينبغي حماية البيانات التي تمر عبر هذه البيئة من التلاعب والتفتيش غير المرخص باستخدام تقنيات الاستيقان والتجفير. ومع أن نظام التشغيل القوي (الذي يشمل عادة برنامجاً للتصفّح) قد يكون إلى حد ما منيعاً ضد التلاعب والاختراق، إلا أنه غير قادر عادة على تحمل هجمات قرصنة خارجية بمساعدة المستعمل أو صادرة عن جهات عدوانية. فالوظائف الحساسة من الناحية الأمنية **لوسطاء ECI** و**مضيف ECI** تعمل في بيئة محمية جيداً من هذه الهجمات. فلو تعرض **المضيف ECI** للاختراق يتعرض أيضاً جميع **الوسطاء ECI** للاختراق. وبالإضافة إلى القدرة على الصمود أمام الهجمات الخارجية، تتوفر الحماية **لوسطاء ECI** باستخدام الآلة الافتراضية **للسطح البيني ECI** [ITU-T J.1013]، راجع أيضاً المعيار [b-ETSI GS ECI 001-4]: أي إنه لا يمكن لهؤلاء الوسطاء النفاذ إلى معلومات **المضيف ECI** ولا إلى معلومات أي وسيط آخر من **وسطاء ECI**، إلا من خلال سطوح بينية محددة لبرمجة التطبيقات في **السطح البيني ECI**. كما أن **المضيف ECI** يضمن إمكانية نفاذ **الوسطاء ECI** إلى **نظام الأمن المعزز** و**مجموعة سلم المفاتيح**. ويوجد في صميم **مجموعة سلم المفاتيح المفتاح السري للشريحة** الذي يمكّن من مخاطبة **معدات منشآت العميل (CPE)** التابعة **للسطح البيني ECI** على نحوٍ فريد. وعادة يتم تنفيذ **مجموعة سلّم المفاتيح** والأجزاء الرئيسية من **نظام الأمن المعزّز** داخل العتاد و/أو داخل برمجيات ثابتة شديدة المتانة.

أما في الممارسة العملية، تقع متانة مكونات الأمن في سلسلة متصلة، يمكن تقسيمها من الناحية المعمارية ضمن تراتبية. وفي سياق ECI، يمكن وصفها على النحو التالي:

0 يتمتع العالم الخارجي (خارج الجهاز) بمستوى متانة 0. لذلك، يحتاج **وسيط ECI** إلى بروتوكول آمن للتواصل مع مركز التحكم. وهذا غير مشمول بمجال تطبيق التوصية بشأن السطح البيني **ECI**.

1 هناك نظام التشغيل وبرامج التشغيل والتطبيقات ومتصفح الجهاز وكلها في مستوى المتانة 1. وكثيراً ما يشار إلى البيئة التي تشغَّل فيها هذه الشفرة على أنها بيئة تنفيذ غنية (REE) وتتكون من معظم برمجيات التطبيقات ونظام التشغيل وبرامج التشغيل. ونظراً لضخامة هذه البرمجيات وخواصها الوظيفية، كثيراً ما تحتاج إلى التصحيح الترقيعي المتكرر لسد الثغرات الأمنية. ويمكن أن يستخدم **مضيف ECI** و**وسيط ECI** الخدمات التي تقدمها بيئة التنفيذ الغنية، مثل التوصيل الشبكي لأمن طبقة النقل (TLS)، ولكن لا يمكن الاعتماد عليها لضمان أي أمن، سواء كان تجفيراً أو استيقاناً لنقطة طرفية.

2 هناك بيئة تنفيذ لوظائف **مضيف ECI** وVM و**وسيط ECI**، والتي يجب أن تعمل جميعها في بيئة تنفيذ آمنة، غالباً على نفس المعالج. وفي مستوى المتانة 2، لا تنفَّذ هذه البيئات إلا الشفرة المستيقَنة، وتمتلك ذاكرة يتولى عتاد إنفاذها، وعزلاً

للجهاز عن بيئة التنفيذ الغنية (REE)، بما في ذلك النواة وبرامج التشغيل. ويشار إليها أحياناً باسم بيئات التنفيذ الموثوقة (TEE). ويمكن أيضاً تنفيذ هذا المستوى على معالج مخصص أو معالج أمني ذي عزل للذاكرة.

3 يتطلب **المسار الفيديوي الآمن** مستوىً أمنياً أعلى من **مضيف ECI** و**وسيط ECI**، وهو مستوى المتانة 3. ويمكن تنفيذه في توليفة من العتاد الآمن والبرمجيات الثابتة الآمنة.

4 الأنظمة الفرعية الأكثر أمناً ومتانة هي أنظمة **سلّم المفاتيح** ومحمِّل الإقلاع ومعالجة الشهادات وإبطالها، وهي تعمل بمستوى المتانة 4. وهي مدمجة بشكل مثالي في عتاد آمن ولكن نظراً لطبيعة التوصية بشأن السطح البيني **ECI**، يمكن لأجزاء أن تتطلب تشغيل البرمجيات الثابتة على معالج أمني مخصص. وتُتطلب تدابير محددة للحفاظ على سرية الأمن المتقدم وأسرار سلم المفاتيح والحسابات التي تنطوي عليها.

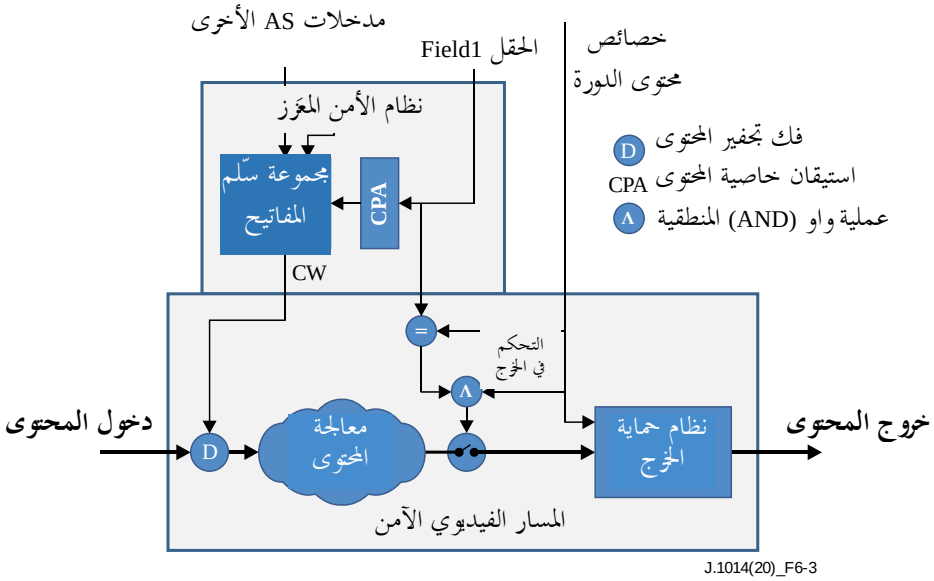
3.6 المسار الفيديوي الآمن والتحكم في نظام حماية الخرج

بمجرد فك تجفير المحتوى، يحميه **المسار الفيديوي الآمن** من الاستخدام غير المشروع. ويمكن أن يؤدي **المسار الفيديوي الآمن** العديد من العمليات الفيديوية (والسمعية) على المحتوى غير المجفَّر من قبيل: فك تعدد الإرسال، وفك التشفير، والمقايسة، وإعادة التشفير، والتشفير المائي، وما إلى ذلك. وجميع العمليات والأجزاء المكونة لها (بما في ذلك الذاكرات العابرة المعنية والمنافذ المستخدمة لنقل المحتوى بين أنظمة فرعية مختلفة من معدات منشآت العميل (CPE)) وسطوح التحكم البينية (خصائص المحتوى والسطوح البينية المحتملة مسجلة الملكية أو المصممة وفق خصوصية التنفيذ) التي تتحكم في خصائص المحتوى وخطوات المعالجة الحرجة وإعدادات حماية مخرجات المحتوى، جميعها تعتبر جزءاً من **المسار الفيديوي الآمن**.

وتحدد خصائص المحتوى [ITU-T J.1012] العمليات المسموح بها والمخرجات وأنظمة حماية الخرج والإعدادات المناسبة لحماية المحتوى. ويقدم نظام الأمن المعزَز آلية تجفير تسمى الاستيقان من حماية المحتوى (انظر الفقرة 3.2.8) للاستيقان الضمني من خصائص المحتوى باستخدام مجموعة سلم المفاتيح عند حساب كلمة التحكم. ويمكن لنظام حماية المحتوى اختيار إعدادات استيقان خاصية المحتوى. وتتمتع **خصائص المحتوى**، غير المستيقَنة بشكل ضمني بواسطة وظيفة استيقان خاصية المحتوى، بمستوى متانة **وسيط ECI** و**مضيف ECI**.

ويجدر بالذكر أن ثمة مواصفات تكميلية مطلوبة لتقديم تفاصيل كافية عن متانة تنفيذ **مسار فيديوي آمن** وأنظمة حماية الخرج.

ويقدم الشكل 6-3 تفاصيل المنطق الناظم لتأمين تسليم **خصائص المحتوى**. إذ يقوم **وسيط ECI** أولاً بتعيين **خصائص محتوى** الدورة للقسم التالي الذي ستُفك شفرته من المحتوى. وبعد ذلك، يقدم **وسيط ECI** المدخلات لحساب كلمة التحكم بما في ذلك الحقل Field1 (كجزء من متجه المدخلات **elk**: انظر الفقرات 1.7 و3.2.8 و7.4.2.8). وتُختار حقول خصائص المحتوى الخاصة بالحقل Field1 بأول بايتتين من الحقل Field1 في وظيفة استيقان خاصية المحتوى وتُستيقن لاحقاً بشكل ضمني باستخدام هذه القيمة كمدخل قيمة مفتاح متناظرة في حساب سلم المفاتيح لكلمة التحكم (CW). وسيؤدي إدخال قيمة مفتاح غير متناظرة إلى خطأ كلمة التحكم، فيتعذر بالتالي فك تجفير المحتوى. ثم تقارَن قيم Field1 عبر المسار الفيديوي الآمن (SVP) مع خصائص المحتوى المقابلة التي عينها وسيط ECI. وفي حال وجود أي تناقض لا يتعذر الخرج. ويمكن أيضاً حظر خرج نظام حماية معين بقيمة حقل التحكم المقابلة في الخرج.



الشكل 6-3 – المسار الفيديوي الآمن واستيقان خاصية المحتوى

4.6 مبادئ التوصيف

1.4.6 الحرية في التنفيذ

تعرّف هذه التوصية الحالات والوظائف التي تعمل على **نظام الأمن المعزّز** وتؤدي إلى حالة جديدة. ولا تحدد هذه التوصية التمثيل المحدد لحالة تنفيذ معينة، علماً بأنه يمكن تحديده بالكامل من خلال التنفيذ طالما أمكن إعادة تجزئة سلوك التنفيذ إلى حالات وتتابعات من الانتقال بين الحالات باستخدام وظائف الانتقال كما هي معرّفة في التوصية الحالية.

ملاحظة – في الكثير من الحالات تعتبر وظيفة **سلم المفاتيح** كما هي معرّفة في التوصية [ITU-T J.1015] جزءاً كبيراً من وظيفة الانتقال بين الحالات.

فعلى سبيل المثال، يمكن أن تكون إحدى عمليات تنفيذ الأمن المعزز مزودة بنظام فرعي سريع لمعالجة الشهادات (CPS) قادر على إعادة الاستيقان من المفتاح العمومي لعمليات المنصة (POPK) من **سلسلة شهادات عمليات المنصة** لكل تطبيق من تطبيقات **مجموعة سلّم المفاتيح**. وفي هذه الحالة لا يطلب من **فتحة الأمن المعزز** تخزين المفتاح POPK كقيمة مستيقن منها بطريقة عصية على التلاعب. وعلى نحو مماثل، قد تقرر بعض عمليات التنفيذ أن تحسب المفتاح LK_1 (المفتاح المتناظر الأعلى مستوى في **سلم المفاتيح**) فور استخدام عمليتي تجفير لاتناظريتين **لفتحة الأمن المعزز**، بينما يمكن لعمليات تنفيذ أخرى، قادرة على إجراء عمليات تجفير لاتناظرية بسرعة كافية، أن تعيد حساب المفتاح LK_1 انطلاقاً من المدخلات الأصلية لكل تطبيق من تطبيقات **سلّم المفاتيح**.

2.4.6 أسلوب التوصيف وعلاقته بالسطح البيني AS-API

لا يوجد سطح بيني مباشر لبرمجة التطبيقات بين **الوسيط ECI** و**نظام الأمن المعزز**. ويقوم **المضيف ECI** بدور القناة. ومع ذلك فإن تعاريف العمليات التي تجري في **فتحة الأمن المعزز** (**الفتحة AS**) تقابل مباشرة رسائل **السطح البيني AS-API** كما هي معرّفة في التوصية [ITU-T J.1012]، باستثناء المعلمة slotId التي لا تحتاج إليها رسائل السطح البيني AS-API التابع في **وسيط ECI**. ويوفر **المضيف ECI** المعلمة slotId إلى **نظام الأمن المعزز**.

وتعرّف المعاملات التي يقوم بها **المضيف ECI** (بالنيابة عن **وسيط ECI**) على **الفتحة AS** بوصفها إعلانات لوظائف مكتوبة بلغة C. وتقدم هذه الوظائف وصفاً لمعاملة غير قابلة للتجزئة تتعلق بحالة **الفتحة AS**. وقد يسفر ذلك عن حالة جديدة للفتحة. وليس التمثيل المحدد لمعلمات الوظائف نتيجة مباشرة للوظيفة المحددة في هذه التوصية، باستثناء ما يتعلق بوظائف التجفير. ومع ذلك، يعتبر التمثيل مهماً لتعريف السطح البيني **AS API** في التوصية [ITU-T J.1012].

7 تطبيق سلّم المفاتيح والوظائف المرتبطة به

1.7 لمحة عامة

يقوم **سلّم المفاتيح** و**آلية الاستيقان** المعرّفان في التوصية [ITU-T J.1015] بدور مركزي في جميع حسابات المفاتيح السرية المنفذة تنفيذاً قوياً في إحدى **معدات منشآت العميل (CPE) المطابقة للسطح البيني ECI**. ويطبّق **نظام الأمن المعزز** هاتين الوظيفتين على النحو المحدد في التوصية [ITU-T J.1015] مع مدخلات ومخرجات على النحو المحدد في هذه التوصية. ويجب أن يقوم **نظام الأمن المعزز** بمراقبة جميع مدخلات **مجموعة سلم المفاتيح**؛ فأي رصد أو تلاعب لن يكون ممكناً وفقاً للقواعد المعمول بها في مجال الامن المعزّز ومواصفات **مجموعة سلم المفاتيح** [ITU-T J.1015].

2.7 نظام الأمن المعزز واستيقان بيانات الوسيط

يمكن تزويد **نظام الأمن المعزز** ببيانات من **وسيط ECI**. ويوفر **نظام الأمن المعزز** وسيلة للتحقق من موثوقية هذه البيانات باستخدام الدخل AD **لمجموعة سلم المفاتيح**.

ويقوم نظام الأمن المعزز بحساب الدخل AD **لمجموعة سلم المفاتيح** باعتباره نتيجة لتطبيق دالة الاختزال (hash) على بيانات إضافية يتعين الاستيقان منها بالترافق مع حساب كلمة التحكم (CW) أو مفتاح الاستيقان (AK). وباتباع ترميز سلسلة البتات الوارد في التوصية [ITU-T J.1015] يتم حساب المفتاح AD على النحو التالي:

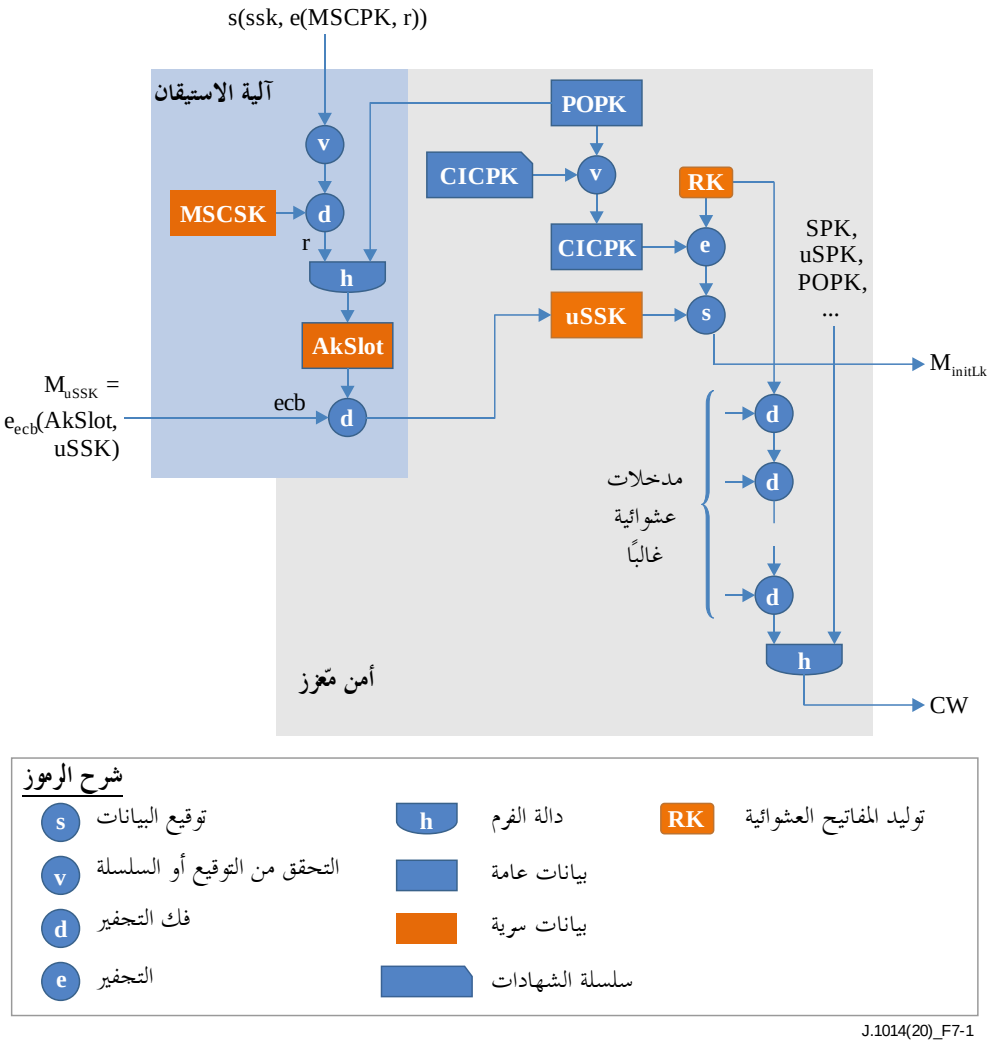
$$AD = hash(\ ACF \ ||\ lm \ ||\ ARK \ ||\ P_1 \ ||...||\ P_m \ ||\ C_1 \ ||...||\ C_m \ ||\ XT\ )$$

حيث يرد تعريف دالة "الاختزال" (*hash*) في الفقرة A.1. وlm هو دخل مؤلف من 8 بتات يتضمن تمثيل m في النظام الثنائي. وتقابل قيمة m القيمة التي تأخذها m في تعريف **مجموعة سلم المفاتيح**. أما طول كل P_i فيبلغ 048 2 بتة لأغراض نقل المفاتيح العمومية (قيم المفاتيح POPK). ويعرّف طول C_i بأنه مساوٍ لطول البنية SessionConfig في الفقرة 5.2.2.8، وطول XT بأنه يساوي 256، ويعمل كآلية تمديد عامة الأغراض. ويجب أن يضبط على الرقم 0. وARK هو عدد مؤلف من 128 بتة مخصص لتمثيل قيمة عشوائية أو قيمة كلها أصفار إذا لم يكن هناك حاجة لدخل عشوائي. أما حقل المراقبة الأمنية المعززة (ACF) فيمثل قيمة تحكم تحدد أسلوب التشغيل.

3.7 أسلوب المخدّم الصغير اللاتناظري

يطبق **نظام الأمن المعزز آلية الاستيقان** بهدف تحميل مفتاح سري للمرسل في **مخدم صغير** داخل **فتحة الأمن المعزز** لغرض القيام باستيقان لاتناظري بين **مخدمات الصغيرة** و**وسطاء صغار**.

ويبين الشكل 7-1 المبدأ الأساسي للحسابات الإجمالية في أسلوب الاستيقان اللاتناظري.



الشكل 7-1 – حساب أسلوب المخدّم الصغير اللاتناظري

وتستعمل **آلية الاستيقان** لحساب مفتاح استيقان **فتحة الأمن المعزز** AkSlot، وذلك باستخدام أدوات من بينها المفتاح السري لشريحة **المخدم الصغير** المبين هنا بالرمز MSCSK. ويستعمل المفتاح AkSlot لتحميل المفتاح السري uSSK **للمخدم الصغير**. ويستخدم **النظام الفرعي لمعالجة الشهادات** للاستيقان من المفتاح العمومي للشريحة CICPK لدى **الوسيط الصغير المستهدف** باستخدام المفتاح POPK كجذر و**سلسلة شهادات** تتضمن المفتاح CICPK في آخر **شهادة** في السلسلة. ويتولد مفتاح عشوائي RK ويستعمل المفتاحان CICPK وuSSK لتوليد رسالة الاستهلال **لسلم مفاتيح الوسيط الصغير** M_{initLk}. كما يستعمل المفتاح العشوائي كمفتاح متناظر في أعلى مستوى من **سلم المفاتيح** مع البنية ذاتها ودالة الاختزال ذاتها المعرفتين في الفقرة 1.7 من التوصية [ITU-T J.1015]. وتستعمل كلمة التحكم (CW) المحسوبة للتجفير بواسطة **المخدم الصغير**. ويمكن استعمال **سلم المفاتيح** النظامي **لوسيط صغير** من أجل حساب كلمة التحكم بهدف تجفيرها.

وتحدد الفقرة 10.4.2.8 مواصفات حساب المفتاح uSSK (الوظيفة **IdUssK**).

ويتم حساب الرسالة M_{initLk} على النحو الوارد في المخطط أدناه باستخدام اصطلاحات المواصفة الواردة في الفقرة 1.7 من التوصية [ITU-T J.1015]:

- Mkey = cl-chipset-ID || E(ClCPK,LK)
- MinitLk =(Mkey || S(uSSK,Mkey))

- حيث تمثل || وظيفة تسلسل البتات، وcl-chipset-ID معرّف هوية الشريحة في معدات منشآت الوسيط، وE() وظيفة التجفير اللاتناظري، وS() وظيفة التوقيع اللاتناظري كما هي محددة في الفقرة 2.7 من التوصية [ITU-T J.1015].

وأثناء تحميل المفتاح uSSK، تقوم **الفتحة AS** بتوليد مفتاح عشوائي (RK) جديد طبقاً للفقرة A.3.

وتتطابق عملية حساب CW انطلاقاً من مفتاح الوصلة (LK) ومدخلاته مع آلية **سلم المفاتيح** المحددة في الفقرة 2.7 من التوصية [ITU-T J.1015] والمدخلات ذاتها المحددة هنا والخرج ذاته (CW, CW-URI)، ولكن مع الاستعاضة عن حساب LK1 بمفتاح الدورة العشوائي RK الذي تولده **الفتحة AS للمخدم الصغير**.

4.7 السطح البيني لمسار فيديوي آمن

يستطيع **سلم المفاتيح** الذي يشتمل على تمديد أسلوب **المخدم الصغير** اللاتناظري أن يقوم بحساب كلمات التحكم مع الكمية التكميلية CW-URI. وتنتقل هذه الكلمات بشكل آمن إلى مورد التجفير أو فك التجفير في النظام الفرعي لمعالجة المحتوى الذي يمكنه (بصورة مؤقتة) تخزين المعلومات المتعلقة بالكلمة CW وCW-URI بالترافق مع المعلومات المتعلقة بمزامنة المفاتيح. وفي تطبيقات تدفق النقل، تتألف المعلومات المتعلقة بمزامنة المفاتيح من البتة الحالية/التالية، وتحدد بالتالي موقعين لخزن قيم CW. وفي التطبيقات القائمة على الملف، يتاح لمورد التجفير وفك التجفير موقع واحد للكلمة CW.

وهناك معلومات أخرى تصاحب كلمات التحكم لدى انتقالها من **الفتحة AS** إلى **مسار فيديوي آمن**، وهي:

- تطبيق الكلمة CW بوصفها كلمة تحكم بالتجفير أو فك التجفير.

ملاحظة – يعتبر ذلك إلى جانب CW-URI بمثابة ترخيص بتطبيق كلمة التحكم.

- معلومات متعلقة بالاستيقان من التصدير.
- خصائص المحتوى.
- الخاصية مفرد/مزدوج لكلمة التحكم في حالة إزالة التخليط في أسلوب تدفق النقل.

5.7 تعريف دخل–خرج مجموعة سلم المفاتيح في مجال الأمن المعزز

تحدد هذه الفقرة التقابل بين متغيرات وبنى الأسلوب في اللغة C بوصفها تمثيلاً لمدخلات **سلم المفاتيح** و**آلية الاستيقان** وتمديداتها كما هي معرفة في الفقرتين 2.7 و3.7. وتستخدم رموز أسماء المدخلات في القسم المتبقي من هذه التوصية لتحديد التطبيقات المتنوعة.

ويحدد التقابل بين بنى اللغة C وسلسلة الأثمونات بواسطة القواعد (وفق الترتيب الأقل أهمية) التالية:

- يتم التقابل بين حقول البتات والهياكل بدءاً من الحقل الأول وأدنى بتة (0) في الأثمون الأول.
- البنى التي لا يكون طولها مضاعفاً للرقم 8 بتات وتملأ في نهاية المضاعف التالي للرقم 8 بتات ببتات محجوزة يجب أن تضبط على الصفر عند المضاعف التالي للرقم 8 بتات.
- يتم تقابل الكيانات المؤلفة من 16 بتة و32 بتة و64 بتة وفق الترتيب الأقل أهمية (البايت الأقل دلالة أولاً).
- يتم تقابل الصفائف بحسب الترتيب المتزايد للمؤشر.

ويتم تقابل تتابعات الأثمونات مع سلاسل البتات باستخدام الوظيفة OS2BSP المعرفة في التوصية [ITU-T J.1015].

ملاحظة – تضمن القواعد الواردة أعلاه أن ترتيب أرقام البتات المستعمل في القيم الصحيحة المتمثلة بمتغيرات c مساوٍ لترتيب المدخلات المناظرة **لمجموعة سلم المفاتيح** المستعمل في التوصية [ITU-T J.1015].

ويرد اصطلاح تسمية المتغيرات في الجدول 7-1.

الجدول 7-1 – اصطلاح تسمية المتغيرات C في السطح البيني لسلم المفاتيح

اصطلاح تسمية المتغيرات–C	البتات	دخل أو خرج مجموعة سلم المفاتيح
ulong **cwUri**;	64	**CW-URI**
uchar **ad**[32]; /* not used directly, see clause 7.2 */	256	**AD**
ulong **spkUri**;	64	**SPK-URI**
typdef uchar PubKey[256]; PubKey **spk**[16]; /* (spk[i-1] == SPK_i) */	2 048 × 16	**SPK_i, i=1..16**
uchar **nSpk**;		**m**
typedef struct InputV{ ulong chipsetId; uchar elk1[256]; uchar signature[256]; } InputV; InputV **inputV**;	64+ 2 048 + 2 048	input to V
typedef uchar SymKey[32]; Symkey **elk**[i]; /* (elk[i-1] == E(LK_i,LK_{i+1})) */ /* **C-input** == E(LK_{t-1},LK_{t1}), i.e. the one but last input ***/**	256 × 24	**E(LK_i,LK_{i+1}), i=1..24; LK_{t+1}=r**
uchar **nElk**;		**t**
value set to 0		**τ_b**
ulong **chipsetId**;	64	**Chipset-ID**
uchar **challenge**[16];	128	**Challenge**
uchar **response**[16];	128	**Response**
The inputs and outputs identified in the present Recommendation are defined below.		
uchar **acf**[15]; /* operation mode */	128-8	**ACF**
uchar **ark**[16];	128	**ARK**
PubKey **pk**[32]; /* first **m** values are applied */	2 048 × 32	**P_i**
SessionConfig **config**[?]; /* SessionConfig is defined in clause 8.2.2.6 */	sizeof(SessionConfig) × 32	**C_i**
uchar **XT**[32]; /* value always set to 0 */	256	**XT**
InputV **mInitLk**;	64+ 2 048 + 2 048	M_{initLk}

تعرّف وظائف c التالية باستخدام متغيرات الدخل الواردة أعلاه للحصول على نتائج.

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

الدلالات:

تقوم هذه الوظيفة بحساب وظيفة المجموعة V والمجموعة C في **سلم المفاتيح** لإنتاج lk1.

وفي حالة مخدّم لاتناظري تقوم الوظيفة التالية بحساب الرسالة الاستهلالية **للوسيط الصغير المستهدف** كما هي معرفة في الفقرة 3.7:

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

الدلالات:

تقوم هذه الوظيفة بحساب الرسالة الاستهلالية iniLk1 وفقاً للفقرة 3.7.

ويقوم ما يلي بالوظائف المتبقية **لسلم المفاتيح**:

```
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])
```

الدلالات:

تقوم هذه الوظيفة بحساب القسم المتبقي من **سلم المفاتيح** باستخدام lk1 كنتيجة للمجموعة D في **سلم المفاتيح** للحصول على نتيجة CW في **مسار فيديوي آمن**.

ويقوم ما يلي بوظائف **آلية الاستيقان** اللازمة لحساب مفتاح الاستيقان AK:

```
SymKey AuthMech(InputV inputV, uchar acf[15], uchar ark[16],
        PubKey pk[16], SSCnfg clCnf[16], char XT[32], ulong spkUri, uint nSpk,
        uint spkIndx, PubKey spk[16])
```

الدلالات:

تقوم هذه الوظيفة بحساب **آلية الاستيقان** حتى المفتاح AK لإعطاء النتيجة.

ومن أجل استخدام المفتاح AK المحسوب تعرّف الوظيفة التالية باستخدام السطح البيني للتحقق من الردود ومجموعة الوظائف d في **آلية الاستيقان** الواردة في الفقرة 8 من التوصية [ITU-T J.1015].

```
uchar[16] AuthMechResponse(SymKey ak, uchar[16] challenge)
```

الدلالات:

تقوم هذه الوظيفة بحساب الرد على تحقق يستخدم المفتاح AK كمفتاح استيقان على النحو المعرف في **آلية الاستيقان**.

6.7 تعريف حقل المراقبة الأمنية المعززة (ACF)

يسهم الدخل المتمثل بحقل المراقبة الأمنية المعززة (ACF) إلى **مجموعة سلم المفاتيح** في تحديد الخصائص الرئيسية لأسلوب التشغيل. وتحدد قيمة المعلمة acf[0] في الجدول 7-2.

الجدول 7-2 – قيمة ACF[0] لتطبيق سلم المفاتيح

Acf[0]	القيمة	الوصف
AcfCw1Mode	0x11	عملية **سلم المفاتيح** كما هي محددة في التوصية الحالية. acf[1]..acf[14] تساوي 0x00.
AcfAk1Mode	0x12	عملية **آلية الاستيقان** كما هي محددة في هذه التوصية. يشار إلى قيمة acf[1] بأنها **AkModeField**. وتحدد القيم المعتمدة في الجدول 7-3. acf[2]..acf[14] تساوي 0x00.
محجوز	قيمة أخرى	محجوز لاستعمال لاحق

وفيما يلي التعريف c التكميلي لهذه الطريقة AcfCw1Mode لتطبيقه كمعلمة **سلم المفاتيح**:

```
const uchar acfCw1Mode= { AcfCw1Mode, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
```

وفيما يلي التعريف c التكميلي لهذه الطريقة AcfAk1Mode لتطبيقه كمعلمة **سلم المفاتيح**:

```
const uchar acfAk1Mode= { AcfAk1Mode, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
```

الجدول 7-3 – تعريف الحقل AkModeField في الأسلوب AcfAk1Mode

السجل	البتة	القيمة		الوصف
AkUseFlag	8	0b0	**AkUseAS**	تطبيق الفتاح AK **لنظام الأمن المعزز** فقط
		0b1	**AkUseCl**	تطبيق الفتاح AK **لوسيط ECI** فقط
AkOnline	7	0b0	**AkOffline**	يحدد المفتاح AK في أسلوب "خارج الشبكة" أحادي الاتجاه فقط. يمكن إجراء حساب مسبق للتحقق من الردود.
		0b1	**AkOnline**	يحدد المفتاح AK باستخدام مفتاح AKRK عشوائي مرة واحدة يتطلب إجراء حساب التحقق من الردود "على الشبكة"
AkAsAppl فقط إذا كان AkUseFlag= **AkUseAS** محجوز في الحالات الأخرى	0..3	0x0	**AkConfigAuth**	الاستيقان من عنصر تشكيلة **الفتحة AS**
		0x1	**AkLdUssk**	استعمال المفتاح AK لفك تجفير وتحميل المفتاح uSSK التابع **لمخدّم صغير**
		0x2	**AkClImg**	استعمال المفتاح AK لفك تجفير المفتاح اللازم لفك تجفير صورة **وسيط ECI** المقرر تحميلها
		0x3 0xF	محجوز	محجوز لاستعمال لاحق
RFU	غير ذلك	0		محجوز لاستعمال لاحق. تضبط القيمة على الصفر.

8 فتحة الأمن المعزّز

1.8 مقدمة إلى فتحة الأمن المعزّز

يشمل **نظام الأمن المعزّز** معلومات متعلقة بحالة كل وسيط محمّل من **وسطاء ECI**. ويتمثل تحديد الهوية الذي يربط **وسيط ECI بفتحة الأمن المعزز (AS)** بالمفتاح العمومي **لعمليات المنصة** (POPK) الخاص **بالوسيط ECI**. ويقوم **المضيف ECI** بتحميل المفتاح POPK الخاص **بوسيط ECI** في **فتحة AS** متاحة. ومن الآن فصاعداً تصبح حالة **الفتحة AS** مرتبطة بهذا **الوسيط ECI**. وستستخدم أي عملية مجدية **للفتحة AS** المفتاح POPK كدخل، مما يجعل النتيجة خاصة **بالوسيط ECI** المحدد ولا معنى لها بالنسبة للآخرين.

وتضمن **متانة المضيف ECI** عدم إمكانية نفاذ **وسيط ECI** إلا إلى المعلومات المتعلقة بالفتحة المخصصة له: وفي حال تأدية **المضيف ECI** لوظائفه بشكل صحيح، فإنه يضمن أن **الوسيط ECI** المعين فقط سيتمكن من النفاذ إلى **الفتحة AS** الخاصة به. وإذا تعرض **المضيف ECI** للخطر بشكل أو بآخر، فإن "آلية إغلاق" المفتاح POPK تضمن أن مدخلات **الفتحة AS** التي تشكل مجموعة متسقة هي وحدها التي يمكنها إعطاء نتائج مجدية على شكل مفاتيح لفك التجفير (CW) و**خصائص المحتوى** المرتبطة بها أو مفتاح للاستيقان (AK).

وإذا قرر **المضيف ECI** إعادة تحديد الغرض من **الفتحة AS** لوسيط آخر من **وسطاء ECI**، تمحى أي حالة مجمّعة متعلقة **بالوسيط ECI** (POPK) السابق.

2.8 تعريف فتحة الأمن المعزّز

1.2.8 لمحة عامة

تعرّف **فتحة الأمن المعزز** بدلالة ***متغيرات الحالة ومتغيرات الدخل*** المتعلقة ***بوظائف تعديل الحالة***. وقد تم اختيار تمثيل قيم الحالة والدخل والخرج في هذه الفقرة بحيث تعرّف العمليات التي تجري عليها بدلالة التمثيلات الثنائية المحددة هنا. ويتعلق ذلك على وجه الخصوص بإدراجها في الحسابات التجفيرية وحسابات **سلم المفاتيح**. ويمكن لعمليات التنفيذ الفعلية أن تختار تمثيلات الحالات الخاصة بها على أن تقوم بتحويل أي تمثيل مخصص إلى التمثيل المحدد هنا لدخل أي عملية تجفيرية.

ويجب أن تكون جميع متغيرات الحالة المتعلقة **للفتحة AS** محمية بقوة ضد أي تعديلات الضارة. فبعض متغيرات الحالة تنطوي على معلومات يجب أن تبقى سرية: ويجب أن تكون هذه السجلات محمية بقوة من النفاذ غير المرخّص. وتعرّف هذه المتغيرات باستعمال عبارة "سري" كجزء من التعريف بلغة C. وينبغي الحفاظ على سرية أي عملية حسابية تستند إلى قيمة متغير سري إلا إذا كانت النتيجة مشتركة صراحة. ويجب أن يتمتع أي موقع لخزن قيمة و/أو عملية حساب سرية **بالمتانة** نفسها المطلوبة **لمجموعة سلم المفاتيح** [ITU-T J.1015].

وتعمل **الفتحة AS** في دورات. وتعمل كل دورة وفقاً لإعدادت التشكيلة الخاصة بها التي تشكل جزءاً من حالة الدورة. ويحدد **الوسيط ECI** تشكيلة الدورة التي يجب الاستيقان منها قبل الاستعمال باستخدام **آلية الاستيقان** أو خصائص الاستيقان الضمنية **لسلم المفاتيح**.

وتحدد جميع متغيرات الحالة ووظائفها بدلالة اللغة C [ISO/IEC 9899]. ولا يراعى ترتيب التتابعات في اللغة C بدقة بمعنى أنه يمكن تعريف العناصر بعد استعمالها. ويتم نسخ الصفائف ذات الحجم الثابت مع بيان assignment واحد (بدلاً من نسخ قيمة المؤشر) كما لو أنها مدرجة في بنية "struct".

وفيما يتعلق بالأخطاء، فقد وضعت الشفرة بشكل قابل أكثر للعرض عن طريق تحديد تدقيق ضمني للأخطاء. فتخصيص قيمة محجوزة لأحد متغيرات الحالة أو لحقل فيه (انظر تعريف الحقل) يعتبر خطأ. وإذا كان الجانب الأيمن لعبارة البيان هذه يستند إلى معلمة واحدة من معلمات الوظيفة يجاب بوجود خطأ في هذه المعلمة: القيمة -i للمعلمة i.

وتحدد جميع القيم المبدئية للحقول والمتغيرات بقيمة 0، إلا إذا حُدِّد خلاف ذلك صراحة.

2.2.8 تعريف حالة الفتحة AS

1.2.2.8 حالة الفتحة والدورة

تحدد حالة **الفتحة AS** على شكل حالة مشتركة للفتحة وحالة الدورة لكل دورة من تجفير أو فك تجفير. وفيما يلي تعريف لبنية حالة الفتحة. ويرد تعريف الحقول في الجدول 8-1.

```
#define NSLOTS       /* (maximum) number of slots */
#define NSESSIONS    /* (maximum) number of sessions */
#define MaxSpkEncr 4 /* maximum number of encryption SPK values */

typedef SessionState {
    bool                active;
    uint                configAuthMode:4;
    uint                mh;
    SessionConfig   config;
    PubKey              spk;
    ulong               spkUri;
    uchar               spkIndx;
    int                 coupledSessionId;
    uint                nEncr;
    PubKey              encrSpk[MaxSpkEncr];
    PubKey              encrPopk[MaxSpkEncr];
    ulong               encrCwUri;
    Secret SymKey   lk1;
    Secret PrivKey   ussk;
    RkState rkState;
    importExportState ies;
} SessionState ;

typedef struct SlotState {
    uint                version:4;
    uint                slotMode:4;
    uint                clientCheckFlag:1;
    uint                reserved:3;
    uint                POClRLVnr: 24;
    PubKey              popk;
    SymKey              slotRk;
    Secret SymKey   akClient;
    SessionState    se[NSESSIONS];
} FixedSlotState;

SlotState ss[NSLOTS]
```

الجدول 8-1 – تعريف بنية حالة الفتحة AS

الحقل	الوصف
active	يكون صواباً إذا كانت الدورة نشطة، وخطأ خلاف ذلك. الحالة المبدئية خطأ.
configAuthMode	الأسلوب الذي تم بموجبه الاستيقان من تشكيلة الفتحة. القيم المسموح بها هي: **ConfigAuthModeNone**: 0x0، لم يتم الاستيقان من تشكيلة الفتحة. **ConfigAuthModeAk1**: 0x1، تم الاستيقان من تشكيلة الفتحة باستخدام آلية مفتاح الاستيقان (AK) كما هي معرفة في الفقرة 8.4.2.8. جميع القيم الأخرى محجوزة.
Mh	معالجة الوسائط التي ترتبط بها دورة **الفتحة AS**.
clientCheckFlag	جرى تحميل **وسيط ECI** جديد. تبدأ عملية التحقق من **POClRLVnr** عند استهلال الدورة. القيمة المبدئية 0b1.
Reserved	الحقل محجوز؛ يضبط على الصفر.
POClRLVnr	رقم صيغة **قائمة الإبطال** المتعلقة بوسطاء **عمليات المنصة** التي استخدمت للتحقق من **الوسيط ECI** قبل تحميله. وستتم مقارنته عند استهلال كل دورة **للوسيط ECI** مع أدنى صيغة يتوقعها **الوسيط ECI**.
slotConfig	تشكيلة الفتحة.
spk	المفتاح العمومي المستخدم لحساب LK1 وAK.
spkUri	استخدام سجل المعلومات المتعلقة بقواعد استعمال المتجه SPK لحساب LK1.
spkIndx	مؤشر اختيار موقع سجل SPK في متجه SPK لحساب LK1.
coupledSessionId	لا ينطبق إلا إذا كانت دورة **الفتحة AS** في أسلوب فك التجفير. دورة (فك تجفير) ثانية مقترنة بهذه الدورة. تجمع تدفقات المحتوى التي أزيل تجفيرها وتقارن **خصائص المحتوى**. القيمة المبدئية -1.
nEncr	عدد قيم مدخلات SPK/POPK المستخدمة في التجفير (باستثناء spk الخاص بالفتحة).
encrSpk	قيم SPK لتجفير المحتوى بواسطة **سلم المفاتيح**.
encrPopk	قيم POPK لتجفير المحتوى بواسطة **سلم المفاتيح**.
encrCwUri	قيمة CwUri اللازمة لتجفير المحتوى بواسطة **سلم المفاتيح**.
lk1	أعلى مستوى لمفتاح الوصلة اللازم لحساب كلمات التحكم باستخدام **سلم المفاتيح**.
Ussk	المفتاح السري **لمخدم الصغير** (لتطبيقات **المخدم الصغير**)
rkState	حالة المفتاح العشوائي للدورة.
ies	حالة الاستيراد/التصدير للدورة.
version	صيغة حالة الفتحة. القيم المسموحة هي: 0x1: الصيغة 1 جميع القيم الأخرى محجوزة.
slotMode	الأسلوب الي تعمل بموجبه الفتحة. القيم المسموحة هي: **SlotModeDecr:** 0x1، الفتحة مشغلة بأسلوب فك التجفير. **SlotModeEncr**: 0x2، الفتحة مشغلة بأسلوب التجفير. جميع القيم الأخرى محجوزة.
popk	مفتاح عمومي **للوسيط ECI** الذي يستخدم هذه الفتحة.
slotRK	عدد عشوائي يستخدم في بروتوكولات التحقق من الردود على الشبكة، مثلاً مع **مخدم التزويد**. تضبط القيمة عند استهلال الفتحة.
akClient	مفتاح استيقان لأغراض معالجة الوسطاء.
se	حالة الدورة (لجميع الدورات في **الفتحة AS**)
ss	حالة **الفتحة AS** (لجميع الفتحات)

وتكون القيمة المبدئية لكل عنصر من عناصر الحالة عند الاستهلال مساوية للصفر، إلا إذا حُدّد خلاف ذلك.

2.2.2.8 تشكيلة فك التجفير

تحدد الشفرة بلغة CC الواردة أدناه حالة التشكيلة المطبقة على دورة **الفتحة AS** في أسلوب فك التجفير ويرد وصفها في الجدول 8-2. وتحدد هذه الحالة تفاصيل تشغيل دورة الفتحة AS عندما تعمل بأسلوب فك التجفير. ويمكن الاستيقان من هذه البيانات بتطبيق **آلية استيقان** مناسبة أو بإجراء حسابات **سلم المفاتيح**.

```
typedef struct DecryptConfig {
    uint                configVersion:4;
    uint                reserved1:4;
    uint                klModeAuth:1;
    uint                akModeAuth:1;
    uint                rkKlMode:1;
    uint                spk0NoDecrypt:1;
    uint                reserved2:6;
    RKMode              rkDecrMode;
    EciRootState    minEciRootState;
    uint                minClientVersion:24;
} DecryptConfig;
```

الجدول 8-2 – تعريف بنية التشكيلة DecryptConfig

الحقل	الوصف
configVersion	صيغة تشكيلة فك التجفير. القيمة المحددة هي 0x1: الصيغة 1. جميع القيم الأخرى محجوزة. تمتنع دورة **الفتحة AS** عن تنفيذ أي وظيفة للانتقال من حالة إلى حالة إذا لم تكن قيمة هذا الحقل مسموحة.
reserved1	حقل محجوز؛ يجب أن يضبط على 0.
klModeAuth	إذا ضبطت هذه البتة، يجب أن تطبق **الفتحة AS** التشكيلة ClientConfig من أجل الاستيقان في جميع حسابات **سلم المفاتيح**. ويتم الاستيقان من هذه البتة بحد ذاتها في جميع حسابات **سلم المفاتيح**.
akModeAuth	إذا ضبطت هذه البتة، يجب أن تتحقق دورة **الفتحة AS** من أن الأسلوب configAuthMode مضبوط على ConfigAuthModeAk1 قبل السماح بإجراء أي حساب **لسلم المفاتيح**. ويتم الاستيقان من هذه البتة بحد ذاتها في جميع حسابات **سلم المفاتيح**.
rkKlMode	إذا ضبط هذا العلم، يجب تطبيق الحقل **slotRK** في جميع حسابات **سلم المفاتيح** بالنسبة **للفتحة AS**.
spk0NoDecrypt	إذا ضبط هذا الحقل، لا يسمح باستعمال spk[0] (spkIndx==0) كدخل لوظيفة **سلم المفاتيح**) من أجل الاستيقان من المفتاح LK1 الخاص بالفتحة عند العمل بأسلوب فك التجفير.
reserved2	حقل محجوز؛ يجب أن يضبط على 0.
rkDecrMode	يحدد تطبيق مفتاح عشوائي للدورة من أجل حسابات **سلم المفاتيح**. انظر الفقرة 5.2.2.8.
minEciRootState	أدنى قيمة لصيغة جذر **السطح البيني ECI** وصيغة **قائمة الإبطال** الجذرية. وإذا استخدَم النظام الفرعي لمعالجة الشهادات (CPS) في هذه البنية قيماً أقل **لمفتاح جذري للسطح البيني ECI** أو **لقائمة إبطال** جذرية لأغراض الاستيقان من **السطح البيني ECI** لا يسمح بإجراء حسابات سلم **المفاتيح** للدورة.
minClientVersion	صيغة **الوسيط ECI**. تستخدم للتحقق من أرقام صيغة **قائمة الإبطال** المتعلقة بالمفتاح POPK.

3.2.2.8 تشكيلة التجفير

تحدد الشفرة بلغة C الواردة أدناه حالة التشكيلة المطبقة على دورة **الفتحة AS** في أسلوب التجفير ويرد وصفها في الجدول 8-3.

```
typedef struct EncryptConfig {
    uint            configVersion:4;
    uint            reserved1:4;
    uint            microServerVersion:24;
    uint            asymKlMode:1;
    unit            rkKlMode:1;
    uint            reserved2:22;
    RkMode          rkEncrMode;
    uchar           basicUriTrfr;
    uint            contPropControl;
    ContProp        defaultCP;
    EciRootState    minEciRootState;
} EncryptConfig;
```

الجدول 8-3 – تعريف بنية التشكيلة EncryptConfig

الحقل	الوصف
configVersion	صيغة تشكيلة التجفير. القيمة المحددة هي 0x1: الصيغة 1. جميع القيم الأخرى محجوزة. تمتنع **الفتحة AS** عن تنفيذ أي وظيفة للانتقال من حالة إلى حالة إذا لم تكن قيمة هذا الحقل مسموحة.
Reserved1	حقل محجوز: يجب أن يضبط على 0.
microServerVersion	رقم الصيغة الخاص **بتشكيلة المخدم الصغير**. يستخدم أيضاً كرقم أدنى صيغة **لقائمة الإبطال** من أجل الاستيقان من **الوسيط الصغير** في أسلوب **المخدم الصغير** اللاتناظري.
asymKlMode	إذا ضبط هذا العلم، يجب أن يعمل **سلم المفاتيح** وفقاً **لآلية الاستيقان** اللاتناظرية من الوسيط المحددة في الفقرة 3.7.
rkKlMode	إذا ضبط هذا العلم، يجب تطبيق الحقل slotRK في أي من حسابات **سلم المفاتيح**.
Reserved2	حقل محجوز: يجب أن يضبط على 0.
rkEncrMode	يحدد تطبيق مفتاح عشوائي للدورة من أجل حسابات **سلم المفاتيح**. انظر الفقرة 5.2.2.8.
basicUriTrfr	يحدد تحولات الحالة للمعرّف URI الأساسي من وصلة الاستيراد قبل تطبيق المعرّف URI الأساسي بوصفه خاصية من خصائص المحتوى المجفر. انظر الجدول 8-5 للاطلاع على القيم.
contPropControl	يحدد كيفية حساب **خصائص المحتوى** المجفر. انظر الجدول 8-4.
defaultCP	قيمة مبدئية لجميع حقول خصائص المحتوى. ويخضع تطبيقها في حساب **سلم المفاتيح** لمراقبة الحقل contPropControl.
minEciRootState	أدنى قيمة للصيغة الجذرية **للسطح البيني ECI** وصيغة **قائمة الإبطال** الجذرية. وإذا استخدم النظام الفرعي لمعالجة الشهادات (CPS) في هذه البنية قيما أقل للمفتاح الجذري **للسطح البيني ECI** أو **لقائمة إبطال** جذرية لأغراض الاستيقان من **السطح البيني ECI** لا يسمح بإجراء حسابات **سلم المفاتيح**.

والحقل contPropControlFields هو عبارة عن صفيف من 16 حقلاً يتكون كل حقل منها من بتّتين. وتبين الحقول المكونة من بتتين كيفية التحكم **بخصائص محتوى** الحقل 1 في الخرج المجفر. ويرد وصفه في الجدول 8-4. وتقابل البتتان 2n و2n+1 للحقل CpControlFlag البايت n للحقل Field1.

الجدول 8-4 – تعريف الحقل CpCtrl

اسم العلم	القيمة	الوصف
CpCtrlCopy	0b00	ينسخ بايت خاصية المحتوى CP Field1 من وصلة الاستيراد
CpCtrlDef	0b01	يضبط بايت خاصية المحتوى CP Field1 على قيمة البايت defaultCP المقابل
CpCtrlMS	0b10	يضبط بايت خاصية المحتوى CP Field1 بواسطة **مخدم صغير**
Reserved	0b11	القيمة محجوزة

يقوم الحقل basicUriTrfr بتعديل السلوك الوارد أعلاه للحقل CPControlFlags المتعلق بالحقل BasicUri حين تكون حالة CPControlFlag مساوية **CPCopy**. ويحدد الجدول 8-5 السلوك البديل.

الجدول 8-5 – قيم الحقل BasicUriTrfr ووصفه

اسم العلم	القيمة	الوصف
JustCopy	0x00	ينسخ بايت خاصية المحتوى CP Field1 من وصلة الاستيراد
NoMoreCopy	0x01	تتحول حالة basicURI للحقل **RedistributionProtected** إلى حالة **ViewOnly**
Reserved	Other	محجوز لاستعمال لاحق
ملاحظة – عند ضبط الحالة BasicUriTrfr على NoMoreCopy، لا يسمح نظام **الوسيط الصغير** بالتدفق إلا لمحتوى محمي عند دخل **المخدم الصغير**.		

4.2.2.8 التحكم بمفتاح الدورة العشوائي

تحدد البنية RKMode المعرّفة في الشفرة بلغة C أدناه والجدول 8-6 الأسلوب الذي يتعين فيه تطبيق مفتاح الدورة العشوائي في **سلم المفاتيح**.

```
typedef struct RKMode {
    uint        mode:2;
    uint        limit:6;
} DecryptConfig;
```

الجدول 8-6 – بنية المفتاح العشوائي لدورة فك التجفير والتجفير

الحقل	الوصف
mode	يحدد أسلوب تطبيق مفتاح الدورة العشوائي. والقيم هي: • **RKModeNone** :0b00، لا يدرج مفتاح دورة عشوائي. • **RKModeDataLimit** :0b10، يطبق مفتاح الدورة العشوائي مع حد للبيانات. • **RKModeTimeLimit** :0x11، يطبق مفتاح الدورة العشوائي مع حد زمني. • 0b01: قيمة محجوزة.
limit	تحدد القيمة الحد المطبق فيما يتعلق بعدد الثواني الفعلية أو الحجم بالكيلوبايت للبيانات التي تم فك تجفيرها أو تجفيرها منذ استهلال المفتاح العشوائي. وتحدد الدالة ()limitValue القيمة الحدية الفعلية المطبقة. والقيمة 63 محجوزة.

```
uint limitValue(uint limit) {
    uint val;

    if (limit==0) return 1;
    limit -=1;
    if (limit&0b1 == 0b0) val=2 else val=3;
    return val * (1<<(limit>>1));
}
```

5.2.2.8 تشكيلة الدورة الكاملة

تحدّد البنية SessionConfig المعرّفة أدناه المعلومات الكاملة للتحكم بالتشكيلة المتعلقة بدورة **فتحة AS** تعمل بأسلوب فك التجفير أو التجفير. وهي تشمل في حالة **فتحة AS** تعمل بأسلوب التجفير المعلومات المتعلقة بالتشكيلة اللازمة لفك تجفير لاحق.

```
typedef struct SessionConfig {
    EncryptConfig   encryptConfig; /* configuration for encryption */
    DecryptConfig   decryptConfig; /* configuration for decryption */
} SessionConfig;
```

وتحدد الشفرة بلغة C أدناه والجدول 8-7 بنية الحالة cpsEciRootState التي تعرّف الحالة الجذرية **للسطح البيني ECI** اللازمة للتحقق من صلاحية **سلاسل شهادات السطح البيني ECI**.

```
typedef struct EciRootState {
    uchar           rootVersion;
    uint            rlVersion:24;
} EciRootState;

EciRootState cpsEciRootState; /* contains the minimum value from the CPS */
```

الجدول 8-7 – وصف حقول البنية EciRootState

الحقل	الوصف
rootVersion	صيغة **الشهادة الجذرية للسطح البيني ECI**
rlVersion	صيغة **قائمة الإبطال** المطبقة مع **الشهادة الجذرية**
ملاحظة – تطبق البنية EciRootState عادة كحد أدنى (قيمة دنيا) يمكن السماح به للصيغة الجذرية **للسطح البيني ECI** وصيغة **قائمة الإبطال** عند تحميل المعلومات المتعلقة **بالمضيف ECI** و**الوسيط ECI**.	

وتعرّف الدالة التالية التي تتحقق مما إذا كانت الحالة cpsEciRootState كافية لمتابعة حساب أحد المفاتيح:

```
bool cpsEciRootStateOk(uint slotId, uint sessionId) {
  if (ss[slotId].slotMode == SlotModeDecr)
    return
        (cpsEciRootState.rootVersion >=
         ss[slotId].se[sessionId].config.decryptConfig.minEciRootState.rootVersion)
     && (cpsEciRootState.rlVersion >=
         ss[slotId].se[sessionId].config.decryptConfig.minEciRootState.rlVersion);

  if (ss[slotId].slotMode == SlotModeEncr)
    return
        (cpsEciRootState.rootVersion >=
         ss[slotId].se[sessionId].config.encryptConfig.minEciRootState.rootVersion)
     && (cpsEciRootState.rlVersion >=
         ss[slotId].se[sessionId].config.encryptConfig.minEciRootState.rlVersion);

  /* following should not occur */
  return false;
}
```

الشروط المسبقة:

- تعطى قيمة أولية لمعرّف الهوية slotId **للفتحة AS**.

6.2.2.8 حالة مفتاح الدورة العشوائي

لكل دورة فك تجفير أو تجفير مرتبطة **بالفتحة AS**، تخزن **الفتحة AS** معلومات تتعلق بحالة المفتاح العشوائي كما هي معرفة في الشفرة بلغة c أدناه وموصوفة في الجدول 8-8.

```
typedef struct RkState {
    SymKey      rkCurrent;
    SymKey      rkNext;
    ulong       limitCounter;
} RkState;
```

الجدول 8-8 – وصف حقل الحالة المتعلقة بالفتاح العشوائي RkState

الحقل	الوصف
rkCurrent	مفتاح عشوائي حالي يستخدم لإدراجه في **سلم المفاتيح** لحساب كلمة التحكم.
rkNext	القيمة التالية للمفتاح العشوائي التي يتعين إدراجها في **سلم المفاتيح** لحساب كلمة التحكم.
limitCounter	عدّاد يدل على حالة استعمال المفتاح الحالي بوحدات متصلة بالقيمة الحدية المطبقة على المفتاح. تحصي القيمة الوحدات المتبقية التي لا يزال من الممكن تجفيرها أو فك تجفيرها بناء على كلمة تحكم محسوبة مع الحقل rkCurrent.

يزاد الحقل **limitCounter** لدى تطبيقه على كلمة التحكم.

ملاحظة – يمكن لعمليات التنفيذ أن تنفذ العدّاد على نحو فعال كجزء من **مسار فيديوي آمن**.

7.2.2.8 حالة الاستيراد/التصدير

يوجد لكل دورة تجفير دورة فك تجفير واحدة مرتبطة بها تستورد منها المحتوى الذي يتعين إعادة تجفيره. ويجب ان يكون الاستيراد ممكناً في وقت واحد لمجموعتي تصدير (على الأقل) في دورات التصدير، ما يسمح بتحول سلس.

ويمكن جمع دورتين من دورات فك التجفير معاً. ويسمح ذلك بدمج تدفقات فرعية مختلفة تتطلب كلمات تحكم أخرى (تحسب بواسطة **الفتحة AS** نفسها) ضمن تدفق مركب واحد مع مجموعة من **خصائص المحتوى** قبل تحويلها إلى نواتج معيارية صناعية أو تصديرها. وكجزء من عملية الدمج، يتحقق **نظام الأمن المعزّز** من أن **خصائص المحتوى** المتعلقة بالتدفق المدمج متساوية.

ملاحظة – إن مقارنة **خصائص المحتوى** يمكن أن تشمل أيضاً معرف هوية مجموعة التصدير، الذي يضمن أن معالجة سلسلة التصدير اللازمة لكلا الدورتين المقترنتين متساوية.

وبالاقتران مع حالة المفتاح العشوائي، توجد في **الفتحة AS** حالة حدود الدورة. وتحدد أدناه حالة الدورة 'se' بلغة c. وترد مواصفات الحقل في الجدول 8-9.

```
#define MaxExpGroupIds 2

typedef struct ImportExportState {
    int         importSlotId;
    int         importSession;
    uchar       expGrpId[MaxExpGrpId];
    bool        importPermitted[MaxExpGrpId];
    RkState rkState;
} ImportExportState;

#define ImportNone -1
```

الجدول 8-9 – تعريف بنية الحالة ImportExportState

الحقل	الوصف
importSlotId	لا يطبق إلا إذا كانت **الفتحة AS** تعمل بأسلوب التجفير. وتتمثل القيمة برقم الفتحة التي يستورد منها المحتوى ("فتحة الاستيراد"). والقيمة المبدئية هي -1.
importSession	لا يطبق إلا إذا كانت **الفتحة AS** تعمل بأسلوب التجفير. وتتمثل القيمة برقم الدورة في فتحة الاستيراد التي يستورد منها المحتوى. والقيمة المبدئية هي -1.
expGrpId[eid]	لا يطبق إلا إذا كانت **الفتحة AS** تعمل بأسلوب التجفير. وهو معرّف هوية مجموعة التصدير في الفتحة AS القائمة بالتصدير والتي يستورد منها المحتوى. والقيمة 0x00 محجوزة.
importPermitted[eid]	لا يطبق إلا إذا كانت **الفتحة AS** تعمل بأسلوب التجفير. يضبط على القيمة "صح" إذا كان الحقل **expGrpId[eid]** مسموحاً من **الفتحة AS** القائمة بالتصدير. والقيمة المبدئية هي خطأ.
rkState	حالة مفتاح الدورة العشوائي لهذه الدورة.

يلغي **نظام الأمن المعزز** دورة الاستيراد (يضبط الحقل ImportPermitted المقابل على خطأ) إذا أعيد ضبط أو استهلال دورة فك التجفير المقابلة. ويعيد **نظام الأمن المعزز** ضبط جميع دورات **الفتحة AS** عند إعادة ضبطها أو استهلالها.

3.2.8 الاستيقان من خصائص المحتوى

يزود **الوسطاء ECI** الذين يقومون بوظائف فك التجفير **المضيف ECI** بقيم خصائص المحتوى من خلال السطح البيني لبرمجة التطبيقات (API) الخاص بخصائص المحتوى. ويُدخل **المضيف ECI** هذه القيم في **نظام الأمن المعزز** إلى جانب البيانات اللازمة لحساب كلمة التحكم المتعلقة بالمحتوى المستخدم. ويضمن **نظام الأمن المعزز** الإنفاذ الصحيح **لخصائص المحتوى** ويتحقق من صلاحية **خصائص المحتوى** باستخدامها لحساب الدخل C **لمجموعة سلم المفاتيح** في **نظام الأمن المعزز**.

وتَستخدم **المخدّمات الصغيرة خصائص المحتوى** التي انتقلت إلى **نظام الأمن المعزز** أو إلى **الوسيط ECI** و/أو عولجت فيه باستخدام نفس الآلية التي وردت أعلاه لحساب الدخل C **لسلم المفاتيح**. وتقوم **الفتحات AS** المستخدمة بأسلوب التجفير بمقارنة **خصائص المحتوى** التي وفرها **المخدم الصغير** بتلك المحالة من مورد فك التجفير طبقاً لإعدادات تشكيلة **المخدم الصغير**. ويتوقف التجفير فوق اكتشاف عدم تطابق.

ولأغراض الاستيقان والتحقق، يتم جمع **خصائص المحتوى** في تتابع من البايتات على مرحلتين. وتجمع المرحلة الأولى الحقول الأصغر الثابتة الطول لخصائص المحتوى في الحقل *field1*. ويتحكم البايت fieldControl بوجود حقول بحجم البايت لخصائص المحتوى من أجل الاستيقان. وفي المرحلة الثانية تجمع الحقول الأطول لخصائص المحتوى في تتابع البايتات *field2*. ويرتب *field1* الحقلان و*field2* في تسلسل يمثل دخل دالة الاختزال التي تكثّف جميع الحقول في قيمة مؤلفة من 128 بتة وتضعها في الدخل C **لسلم المفاتيح**. ويعرض الجدول 8-10 بنية الحقل field1.

الجدول 8-10 – تعريف بنية الحقل field1

الاسم	النوع	رقم البايت	الوصف
fieldControl	FieldControl	0,1	يحدد هذا الحقل قيمة مؤلفة من 16بتة حيث تكون البتات الأقل دلالة في البايت 0. انظر الجدول 8-11.
basicUri	byte	2	تقابل قيمة هذا الحقل مواصفة نوع BasicUri، التوصية [ITU-T J.1012]، الجدول 1-1.5.2.8.9.
outputControl	byte [2]	4-3	تقابل قيمة هذا الحقل مواصفة متجه التحكم في الخرج، التوصية [ITU-T J.1012]، الجدول 1-1.6.2.8.9.
standardUri	byte [3]	7-5	تقابل قيمة هذا الحقل مواصفة URI المعيارية، التوصية [ITU-T J.1012]، الجدول 1-1.3.2.8.9.
exportGroup	byte	8	يفسّر على أنه عدد صحيح غير جبري يمثل معرّف هوية مجموعة التصدير التي تطبق على المحتوى. ويفسّر **المضيف ECI** القيمة 0 على أن التصدير غير مسموح؛ والقيم من 0x80 إلى 0xFF محجوزة.
parentalAuth	byte	9	يقابل الحقل ParCond.basicCondition كما هو معرف في التوصية [ITU-T J.1012]، الفقرة 1-1.8.2.8.9، مع ضبط البتات [5..0] على 0b000000.
Reserved	byte[6]	15-10	يقوم **المضيف ECI** بضبط البايتات على القيمة 0x00 عملاً بهذه التوصية.

الجدول 8-11 – تعريف بنية الحقل FieldControl

الاسم	البتة (البتات)	الوصف
bit-<n>	16-2	تتحكم هذه البتة في التحقق من صلاحية البايت <n> في الحقل field1. فإذا كانت القيمة 0b1 فهذا يدل على أن قيمة البايت <n> متحقق منها وتساوي الحقل المبين، وإذا كانت القيمة 0b0 فهذا يدل على أن قيمة البايت <n> غير متحقق منها وأن القيمة 0x00 تستخدم بدلاً منها للبايت <n> في الحقل field1. وتضبط البتة 2 على القيمة 0b1 عند استعمالها كدخل لحساب كلمة التحكم في فك التجفير. يضمن ذلك أن basicUri مستيقن منه دائماً بالنسبة للقيمة التي استخدمت وقت تجفير المحتوى.
Field2ctrl	1-0	تدل القيمة 0b00 على أن الحقل field2 غير موجود. وتدل القيمة 0b01 على أن الحقل موجود وأنه يستخدم التجفير كما هو محدد أعلاه. والقيمتان 0b10 و0b11 محجوزتان ويجب عدم استعمالهما.

يَستخدم تعريف الحقل Field2 بنية وسم أو طول أو قيمة مع حقل للطول الإجمالي لضمان السلامة العامة. ويرد أدناه في هذه الفقرة تعريف لبنية الحقل Field2.

وتقوم الوظيفة computeField1Decrypt بحساب منطق الاختيار في الخطوة التالية من الاستيقان.

```
void computeField1Decrypt(uchar field1[16], uchar result1[16]) {
    int i;
    ushort fieldControl = field1[0] + field1[1]<<8;

    result1[0] = field1[0];
    result1[1] = field1[1];
    for (i=2; i<16; i++)
            if (fieldControl>>i & 0b1)
                result1[i] = field[i];
            else
                result1[i] = 0x00;
}
```

وتقوم **الفتحة AS** التي تعمل بأسلوب التجفير بحساب دخل الاستيقان من خصائص المحتوى الذي يرمز إليه بالرمز result1 وأحد حقول cpMask لمقارنة بايتات الحقل field1 بالحقل field1 الذي ينتمي إلى محتوى دورة الاستيراد، وذلك على النحو التالي:

```
    void computeField1Encrypt(
                    uchar msField1[16],/* field1 for CP from Micro Server Client */
result1[16],            /* result CP for authentication in computing CW */
                    ushort cpMask,              /* result mask for comparing msField1 to client's
                                                   version of CP field1 */
                    EncryptConfig ssEncrypt /* encryption configuration of the AS slot */
    ) {
        int i;
        uchar cp[16]; /* CP value to be computed */

        /* set control bytes of content properties */
        cp[0] = ssEncrypt.defaultCp[0];
        cp[1] = ssEncrypt.defaultCp[1];
        mask = 0x0000;

        /* process the contPropControl rules to compute cp */
        for (i=2; i<16; i++) {
                switch (ssEncrypt.contPropControl>>(2*i) && 0b11) {
                    case CpCtrlCopy: /* shall be copied from import Client */
                        if (i==2) { /* basic URI byte */
                            /* process basicUriTrfr */
                            switch (ssEncrypt.basicUriTrfr) {
                            case BasicUriTrfrNoChange:
                                cp[i]= msField1[i];
                                break;
                            case BasicUriTrfrNoMoreCopy:
                                if ((clField1[2]&0b11) == RedistributionProtected)
                                    cp[i]= (msField1[1] & 0xFC) + ViewOnly;
                                else
                                    cp[i]= msField1[i];
                                break;
                        } else {   /* all other CP bytes */
                            cp[i]= msField1[i];
                        }
                        cpMask += 1<<i; /* msField1 byte i to be compared to imp client */
.                       break ;
                    case CpCtrlDef: /* shall be set to default CP from configuration *?
                        cp[i] = ssEncrypt.defaultCP[i] ;
                        break ;
                    case CpCtrlMS: /* shall be defined my software Micro Client */
                        cp[i] = msField1[i];
                        break;
                }
        }

        /* compute input to authentication function same was as for decryption */
        computeField1Decrypt(cp, result1);
    }
```

والحقل Field2 هو تتابع منظم من البايتات معرّف على النحو التالي:

```
typdef struct Field2 {
    uint length;        /* number of bytes in content, shall be a multiple of 4 */
    byte content[]; /* content defined below */
} Field2;
```

يتضمن حقل المحتوى في بنية الحقل Field2 تتابعاً من البنى LargeProperty لكل واحدة منها وسم فريد. وتعرّف LargeProperty بالشفرة أدناه بلغة C:

```
typdef struct LargeProperty {
    uint  propertyTag;  /* see Table 8.2-12  */
    uint  length;       /* length of property field in bytes
    byte  property[];   /* contains the actual property value */
    byte  padding[];    /* additional bytes set to 0x00 to make LargeProperty a
                               multiple of 4 bytes large */
} LargeProperty;
```

وترد في الجدول 8-12 قيم الحقل largePropertyTag وتعاريف الحقول المقابلة.

الجدول 8-12 – قيم حقل الوسم largeProperty ومعناه

قيمة propertyTag	الخاصية
0x00000000	محجوز
0x00000001	يقابل **بيانات** المعلمات الخاصة بالرسالة **setDcrMarkBasic** على النحو المعرّف في التوصية [ITU-T J.1012]، الفقرة 5.7.2.8.9.
0x00000002	يقابل **بيانات** المعلمات الخاصة بالرسالة **setDcrMarkExt** على النحو المعرّف في التوصية [ITU-T J.1012]، الفقرة 6.7.2.8.9.
0x00000003	يقابل المعلمة **custURI** الخاصة بالرسالة **setDcrCustUri** على النحو المعرّف في التوصية [ITU-T J.1012]، الفقرة 1.4.2.8.9.
قيمة أخرى	محجوز لاستعمال لاحق

يمكن **لنظام الأمن المعزز** أن يرفض أي بيانات تتجاوز قدرته على معالجة الحقل *field2*.

ويتحقق **نظام الأمن المعزز** من اتساق أي معلمة من معلمات بيانات الحقل Field2 باستخدام عمليات التحقق التالية:

- طول البنى LargeProperty المكونة مساوٍ لطول حقل البنية Field2.
- بايتات التحشية لجميع بنى LargeProperty المكوّنة هي 0x00.

ويحسب الدخل C من البيانات المرتبطة **لسلم المفاتيح** من النتيجة result1 وfield2 وفقاً للشفرة التالية بلغة C:

```
void computeInputC(uchar result1[16], unchar *field2, uchar input_C[16])
{
    uchar hash2[16], hashIn[32];
    uint i, length;

    if (result1[0] & 0b11 == 0x00) {
        /* no field2 to be included */
        for (i=0; i<16; i++) hashIn[i] = result1[i];
        asHash(hashIn, 16, 128, input_C);
    } else if result[0] & 0b11 = 0x01) {
        /* field2 to be included for input-C */
        length = (Field2 *)field2->length + 4;
        asHash(field2, length, 256, hash2);
        for (i=0; i<16; i++) hashIn[i] = result1[i];
        for (i=0; i<32; i++) hashIn[16+i] = hash2[i];
        asHash(hashIn, 48, 128, input_C);
    }
}
```

حيث تمثل asHash دالة الاختزال المعرّفة في الفقرة 1.A والمتعلقة بتتابع البايتات في المعلمة الأولى، وطول تتابع البايتات في المعلمة الثانية، وطول النتيجة بالبتات في المعلمة الثالثة، والنتيجة في المعلمة الأخيرة.

ويجب أن تكون **المتانة** المتعلقة بحساب الاختزال الخارجي (الحساب المباشر) بنفس ارتفاع درجة متانة حساب الاختزال الداخلي على الأقل. ويعكس قياس **متانة** الاختزال الجهد المطلوب لخلق تباين بين أي من مدخلات دالّة الاختزال واستخدام هذه المدخلات كخصائص محتوى بالإضافة إلى معالجة دالة الاختزال و/أو مخرجاتها.

ومن الأمثلة على اختلاف مستويات **المتانة** في عمليتي الحساب لدالة الاختزال إمكانية إجراء حساب الاختزال الخارجي بواسطة مجموعة من التجهيزات المتينة بينما يمكن إجراء حساب الاختزال الداخلي بواسطة برمجيات قوية.

4.2.8 وظائف فتحة الأمن المعزز

1.4.2.8 لمحة عامة

يمكن **لنظام الأمن المعزز** أن يقوم بوظائف متنوعة بالنيابة عن **الوسطاء ECI** من خلال العمل عبر **المضيف ECI**. تشكل هذه الوظائف الواردة في التوصية [ITU-T J.1012] الأساس للسطح البيني لبرمجة التطبيقات (API) في نظام الأمن المعزز. ويقوم "حدث" ما بإبلاغ **الوسيط ECI** بواقعة تحصل بطريقة غير متزامنة. ولا يكون الرد ممكناً. وتعيّن جميع الوظائف الأخرى لتكون إما رسائل غير متزامنة أو متزامنة تستهل من جانب **الوسيط ECI**؛ وتبين قيمها المرجَعة حالة الردود. وترد هذه الوظائف في الجدول 8-13.

الجدول 8-13 – لمحة عامة على وظائف الأمن المعزّز

اسم الوظيفة	الوصف	الفقرة
reqAsInitSlot	استهلال **الفتحة AS**	2.4.2.8
reqAsAStartDecryptSession	بدء دورة فك تجفير في **الفتحة AS**	3.4.2.8
reqAsCoupleDecryptSession	جمع دورتي فك تجفير في دورة واحدة	3.4.2.8
reqAsDecoupleDecryptSession	فصل دورتين مجموعتين من دورات فك التجفير	3.4.2.8
reqAsStartEncryptSession	بدء دورة تجفير	3.4.2.8
callAsNextKeySession	تغيير إلى المفتاح العشوائي التالي	3.4.2.8
reqAsStopSession	وقف دورة	3.4.2.8
reqAsExportConnSetup	إعداد **وصلة تصدير** من دورة فك تجفير إلى دورة تجفير	4.4.2.8
reqAsExportConnEnd	إنهاء دورة تصدير قائمة	4.4.2.8
reqAsLoadLk1	تحميل مفتاح الوصلة الأعلى مستوى في **سلم المفاتيح** من أجل الدورة	5.4.2.8
reqAsComputeEncrCw	حساب كلمة التحكم في التجفير	6.4.2.8
reqAsComputeDecrCw	حساب كلمة التحكم في فك التجفير	7.4.2.8
reqAsComputeAkClient	حساب مفتاح الاستيقان لاستعماله من قبل **الوسيط ECI**	8.4.2.8
reqAsClientChalResp	استعمال مفتاح الاستيقان بالنيابة عن **الوسيط ECI**	8.4.2.8
reqAsAuthDecrConfig	الاستيقان من تشكيلة الدورة **بآلية الاستيقان** (أسلوب فك التجفير)	9.4.2.8
reqAsAuthEncrConfig	الاستيقان من تشكيلة الدورة ومعلمات التجفير **بآلية الاستيقان** (أسلوب التجفير)	9.4.2.8
reqAsLdUssk	تحميل المفتاح السري **لمخدم صغير**	10.4.2.8
reqAsMInikLk1	حساب رسالة الاستهلال **لوسيط صغير** لاتناظري	11.4.2.8
reqAsClientImageDecrKey	حساب مفتاح فك التجفير لصورة **الوسيط ECI**	12.4.2.8
getAsSlotRk	عرض مفتاح الفتحة العشوائي	13.4.2.8
getAsSessionRk	عرض مفتاح الدورة العشوائي	13.4.2.8
getAsSessionLimitCounter	عرض الوحدات المتبقية من مفتاح الدورة العشوائي	13.4.2.8
setAsSessionLimitEvent	إرسال حدث عند بلوغ القيمة الحدية للوحدات المتبقية	13.4.2.8
reqAsEventSessionLimit	رسالة حدث عند بلوغ القيمة الحدية	13.4.2.8
getAsClientRnd	الحصول على رقم عشوائي جديد لتطبيقات **الوسيط ECI**	13.4.2.8
getAsSC	الحصول على الحالة الراهنة لحقل التحكم في تخليط المحتوى في الدورة	9.9
reqAsEventCpChange	رسالة حدث بشأن تغير خصائص المحتوى في محتوى مستورد خلال دورة تجفير	9.9
setAsPermitCPChange	تفعيل/عدم تفعيل تغيرات خصائص المحتوى المستورد التي تؤثر على اختيار كلمة التحكم للتجفير في دورة تجفير	9.9
setAsSC	ضبط حقل التحكم في التخليط لمحتوى مجفر في دورة تجفير	9.9
reqAsEventSC	رسالة حدث بشأن تغير حقل التحكم في التخليط في الدورة	9.9

وتتضمن شبه الشفرة في الفقرات الفرعية لهذه الفقرة شفرات أخطاء بشكل قيمة مرجَعة لهذه الوظائف. وتحدد قيم شفرة الأخطاء في الفقرة 15.4.2.8 بما في ذلك وصفها اللفظي.

2.4.2.8 استهلال تشغيل الفتحة AS

في وقت التحميل يقوم **المضيف ECI** بحجز **فتحة AS** في **نظام الأمن المعزز** بالنيابة عن كل وسيط من **الوسطاء ECI** المقرر تحميلهم. ويستدعي **المضيف ECI** الوظيفة reqAsInitSlot كما هي معرفة أدناه. ويجب أن تضبط جميع المعلومات المتعلقة بحالة **الفتحة AS** على حالتها المبدئية؛ ويجب إعادة ضبط أي وصلة من **وصلات التصدير**. ويقوم **المضيف ECI** بتحميل **الوسيط ECI** باستخدام أداة التحميل (انظر الفقرة 11). ويجب أن تعبّر الوظيفة *POClRLVnr الخاصة* ***بالفتحة AS*** *عن رقم أدنى صيغة* ***لقائمة إبطال*** *شهادات عمليات المنصة المستخدمة للتحقق من صلاحية صورة الوسيط. ويتم التحقق من هذه القيمة عندما يستهل* ***الوسيط ECI*** *دورة ما.*

```
int  reqAsInitSlot(uint slotId, ECI_Certificate_Chain popkChain,
       uint slotVersion, slotMode)
```

الدلالات:

يجب أن تضبط جميع المعلومات المتعلقة بمعرّف هوية **الفتحة AS (slotId)** على الحالة المبدئية؛ ويجب إعادة ضبط أي وصلة من **وصلات التصدير**.

ويتطلب تحميل المفتاح POPK تزويد النظام الفرعي لمعالجة الشهادات (CPS) بسلسلة المعالجة. وتحدد الفقرة 4.10 قواعد معالجة سلاسل المفاتيح POPK. وتعرّف سلسلة شهادات السطح البيني ECI الفقرة 1.4.5 من التوصية [ITU-T J.1012]. ويجب تنفيذ الشفرة بلغة C فور نجاح التحقق من صلاحيتها:

```
/* initialise the slot state */
ss[slotId].popk = /* validated value of popk returned by CPS */;
ss[slotId].POClRLVnr = /* value used for client image verification */;
ss[slotId].version = slotVersion;
ss[slotId].slotMode = slotMode;
ss[slotId].configAuthMode = ConfigAuthModeNone;
ss[slotId].rkSlot = rnd128();
return ErrOk;
```

تُرجع الوظيفة rnd128() عدداً عشوائياً مؤلفاً من 128 بتة كما هو معرّف في الفقرة 3.A على شكل صفيف من 16 uchar.

3.4.2.8 التحكم في الدورة والمفتاح العشوائي للفتحة AS

تدعم **الفتحة AS** حالات للدورة تختلف باختلاف الدورات المتزامنة. وتقوم الوظائف التالية ببدء وإيقاف الدورات الخاصة بفتحة ما:

```
int reqAsAStartDecryptSession(uint slotId, ushort mh, PubKey spk,
       SessionConfig config, uint *sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (ss[slotId].slotMode != slotModeDecr) return ErrSlotMode;

/* check if a valid client revocation list was used */
if (config.decryptConfig.clientVersion >
    ss[slotId].clientPOClRLVnr) return ErrRevocEnforce;

/* locate any free sessionId; any algorithm is ok */
int i=0;
while (i<NSESSIONS && ss[slotId].se[i].active) i++;
if (i==NSESSIONS) return ErrNoMoreSessions;
/* i contains a non-active session administration block */
*sessionId = i;

/* initialise session state */
```

```
ss[slotId].se[i].active = true;
ss[slotId].se[i].mh = mh;
ss[slotId].se[i].coupledSessionId = -1;
ss[slotId].se[i].importPermitted = false;
ss[slotId].se[i].spk = spk;
ss[slotId].se[i].config = config;
ss[slotId].se[i].rkState.rkCurrent = rnd128();
ss[slotId].se[i].rkState.rkNext      = rnd128();
ss[slotId].se[i].rkState.limitCounter =
            limitValue(config.decryptConfig.rkDecrMode.limit);

if (!cpsEciRootStateOk(sdlotId,i)){
    ss[slotId].se[i].active = false;
    return ErrRevocEnforce;
}

return ErrOk;
```

الشروط المسبقة:

- أن يستهلّ بنجاح تشغيل **الفتحة AS**.

ملاحظة – تسمح المعلمة mh (مشغّل الوسائط) **للمضيف ECI** بتحديد دورة فك التحفير في الأمن المعزز المرتبطة بدورة فك تحفير المحتوى التي باشر بها. ولا تستخدم من **نظام الأمن المعزز** نفسه.

وتتوفر الوظيفة coupleDecrypSessions من أجل جمع دورتين مستهلتين. تقرن الدورة الثانية بالأولى؛ وتصبح الأولى المشغل الرئيسي للمحتوى الموحّد.

```
int reqAsCoupleDecryptSession(uint slotId, uint sId1, uint sId2)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (ss[slotId].slotMode != slotModeDecr) return ErrSlotMode;
if (!ss[slotId].se[sId1].active) return ErrParam2;
if (!ss[slotId].se[sId2].active) return ErrParam3;
if (ss[slotId].se[sId1].coupledSessionId != -1) return ErrSession1Coupled;
if (ss[slotId].se[sId2].coupledSessionId != -1) return ErrSession2Coupled;

se[slotId][sId1] = sId2;
/* the Secure Video Path is informed on the session coupling &/

return ErrOk;
```

الشروط المسبقة:

- أن يستهلّ بنجاح تشغيل دورتي **الفتحة AS**.

ويمكن استدعاء الوظيفة التالية لفصل دورة مقترنة:

```
int reqAsDecoupleDecryptSession(uint slotId, uint sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (ss[slotId].slotMode != slotModeDecr) return ErrSlotMode;
if (!ss[slotId].se[sessionId].active) return ErrParam2;
if (se[slotId][sessionId].coupledSessionId == -1)
    return ErrSessionNotCoupled;

ss[slotId].se[sessionId].ies.coupledSessionId = -1;
/* the Secure Video Path is informed on the session decoupling */

return ErrOk;
```

الشروط المسبقة:

- أن تكون الدورتان مجموعتان مسبقاً.

فيما يلي الوظيفة التي تطلق دورة التجفير:

```
int reqAsStartEncryptSession(uint slotId, ushort mh, uint importSlotId,
        int importSessionId, PubKey spk, SessionConfig config,
        uint nEncr, PubKey encrSpk[MaxSpkEncr],
        PubKey encrPopk[MaxSpkEncr], ulong encrCwUri, uint *sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
If (ss[slotId].slotMode != slotModeEncr) return ErrSlotMode;
if (0 > nEncr || nEncr >= MaxEncr) return ErrParam4;

/* locate free sessionId; any algorithm is ok */
int i=0;
while (i<NSESSIONS && ss[slotId].se[i].active) i++;
if (i==NSESSIONS) return ErrNoMoreSessions;
/* i contains a non-active session administration block */

/* check if a valid client revocation list was used */
if (config.encryptConfig.microServerVersion >
    ss[slotId].clientPOClRLVnr) return ErrRevocEnforce;

*sessionId = i;

/* initialise session state information */
ss[slotId].se[i].active=true;
ss[slotId].se[i].mh = mh;
ss[slotId].se[i].spk = spk;
ss[slotId].se[i].config = config;
ss[slotId].se[i].encrCwUri = encrCwUri;

int j;
for (j=0; j<nEncr; j++) {
  ss[slotId].se[i].encrSpk[j] = encrSpk[j];
  ss[slotId].se[i].encrPopk[j] = encrPopk[j];
}

/* initialise random key state */
ss[slotId].se[i].rkState.rkCurrent = rnd128();
ss[slotId].se[i].rkState.rkNext    = rnd128();
ss[slotId].se[i].rkState.limitCounter =
            limitValue(config.encryptConfig.rkEncrMode.limit);

/* initialise import state */
ss[slotId].se[i].importSlotId = importSlotId;
ss[slotId].se[i].importSession = importSessionId;

if (!cpsEciRootStateOk(sdlotId,i)){
    ss[slotId].se[i].active = false;
    return ErrRevocEnforce;
}

return i;
```

الشروط المسبقة:

- أن يستهلّ بنجاح تشغيل **الفتحة AS**.

يمكن **للمضيف ECI** تحديث حالة المفتاح العشوائي (جعل الحالة التالية هي الحالة الراهنة) لدورة ما باستخدام الوظيفة التالية:

```
int callAsNextKeySession(uint slotId, uint sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (!ss[slotId].se[sessionId].active) return ErrNoSuchSession;

ss[slotId].se[sessionId].rkCurrent = ss[slotId].se[sessionId].rkNext;
ss[slotId].se[sessionId].rkNext = rnd128();
if (ss[slotId].slotMode == SlotModeEncr)
    se[slotId][sessionId].limitCounter =
       limitValue(
          ss[slotId].se[sessionId].config.encryptConfig.rkEncrMode.limit)
else if (ss[slotId].slotMode == SlotModeDecr)
    se[slotId][sessionId].limitCounter =
       limitValue(
          ss[slotId].se[sessionId].config.decryptConfig.rkDecrMode.limit);

return ErrOk;
```

الشروط المسبقة:

- أن تستهلّ بنجاح دورة **الفتحة AS**.

عندما يعمل **مسار فيديوي آمن** بأسلوب تدفق النقل فإنه يوعز إلى **الوسيط ECI** بأن كلمة التحكم قد تغيرت من الحالة الراهنة إلى الحالة التالية (انظر التوصية [ITU-T J.1012]). ويمكن **للوسيط ECI** أن يستخدم هذه الرسالة لإطلاق عملية حساب كلمة التحكم التالية.

ويمكن **للمضيف ECI** أن يوقف الدورة أن ينهي بالتالي أي **وصلات تصدير** عالقة من تلك الدورة بواسطة الوظيفة التالية:

```
int reqAsStopSession(uint slotId, uint sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
int i, j;

ss[slotId].se[sessionId].active = false;

/* decouple from any coupled decryption sessions */
for (j=0; j<NSESSESIONS; j++)
    if (ss[slotId].se[j].coupledSessionId == sessionId)
        ss[slotId].se[j].coupledSessionId = -1;
        /* the Secure Video Path is informed of decoupling */

/* cancel all export sessions */
if (ss[slotId].slotMode == SlotModeDecr)
    for (i=0; i<NSLOTS; i++)
        for (j=0; j<NSESSIONS; j++)
            if (ss[i].se[j].importSlot == slotId &&
                ss[i].se[j].importSession == sessionId)
            {
                for (k=0; k<MaxExpGrpId; k++)
                    ss[i].se[j].importPermitted[k]= false;
                ss[i].se[j].importSlotId= -1;
                ss[i].se[j].importSession= -1;
            }

return ErrOk;
```

الشروط المسبقة:

- أن تستهلّ بنجاح دورة **الفتحة AS**.

4.4.2.8 التحكم في تصدير الفتحة AS

تسمح **آلية الاستيقان** الخاصة بالتصدير **للمضيف ECI** باستحداث **وصلة تصدير** من دورة **الفتحة AS** في **وسيط ECI** يقوم بفك التجفير إلى دورة **الفتحة AS لمخدم صغير**، ما يسمح بالتالي بنقل المحتوى من **الوسيط ECI** القائم بفك التجفير إلى **المخدم الصغير**. ويستخدم **نظام الأمن المعزز النظام الفرعي لمعالجة الشهادات** لمعالجة التصدير المطلوب، وسلاسل الاستيقان من الاستيراد والتصدير باستخدام المفتاح POPK الخاص بدورات التصدير في **الفتحة AS** والوظيفة minClientVersion كأساس للتحقق من صلاحية سلسلة التصدير والاستيراد اللاحق. وتتمثل النتيجة النهائية في التحقق الإيجابي من صحة عنصر **وصلة التصدير** الخاص بمعرف مجموعة التصدير، أو رفض الوصلة. وتستحدث الوصلة الفعلية من دورة (تصدير) إلى دورة استيراد.

```
int reqAsExportConnSetup(uint slotId, uint sessId, uint impSlotId,
        uint impSessId, uint grpIndx, CertSerialChain expCh,
        CertSerialChain impCh, CertSerialChain auth[])
```

الدلالات:

expCh هي سلسلة التصدير من المفتاح POPK إلى شهادة TPEGC أو شهادة ESC. وتمثل ImpCh سلسلة الاستيراد من TPEGC إلى ESC.

ملاحظة – يمكن أن تبقى السلسلة impCh فارغة. و[]auth هي تتابع من سلاسل الاستيقان من التصدير اللازمة للاستيقان المشترك من أجزاء من سلسلة الاستيراد.

ويرد في الفقرة 2.4.2.5.9 من التوصية [ITU-T J.1012] تعريف بنية السلسلة CertSerialChain.

تتحقق **الفتحة AS** أولاً من السلسلة impCh باستخدام سلاسل الاستيقان من التصدير []auth و**المفتاح الجذري للسطح البيني ECI** الذي تم تثبيته ورقم صيغة **قائمة الإبطال**.

بعد ذلك تطلب **الفتحة AS** من **النظام الفرعي لمعالجة الشهادات** تجهيز سلسلة التصدير والاستيراد باستخدام المفتاح POPK والمفتاح **POPK** لسجلات حالة الأمن المعزز والوظيفة **ExportRlVersion** كجذر. ويخزن معرّف هوية أول **شهادة** في سلسلة التصدير في الوظيفة expGrpId.

وعند نجاح عملية الاستيقان، يضاف أحد عناصر التصدير إلى حالة دورة **الفتحة AS**، ويتضمن معرّف هوية مجموعة التصدير ومعرّف هوية الفتحة بالإضافة إلى معرّف هوية **دورة الوسيط ECI** الذي تم الاستيقان منه. وتنفّذ الشفرة التالية بلغة C لمعالجة وصلة الاستيراد. ويمكن حساب الاستيقان لمعرّفي هوية مجموعتي التصدير بحيث يمكن تحقيق تغير سلس من مجموعة تصدير إلى أخرى في خصائص المحتوى.

```
/* the CPS delivers the following variables on successful processing of the
   Export import chains */
PubKey impSpk;           /* the spk of the importing system */
uint   impConfigVersion; /* the config. Version nr of the export system */
uint   expGrpId;         /* the export group for which the export connection     is valid */


/* check if potential import slot is in decent state */
if (!( ss[impSlotId].slotMode == SlotModeEncr &&
       ss[impSlotId].se[impSessId].active &&
       ss[impSlotid].se[impSessId].spk == impSpk
       ss[impSlotId].se[impSessId].encryptConfig.microServerVersion >=
         impConfigVersion
   ) ) return ErrExportSlotBadState;
}

/* check if another import connection already exists */
if (ss[impSlotId].se[impSessId].ies.importSlotId != ImportNone)
  return ErrExportOngoing;
/* Set the import/export state of the import-session to reflect the export connection */
ss[impSlotId].se[impSessId].ies.importSlotId = slotId;
ss[impSlotId].se[impSessId].ies.importSession = sessId;
ss[impSlotId].se[impSessId].ies.expGrpId[grpIndx] = expGrpId;
ss[impSlotId].se[impSessId].ies.importPermitted[grpIndx] = true;
return ErrOk;
```

الشروط المسبقة:

- أن تستهلّ بنجاح دورة **الفتحة AS**.

بعد إعداد **وصلة تصدير** معينة يمكن إنهاؤها أيضاً من جانب الاستيراد (ما يؤدي إلى وقف دورة التجفير بشكل فعال):

```
int reqAsExportConnEnd(uint slotId, uint sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (!(ss[slotId] != SlotModeEncr)) return ErrImportSlotBadState;
if (!(ss[slotId].se[sessionId].active)) return ErrParam2;
if (ss[slotId].se[sessionId].ies.slotId == -1) return ErrNoExport;

ss[slotId].se[sessionId].ies.importSlotId = -1;
ss[slotId].se[sessionId].ies.importSession = -1;
for (int i=0; i< MaxExpGrpId; i++)
  ss[slotId].se[sessionId].ies.importPermitted[i] = false;
return ErrOk;
```

الشروط المسبقة:

- أن تكون دورة **الفتحة AS** معدّة للاستيراد.

5.4.2.8 استهلال سلم المفاتيح LK1

بغية إجراء عمليات آلية **سلم المفاتيح** في **فتحة AS** معينة، يستطيع **المضيف ECI** تحميل مفتاح الوصلة الأعلى مستوى LK1 من أجل الحسابات اللاحقة **لخرج سلم المفاتيح**.

```
int reqAsLoadLk1(uint slotId, uint sessId, InputV inputV,
             ulong spkUri, uchar spkIndx)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (ss[slotId].slotMode == SlotModeEncr) spkIndx = 0;
if (spkIndx >= 16) return ErrParam5;
/* check if spkUri in set_1 */
if ((spkUri>>spkIndx & 0b1) != 0b1) return ErrSpkUriViolation;
if (!ss[slotId].se[sessId].active) return ErrParam2;
if (spkIndx==0 && ss[slotId].slotMode==SlotModeDecr &&
    ss[slotId].se[sessId].config.decryptConfig.spk0NoDecrypt)
  return ErrSpk0NoDecrypt;

ss[slotId].se[sessId].spkUri = spkUri;
ss[slotId].se[sessId].spkIndx = spkIndx;

if (ss[slotId].slotMode == slotModeEncr &&
    ss[slotId].se[sessId].config.encryptConfig.asymKlMode
){
   ss[slotId].lk1= rnd128();
   return ErrOk;
}

ss[slotId].se[sessId].lk1 =
    blockV_blockC_keyladder(inputV,ss[slotId].se[sessId].spk);
return ErrOk;
```

الشروط المسبقة:

- أن تستهلّ دورة **الفتحة AS**.

6.4.2.8 حساب كلمات التحكم المتعلقة بالتجفير

يمكن حساب كلمات التحكم حالما يتم إعداد حقل الحالة lk1 المتعلق **بالفتحة AS**. ويدل المؤشر cwIndx على كلمة التحكم المفردة أو المزدوجة التي تم حسابها. وقد تكون القيمة 0 (مزدوجة) أو 1 (مفردة)، ويجب أن تكون دائماً 0 لفك التجفير القائم على الملف.

```
int reqAsComputeEncrCw(uint slotId, uint sessId, ulong cwUri, uint nElk,
    SymKey elk[24], uchar XT[32], uint rkIndx, Field2 field2,
    uint cwIndx)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
PubKey spk[MaxSpkEncr+1], popk[MaxSpkEncr+1]; /* temporary variables */
SessionConfig config[MaxSpkEncr+1]; /* temporary variable */

/* basic consistency checks */
if (!ss[slotId].se[sessId].active) return ErrParam2;
if (ss[slotId].slotMode != SlotModeEncr) return ErrSlotMode;
if (ss[slotId].se[sessId].config.encryptConfig.rkEncrMode.mode==0b00) {
    if (nElk<2) return ErrParam4;
} else {
    If (nElk<3) return ErrParam4;
}

/* verify if the slot configuration has been authenticated */
if (ss[slotId].se[sessId].configAuthMode != ConfigAuthModeAk1)
    return ErrNoConfigAuth;

/* verify if the CPS ECI Host Root state is sufficient to proceed */
if (!cpsEciRootStateOk(slotId,sessId)) return ErrRevocEnforce;


/* check if random slot-session key has to be applied */
SymKey rkAppl; /* random key that may have to be applied */
if (rkIndx == 0) {
        rkAppl = ss[slotId].se[sessId].rkState.rkCurrent;
} else if (rkIndx == 1) {
        rkAppl = ss[slotId].se[sessId].rkState.rkNext;
} else {
        return ErrParam7;
}

/* insert random slot key and random session key if required */
if (ss[slotId].se[sessId].config.encryptConfig.rkKlMode) {
        elk[0] = ss[slotId].slotRk;
}
if (ss[slotId].se[sessId].config.encryptConfig.rkEncrMode.mode != RKModeNone) {
        if (nSpk < 3) return ErrNoSlotRkInsert;
        elk[nSpk-1] = rkAppl;
}

/* compute input-C, insert in key ladder */
uchar result1[16], seField1[16];
ushort cpMask;

computeField1Encrypt(elk[nElk-2], result1, cpMask,
ss[slotId].se[sessId].config.encryptConfig);
computeInputC(result1, field2, elk[nElk-2]);

/* use ARK with value 0 */
uchar ark[16] = (uchar){0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

/* define spk, popk and config inputs to key ladder; using slot's spk/popk in position 0 and a
replication of the slot configuration */

spk[0]  = ss[slotId].se[sessId].spk;
popk[0] = ss[slotId].popk;
config[0] = ss[slotId]. se[sessId].config;
int i;
int nSpk = slot[slotId]. se[sessId].config.EncryptConfig.nEncr + 1;
for (i=0; i<nSpk-1; i++) {
  spk[i+1]  = ss[SlotId]. se[sessId].encrSpk[i];
```

```
  popk[i+1] = ss[SlotId]. se[sessId].decrSpk[i];
  config[i+1] = ss[slotId]. se[sessId].config;
}

/* define spkUri values */
ulong spkUri = (0x1<<(nSpk+1)) - 1; /* all SPKs can be used for decoding
 keys */

/* perform the key ladder calculation */
bool asym = ss[slotId].se[sessId].config.encryptConfig.asymKlMode;
Secret SymKey cw =
  KeyLadder(ss[slotId].se[sessId].lk1, ss[slotId].se[sessId].encrCwUri,
            AcfCw1Mode, ark, popk, config XT, ss[slotId].spkUri, nSpk, spk,
            nElk,  elk, asym);

/* cw is sent to the encryption resource along with cwUri, msField1, cpMask and cwIndx */

return ErrOk;
```

الشروط المسبقة:

- أن يكون المفتاح LK1 قد تم تحميله.
- أن يكون قد تم الاستيقان من **الفتحة AS** عند الاقتضاء.

7.4.2.8 حساب كلمات التحكم المتعلقة بفك التجفير

يمكن حساب كلمات التحكم حالما يتم إعداد حقل الحالة lk1 المتعلق **بالفتحة AS**. ويدل المؤشر cwIndx على كلمة التحكم المفردة أو المزدوجة التي تم حسابها. وقد تكون القيمة 0 (مزدوجة) أو 1 (مفردة)، ويجب أن تكون دائماً 0 لفك التجفير القائم على الملف.

```
int reqAsComputeDecrCw(uint slotId, sessionId, ulong cwUri, uint nSpk,
    uint nElk, SymKey elk[24], PubKey spk[16], PubKey popk[16], SSConflig config[16], uchar
    XT[32], uint rkIndx, Field2 field2, uint cwIndx)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
/* basic consistency checks */
if (!ss[slotId].se[sessionId].active) return ErrParam2;
if (ss[slotId].slotMode != SlotModeDecr) return ErrSlotMode;
if (ss[slotId].se[sessionId].spkIndx >= nSpk) return ErrParam4;
if (ss[slotId].se[sessionId].config.decryptConfig.rkDecrMode.mode==0b00) {
    if (nElk<2) return ErrParam5;
} else {
    if (nElk<3) return ErrParam5;
}
uint si = ss[slotId].se[sessionId].spkIndx ;

/* verify if the slot configuration has been authenticated if so required */
if ( ss[slotId].se[sessionId].config.decryptConfig.akModeAuth &&
     ss[slotId].se[sessionId].configAuthMode != ConfigAuthModeAk1
   ) return ErrNoConfigAuth;

/* verify if the CPS ECI Host Root state is sufficient to proceed */
if (!cpsEciRootStateOk(slotId,sessionId)) return ErrRevocEnforce ;

/* ensure proper slot spk, popk and slotConfig are applied */
spk[si]= ss[slotId].se[sessionId].spk;
popk[si] = ss[slotId].se[sessionId].popk;

/* only authenticate the slot's decrypt configuration if required */
if ( ss[slotId].se[sessionId].config.decryptConfig.klModeAuth )
    ssConfig[si].decryptConfig = ss[slotId].ssConfig.decryptConfig;

/* in all cases authenticate the klModeAuth and akModeAuth fields */
config[si].decryptConfig.klModeAuth=
     ss[slotId].se[sessionId].config.decryptConfig.klModeAuth;
config[si].decryptConfig.akModeAuth =
     ss[slotId].se[sessionId].config.decryptConfig.akModeAuth;
```

```
/* check if random slot-session key may have to be applied */
SymKey rpAppl; /* random key that may have to be applied */
if (rkIndx == 0) {
        rpAppl = ss[slotId].se[sessionId].rkState.rkCurrent;
} else if (rkIndx == 1) {
        rpAppl = ss[slotId].se[sessionId].rkState.rkNext;
} else {
        return ErrParam11;
}

/* insert random slot key and random session key if required */
if (ss[slotId].se[sessionId].config.decryptConfig.rkKlMode) {
        elk[0] = ss[slotId].slotRk;
}
if (ss[slotId].se[sessionId].config.decryptConfig.rkDecrMode.mode != RKModeNone) {
        if (nSpk < 2) return ErrNoSlotRkInsert;
        elk[nSpk-2] = rpAppl;
}

/* compute input-C, i.e. elk[nElk-1] for content Property authentication */
/* veryfy basicUri control bit is set */
if (((elk[nElk-1][0]>>2)&0b1) != 0b1) return ErrBasicUriCtrl;
uchar result1[16];
computeField1Decrypt(elk[nElk-2],result1);
computeInputC(result1, field2, elk[nElk-2]);

/* use ARK with value 0 */
uchar ark[16] = (uchar){0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

/* perform the key ladder calculation */
Secret SymKey cw =
  KeyLadder(ss[slotId].se[sessionId].lk1, cwUri, AcfCw1Mode, ark,
            popk, ssConfig, XT, ss[slotId].se[sessionId].spkUri, nSpk,
            spk, nElk, elk, false);

/* cw is passed to the decryption resource session along with cwUri, result1 and cwIndx and the
sessions states media handle value */

return ErrOk;
```

الشروط المسبقة:

- أن يكون المفتاح LK1 قد تم تحميله.
- أن يكون قد تم الاستيقان من **الفتحة AS** عند الاقتضاء.

8.4.2.8 حساب akClient وتطبيقاته

تسمح **آلية الاستيقان** المتعلقة **بمجموعة سلم المفاتيح** بإجراء حساب آمن للمفاتيح المؤمنة لاستخدامها من جانب **الوسيط ECI** الذي يستعمل **آلية الاستيقان**:

```
int reqAsComputeAkClient(uint slotId, InputV inputV, uint nSpk,
    uchar spkIndx, PubKey spk[16], PubKey popk[16], SessionConfig akCnf[16],
    ulong spkUri,uchar XT[32], bool online)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
/* basic consistency checks */
if (ss[slotId].slotMode==SlotModeEncr) spkIndx = 0;
if (spkIndx >= 16) return ErrParam4;
/* check if spkUri in set_1 */
if ((spkUri>>spkIndx & 0b1) != 0b1) return ErrSpkUriViolation;
if (ss[slotId].slotMode == SlotModeEncr) {
  if (akCnf[spkIndx].encryptConfig.configVersion != 0x1) return ErrParam7;
  if (akCnf.encryptConfig.microServerVersion >
      ss[slotId].clientPOClRLVnr) return ErrRevocEnforce;
  if ((cpsEciRootState.rootVersion <
      akCnf[spkIndx].encryptConfig.minEciRootState.rootVersion)
   || (cpsEciRootState.rlVersion <
```

```
      akCnf[spkIndx].encryptConfig.minEciRootState.rlVersion))
    return ErrRevocEnforce;
}
if (ss[slotId].slotMode == SlotModeDecr) {
  if (akCnf[spkIndx].decryptConfig.configVersion != 0x1) return ErrParam7;
  if (akCnf.decryptConfig.minClientVersion >
      ss[slotId].clientPOClRLVnr) return ErrRevocEnforce;
  if ((cpsEciRootState.rootVersion <
      akCnf[spkIndx].decryptConfig.minEciRootState.rootVersion)
   || (cpsEciRootState.rlVersion <
      akCnf[spkIndx].decryptConfig.minEciRootState.rlVersion))
    return ErrRevocEnforce;
}
/* ensure proper slot spk and popk are applied */
popk[spkIndx] = ss[slotId].popk;

/* ensure proper ACF and ARK are applied */
uchar ark[16] ;
uchar acf[15] = acfAk1Mode ;
acf[1] = AkUseCl;
if (online) {
acf[1] += AkOnline;
ark = ss[slotId].slotRk;
} else {
    acf[1] += AlOffline;
    ark = {0} ;
}

/* perform the authentication mechanism */
ss[slotId].akClient =
      AuthMech(inputV,acf,ark,popk,akCnf,XT,spkUri,nSpk,spkIndx,spk);
return ErrOk;
```

الشروط المسبقة:

- أن يكون قد تم استهلال تشغيل الفتحة.

ومن أجل استخدام المفتاح AK **للوسيط ECI**، تعرّف الوظيفة التالية:

```
int reqAsClientChalResp(int slotId, uchar challenge[16],
    uchar *(response[16]));
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
*response = AuthMechResponse(ss[slotId].akClient, challenge);
return ErrOk;
```

الشروط المسبقة:

- أن يكون قد تم استهلال تشغيل الفتحة.
- أن يكون قد تم بنجاح حساب AkClient الخاص بالفتحة.

9.4.2.8 الاستيقان من تشكيلة دورة الفتحة AS

تسمح **آلية الاستيقان** المتعلقة **بمجموعة سلم المفاتيح** بالاستيقان من تشكيلة الدورة بواسطة مخدّم التجهيز. ويمكن **لمخدّم التجهيز** أن تصدر معلومات خارج الشبكة تتعلق بالاستيقان أو تطلب تنفيذ الاستيقان على الشبكة بإعداد AkOnline في حقل المراقبة الأمنية المعززة (ACF). وتتاح وظيفتان منفصلتان للاستيقان من فتحة فك التجفير وفتحة التجفير.

```
int reqAsAuthDecrConfig(uint slotId, uint sessId, InputV inputV,
  uint nSpk, uchar spkIndx, PubKey spk[16], PubKey popk[16], SSCnfg clCnf[16],
  ulong spkUri, uchar XT[32], bool online, uchar verifier[16])
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
/* basic consistency checks */
if (!ss[slotId].se[sessionId].active) return ErrParam2;
if (ss[slotId].slotMode!=SlotModeDecr) return ErrSlotMode;
if (spkIndx >= 16) return ErrParam5;
/* check if spkUri in set_1 */
if ((spkUri>>spkIndx & 0b1) != 0b1) return ErrSpkUriViolation;
if (spkIndx==0 && ss[slotId].slotMode==SlotModeDecr &&
    ss[slotId].se[sessId].config.decryptConfig.spk0NoDecrypt) return ErrSpk0NoDecrypt;

/* verify if the CPS ECI Host Root state is sufficient to proceed */
if (!cpsEciRootStateOk(slotId)) return ErrRevocEnforce;

/* ensure proper slot spk, popk and config are applied */
popk[spkIndx] = ss[slotId].popk;
spk[spkIndx] = ss[slotId].se[sessId].spk;
clCnf[spkIndx] = ss[slotId].se[sessId].config;

uchar ark[16];
uchar acf[15] = acfAk1Mode;
acf[1] = AkUseAS + AkConfigAuth;
if (online) {
        acf[1] = AkOnline;
        ark = ss[slotId].slotRk;
} else {
        acf[1] = AkOffline;
        ark = {0};
}

/* perform the authentication mechanism */
Secret SymKey ak =
      AuthMech(inputV,acf,ark,popk,clCnf,XT,spkUri,nSpk,spkIndx,spk);

uchar response[16] = AuthMechResponse(ak, verifier);

if (response == {0}) {
        ss[slotId].se[sessId].configAuthMode = ConfigAuthModeAk1;
        return ErrOk;
} else {
        ss[slotId].se[sessId].configAuthMode = ConfigAuthModeNone;
        return ErrSlotConfigAuthFail;
}
```

الشروط المسبقة:

- تحميل المفتاح LKI الخاص بدورة **الفتحة AS**

ويشمل الاستيقان في حالة التجفير التحقق من الحالة الخاصة بالتجفير.

```
int reqAsAuthEncrConfig(uint slotId, uint sessId, InputV inputV,
       uchar XT[32], bool online, uchar verifier[16])
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
PubKey spk[MaxSpkEncr+1], popk[MaxSpkEncr+1]; /* temporary variables */
SessionConfig config[MaxSpkEncr+1]; /* temporary variable */

/* basic consistency checks */
if ((ss[slotId.SlotMode != SlotModeEncr) return ErrSlotMode;

/* verify if the CPS ECI Host Root state is sufficient to proceed */
if (!cpsEciRootStateOk(slotId,sessId)) return ErrRevocEnforce;

/* define spk, popk and config inputs to key ladder; using slot's spk/popk in position 0 and a
replication of the slot configuration */

spk[0]  = ss[slotId].se[sessId].spk;
popk[0] = ss[slotId].popk;
config[0] = ss[slotId].se[sessId].config;
```

```
int i;
int nSpk = slot[slotId]. config.EncryptConfig.nEncr + 1;
for (i=0; i<nSpk-1; i++) {
  spk[i+1] = ss[SlotId].encrSpk[i];
  popk[i+1] = ss[SlotId].encrPopk [i];
  config[i+1] = ss[slotId].se[sessId].slotConfig;
}

/* define spkUri values */
ulong spkUri = (0x1<<(nSpk)) - 1;
                      /* all SPKs can be used for decoding content */

uchar ark[16];
uchar acf[15] = acfAk1Mode;
acf[1] = AkUseAS + AkConfigAuth;
if (online) {
        acf[1] = AkOnline;
        ark = ss[slotId].slotRk;
} else {
        acf[1] = AkOffline;
        ark = {0};
}

/* perform the authentication mechanism */
Secret SymKey ak =
      AuthMech(inputV,acf,ark,popk,clCnf,XT,spkUri,nSpk,spkIndx,spk);

uchar response[16] = AuthMechResponse(ak, verifier);

if (response == {0}) {
        ss[slotId].se[sessId].configAuthMode = ConfigAuthModeAk1;
        return ErrOk;
} else {
        ss[slotId].se[sessId].configAuthMode = ConfigAuthModeNone;
        return ErrSlotConfigAuthFail;
}
```

الشروط المسبقة:

- تحميل المفتاح LKI الخاص بدورة **الفتحة AS**

10.4.2.8 تحميل المفتاح السري لمخدّم صغير

يمكن لوسيط **المخدم الصغير** أن يستخدم **الفتحة AS** التي تعمل بأسلوب المخدم اللاتناظري وتحميل قيمة المفتاح السري **للمخدم الصغير** ussK لإنشاء وصلة آمنة فيما بعد للشريحة الخاصة **بالوسيط الصغير**.

```
int reqAsLdUssk(uint slotId, uint sessId, InputV inputV,
                uchar XT[32], bool online, uchar mUssk[NUSSK])
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
PubKey spk[MaxSpkEncr], popk[MaxSpkEncr];
SessionConfig config[MaxSpkEncr];

/* basic consistency checks */
if (ss[slotId].slotMode!=SlotModeEncr) return ErrSlotMode;
if (!ss[slotId].se[sessId].config.encryptConfig.asymKlMode)
        return ErrSlotModeUndefined;

/* verify if the CPS ECI Host Root state is sufficient to proceed */
if (!cpsEciRootStateOk(slotId,sessId)) return ErrRevocEnforce;

spk[0]  = ss[slotId].se[sessId].spk;
popk[0] = ss[slotId].popk;
config[0] = ss[slotId]. se[sessId].config;
int i;
int nSpk = slot[slotId]. se[sessId].config.EncryptConfig.nEncr + 1;
for (i=0; i<nSpk-1; i++) {
  spk[i+1]  = ss[SlotId]. se[sessId].encrSpk[i];
  popk[i+1] = ss[SlotId]. se[sessId].decrSpk[i];
  config[i+1] = ss[slotId]. se[sessId].config;
}
```

```
/* define spkUri values */
ulong spkUri = (0x1<<(nSpk+1)) - 1; /* all SPKs can be used for decoding
 keys */

uchar ark[16];
uchar acf[15] = acfAk1Mode;
acf[1] = AkUseAS + AkLdUssk;
if (online) {
        acf[1] = AkOnline;
        ark = ss[slotId].slotRk;
} else {
        acf[1] = AkOffline;
        ark = {0};
}

/* perform the authentication mechanism */
Secret SymKey ak =
      AuthMech(inputV,acf,ark,popk,config,XT,spkUri,nSpk,0,spk);

/* perform AES ECB decoding of ussk */
int i,j;
uchar response[32];
for (i=0; i<NUSSK; i+=32){
response = AuthMechResponse(ak, &(mUssk[i]));
for (j=0; j<32; j++) ss[slotId].se[sessId].ussk[i+j] = response[j];
}
return ErrOk;
```

الشروط المسبقة:

- أن يكون قد تم الاستيقان من تشكيلة الدورة.

11.4.2.8 توليد MinitLk1 للوسطاء الصغار

في أسلوب **المخدم الصغير** اللاتناظري، يمكن **للفتحة AS** أن تولّد رسائل استهلال **مجموعة سلم المفاتيح** المتعلقة **بالوسطاء الصغار**:

```
InputV reqAsMInikLk1(uint slotId, uint sessId, ECI_Certificate_Chain ClCPK)
```

الدلالات:

يرد تعريف سلسلة شهادات السطح البيني ECI في الفقرة 1.4.5 من التوصية [ITU-T J.1012] وتتضمن **سلسلة الشهادات** اللازمة للتحقق من صلاحية **وسيط صغير**. تستخدم هذه الوظيفة أولاً النظام الفرعي لمعالجة الشهادات للتحقق من صلاحية المفتاح العمومي للشريحة CICPK باستخدام الوظيفة slot[slotId] مع المفتاح POPK **كشهادة سلَف** واستخدام ss[SlotId].se[sessId]. config.encryptConfig.microServerVersion كرقم أدنى صيغة **لقائمة الإبطال** من أجل أول **شهادة** في السلسلة. وإذا نجحت عملية التحقق من الصلاحية يتضمن المتغير clcpk المفتاح العمومي للشريحة الخاصة بالوسيط، وتنفّذ الشفرة التالية بلغة C.

```
return asymInitLk1(ss[slotId].lk1, slot[slotId].ussk, clcpk);
```

الشروط المسبقة:

- أن تعطى قيمة أولية للمفتاح Ussk.
- أن تكون الدورة بأسلوب تجفير لاتناظري.

12.4.2.8 حساب مفتاح فك التجفير المتعلق بصورة الوسيط ECI

بغية القيام بتحميل صورة مجفرة يمكن **للفتحة AS** أن توفر مفتاح استيقان يمكن بواسطته فك تجفير المفتاح اللازم لفك تجفير الصورة. وينبغي أن تنفّذ هذه الوظيفة قبل استهلال عمل الفتحة:

```
int reqAsComputeImageKey(uint slotId, InputV inputV,
        symKey eKey , bool online, ECIRootState min_root_state)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
/* a default slot configuration state is used */
SessionConfig config = {
  .decryptConfig = {
    .configVersion = 0x1,
    .reserved1 = 0x0,
    .klModeAuth = 0x0,
    .akModeAuth = 0x0,
    .rkKlMode = 0x0,
    .spk0NoDecrypt = false,
    .reserved2 = 0b000000,
    .rkDecrMode = { 0 },
    .minEciRootState = min_root_state,
    .expRlVersion = 0x0
  },
  .encryptConfig = { 0 }
} ;

if (!(cpsEciRootState.rootVersion >= min_root_state.rootVersion &&
     (cpsEciRootState.rlVersion >= min_root_state.rlVersion))
    return ErrRevocEnforce;

/* create straightforward popk/spk, XT, clCnf, */
PubKey popkArr[1]; /* also used for spk */
popkArr[0] = ss[slotId].popk;
SessionConfig cnf[1];
cnf[0] = config;
uchar XT[32] = {0};
ulong spkUri= 0x1;

uchar ark[16];
uchar acf[15] = acfAk1Mode;
acf[1] = AkUseAS + AkClImg;
if (online) {
        acf[1] = AkOnline;
        ark = ss[slotId].slotRk;
} else {
        acf[1] = AkOffline;
        ark = {0} ;
}

/* perform the authentication mechanism */
Secret SymKey ak =
      AuthMech(inputV,acf,ark,popkArr,cnf,XT,spkUri,1,0,popkArr);

Secret SymKey dImgKey = AuthMechResponse(ak, eImgKey);
/* dImgKey is subsequently used by the client loader to decrypt the client image using AES CBC mode
with IV=0 */

return ErrOk;
```

الشروط المسبقة:

- تضبط الفتحة على الحالة المبدئية؛ ويضبط المفتاح slotRk على قيمة عشوائية جديدة.

ملاحظة – لا تنفّذ هذه الوظيفة بناء على طلب **الوسيط ECI**.

13.4.2.8 عرض المعلومات المتعلقة بالأمن المعزّز

يوفر **نظام الأمن المعزّز** نفاذ **الوسيط ECI** إلى البيانات التي يولّدها ويوفر له وظيفة مفتاح عشوائي عامة الغرض.

الملاحظة 1 – لا تولّد الوظيفتان "get" و"set" المعرّفتان في هذه الفقرة أخطاءً تلقائية في قيم معلمات غير محددة، ولكن في حالة وظائف "get" يكون المرتجع ببساطة قيمة غير محددة، ولا يكون لها أي تأثير في حالة وظائف "set".

وتعرض الوظيفة التالية المفتاح العشوائي **للفتحة AS** (يستعمل عادة كقيمة ظرفية للدورات):

```
SymKey getAsSlotRk(uint slotId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
return ss[slotId].slotRk;
```

ويُرتجع رقم في الحالة التي لا يكون فيها تشغيل الفتحة قد استُهل.

وتعرض الوظيفة التالية حالة المفتاح العشوائي للدورة:

```
SymKey getAsSessionRk(uint slotId, uint sessionId, uint rkIndx
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
if (rkIndx == 0)
  return se[slotId][sessionId].rkState.rkCurrent;
else
  return se[slotId][sessionId].rkState.rkNext;
```

ويُرتجع رقم في الحالة التي لا يكون فيها تشغيل الفتحة قد استُهلّ.

ويمكن عرض العدّاد الحدي لمفتاح الدورة العشوائي:

```
ulong getAsSessionLimitCounter (uint slotId, uint sessionId)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
return se[slotId][sessionId].rkState.limitCounter;
```

ويُرتجع رقم في الحالة التي لا يكون فيها تشغيل الفتحة قد استُهلّ.

ويمكن ضبط قيمة حدية للعدّاد ينشأ عندها حدث ما (مثلاً تجديد الرقم العشوائي في وقت مناسب بما فيه الكفاية):

```
ulong setAsSessionLimitEvent(uint slotId, uint sessionId, ulong eventLimit)
```

الدلالات:

ينشأ الحدث eventSessionLimitCounter مرة واحدة حين تكون الحالة التالية صحيحة بعد استدعاء هذه الوظيفة:

```
se[slotId][sessionId].rkState.limitCounter <= eventLimit;
```

الملاحظة 2 – يتجاوز النداء الثاني النداء السابق. فاستدعاء هذه الوظيفة للمرة الثانية مع قيمة كبيرة جداً للحد eventLimit يلغي الحدث بالفعل (إلا إذا كان الحدث قد نشأ بالفعل).

وينشأ الحدث التالي عند بلوغ حدّ الحدث في دورة ما:

```
reqAsEventSessionLimit(uint slotId, uint sessionId)
```

الملاحظة 3 – يترجم هذا الحدث إلى رسالة غير متزامنة من دون رد مقابل في التوصية [ITU-T J.1012].

14.4.2.8 توليد الأرقام العشوائية للوسطاء

يمكن **للوسيط ECI** أن يطلب عن طريق استدعاء الوظيفة التالية رقماً عشوائياً مؤلفاً من 128 بتة يولّده **نظام الأمن المعزّز**:

```
SymKey getAsClientRnd()
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
return rnd128();
```

15.4.2.8 رموز الأخطاء

يرد في الجدول 8-14 قيم رموز الأخطاء التي ترجعها الدالة المعرّفة في الفقرة 4.2.8.

وتتقيد شفرات الخطأ هذه باصطلاح رموز الأخطاء المتعلقة بالرسائل بين **مضيف ECI** و**وسيط ECI** على النحو المعرّف في الفقرة 9 من التوصية [ITU-T J.1012].

الجدول 8-14 – تعريف رموز إعادة الخطأ

رمز إعادة الخطأ	القيمة	الوصف
ErrSlotMode	-256	**الفتحة AS** ليست بالأسلوب المناسب لهذه العملية
ErrNoMoreSessions	-257	لا يتوفر المزيد من الدورات
ErrSession1Coupled	-258	تم ضم الدورة الأولى بالفعل
ErrSession2Coupled	-259	تم ضم الدورة الثانية بالفعل
ErrSessionNotCoupled	-260	لم يتم ضم الدورة
ErrNoSuchSession	-261	الدورة غير موجودة
ErrExportNoSlot	-262	فتحة التصدير غير معروفة
ErrExportSlotBadState	-263	فتحة التصدير في حالة غير مناسبة
ErrExportOngoing	-264	يوجد بالفعل **وصلة تصدير** لفتحة التصدير
ErrImportSlotBadState	-265	فتحة الاستيراد ليست بأسلوب التجفير
ErrNoExport	-266	لا يوجد تصدير جار في الدورة
ErrSpkUriViolation	-267	قيمة SpkUri للمفتاح العمومي للمرسل (SPK) لا تناسب أسلوب الفتحة
ErrSlotModeUndefined	-268	قيمة أسلوب الفتحة لا تناسب هذه العملية
ErrRevocEnforce	-269	**إبطال السطح البيني ECI** لا يسمح للفتحة بالعمل
ErrNoConfigAuth	-270	لم يتم الاستيقان من تشكيلة الفتحة بصورة مناسبة
ErrNoSlotRkInsert	-271	طول المتجه ELK غير كافٍ لإدراج مفتاح عشوائي
ErrSpk0NoDecrypt	-272	لا يمكن استعمال spk[0] لتوليد كلمات تحكم بفك التجفير
ErrBasicUriCtrl	-273	بتة التحكم في الحقل field1 للمعرّف الأساسي URI غير مضبوطة
ErrOk	0	استدعاء ناجح
ErrSlotConfigAuthFail	-274	فشل الاستيقان من تشكيلة الدورة المتعلقة بالفتحة
ErrParam<N>	-<N>	خطأ في معلمة الدخل **N** (قيمة المعلمة ErrParam1 تساوي -1 وتشير إلى خطأ في المعلمة 1)
	MaxInt..1	استدعاء ناجح، القيمة تحددها تعاريف الرسائل

9 التخليط/إزالة التخليط وتصدير المحتوى

1.9 الوظيفية الأساسية

يستطيع مسار فيديوي آمن أن يفك تجفير المحتوى. ويترافق هذا المحتوى مع **خصائص المحتوى** و**وصلات التصدير**. ويمكن إحالة المحتوى إلى نقاط خرج معيارية إذا سمحت بذلك **خصائص المحتوى**، ويمكن إعادة تجفيره بواسطة **مخدم صغير** في حالة وجود **وصلة تصدير** مطابقة.

وبهدف إدارة الموارد يحدد **السطح البيني ECI** موارد التجفير وفك التجفير. ويستخدم المورد لفك تجفير أو تجفير محتوى مأخوذ من دورة وسائط واحدة مجفرة أو من المقرر تجفيرها بواسطة كلمة تحكم واحدة كل مرة، ويتم توصيل مورد فك التجفير أو التجفير **بفتحة AS** واحدة لفك التجفير أو التجفير. وفي حالة فك تجفير تدفق النقل، يكون لمورد فك التجفير ذاكرتا خزن احتياطي

مخصصتان لكلمة التحكم المفردة وكلمة التحكم المزدوجة. ويتم اختيار كلمة التحكم المفردة أو المزدوجة بواسطة التدفق الذي يتعين إزالة تخفيره. ويمكن أن يتلاءم ذلك مع الحاجة إلى تغيير كلمة التحكم بشكل فوري إذا تغيرت **خصائص المحتوى** المقرر تخفيره. وبالنسبة لفك التجفير أو التجفير القائم على ملف، يوفر **المضيف ECI** التزامن بين كلمة التحكم والمحتوى المقرر فك تخفيره، الذي قد يكون أسرع مما هو في الوقت الفعلي. ولا يتطلب مورد فك التجفير وإعادة التجفير القائم على ملف إلا ذاكرة خزن احتياطي واحدة لكلمة التحكم.

ملاحظة – إن تدفقات النقل التي تتطلب اثنتين أو أكثر من كلمات التحكم لإزالة تخليط مختلف التدفقات الأولية تحتاج إلى موارد متعددة لإزالة التخليط وبالتالي إلى دورات متعددة.

ولا يحدد **السطح البيني ECI** أي خصائص تتعلق بالخزن الاحتياطي أو بالمعالجة الفورية (المكثفة ربما) مثل تحويل الشفرة أو التشفير المائي يمكن إجراؤها على المحتوى الذي أزيل تخفيره بالانتقال من مورد فك التجفير إلى مورد التجفير. وقد تسبب هذه العملية تأخراً كبيراً. ويمكن لمصنّعي معدات منشآت العميل (CPE) أن يختاروا عمليات تنفيذ مناسبة تسبب تفاوتاً زمنياً بين مورد فك التجفير ومورد موصول لإعادة التجفير. وتتزامن فتحة إعادة التجفير و**الوسيط ECI** مع تجفير المحتوى.

2.9 مواصفات المخلّط ومزيل التخليط

تدعم وظيفة إزالة التخليط في معدات منشآت العميل المطابقة **للسطح البيني ECI** خوارزميات إزالة التخليط التالية في أسلوب النقل TS.

- CSA1/2، في الأسلوبين PES (التدفق الأولي بأسلوب الرزم) وTS (تدفق النقل) على النحو المعرف في المعيار [ETSI ETR 289] والمعيار [b-ETSI DVB CSA].
- CSA3، في الأسلوبين PES وTS على النحو المعرف في المعيار [ETSI TS 100 289] والمعيار [b-ETSI DVB CSA3].
- الأسلوب DVB-CISSA PES والأسلوب TS [ETSI TS 103 127].

وتدعم وظيفة إزالة التخليط في معدات منشآت العميل المطابقة **للسطح البيني ECI** خوارزميات إزالة التخليط التالية في أسلوب الملف.

- يستخدم الأسلوب CENC AES128 CTR والأسلوب AES128-CBC (وكلاهما يستخدم التجفير بعينة كاملة وعينة جزئية) على النحو المعرّف في المعيار [ISO/IEC 23001-7]. ويستخدم الأسلوب CENC والمعيار [ISO/IEC 23009-4] في حالة النقل MPEG-DASH.

وتدعم وظيفة إزالة التخليط في معدات منشآت العميل المطابقة **للسطح البيني ECI** خوارزميات إزالة التخليط التالية في أسلوب النقل TS.

- الأسلوب DVB-CISSA PES والأسلوب TS [ETSI TS 103 127].

وتدعم وظيفة إزالة التخليط في معدات منشآت العميل المطابقة **للسطح البيني ECI** خوارزميات إزالة التخليط التالية في أسلوب الملف:

- CENC AES128 CTR وأسلوب CBC (وكلاهما يستخدم التجفير بعيّنة كاملة وعيّنة جزئية) على النحو المعرّف في المعيار [ISO/IEC 23001-7] والمعيار [ISO/IEC 23009-4]. ويولد **مسار فيديوي آمن** متجه استهلال فريد للمحتوى المجفر بواسطة كلمة تحكم واحدة للأسلوب AES-CTR ويتبع قواعد التعريف الرابع للأسلوب AES-CBC كما هو معرف في المعيار [ISO/IEC 23009-4]. ويمكن **للمضيف ECI** النفاذ إلى متجهات الاستهلال لاستعمالها في ترزيم المحتوى.

3.9 التحكم في التصدير

تستخدم **وصلات التصدير** التي تم الاستيقان منها في دورات فك التجفير في **الفتحة AS** كبطاقات للترخيص لمورد فك التجفير بالاستيراد والتصدير. ويسمح مورد فك التجفير بتصدير المحتوى الذي أزيل تجفيره إلى مورد تجفير إذا سمحت **وصلة التصدير** التي توفرها دورة **الفتحة AS** المرتبطة بها بذلك لمعرّف مجموعة التصدير وأشارت **خصائص المحتوى** المجفّر إلى معرّف مجموعة التصدير المقابلة على النحو المحدد في الفقرة 4.4.2.8. ولا يسمح مورد فك التجفير بتصدير محتوى تمت إزالة تجفيره إذا لم يتم التحقق من صلاحية **وصلة التصدير** الخاصة بمجموعة التصدير التي اختارها معرّف مجموعة التصدير في **خصائص المحتوى** باعتبارها **وصلة تصدير** توفرها **الفتحة AS** المرتبطة بها.

4.9 التحكم في الخرج

تستخدم **خصائص المحتوى** المتعلقة بالتحكم في الخرج في عدم تفعيل أو تفعيل تصدير المحتوى بموجب تكنولوجيات الحماية المعيارية للصناعة على وصلات الخرج في معدات منشآت العميل. ويجب على مورد فك التجفير أن يسمح بتصدير محتوى أزيل تجفيره إلى خرج ما إذا سمحت بذلك معلومات التحكم في الخرج الصادرة عن دورة **الفتحة AS** المرتبطة بها. ولا يسمح مورد فك التجفير بتصدير محتوى أزيل تجفيره إلى خرج ما إذا لم تتضمن معلومات التحكم المتعلقة بالخرج إذناً من دورة **الفتحة AS** المرتبطة بها.

5.9 مقارنة خصائص المحتوى في الدورات المجمّعة

يتحقق **مسار فيديوي آمن** من أن **خصائص المحتوى** كما هي معرّفة في الحقل field1 لدورة معينة باستثناء البايتين الأولين مساوية **لخصائص محتوى** أي دروة مقترنة. ويسمح بتصدير المحتوى وإنتاج محتوى دورة مقترنة ذات **خصائص محتوى** متساوية. ومن الآن فصاعداً تعالج التدفقات الموحدة كدورة واحدة من منظور حماية **السطح البيني ECI**. ويمنع دمج تدفق مقترن إذا كانت **خصائص المحتوى** في الحقل field1 باستثناء البايتين الأولين غير متساوية.

6.9 انتشار خصائص المحتوى عند التصدير

تنقل دورة مورد فك التجفير **خصائص المحتوى** المتعلقة بالحقل field1 التي حددها الوسيط واستيقن ضمناً منها (جزئياً) **سلم المفاتيح** مع المحتوى إلى موارد دورة إعادة التجفير التي استوردت المحتوى الذي أزيل تجفيره على النحو المعرف في الفقرة 6.4.2.8. وتقوم دورات التجفير التي تلقت المحتوى الذي أزيل تجفيره بمقارنة بايتات الحقل field1 المعينة بالقيمة التي حددها الحقل field1 لتجفير المحتوى وتطبق في الوقت نفسه قناعاً لاختيار الحقول التي تحتاج إلى نقل كما تحدده الوظيفة، ما يضمن بالتالي أن بايتات الحقل field1 المتعلقة بالوسيط القائم بفك التجفير قد انتقلت إلى المحتوى المجفر.

وتقوم دورة مورد التجفير بتنفيذ الشفرة التالية بلغة C على كل تغير يطرأ على قيم الدخل impField1 وexpField1 وcpMask:

```
uchar impField1[16];  /* field1 values for the imported content */
uchar expField1[16];  /* field1 values from the encryption CW computation */
ushort cpMask;        /* comparison mask */

bool propOk = true;  /* indicates if propagation of imported content is Ok */
int i;

for (i=2; i<16; i++)
    propOk &&= !(cpMask>>I & 0b1) || (impField1[i] == expField1[i]);

if (propOk) /* re-encrypt content */
else /* do not re-encrypt content */
```

7.9 إنفاذ المعرّف URI الأساسي عند التصدير

يتم التحكم بنقل المعرّف URI الأساسي من دورة التجفير في **الفتحة AS** إلى دورة إعادة التجفير في **الفتحة AS** عن طريق الآليات التالية، حيث يمثل slotId معرّف هوية فتحة التجفير وsessionId معرّف هوية دورة التجفير:

ليست الحقوق التي تخصصها **الفتحة AS** القائمة بالتجفير إلى المحتوى المتعلق بالمعرّف URI الأساسي أكثر يسراً من تلك المخصصة إلى المحتوى المنقول.

يتم الاستيقان من **المخدم الصغير**: ss[slotId].se[sessionId].config.decryptConfig,akModeAuth تساوي 0b1.

إذا لم يسمح المعرّف URI الأساسي بإعادة عرض المحتوى (أي أسلوب التدفق) يتم التحقق مما يلي عند التصدير:

- يجب أن لا تتساوى ss[slotId].se[sessionId].config.decryptConfig.rkDecrMode.mode مع RKModeNone (أي يطبق رقم عشوائي مرة واحدة لمنع إعادة عرض محتوى مسبق التشفير عند إعادة تشغيل النظام)؛
- ويجب أن تضبط ss[slotId].se[sessionId].klModeAuth (القيمة 0b1) للتاكد من أن التشكيلة decryptConfig التي يستعملها المخدم، بما في ذلك إدراج مفتاح عشوائي عند **الوسيط الصغير**، قد تم الاستيقان منها ويستخدمها **الوسيط الصغير** استناداً إلى حساب **سلم المفاتيح**.

8.9 تطبيق خصائص المحتوى على النواتج المعيارية الصناعية

يستخدم الناتج المعياري، الذي هو عبارة عن ناتج مادي مقترن بنظام لحماية المعايير الصناعية، **خصائص المحتوى** لاختيار الوضع المناسب لحماية الناتج أو لعدم تفعيله إذا لم يكن الوضع المناسب ممكناً. وتحدد القواعد الدقيقة له بقواعد الامتثال.

ويجب أن تكون **متانة** تنفيذ المعرّف URI الأساسي و**خصائص المحتوى** المتعلقة بالتحكم في الناتج بنفس مستوى متانة تنفيذ **مسار فيديوي آمن**.

ويجب أن تكون **متانة** إنفاذ المعرّف URI المعياري بنفس درجة ارتفاع تنفيذ **المضيف ECI** على الأقل، باستثناء الوظائف التي تكون متطلبات تنفيذها معقدة.

9.9 مزامنة كلمات التحكم

في حالة معالجة التدفقات TS (تدفقات النقل) يوفر **مسار فيديوي آمن** وظائف تسمح بالتحكم في تغيرات كلمات التحكم (في حالة التجفير) ويقدم إشعارات بشأن تغيرات حقل التحكم في التخليط. وتلتزم الوظائف والأحداث الواردة في هذا القسم بالاصطلاحات المحددة في الفقرة 4.2.8.

ويمكن **للفتحة AS** أن توفر كلمة تحكم "مفردة" وكلمة تحكم "مزدوجة" لتطبيقها في تجفير المحتوى أو فك تجفيره.

في حالة فك التجفير، يُبلغ حقل التحكم في التخليط وظيفة فك التجفير بكلمة التحكم التي يتعين استخدامها. ولا تستعمل أي كلمة تحكم إذا تبين أن المحتوى غير مخلّط. وتساوي قيمة النتيجة حقل التحكم في التخليط، والقيم محددة في الفقرة 1.5 من المعيار [ETSI TS 100 289].

وتعرض الوظائف التالية الحالة الراهنة لحقل التحكم في التخليط داخل التدفق:

```
uint getAsSC(uint slotId, uint sessionId)
```

وفي حالة التجفير يمكن أن تتغير كلمة التحكم المستخدمة على أساس حدثين:

1) حدوث تغير في **خصائص المحتوى** المستورد، الأمر الذي يسبب تغيراً في كلمة التحكم المستخدمة للتجفير. ويمكن تأخير حدوث هذا التغير بواسطة **الفتحة AS** من أجل إكمال تغير جارٍ في كلمة التحكم يسببه الحدث التالي.

   إشارة من دورة **الفتحة AS** تفيد بضرورة تغيير كلمة التحكم.

في الحالة التي لا يتم فيها تخليط المحتوى المستورد، لا يطبق أي تخليط من أجل التجفير ويضبط حقل التحكم في التخليط على القيمة 0b00 في موقع أول تغيير ممكن. وبالعكس من ذلك، ففي الحالة التي يتبدل فيها المحتوى المستورد من حالة عدم التخليط إلى حالة التخليط، يتم تخليط المحتوى بواسطة كلمة التحكم التالية؛ ويتم اختيار المفتاح المعاكس مقارنة بالمحتوى الذي تم تخليطه قبل الجزء الواضح من المحتوى.

ويعرّف الحدث الذي يشير إلى تغير خصائص المحتوى المستورد على النحو التالي:

reqAsEventCpChange(uint **slotId**, uint **sessionId**)

الدلالات:

يشير الحدث إلى تغير في **خصائص المحتوى** المستورد إذا كان هذا المحتوى بحاجة إلى تجفير.

ولا يسمح **مسار فيديوي آمن** بوجود تباين بين معلمات التجفير و**خصائص المحتوى** المستورد لفترة أطول. وتُقترح قيمة قصوى في الفقرة 2.6.6 من الإضافة [b-ITU-T J. Suppl. 7].

الملاحظة 1 – لا ينشأ هذا الحدث عندما يتغير المحتوى المستورد من مجفر إلى غير مجفر. ولا تطبق **خصائص المحتوى** على المحتوى غير المجفر.

ويسمح **مسار فيديوي آمن للفتحة AS** بتقييد أي تغير تلقائي للوظيفة eventCpChange المتعلقة **بخصائص المحتوى** بناء على التعليمة التالية:

setAsPermitCPChange(uint **slotId**, uint **sessionId**, bool **permit**)

الدلالات:

تحدد هذه الوظيفة الإذن الذي يسمح لتغير تلقائي في خصائص المحتوى المستورد بإحداث تغيير في كلمة التحكم في المحتوى المجفّر.

الملاحظة 2 – ينبغي أن تسبق هذه الوظيفة أي كلمة تحكم تالية لم تحسب على أساس **خصائص المحتوى** المقبلة، التي تعكس مثلاً تغيراً ظرفياً أو عشوائياً في المفتاح.

الملاحظة 3 – إذا لم يتم تفعيل إذن التغير (permit==false) ينبغي استعادته ضمن الفترة المسموحة للتفاوت مع **خصائص المحتوى** المقرر استيراده بحيث ينشأ "تعتيم" في التدفق الذي أعيد تجفيره.

وتسمح الوظيفة التالية بضبط حقل التحكم في التخليط خلال التجفير على حالة معينة.

setAsSC(uint **slotId**, uint **sessionId**, uint **scramblingControlField**)

الدلالات:

تضبط قيمة حقل التحكم في التخليط على القيمة scramblingControlField في أول نقطة تغير ممكنة في التدفق. ولا يسمح للحقل scramblingControlField إلا بأخذ القيمتين 0b10 و0b11 (تخليط مع مفتاح مزدوج ومفرد على التوالي).

ويضبط حقل التحكم في التخليط المتعلق بالتدفق المجفّر على القيمة 0b00 (بدون تخليط) إذا كان المحتوى المستورد في حالة عدم التجفير.

وتعرّف الوظيفة التالية لدورتي فك التجفير والتجفير:

reqAsEventSC(uint **slotId**, uint **sessionId**, uint **scramblingControlField**)

الدلالات:

ينشأ الحدث عند تغير حالة حقل التحكم في التخليط.

10 النظام الفرعي لمعالجة الشهادات

1.10 القواعد الأساسية لمعالجة سلاسل الشهادات

يمكن **للنظام الفرعي لمعالجة الشهادات** أن يعالج **سلاسل الشهادات** للاستيقان من البنود استناداً إلى مفتاح عمومي أولي ورقم أدنى **قائمة إبطال**. وتكون معالجة معظم سلاسل الشهادات عامة. وتحدد هذه الفقرة قواعد المعالجة العامة **لسلاسل الشهادات**. وتحدد الفقرات التالية قواعد المعالجة الخاصة بأنواع مختلفة من السلاسل.

ويرد في الفقرة 4.5 من التوصية [ITU-T J.1012] تعريف **لسلاسل الشهادات** المحددة أدناه.

وتستخدم تعاريف قواعد النظام الفرعي لمعالجة الشهادات نهجاً تدريجياً لمعالجة **سلاسل الشهادات** يبدأ عند بداية السلسة (أول **قائمة إبطال**) باستخدام مفتاح عمومي أولي ورقم أدنى **قائمة إبطال**. وتتمثل الخطوة الأولى بالتحقق من **قائمة الإبطال**. وتتمثل الخطوة الثانية بالتحقق من الشهادة التالية في السلسلة. وبعد إجراء الخطوتين 1 و2 مرة واحدة، يتم تحديد مفتاح عمومي جديد ورقم جديد **لقائمة الإبطال** من أجل معالجة القسم المتبقي من السلسلة. وتكرر الخطوتان 1 و2 حتى تتم معالجة السلسلة بكاملها. وبوجه عام، يوصى بأن تقوم وظائف البرمجيات التي تقدم السلاسل بالتحقق المسبق من صلاحية هذه السلاسل بغية تجنب فشل النظام الفرعي لمعالجة الشهادات في معالجة سلسلة ما بصورة غير متوقعة.

وفيما يلي الخطوات العامة لمعالجة **سلسة الشهادات**:

1) يقوم النظام الفرعي لمعالجة الشهادات بعملية التحقق التالية من **قائمة الإبطال**:

أ) يتحقق النظام الفرعي لمعالجة الشهادات من الحقل **Revocation List format_version** لمواءمة صيغة يمكنه تفسيرها (انظر القواعد الخاصة بمعالجة السلاسل) ومن الحقلين **rl_id.type** و**rl_id.rl_indicator** لمواءمة القيم المتوقعة.

ب) إذا كانت الشهادة السلَف شهادة جذرية (**root_version_indicator**=1) يختار **المضيف ECI الشهادة الجذرية** مع **root_version** لتكون الشهادة السلَف، وإلا تستخدم الشهادة التي جرى تحميلها مسبقاً أو الشهادة السابقة.

ج) يتحقق النظام الفرعي لمعالجة الشهادات من توقيع **قائمة الإبطال** بواسطة آخر مفتاح عمومي أقرت صلاحيته.

د) يتحقق النظام الفرعي لمعالجة الشهادات مما إذا كان طول **قائمة الإبطال** يقابل قيم الحقل الخاص بها ومما إذا كان لأي حقل متغير الطول الطول المناسب.

هـ) يتحقق النظام الفرعي لمعالجة الشهادات مما إذا كان قد تم إبطال صلاحية رقم صيغة **قائمة الإبطال** بواسطة رقم أدنى **قائمة إبطال**.

2) يقوم النظام الفرعي لمعالجة الشهادات بعملية التحقق التالية من **الشهادة**:

أ) يتحقق النظام الفرعي لمعالجة الشهادات من عدم إبطال <type, entity_ id, version> ***next*** **للشهادة** في السلسلة وفقاً **لقائمة الإبطال** الأخيرة ويحدد الصيغة الدنيا **لقائمة الإبطال** التي ترافق **الشهادة** وفقاً للحقلين **base_rl_version** و**min_rl_version** في آخر **قائمة إبطال**.

ب) يتحقق النظام الفرعي لمعالجة الشهادات من الحقل **Revocation List format_version** لمواءمة صيغة يسمح له بتفسيرها.

ج) يتحقق النظام الفرعي لمعالجة الشهادات مما إذا كان طول **الشهادة** يقابل قيم الحقل الخاص بها ومما إذا كان لأي حقل متغير الطول المناسب.

د) يتحقق النظام الفرعي لمعالجة الشهادات من **توقيع الشهادة** بواسطة المفتاح العمومي.

وبعد إجراء الخطوتين 1 و2 يتم تحديث المفتاح العمومي و**قائمة الإبطال** الدنيا. ويكون المفتاح العمومي مساوياً لحقل المفتاح العمومي **للشهادة** التي تمت معالجتها في الخطوة 2، ورقم أدنى قائمة إبطال تم الحصول عليه في الخطوة 2أ.

ولا تحتاج جميع **الشهادات** لأن تكون مصحوبة **بقائمة إبطال**. فإذا كانت البتة الأكثر دلالة لحقل النوع في معرّف هوية الشهادة مساوية للصفر، يقتضي **النظام الفرعي لمعالجة الشهادات** أن تكون **قائمة الإبطال** مصحوبة **بشهادة** لمواصلة معالجة السلسلة. ولا تطبق أي معالجة **لقائمة الإبطال** ورقم الصيغة وأرقام صيغة **قائمة الإبطال** الواردة في الخطوات أعلاه إذا لم يكن هناك حاجة إلى **قائمة الإبطال**.

2.10 قواعد خاصة بسلاسل صور المضيف

يطبق النظام الفرعي لمعالجة الشهادات عملية التحقق الخاصة من صلاحية سلاسل صور المضيف:

1) تكون **قائمة الإبطال** الأولى من النوع 0x1 (**قائمة الإبطال** الخاصة بالمصنّع).
2) تكون **الشهادة** الأولى من النوع 0x1 (**الشهادة** الخاصة بالمصنّع).
3) تكون **قائمة الإبطال** الثانية من النوع 0x0 (**قائمة الإبطال** الخاصة **بالمضيف ECI**).
4) تكون **الشهادة** الثانية من النوع 0x0 (**الشهادة** الخاصة **بالمضيف ECI**).
5) وهناك **شهادة** ثالثة ممكنة من النوع 0x98 (شهادة سلاسل صور **المضيف**).

ويستخدم المفتاح العمومي لآخر **شهادة** (إما **شهادة المضيف ECI** أو **شهادة** سلاسل صور **المضيف ECI**) للتحقق من صلاحية الصورة الفعلية **للمضيف ECI**.

3.10 قواعد خاصة بسلاسل صور الوسيط

يطبق النظام الفرعي لمعالجة الشهادات عملية التحقق الخاصة من صلاحية سلاسل صور الوسيط:

1) تكون **قائمة الإبطال** الأولى من النوع 0x2 (**قائمة الإبطال** الخاصة بالبائع).
2) تكون **الشهادة** الأولى من النوع 0x2 (**الشهادة** الخاصة بالبائع).
3) تكون **قائمة الإبطال** الثانية من النوع 0x0 (**قائمة الإبطال** الخاصة **بالوسيط ECI**).
4) تكون **الشهادة** الثانية الممكنة من النوع 0x1 (**الشهادة** الخاصة بسلسلة الوسطاء).

ويستخدم المفتاح العمومي لآخر **شهادة** (إما **شهادة** البائع أو شهادة سلسلة الوسطاء) للتحقق من صلاحية الصورة الفعلية **للمضيف ECI**، مع الأخذ في الاعتبار رقم آخر صيغة لقائمة **إبطال الوسيط** من أجل التحقق من صيغة الصورة إذا كانت آخر **شهادة** هي **شهادة** البائع.

4.10 قواعد خاصة بشهادات عمليات المنصة

يطبق النظام الفرعي لمعالجة الشهادات عملية التحقق الخاصة من صلاحية **سلاسل شهادات عمليات المنصة**:

1) تكون **قائمة الإبطال** الأولى من النوع 0x3 (قائمة الإبطال الخاصة **بالمشغّل**).
2) تكون **الشهادة** الأولى من النوع 0x3 (**شهادة المشغّل**).
3) تكون **قائمة الإبطال** الثانية من النوع 0x0 (قائمة إبطال **عمليات المنصة**).
4) تكون **الشهادة** الثانية من النوع 0x0 (**شهادة عمليات المنصة**).

5.10 قواعد خاصة بسلاسل التصدير/الاستيراد

1.5.10 معالجة سلسلة تراخيص التصدير

يوفر النظام الفرعي لمعالجة الشهادات سلسلة الاستيقان من التصدير والقسم المقابل من سلسلة التصدير الخاصة بطرف ثالث.

يبدأ النظام الفرعي لمعالجة الشهادات بالصيغة الجذرية الدنيا وصيغة **قائمة الإبطال** المعرّفة في .[slotId]ss.[sessionId]se. config.decryptConfig.minEciRootState. ويقوم بمعالجة سلسلة **شهادة مشغل** تراخيص التصدير (EAOC) و**شهادات** ترخيص التصدير (EAC) و**قوائم الإبطال** المرتبطة بها والتحقق من القواعد الخاصة التالية المتعلقة بهذه السلسلة:

- يكون معرّف هوية قائمة الإبطال الجذرية 0x4 (قائمة إبطال **مشغلي** تراخيص التصدير).

- يكون معرّف هوية **الشهادة** التالية (EAOC) 0x4.
- يكون معرّف هوية **قائمة الإبطال** التالية (REAOC RL) 0x0.
- يكون معرّف هوية **الشهادات** اللاحقة (EAC) في السلسلة 0x0.
- يكون محتوى حقل تمديد **الشهادة** مساوياً **لشهادة** سلسلة التصدير المقابلة في سلسلة التصدير.
- يكون معرّف هوية **قائمة الإبطال** اللاحقة (EAC-RL) في السلسلة 0x0.
- يجب أن يتم التحقق من صلاحية جميع **الشهادات** في سلسلة التصدير بالتتابع بواسطة سلسلة تراخيص التصدير.
- يجب أن تكون **الشهادة** الأولى في فرع سلسلة التصدير الخاصة بطرف ثالث من نوع TPEGC (يساوي معرّف هوية الشهادة 0x5).
- يجب أن تكون **الشهادة** الأخيرة في قسم سلسلة التصدير الخاص بطرف ثالث من نوع TPEGC أو ESC أو ERC (يساوي معرّف هوية الشهادة 0x5 أو 0xE أو 0xF على التوالي).
- يجب أن تكون **الشهادات** الوسيطة كلها من نوع EGC (يساوي معرّف هوية الشهادة 0x4).
- إذا كانت **الشهادة** الأخيرة من نوع TPEGC، يجب أن يكون ذلك بداية القسم التالي من سلسلة التصدير. ويجب أن تتكرر عملية التحقق الواردة أعلاه في جميع الأقسام اللاحقة من سلاسل الاستيقان من التصدير وأقسام الطرف الثالث من سلسلة التصدير حتى يتم التحقق من صلاحية سلسلة التصدير الخاصة بطرف ثالث بشكل كامل (الانتهاء بشهادة ESC أو ERC).

2.5.10 التحقق من سلسلة التصدير

يبدأ النظام الفرعي لمعالجة الشهادات بالمفتاح العمومي لشهادة عملية المنصة (POC)، وبمؤشر مجموعة التصدير التي يتعين أن يكون التصدير من أجلها، ورقم أدنى صيغة **لقائمة الإبطال** الذي ينبغي تطبيقه على **قائمة إبطال** شهادة عمليات المنصة المبين في الحقل ss[slotId].se[sessionId].config.decryptConfig.minClientVersion المتعلق بحالة **الفتحة AS**.

ملاحظة – يعتمد هذه التحقق من الصلاحية على استيقان مناسب من المفتاح POPK ومن صيغة **قائمة الإبطال**. وينبغي أن يتم ذلك بواسطة الاستيقان بأسلوب مفتاح الاستيقان (AK) أو باستيقان ضمني بواسطة **سلّم المفاتيح** (انظر الفقرة 2.2.2.8، الحقلان klModeAuth وakModeAuth).

ويعالج النظام الفرعي لمعالجة الشهادات الحقول POC-RL وEGC وEGC-RL والحقل اللاحق TPEGC أو ESC **كسلسلة شهادات** منتظمة. ويجب التحقق من القواعد التالية:

- يكون نوع شهادة مجموعة التصدير (EGC) 0x4.
- يكون حقل معرّف الهوية export_group_id في الشهادة مساوياً لمؤشر مجموعة التصدير.
- يكون نوع **قائمة إبطال** الشهادة EGC 0x4.
- يكون نوع الشهادة ECG-RL 0x4.
- يقابل نوع الشهادة TPEGC أو ESC القيمة الواردة في الجدول 2.5-2 من التوصية [ITU-T J.1012].

وترد في الفقرة 3.5.10 معالجة الشهادة TPEGC. وترد في الفقرة 4.5.10 معالجة الشهادة ESC.

3.5.10 التحقق من سلسلة التصدير الخاصة بطرف ثالث

تبدأ معالجة سلسلة التصدير الخاصة بطرف ثالث بالتحقق من صلاحية الشهادة الرئيسية TPEGC ورقم أدنى صيغة **لقائمة الإبطال** الذي تنتهي عنده معالجة **قائمة الإبطال** بشهادة من شهادات نظام التصدير (ESC).

4.5.10 معالجة شهادة نظام التصدير

يستخدم المفتاح العمومي لمرسل **الشهادة** ESC ورقم أدنى صيغة **لقائمة الإبطال** المتعلقة بسلَف الشهادة ESC للتحقق من صلاحية **وصلة التصدير**. ويجب أن يتواءم المفتاح SPK الخاص **بالشهادة** مع الحقل ss[slotId].spk المتعلق بفتحة التصدير المعينة. ويجب أن يكون رقم أدنى صيغة **لقائمة الإبطال** أكبر من الصيغة ss[slotId].ssConfig.microServerVersion.

ملاحظة – يجب الاستيقان من المفتاح SPK المتعلق بفتحة التصدير بواسطة **آلية الاستيقان** من المفتاح AK الخاص **بالفتحة AS** لكي يكون الاستيقان مجدياً.

5.5.10 قواعد معالجة سلسلة الوسطاء المستهدفين

تبدأ معالجة سلسلة الوسطاء **المستهدفين** بالمفتاح POPK و**قائمة الإبطال** الدنيا لحالة التشكيلة **MSConfig لمخدّم صغير**. وتتقيد معالجة سلسلة الوسطاء **المستهدفين** بالنظام الفرعي لمعالجة الشهادات بالقواعد العامة المحددة في الفقرة 1.10. وهي تتبع بالإضافة إلى ذلك القواعد المحددة التالية:

1) تكون قائمة الإبطال الأولى من النوع 0x0 (قائمة الإبطال المستهدفة).

2) تكون الشهادة الأولى من النوع 0x0 (شهادة المجموعة المستهدفة) أو 0x8 (شهادة الوسيط الصغير).

3) تكرر الخطوتان 1 و2 في الحالة التي تكون فيها الشهادة في الخطوة 2 شهادة مجموعة مستهدفة.

ويتمثل المفتاح العمومي **لشهادة الوسيط الصغير** بالمفتاح العمومي للشريحة الذي يجب استخدامه وفقاً للآلية الوارد وصفها في الفقرة 3.7.

6.10 استهلال المفتاح الجذري للسطح البيني ECI في النظام الفرعي لمعالجة الشهادات

في الوقت الذي يستهل به **نظام الأمن المعزز** عمله يقوم **المضيف ECI** بتحميل البرنامج الفرعي لمعالجة الشهادات بآخر المعلومات المتعلقة **بالمفتاح الجذري للسطح البيني ECI** ورقم **قائمة الإبطال** المعتمدَيْن.

```
function InitCPSEciRoot(uchar minRootKeyVersion, uint minRevListNr)
```

الدلالات:

تنفّذ الشفرة التالية بلغة C:

```
cpsEciRootState.rootVersion = minRootKeyVersion;
cpsEciRootState.rlVersion   = minRevListNr;
```

يطبق النظام الفرعي لمعالجة الشهادات الصيغة **rootKeyVersion** بوصفها رقم صيغة **المفتاح الجذري للسطح البيني ECI** ويطبق الرقم **minRevListNr** على جميع السلاسل المتوفرة له لتحميل أوراق اعتماد **السطح البيني ECI**.

ويعاد ضبط جميع الحالات الأخرى **لنظام الأمن المعزّز**.

وتجدر الملاحظة أن تحديد المعلمتَيْن من جانب **المضيف ECI** ينبغي أن يضمن إمكانية تحميل جميع **الوسطاء ECI** وأن **المضيف ECI** لم يتم إبطاله، مع أن أياً من **الوسطاء ECI** لم يتعرض **للإبطال**.

11 أداة التحميل الرئيسية

1.11 مقدمة

يستعمل نظام **السطح البيني ECI** آلية تحميل تسمح **للوسطاء ECI** بالتحقق بشكل آمن من صيغة أوراق اعتماد **المضيف ECI** و**الوسطاء ECI** التي جرى تحميلها لكشف أي مشكلة أمنية معروفة. يمكّن ذلك من تحديث **المضيف ECI** و**الوسطاء ECI** (سواء الصور والمفتاح POPK) مثل أي وظيفة منتظمة لعمل النظام.

وتعتمد أداة التحميل لصور **المضيف ECI** و**الوسطاء ECI** على بعض مبادئ **المتانة** المعرّفة كقواعد محددة في الفقرات التالية. وتعرّف **متانة** تنفيذ هذه القواعد في وثيقة ملائمة تقع خارج نطاق مواصفات **السطح البيني ECI**، ولكن هذه القواعد يجب أن تتسم عموماً **بمتانة** تنفيذ مكافئة. وتعتبر بعض القواعد منفذة بدرجة عالية من **المتانة** (على درجة عالية) ويجب أن تكون أقوى بكثير من تنفيذ **المضيف ECI**.

2.11 قواعد أداة التحميل الخاصة بالمضيف

تمتثل أداة التحميل الخاصة **بالمضيف ECI** للقواعد التالية:

1) يجب على أداة التحميل الخاصة **بالمضيف ECI** أن تضمن الصيغة الجذرية للسطح البيني **ECI** ورقم صيغة قائمة الإبطال الجذرية المستخدمين للتحقق من صلاحية صور **المضيف ECI** مختزنان عند استهلال التشغيل وأنه لن يكون من الممكن تغيير هذا الرقم بعد الآن. تتطلب هذه القاعدة درجة عالية من **المتانة**.

2) يجب ألا يُسمح بتغيير أداة التحميل نفسها الخاصة **بالمضيف ECI**. تتطلب هذه القاعدة درجة عالية من المتانة.

3) يجب ألا يُسمح بتغيير أو رصد صورة **المضيف ECI** فور تحميلها فيه حيثما يكون ذلك مطلوباً لمنع التلاعب بالمعلومات الحساسة أو رصد معلومات سرية.

4) يجب على أي تحقق لاحق من صورة **المضيف ECI** (في حالة أداة تحميل على مراحل) تنفذه برمجيات ناجمة عن صورة سابقة أن يستخدم نفس المفتاح العمومي **ECI HostCertificate** ونفس **قائمة الإبطال** من أجل عملية التحقق.

ويوصى بأن تستخدم أدوات التحميل المرحلية آلية آمنة واحدة للتحقق من صلاحية صور **المضيف ECI** التي استخدمت أيضاً للتحقق من صلاحية أول صورة تم تحميلها **للمضيف ECI**.

3.11 قواعد أداة التحميل الخاصة بالوسيط

توجد **أداة التحميل** الخاصة **بالوسيط ECI** في إطار **المضيف ECI**. يحدد **المضيف ECI** الصيغة الجذرية الدنيا **للسطح البيني ECI** وصيغة **قائمة الإبطال** الجذرية **للسطح البيني ECI** التي يستخدمها للتحقق من صلاحية **سلاسل الشهادات** قبل تحميل أي بند متعلق بالوسيط. ويجب أن تمتثل **أداة التحميل** الخاصة **بالوسيط ECI** للقواعد التالية:

1) يجب إزالة تجفير صورة **الوسيط ECI** أولاً إذا طُلب ذلك كما هو محدد في الفقرة 5.11.

2) يجب التحقق من صلاحية صورة **الوسيط ECI** والمفتاح POPK باستخدام سلاسل تمت معالجتها بالبرنامج الفرعي لمعالجة الشهادات (CPS) كما هو محدد في الفقرة 10. تتطلب هذه القاعدة درجة عالية من **المتانة**.

3) يجب التحقق في نفس الوقت من صورة **الوسيط ECI** أو **شهادة** صور الوسطاء ECI (حسب الاقتضاء) بواسطة المفتاح POPK و**قائمة إبطال** وسطاء **عمليات المنصة**. ويتم التحقق لاحقاً من صلاحية رقم الصيغة الخاصة **بقائمة الإبطال** هذه عند استهلال دورة **الفتحة AS** بواسطة **الوسيط ECI**.

4) يجب ألا يُسمح بتغيير أو رصد صورة من صور **الوسيط ECI** بعد تحميلها.

5) يجب ألا يكون **الوسطاء ECI** قادرين على "تحطيم جهازهم الافتراضي" ورصد أو تغيير سلوك **المضيف ECI** أو **الوسيط ECI**.

4.11 إنفاذ الإبطال

يستخدم **السطح البيني ECI** آلية إنفاذ متينة للتحقق من أوراق اعتماد **المضيف ECI** و**الوسيط ECI**. وبعمل ذلك في إطار القواعد التالية:

1) يجب أن يتوقف مزيل التخليط عن العمل إذا كانت الصيغة الجذرية **للسطح البيني ECI** ورقم أدنى صيغة **لقائمة الإبطال** الجذرية اللازم للتحق من **سلسلة شهادات المضيف ECI** أقل من القيم التي قام **المضيف ECI** بتحميلها عند الاستهلال. تتطلب هذه القاعدة درجة عالية من **المتانة**.

الملاحظة 1 – ينبغي أن يكون ذلك أمراً غير مألوف لأنه ينبغي تحديث **قوائم الإبطال** الجذرية **للمضيف ECI** بانتظام عبر قنوات جميع **المشغلين**، ولأن بإمكان **أداة التحميل** الخاصة **بالمضيف ECI** استخدام **قائمة الإبطال** الجذرية **للمضيف ECI**.

2) ترفض **الفتحة AS** تحميل أي **وسيط ECI** لا يمكن التحقق من صلاحية **سلسلة الشهادات** الخاصة به باستخدام الصيغة الجذرية **للسطح البيني ECI** ورقم أدنى صيغة **لقائمة الإبطال** الجذرية **للسطح البيني ECI** حدده **المضيف ECI** عند الاستهلال كما هو محدد في الفقرة 2.11. تتطلب هذه القاعدة درجة عالية من **المتانة**.

3) ترفض **الفتحة AS** إجراء حسابات المفاتيح إذا كان رقم الصيغة الجذرية الدنيا ورقم الصيغة الدنيا **لقائمة الإبطال** الجذرية اللذين يحتاج إليهما **الوسيط ECI** أقل من الأرقام التي قام **المضيف ECI** بتحميلها عند الاستهلال. ويحدد ذلك في قواعد الحساب المتعلقة بصورة الوسيط ومفاتيح التجفير وفك التجفير الواردة في الفقرة 4.2.8. تتطلب هذه القاعدة درجة عالية من **المتانة**.

الملاحظة 2 – تضمن هذه القواعد أن أنظمة أمن المحتوى قد تحتاج إلى حالة جذرية **للسطح البيني ECI** لتطبيقها من أجل التحقق من جميع البنود المحملة في **المضيف ECI** قبل متابعة أي عملية حساسة أمنياً.

5.11 فك تجفير صورة الوسيط

لأغراض فك تجفير صورة **الوسيط ECI** يمكن **لنظام الأمن المعزز** أن يفك تجفير مفتاح فك تجفير الصورة المجفرة الذي يوفره **مشغل الوسيط ECI** والقيام بفك تجفير صورة **الوسيط ECI** على النحو المحدد في الفقرة 8.7 من التوصية [ITU-T J.1012]. ويجب فك تجفير صور **الوسيط ECI** قبل التحقق من توقيع صورة **الوسيط ECI**. ووظيفة الأمن المعزّز المستخدمة لحساب مفتاح فك تجفير الصورة هي reqAsComputeImageKey المحددة في الفقرة 12.4.2.8. ويتلقى **المضيف ECI** من **المشغّل** الدخل V للمعلومات المطلوبة المتعلقة بالمفتاح المجفر (رسالة دخل إلى **آلية الاستيقان** يمكن من خلالها حساب مفتاح الاستيقان)، والمفتاح eKey (مفتاح فك تجفير الصورة الذي تم تجفيره بمفتاح الاستيقان) والمعلمتين "online" و"min_root_state" المعرفتين في الفقرة 8.7 من التوصية [ITU-T J.1012]، ويستخدم **الفتحة AS** المتوفرة للوسيط للقيام بفك التجفير (انظر الفقرة 8.7 من التوصية [ITU-T J.1012]). ويجب أن تكون أي قيمة ظرفية مستخدمة في دورة تبادل مفتاح فك تجفير الصورة جديدة (قيمة من إعادة استهلال **الفتحة AS**).

12 متطلبات التوقيت

1.12 مقدمة

يتعين على **الوسطاء ECI** إجراء عملياتهم مع التقيد بضغوط الوقت من أجل تلبية متطلبات النظام الأمني الذي يمثلون جزءاً منه. ويعتمد **الوسطاء ECI** على خصائص أداء معينة للوظائف التي يوفرها **نظام الأمن المعزز** (من خلال المضيف ECI). وتحدد هذه الفقرة خصائص التوقيت المتعلقة بوظائف **نظام الأمن المعزز**.

تقسم خصائص التوقيت المتعلقة لنظام الأمن المعزّز الوظائف إلى أربع فئات:

1) الوظائف التي تتطلب *وظائف إدارية* فقط في **الفتحة AS**.

2) الوظائف التي لا تتطلب سوى *عمليات تجفير متناظرة*، مثل عمليات حساب أو فك تجفير **سلم المفاتيح** بمفتاح الاستيقان (AK).

3) الوظائف التي تتطلب بين عملية واحدة وأربع *عمليات تجفير لاتناظرية* في **مجموعة سلم المفاتيح** أو النظام الفرعي لمعالجة الشهادات، مثل تحميل المفتاح LK1 وإجراء وظائف تتطلب حساب المفتاح AK.

4) الوظائف التي تتطلب معالجة **سلاسل شهادات** يمكن أن تكون أطول مثل سلاسل الاستيراد/التصدير وسلاسل الاستيقان من **الوسطاء الصغار**.

ويمكن **للوسيط ECI** أن يستدعي وظيفة من وظائف الفئات الثلاث الأخيرة من خلال رسائل غير متزامنة. أما وظائف الفئة الأولى فقد تكون متزامنة أو غير متزامنة.

وتستغرق عمليات التجفير اللاتناظرية المزيد من الوقت. ويجب على أي عملية تجفير لاتناظرية أن لا توقف عمل وظائف الفئتين الأوليين. وإذا تطلبت وظيفة من الفئة 1 أو 2 نتيجة من عملية في الفئة 3 أو 4، يكون **الوسيط ECI** مسؤولاً عن مزامنة نتيجة الوظيفة في الفئة 3 أو 4. أي إن عليه أن ينتظر حتى تصبح نتيجة العملية اللاتناظرية متوفرة (أي استلام رسالة النتيجة) قبل استدعاء وظيفة تعتمد على النتيجة.

2.12 الوظائف الإدارية

بالنسبة لوظائف الفئة 1)، تطبق المعايير العامة المتعلقة بالرسائل اللاتناظرية.

3.12 وظائف التجفير المتناظرة

يجب أن يقوم **نظام الأمن المعزز** بوظائف تستدعي عمليات تجفير متناظرة على أساس أن كل **دورة فتحة أمن معزز (AS)** يجب أن تكون قادرة على أداء وظيفة واحدة في إطار زمني معين. ويمكن العثور على القيمة المقترحة في الفقرة 3.6.6 من الإضافة [b-ITU-T J Suppl. 7].

4.12 وظائف التجفير اللامتناظرة

يجب أن يقوم **نظام الأمن المعزز** بوظائف تستدعي عمليات تجفير لاتناظرية (مثل استدعاء حسابات المفاتيح المتناظرة لسلم المفاتيح أو استخدام نتيجة **آلية الاستيقان**) على أساس أن كل **دورة فتحة أمن معزز (AS)** يجب أن تكون قادرة على أداء وظيفة واحدة في إطار زمني معين في المرة الواحدة. ويمكن العثور على القيمة المقترحة في الفقرة 4.6.6 من الإضافة [b-ITU-T J Suppl. 7].

الملحق A

تعاريف وظائف التجفير

(يشكل هذا الملحق جزءاً أساسياً من هذه التوصية.)

1.A وظيفة الاختزال

تستند وظائف الاختزال الواردة في هذه التوصية إلى الوظيفة SHA256 المحددة في المعيار [NIST FIPS 180-4] NIST FIPS PUB 180-4.

أما وظيفة *الاختزال* الواردة في الفقرة 2.7 فتساوي ()SHA-256 كما هي معرفة في المعيار [NIST FIPS 180-4].

وتستخدم الوظيفة بلغة c asHash (uchar *data, uint datalength, resultLength, uchar *result) الأثمونات التي تبدأ عند بيانات بطول dataLength مثل سلسلة الأثمونات *dataIn* وتقوم بحساب سلسلة الأثمونات resultOut كسلسلة أثمونات resultLength/8، وتخزنها كنتيجة وفقاً لما يلي:

resultOut = BS2OSP(truncate(SHA-256(OS2BSP(*dataIn*)),resultLength)))

ويكون resultLength عداً مضاعفاً للرقم 8. أما truncate فهي وظيفة تمثل البتر من اليسار لسلسلة البتات (المعلمة 1) التي يبلغ طولها (المعلمة 2) بتات.

وBS2OSP وOS2BSP هما وظيفتان تحولان سلسلة بتات إلى سلسلة أثمونات والعكس بالعكس كما هو محدد في الفقرة 9 من التوصية [ITU-T J.1015].

2.A التجفير اللاتناظري

يرد في الفقرتين 2.10 و3.10 من التوصية [ITU-T J.1015] تعريف لعمليتي التجفير وفك التجفير اللاتناظري.

3.A توليد الأرقام العشوائية

يمتثل توليد الأرقام العشوائية كما هو محدد في هذه التوصية للمرجع [NIST 800-90Ar1]، ويفي بالقواعد التالية:

- كحد أدنى عند إقلاع النظام (إعادة إقلاع **نظام الأمن المعزز** للشريحة)، يتم توليد رقم بداية سري عشوائي فريد. وتعتمد العملية على خصائص الفيزياء (الضوضاء) أو الخصائص الأخرى للشريحة أو بيئتها غير القابلة للتكرار ولا يمكن التعامل معها. ويجب أن تكون أنتروبيا الرقم المولّد 128 بتة على الأقل.
- يجب أن يتم توليد أي أرقام عشوائية بواسطة مولّد أرقام شبه عشوائية محددة استناداً إلى رقم البداية العشوائي الوارد أعلاه وفقاً للمرجع [NIST 800-90Ar1]. ويمكن للشريحة أن تعيد توليد رقم البداية بانتظام و/أو أن تزيد الأنتروبيا على النحو المحدد في الفقرة 7.8 من النشرة [NIST 800-90Ar1] باستخدام مدخلات داخلية (ضوضاء) أو خارجية يصعب التلاعب معها. وكحد أدنى يجب استعمال معرف هوية الشريحة في إحدى السلاسل الخاصة بإضفاء الطابع الشخصي.

ملاحظة – في الكثير من تطبيقات الأمن المعزز لا تكون العشوائية الفعلية لمولد الأرقام العشوائية حرجة، بعكس التفرّد مع الوقت الذي يكون كذلك. وهناك تطبيقات نموذجية ظرفية: مثل الرقم العشوائي للاستيقان على الشبكة من أجل منع إعادة العرض عند فك التجفير وإدراج رقم عشوائي عند تجفير المحتوى. والاستثناء هو المفتاح العشوائي المتولد على شكل LK1 في **الفتحة AS** المتعلقة بالتجفير بأسلوب **المخدم الصغير** اللاتناظري.

التذييل I

تطبيق نموذجي لنظام صغير لإدارة الحقوق الرقمية (DRM)

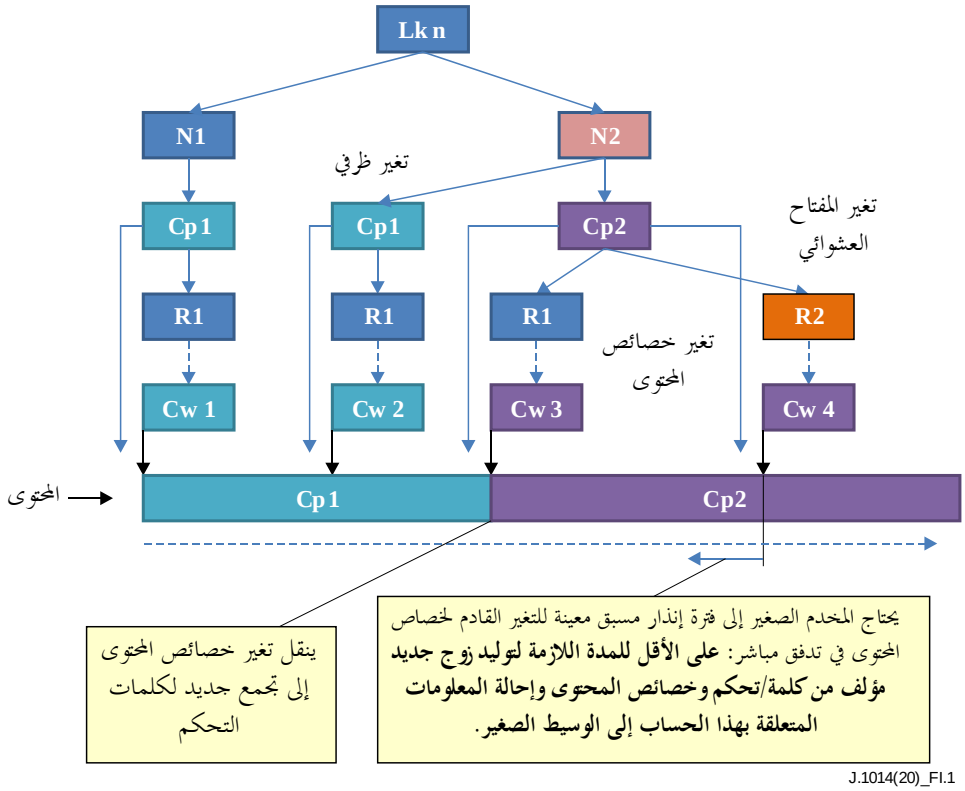
(لا يشكل هذا التذييل جزءاً أساسياً من هذه التوصية.)

1.I مقدمة

يقدم هذا التذييل مثالاً واقعياً لأحد تطبيقات **نظام الأمن المعزز** المتعلق بتنفيذ **نظام صغير لإدارة الحقوق الرقمية (DRM)** يعمل على تدفق TS (تدفق النقل). وتعرض فيه العملية النموذجية لتجفير وفك تجفير **الوسطاء ECI**. وينصب تركيز العرض على التوافق بين الإجراءات المتنوعة وتتابع كلمات التحكم والرسائل الصغيرة DRM المرتبطة بها (من **مخدم صغير** إلى **وسيط صغير** وبالعكس) والتي يتعين توليدها. ويستخدم **النظام DRM الصغير** توليد مفاتيح عشوائية عند التجفير وكذلك توليد مفاتيح ظرفية عند فك التجفير (لمنع إعادة العرض). ويفترض وجود حدّ لكلا النوعين من المفاتيح العشوائية.

2.I سيناريو التطبيق

يبين سيناريو التطبيق الوارد في الشكل 1.I حالة **سلم المفاتيح** في جانب التجفير. ويمثل LK_n المفتاح الثالث والأدنى في تراتبية المفاتيح. ويوجد تحته المفتاح الظرفي (N1 أو N2) من **الوسيط الصغير**، **وخصائص المحتوى** CP1 وCP2 (التي يتم نقلها إلى الدخل C **لسلم المفاتيح** في المرحلة n+2) ومفتاحا البداية العشوائيان R1 وR2 اللذان يشكلان الدخل للمرحلة n+3 **بسلم المفاتيح**. وانطلاقاً من مدخلات **سلم المفاتيح** يتم حساب كلمات التحكم CW1 إلى CW4 وتطبيقها على المحتوى بالترافق مع **خصائص المحتوى** المرتبطة بها.



الشكل 1.I – مثال على تطور تراتبية المفاتيح في حساب كلمات التحكم

تتمثل حالة البداية في المراحل الثلاث السفلى لتراتبية المفاتيح في **المخدم الصغير** بالقيم N1 وCP1 وR1. وانطلاقاً من هذه القيم يتم حساب كلمة التحكم CW1 اللازمة لتجفير المحتوى. وتمثل t1 الحالة الأولية لبتة الأرجحة. وفي هذا المثال يتلقى **المخدم الصغير** أولاً قيمة ظرفية جديدة ويحدد الوقت الذي يحين فيه تطبيقها على المحتوى المستقبلي على شكل كلمة تحكم CW2. فيرسل أولاً رسالة من نوع رسائل مراقبة الأحقية (ECM) إلى **الوسيط الصغير** مع بتة الأرجحة الجديدة t2 ومجموعة المفاتيح المجفرة (t2، N2، CP1، R1)، وينتظر لبعض الوقت للتأكد من إمكانية استلام **الوسيط الصغير** لكلمة التحكم الجديدة وإجراء حساب أولي لها والاستعداد للتغير المقبل. ثم يقوم بعد ذلك بحساب كلمة التحكم الجديدة CW2 نفسها ويلتزم بالتطبيق مطلقاً العنان لتغيير في بتة الأرجحة المتعلقة بالتدفق TS المجفر المرتبط بها.

ويبيّن الحدث التالي في الشكل 1.I تغيراً في **خصائص المحتوى** الذي يتعين تجفيره. حيث يتلقى **المخدم الصغير** رسالة من **المضيف ECI** تفيد بأن **خصائص المحتوى** ستتغير إلى CP2. ويرسل **المخدم الصغير** رسالة ECM إلى **الوسيط الصغير** مع المجموعة الجديدة من المفاتيح المجفرة (t3، N2، CP2، R1) ويجري حساباً أولياً للكلمة CW3 من تتابع المفاتيح هذا. وفي اللحظة التي تطبق فيها **خصائص المحتوى** الجديدة، تتغير بتة الأرجحة في المحتوى المجفر بصورة تلقائية وتطبق بفعالية كلمة التحكم الجديدة المتعلقة بالتجفير CW3 و**خصائص المحتوى** المرتبطة بها CP2.

ويتعلق الحدث الأخير في **المخدم الصغير** بتقرير تغيير المفتاح العشوائي R1 إلى R2. وهذه العملية مطابقة عملياً لعملية تغيير القيمة الظرفية. فيرسل **المخدم الصغير** رسالة ECM مع المجموعة (t4، N2، CP2، R2) إلى **الوسيط الصغير** يسمح فيها بإجراء حساب أولي للكلمة CW4، ويستخدم التأخير للتأكد من إتاحة الوقت الكافي **للوسيط الصغير** لكي يصبح جاهزاً. بعد ذلك يُستخدم **سلم المفاتيح** لحساب الكلمة CW4 ويطبقها على المحتوى، ما يسفر عن تغير حالة بتة الأرجحة في المحتوى إلى t4.

3.I الافتراضات والترميز

يقوم الوسيط القائم بالتصدير مع **الفتحة AS** المرتبطة به الخاصة بفك التجفير بتسليم المحتوى إلى وصلة الاستيراد في **الفتحة AS الخاصة بالتجفير**. ويولد الوسيط القائم بالتصدير رسائل إلى **المضيف ECI** تشير إلى أي تغير في **خصائص المحتوى** قبل حدوثه فعلياً في المحتوى المستورد. ويستخدم ذلك السطح البيني لبرمجة التطبيق في الأمن المعزز الوارد في التوصية [ITU-T J.1012].

ويستخدم الترميز التالي:

<event-name>(parameters) -> <pseudo-code statement>؛ ويدل على أنه في حال حصول حدث ينفذ event-name (استلام رسالة) مع شبه الشفرة التالية.

وتحدد الأحداث التالية:

- **e_cp(cp)**: تستخدم خصائص محتوى جديدة في الحدث المقبل (تغير الكلمة CW) للمحتوى المقرر تجفيره. يأتي هذا الحدث قبل ()e_cpe.
- **e_cpe()**: تغير وشيك لخصائص المحتوى (يحدث في غضون وقت محدد).
- **e_cpch()**: تغيرت للتو خصائص المحتوى المستورد، في الحالة التي لا تعبر فيها كلمة التحكم المستخدمة حالياً عن طلب تغير فوري. يأتي هذا الحدث قبل ()e_cw عند حصول تغير تلقائي لكلمة التحكم ناجم عن تغير في خصائص التحكم.
- **e_nn(nonce)**: وصول رسالة ظرفية جديدة من **الوسيط الصغير** إلى **المخدم الصغير** أو صدورها عن **الوسيط الصغير**.
- **e_cw()**: تغير بتة الأرجحة في المحتوى الذي أعيد تجفيره، وتطبق كلمة التحكم الجديدة CW (التي حسبت مسبقاً) على المحتوى.
- **e_ecm(<parameters>)**: استلام رسالة مع معلمات جديدة بشأن كلمة التحكم الجديدة المقرر استخدامها.
- يمكن إنشاء الاحداث بواسطة أجهزة توقيت.

cw(toggle_bit,random_key,nonce,content_properties) ويقوم بتوليد كلمة تحكم من أجل تجفير أو فك تجفير المحتوى باستخدام المعلمات المعينة. وتولد الرسالة أولاً في **المخدم الصغير** مع المعلمات ذاتها التي أحيلت إلى **الوسيط الصغير**، وتم استقبالها هناك على شكل e_ecm(…).

block_cpch() and unblock_cpch() ويستخدم الرسالة setAsPermitCPChange(…) لمنع أو إزالة منع التغيرات التلقائية في كلمة التحكم في التجفير الناجمة عن تغيرات في **خصائص المحتوى** المستورد.

changeCw(toggleBit) وتفرض تحولاً في كلمة التحكم (بتة الأرجحة في حقل التحكم في التخليط) في جانب التجفير باستخدام الرسالة setAsSC() كما هي محددة في الفقرة 9.9.

startTimer(timerHandle) وتقوم ببدء عمل المؤقت.

ويستخدم الترميز بالأسلوب c للمتغيرات وشبه الشفرة.

4.I شبه الشفرة الخاصة بالمخدم الصغير

يدل أحد التعقيدات الأساسية في **المخدم الصغير** على أن عليه التعامل مع عدة أحداث متزامنة وغير متزامنة يمكن أن تحدث تغيراً في كلمة التحكم:

- وصول محتوى يتطلب خصائص محتوى جديدة (تحدد بتطبيق الوسيط القائم بالتصدير لكلمة تحكم جديدة في المحتوى الذي يجري فك تجفيره)؛
- والانتهاء القريب لصلاحية القيمة الظرفية؛
- والانتهاء القريب لصلاحية المفتاح العشوائي.

ويتعين إعطاء الأولوية لمعالجة تغيير ما في **خصائص المحتوى** نظراً لوجود وقت محدود عادة قبل أن يؤدي تغير كلمة التحكم المقابلة في عملية فك التجفير إلى بدء تطبيق خصائص المحتوى الجديدة. ولذلك ينبغي أن تحدد أوقات انتهاء صلاحية المفتاح الظرفي والمفتاح العشوائي بشكل كاف ومتحفظ نظراً لإمكانية تأجيلها للفترة اللازمة لمعالجة تغير في خصائص المحتوى (بضع ثوانٍ عادة). ويفترض ذلك أن الوقت الذي ينقضي بين تغيرات خصائص المحتوى يكفي دائماً للسماح بالمعالجة اللازمة لتغير مفتاح ظرفي أو عشوائي واحد على الأقل في كلمة التحكم.

وهناك أولويتان لمعالجة التغيرات الوشيكة في المفتاح الظرفي أو العشوائي. في البداية يوضع المؤقت على الأولوية المنخفضة. وفي حالة عدم وجود تغير عالق في **خصائص المحتوى**، يتم تغيير المفتاح الظرفي أو المفتاح العشوائي، وإلا يضبط المؤقت على التغير العالي الأولوية. وقد يؤدي التغير العالي الأولوية في المفتاح الظرفي أو العشوائي إلى إلغاء تغير وشيك في خصائص المحتوى. ولكن **المخدم الصغير** قد يخلص إلى استنتاج خاطئ. وإذا حدث ذلك فإن تغير خصائص المحتوى يحدث قبل تطبيق تغير المفتاح الظرفي أو العشوائي على المحتوى. وفي هذه الحالة يتعين إعادة حساب كلمة تحكم جديدة تتضمن أيضاً **خصائص المحتوى** الجديدة. كذلك فإن التغير في خصائص المحتوى قد يحدث مباشرة بعد تطبيق تغير عالي الأولوية للمفتاح الظرفي أو العشوائي. وفي تلك الحالة تتأخر كلمة التحكم التي تعبر عن **خصائص المحتوى** الجديدة والرسالة ECM التي يقوم **المخدم الصغير** بحسابها.

وإذا أمكن ضبط قيمتي المؤقت TNONCEURGENT وTRKURGENT على قيمة أكثر من 10 ثوان زائد التأخير TECM وكان الوقت بين e_cpch() وe_cw(CPCHANGE) أقل من 10 ثوان، فمن الممكن أن تحدث مثل هذه التصادمات، لأنه يمكن تحديد مواعيد أي تغير في المفتاح العشوائي أو القيمة الظرفية قبل الفترة الممتدة بين e_cpch() وe_cw(CPCHANGE) أو بعد هذه الفترة من دون الأولوية التي تتطلب نشوءها.

تجدر الإشارة إلى أن معالجة المتغيرين rc وrn على النحو المحدد أدناه لا يمكن أن يقوم بها الوسيط مباشرة ولكن يجب القيام بها باستخدام وظائف **نظام الأمن المعزز**.

```
/*
   four priority processing model with small shift of CP change time
   in case priority 4 is required (here & now non-anticipated change in CP):
    1) low priority nonce/rk change
    2) low priority CP change (cp eminent but e_cpch() did not occur)
       adopts any previous nonce or rk changes
    3) high priority nonce or rk change; reverts to old CP value
    4) high priority CP change; adopts pending nonce/rk changes and new cp;
       queues new changes

    Optimization may be possible to try to schedule pending nonce and rk changes
    immediately after a CP change; provides modest performance improvement

    State variable invariants/meanings:
    <x> = cp (content property), n (nonce) or r (random key)
    Invariant: p<x> = change in <x> in next CW (p = pending)
                      (not for low priority <n> or <r>)
               q<x> = queud change for <x>, not pending for next CW
               hpcp = high priority content property change (pcp || qcp)
   During a brief time between changeCw() and e_cw() all changes are queued.
   This temporary state is indicated with dhp==true;
 */

#define TECM 3000 /* delay between sending ecm message and changing CW */
#define TNONCEURGENT (2*TEMC + 1000)
#define TRKURGENT (2*TECM + 1000)
#define TNONCE /* some value; may be dynamically determined*/
#define TRK    /* some value; may be dynamically determined*/

toggle(bool t) { return !t }; /* toggles between true and false */

encryptionSession()
/* case rk & nonce change and cp change; unreliable warning cp change (priority with nonce/RK
change) */
/* first priority on nonce/rk lower than cp, but if urgent it is higher */
{
  SymKey  nc, nn;   /* current and next nonce */
  SymKey  rc, rn;   /* current and next Random Key */
  SymKey  cpc, cpn; /* current and next CP value */
  SymKey  nt, rt;   /* temporary value for nonce, random key */
  TimerHandle t_lpn, t_lpr;
              /* timers for low priority scheduling of nonce and rk change */
  TimerHandle t_n, t_r;
              /* timers for high priority scheduling of nonce and rk change */
  TimerHandle t_ecm_n1, t_ecm_r1;
              /* ecm timers for low priority (1) nonce and rk  ecm */
  Timerhandle t_ecm0, t_ecm1 t_ecm2, t_ecm3;
  TimerHandle t_ecm[4] = {t_ecm0, t_ecm1, t_ecm2, t_ecm3 };
              /* four level 2/3/4 priority level ecm timer pool */
  int t_ecm_cnt = 0;  /* counter for above timer pool allocation */
  bool pn, pr, pcp;   /* true if current CW reflects a change in nonce (nn),
                         random key (rn) or cp (cpn) value */
  bool qn, qr, qcp; /* true if a queued change in nonce, random-key or cp change */
  bool dhp;         /* delay (queue) any new events */
  book hpcp;        /* true if priority 4: high priority CP change */
  int  tCnt1, tCnt234;  /* tCnt<n> is the counter for number of timers
                        in priority <n> that are fired but not yet expired */
  bool t;           /* toggle bit */

/* some macro's are defined to permit reuse of code for processing events */

.* event for next random key */
#define next_r() { rc = rn; rn = rnd128(); startTimer(t_lpr,TRK); }


/* force changeCw on last cascaded higher priority timer unless it is a level 2
   priority cp change in which case the change of CW will be triggered by a CP
   change event */
#define process_emc2_timer(){\
  if (--tCnt234 == 0)\
    if (pn || pr || hpcp){\
      dhp = true; changeCw(toggle(t));\
    } else {\
      /* pcp == true, pn, pr, hpcp == false */\
      unblock_cpch();\
    };\
}

/* on cw-change update state with all processed changes */
#define end_pending() {\
```

```
  t = toggle(t);\
  if (pcp) { cpc = cpn; pcp = false };\
  if (pn)  { nc = nn; pn = false };\
  if (pr)  { next_r(); pr = false };\
}

/* move queued events to pending */
#define queued_to_pending() {\
  if (qcp && (!(qn || qr) || cphp)) {
    /* if priority 2 or 4 */\
    pcp = true; qcp = false\
  };\
  /* priority 3 events can be folded with priority 4 */
  if (qn)  { pn = true; qn = false };\
  if (qr)  { pr = true; qr = false };\
}

/* start cw/ecm for pending changes to cw */
#define start_pending() {\
  cnt = 0;\
  if (pcp) { cpt = cpn; cnt++ } else cpt = cpc;\
  if (pn)  { nt  = nn ; cnt++ } else nt  = nc;\
  if (pr)  { rt  = rn ; cnt++ } else rt  = rc;\
  if (cnt > 0) {\
    block_cpch();\
    cw(t,rt,nt,cpt);\
    tCnt234++;\
    startTimer(t_ecm[t_ecm_cnt++],TECM);\
    if (t_ecm_cnt >=4) t_ecm_cnt = 0 ;\
  }\
}


  /* only permit auto-changes of toggle bit when prepared */
  block_cpch();

  /* receive first cp and nonce values */
  for (int i=0; i<2;) {
    ->e_nn(&nc): i++;
    ->e_cp(&cpc): i++;
  }

  /* initialise state */
  pn = pr = pcp = hpcp = false;
  dhp = false;
  tCnt1 = tCnt2 = 0;
  rc = rnd128(); rn = rnd128() ;
  t = false;          /* should be initialised to first value in content */
  cw(t,rc,nc,cpc) ;   /* will start to be used automatically */

  while (!end_session) {
  ->e_nn(&nn) : startTimer(t_lpn,TNONCE) ;
                /* should occur before nonce limit runs out */
  ->e_cp(&cpn): /* e.g. compute new export licenses */ ;
  ->t_lpn()   : { /* low priority nonce change */
      if (pcp || pn || pr || cphp) {
         /* delay new nonce till urgent */
         startTimer(t_n,TNONCEURGENT);
      } else {
        nc = nn;
        cw(t,rc,nc,cpc);
        startTimer(t_ecm_n1,TECM) ;
        tCnt1++;
      }
    };
  ->t_lpr()   : { /* low priority rk change */
      if (pcp || pn || pr || cphp) {
        /* delay RK till urgent */
        startTimer(t_r,TRKURGENT);
      } else {
        next_r();
        cw(t,rc,nc,cpc);
        startTimer(t_emc_r1,TEMC);
        tCnt1++;
      }
    };
  ->t_emc_n1() :   /* low priority nonce ecm timer expiry */
  ->t_emc_r1() : { /* low priority rk ecm timer expiry */
      if (--tCnt1 == 0 && tCnt234 == 0) {
        changeCw();
```

```
        dhp = true;
      }
    };
  ->e_cpe()   : { /* cp change may occur from now on */
      if (dhp || (pn || qn)) qcp = true; /* assert(!hpcp) */
      else { pcp = true; start_pending() };
    };
  ->t_n()     : { /* urgent nonce change due */
      if (dhp || cphp) qn = true;
      else { pn = true; start_pending() };
    };
  ->t_r()     : { /* urgent random key change due */
      if (dhp || cphp) qr = true;
      else { pr = true; start_pending() };
    };
  ->e_cpch()  : { /* high priority change of CP needed */
      cphp = true;
      if (dhp) qcp = true;
      else {
        pcp = true;
        start_pending();
      }
    };
  ->t_ecm0()  :
  ->t_ecm1()  :
  ->t_ecm2()  :
  ->t_emc3()  : {
      process_timer();
    };
  ->e_cw()    : { /* assert( pcp && !pn && !pr && !cphp ) */
      end_pending();
      queued_to_pending();
      start_pending();
    };
  }
}
```

الملاحظة 1 – في المثال المقدم أعلاه المتعلق **بالمخدم الصغير**، يستطيع **المخدم الصغير** أن يولد عدداً من الرسائل ECM المتتابعة ولكن المختلفة ذات بتة الأرجحة ذاتها والتي يمكن أن يستلمها الوسيط الصغير. والمثال الأقصى هو أن الوظيفة e_n() تحدث أولاً، ثم تحدث e_r() خلال مدة التأخير TDELAY وبعد TDELAY تحدث e_cp(..). وفي هذه الحالة ترسل ثلاث رسائل ECM متتالية، بنفس بتة الأرجحة، حيث تؤدي الأخيرة فقط إلى كلمة تحكم تطبق فعلياً على المحتوى.

الملاحظة 2 – تفترض الشفرة الواردة أعلاه أن تغير خصائص المحتوى e_cp() يعقبه دائماً تغير سريع نسبياً لبتة الأرجحة الفعلية. وإذا أتيحت قيمة cp الجديدة في وقت قريب جداً فلن يكون لذلك أي فائدة **للمخدم الصغير**. فنقطة الانطلاق الأساسية لتوليد كلمة تحكم جديدة هي الحدث المتمثل بتغير وشيك في خصائص المحتوى الوارد. يؤدي ذلك إلى الاستعاضة عن القيمة cp القديمة بالقيمة الجديدة في جميع حسابات كلمات التحكم القادمة.

وتتمثل مدة الإنذار المسبق الدنيا لإطلاق الوظيفة e_n() أو e_r() في عيّنة الشفرة الواردة أعلاه بأسوأ حالة تأخر بين حدث من نوع e_cp() والتغير الفعلي لكلمة التحكم اللاحقة e_cw()، زائد TDELAY × 2 زائد قيمة صغيرة لتأخر الأحداث ومدة المعالجة.

5.I شبه الشفرة الخاصة بالوسيط الصغير

يبدأ **الوسيط الصغير** دورة ما بتوليد رسالتين ظرفيتين متلاحقتين (للقيمة الظرفية الحالية والتالية). فإذا استلم رسالة ECM، فإنه يقوم ببساطة بحساب كلمة التحكم المقابلة. ويستمر بتوليد قيمة ظرفية جديدة وإرسال رسالة ظرفية جديدة بمجرد أن يتبين أن القيمة الظرفية الأخيرة التي أرسلها يجري تطبيقها في رسالة ECM.

الملاحظة 1 – لا يمكن توليد قيم ظرفية آمنة مباشرة بواسطة شفرة **الوسيط ECI** ولكن يتعين استخدام الوظيفة المناسبة **لنظام الأمن المعزّز**.

```
decryptionSession()
{
  Symkey nc, nn, ln;   /* current, next and last nonce */
  SymKey cp, cpp;  /* received and previous cp */
  SymKey r;        /* received random key */
  bool   t;        /* received toggle bit */
  SymKey n;        /* received nonce */
  bool   end_session; /* end of session reached */


  /* initialise and send nonces */
  nn = rnd128();
  e_nn(nn);
```

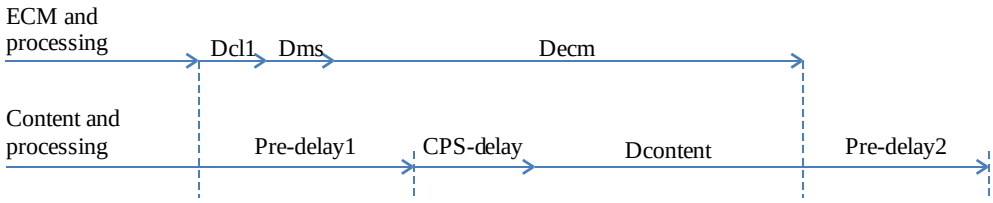
```
  cpp = Reserved; /* undefined value */
  ln  = Reserved;

  while (!end_session) {
    ->e_ecm(&t,&r&n&cp): {
        if (cp!=cpp) { /* new CP; send event to all export connections via host */
          e_cp(cp);
          cpp = cp;
        }
        cw(t,r,n,cp);
      };
    ->e_cw(): {         /* also triggered on first cw application */
        if (n != ln) {   /* new nonce actually used; move nonce forward */
          nc= nn; nn= rnd128();
          e_nn(nn);
          ln = n;
        }
      };
  } /* end while loop */
} /* end decryption session */
```

الملاحظة 2 – ليس من الضروري إعادة إرسال القيم الظرفية الكاملة من **مخدم صغير** إلى **وسيط صغير**. ويمكن استخدام بتة متناوبة بدلاً من مرجع غير مباشر. بالإضافة إلى ذلك، ليس من الضروري قطعاً إرسال جميع المعلمات في جميع الرسائل ECM: فالتغيرات فقط هي التي يجب إبلاغها إلى **الوسيط الصغير**، مع ضرورة الإشارة في بعض الحالات إلى أن جميع القيم الظرفية الثلاث لمدخلات **سلم المفاتيح** و**خصائص المحتوى** والمفتاح العشوائي قد تتغير على الفور (انظر الملاحظة 2 في الفقرة 4.I). ويعتبر إرسال بتة الأرجحة إلى جانب ذلك مفيداً للمزامنة ويتفادى عدم تفسير (بقصد أو بغير قصد) أي رسالة ECM مكررة على أنها رسالة لحساب كلمة تحكم تالية.

6.I الأثر التسلسلي لنظام DRM الصغير على التأخر المسبق لرسائل ECM

يعتمد **نظام DRM الصغير** على فترة إنذار مسبق (تأخير مسبق) بوجود تغير مقبل في خصائص المحتوى من **الوسيط الصغير** الذي يستورد المحتوى منه للسماح بإجراء حساب مسبق لإحدى الرسائل ECM وإرسالها إلى **الوسيط الصغير** قرينه. وقد يكون الوقت اللازم للقيام بالمعالجة المطلوبة (الذي قد يكون قصيراً نسبياً: إذ لا يطلب في العادة إجراء حسابات مهمة) بإضافة إلى الوقت اللازم لإحالة هذه الرسالة ECM إلى **الوسيط الصغير** أطول من المسار المستخدم لنقل المحتوى الذي تم تجفيره مؤخراً إلى **الوسيط الصغير**. ويعني ذلك أن أي تأخير مسبق للرسالة ECM الجديدة الذي يتعرض له **الوسيط الصغير** يكون أقصر بصورة مناظرة من التأخير الذي يتعرض له **الوسيط ECI** الذي استُورد منه المحتوى بالأساس.



Dcl1 = Delay ECI Client 1	CPS-delay = Content processing system delay
Dms	Dcontent = Delay Content transfer to Micro Client
Dms = Delay ECM	Pre-delay1 = Predelay ECM at Decryption ECI Client #1
	Pre-delay2 = Predelay ECM at Decryption ECI Client # 2

J.1014(20)_FI.2

الشكل 2.I: العلاقات الزمنية بين التأخير المسبق والتعويض الاختياري عن التأخير

ويمكن **لمسار فيديوي آمن** أن يدخل تأخيراً في نقل المحتوى للتعويض عن التأخير الذي يحصل عند إحالة الرسائل ECM كما هو مبين في الشكل 2.I. ويمكن بعد ذلك اختيار هذا التأخير بحيث يساوي تقريباً فارق التأخير. وإذا كانت الرسائل ECM مدرجة يكون كل من Decm وDcontent متوائمين تقريباً. ومع ذلك ينبغي التعويض عن قيم تأخير المعالجة في **الوسيط ECI** القائم بفك التجفير وتلك التي تحدث في **المخدم الصغير** حتى لحظة الإدراج الفعلي للرسائل ECM في تدفق النقل.

7.I اصطلاح السطح البيني لتوقيت التغير في خصائص المحتوى

كما بيّنت الفقرة 4.I، يحتاج **المخدم الصغير** إلى إنذار مسبق بوجود تغير مقبل في خصائص المحتوى المستورد إليه. ويشار إلى الاصطلاح الخاص بالفترة الزمنية الدنيا المطلوبة لمعالجة التغيير وإرسال رسالة ECM إلى **المخدم الصغير** بالرمز TECM: ويرد مثال على ذلك في الفقرات التالية من هذا التذييل. وبالنسبة لهذا المثال تضبط قيمة TECM على 3 s.

ويتمثل الاصطلاح المتعلق بالحد الأدنى من التأخير الناجم عن الإنذار المسبق في أول **وسيط ECI** يقوم بفك التجفير في سلسلة من **الوسطاء ECI** بالقيمة TECM + TCASCADE. وتعبر الكمية TCASCADE عن التأخير التراكمي الأقصى لمعالجة الرسائل ECM بواسطة **الوسطاء ECI** في سلسلة **أنظمة DRM الصغيرة**. وفي هذا المثال تضبط TCASCADE على القيمة 2 s.

الملاحظة 1 – في الحالات التي يتأخر فيها المحتوى، تعوض هذه القيمة عن تأخير معالجة الرسائل ECM. ومع ذلك فهذا الأمر غير مستحسن في أسلوب التدفق.

ويشار إلى أقصى تأخير في معالجة الرسائل ECM الخاصة **بالوسيط ECI** (Dcl1 غير المعوضة + Dms كما في الفقرة 6.I) بالقيمة TDELAY، وتضبط في هذا المثال على القيمة 0,3 s. وتسمح القيم الواردة في هذا المثال لتسلسل من 6 **أنظمة DRM صغيرة** بالعمل ضمن الفترة TCASCADE (2 s)، تاركاً فترة دنيا TECM للإنذار المسبق **للمخدم الصغير**.

وكما بيّنت الفقرة 4.I، ومن أجل معالجة تغير خصائص المحتوى في المحتوى القادم دون التسبب بتحول في تغير خصائص المحتوى، لا تحتاج **المخدمات الصغيرة** إلى إنذار مسبق بحدوث تغير في خصائص المحتوى فحسب بل تحتاج أيضاً إلى حد أعلى لفترة الإنذار المسبق هذه بحيث يمكنها معالجة تغيرات أخرى في كلمات التحكم (مثل تغيرات المفتاح الظرفي والمفتاح العشوائي) بشكل آمن. وفي هذا المثال يمكن ضبط الحد الأعلى لفترة الإنذار المسبق بشكل آمن على 10 s.

الملاحظة 2 – في حالة عدم التقيد بهذه الاصطلاحات، قد يتمثل التأثير بتحول أقصى لقيمة TECM في موقع تغير خصائص المحتوى في المحتوى الذي أعيد تجفيره في أحد **أنظمة DRM الصغيرة** وتحول أقصى في TECM+6*TCASCADE ضمن تسلسل من 6 **أنظمة DRM صغيرة**.

ويوصى بشدة بتصميم إنذارات منخفضة الأولوية للمفتاح الظرفي والمفتاح العشوائي (t_lpn وt_lpr في الفقرة 4.B) تطلق قبل وقت كافٍ بحيث تسمح لتغير واحد (أو حتى لبضعة تغيرات) في خصائص المحتوى بتأخير معالجة تغيرات المفتاح الظرفي والمفتاح العشوائي. وإذا كانت تغيرات خصائص المحتوى متباعدة زمنياً بشكل كافٍ (وتمت مراعاة TMAXWARN)، فينبغي أن يمنع ذلك أي حالات إلغاء في معالجة تغيرات المفتاح الظرفي والمفتاح العشوائي.

ويعتبر اختيار معلمات التوقيت مهماً من أجل الانتقال السلس للمحتوى بين **الوسطاء ECI**. وترد في الفقرة 6 من الإضافة [b-ITU-T J.Suppl.7] معلومات أوفى عن القيم المقترحة لمعلمات التأخير TECM وTCASCADE وTDELAY وTMAXWARN.

التذييل II

مجالات تحتاج لمزيد من التطوير

(لا يشكّل هذا التذييل جزءاً من هذه التوصية.)

تبيَّن أن هذه التوصية في حاجة إلى المزيد من التطوير وإلى التحقق من أنها تفي بالمتطلبات المحددة في التوصية [ITU-T J.1010]، وأن التوصية [ITU-T J.1010] في حاجة إلى التحديث كي تعبِّر عن متطلبات مواصفة الحماية المعززة للمحتوى (ECP) الصادرة عن منظمة MoviLabs [b-ECP]. وينبغي في المستقبل تحديث التوصيات [ITU-T J.1011] و[ITU-T J.1012] و[ITU-T J.1013] و[ITU-T J.1014] و[ITU-T J.1015] و[b-ITU-T J.1015.1] لكي تعبِّر عن التحديثات المدخلة على التوصية [ITU-T J.1010].

وأعرب عدد من الدول الأعضاء وأصحاب المصلحة من مجموعة متنوعة من الصناعات – بما في ذلك مصنعو الأجهزة والمكونات الإلكترونية ومالكو وحائزو تراخيص المحتوى المحمي بحقوق الطبع والنشر وموردو الخدمات المستقلة عن المشغِّل (OTT) وخدمات التلفزيون الخطي وموردو حلول أنظمة النفاذ المشروط (CAS) وإدارة الحقوق الرقمية (DRM) – المنتشرة في شتى أرجاء العالم، عن مخاوفهم من أن السطح البيني ECI لا يفي تماماً بمتطلبات الحماية ECP ولا بمتطلبات حماية المحتوى لدوائر الصناعة على نطاق أوسع.

وبعبارة أدق، أبدى هؤلاء مخاوفهم في مساهمات قُدمت إلى اجتماع لجنة الدراسات 9 لقطاع تقييس الاتصالات (SG9) (16-23 أبريل 2020). واقترحت مساهمات مقدمة من إسرائيل وأستراليا وشركة Samsung، أحد أعضاء قطاع تقييس الاتصالات، وSky Group وMoviLabs، من المنتسبين إلى لجنة الدراسات 9، أن هناك عدداً من التغييرات يجب إدخالها على توصيات السطح البيني ECI، ولكن تعذر التوصل إلى اتفاق بشأنها. وترد قائمة جرد لهذه البنود في المرجع [b-SG9 Report 17 Ann.1].

وهي تتضمن مقترحات من أجل:

1) تبسيط النظام ECI بتضييق نطاقه؛
2) إلغاء إدارة الحقوق الرقمية (DRM)؛
3) إلغاء إعادة تجفير المحتوى؛
4) إلغاء إدارة البرمجيات؛
5) إضافة سطوح بينية لبرمجة التطبيق (API) توخياً لتأمين عمليات التخزين والتجفير؛
6) السماح بسلالم المفاتيح الخاصة بالبائعين؛
7) استخدام متطلبات TEE للتوصية J.1207؛
8) إضافة تنفيذ TEE إلى VM؛
9) تعزيز قوة خوارزميات التجفير، باستعمال SHA-384 مثلاً؛
10) استعمال شهادات معيارية، مثل التوصية ITU-T X.509؛
11) إعادة النظر في الاتصالات بين الوسطاء؛
12) إجراء مزيد من الاتصالات مع معهد ETSI؛
13) إجراء المزيد من استعراضات النظراء؛
14) استقصاء بدائل لنموذج سلطة الاستوثاق؛
15) تحديد المزيد من الجوانب التقنية لقواعد التزام السطح البيني ECI ومتانته؛

16) إضافة متطلبات من أجل التنوع، مثل التحديد العشوائي لحيز العنوان؛

17) إضافة متطلبات بشأن التحقق من سلامة وقت التشغيل.

وتبين هذه المقترحات أن حماية المحتوى والتهديدات الخاصة بإصابتها بالخلل تتطور باستمرار. وقد صمم السطح البيني ECI في الأصل قبل الموافقة على توصية قطاع تقييس الاتصالات هذه بعقد من الزمن تقريباً. والأنظمة من شاكلة السطح البيني ECI تحتاج إلى التقييم بصورة منتظمة على محك أحدث تقنيات الهجمات ومتطلبات حماية الصناعة، على السواء.

وهناك آليات أخرى لتمكين قابلية التشغيل البيني. وبالنسبة لحالة استعمال إدارة الحقوق الرقمية، بوجه خاص، نشرت معظم خدمات الفيديو عبر الإنترنت حلولاً أخرى لتحقيق قابلية التشغيل البيني ولتلبية هذه الاحتياجات.

وزيادة الوضوح مهمة في هذا الصدد لأن الكثير من الدول الأعضاء تعتبر معايير الاتحاد مصادر توجيه مؤثرة في تطوير أسواقها وصناعاتها. وتؤكد قائمة الشواغل أن تنفيذ السطوح ECI في الأسواق المحلية لهذه الدول يمكن أن ينطوي على تقدير كامل لآثار توصية قطاع تقييس الاتصالات هذه ويضمن أخذ هذه القضايا بعين الاعتبار عندما تتطلب التشريعات أو اللوائح أو الاحتياجات السوقية من معدات التلفزيون الرقمي الاستهلاكية بأن تكون قابلة للتشغيل البيني. ويضمن ذلك أيضاً في حالة مصنعي المعدات التكنولوجية، الذين قد يفضلون استخدام مجموعة فريدة من المتطلبات أو معايير أخرى لتصميم المنتجات، إمكانية أن يأخذوا هذه القضايا بعين الاعتبار عند تطوير منتجات لأسواق مختلفة.

بيبليوغرافيا

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

[b-ITU-T J Suppl. 7] Supplement 7 to the ITU-T J series Recommendation(2020), *Embedded Common Interface for exchangeable CA/DRM solutions; Guidelines for the implementation of ECI*.

[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-ETSI GS ECI 001-1] ETSI GS ECI 001-1 V1.2.1 (2018), *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 1: Architecture, Definitions and Overview*.
https://www.etsi.org/deliver/etsi_gs/ECI/001_099/.../gs_ECI00101v010101p.pdf

[b-ETSI GS ECI 001-2] ETSI GS ECI 001-2 V1.2.1 (2018), *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 2: Use cases and requirements*.
https://www.etsi.org/deliver/etsi_gs/ECI/001_099/.../gs_ECI00102v010201p.pdf

[b-ETSI GS ECI 001-3] ETSI GS ECI 001-3 V1.1.1 (2017), *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 3: CA/DRM Container, Loader, Interfaces, Revocation*.
https://www.etsi.org/deliver/etsi_gs/ECI/001_099/.../01.../gs_ECI00103v010101p.pdf

[b-ETSI GS ECI 001-4] ETSI GS ECI 001-4 V1.1.1 (2017), *Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 4: The Virtual Machine*.
https://www.etsi.org/deliver/etsi_gs/ECI/001_099/.../gs_ECI00104v010101p.pdf

[b-ETSI GS ECI 001-5-1] ETSI GS ECI 001-5-1 V1.1.1 (2017-07), Embedded Common Interface (ECI) for exchangeable CA/DRM solutions; Part 5: The Advanced Security System; Sub-part 1: ECI specific functionalities
https://www.etsi.org/deliver/etsi_gs/ECI/001_099/0010501/01.01.01_60/gs_ECI0010501v010101p.pdf

[b-ETSI GS ECI 001-5-2] ETSI GS ECI 001-5-2 V1.1.1 (2017), *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/.../gs_ECI0010502v010101p.pdf

[b-ETSI DVB CSA] ETSI: *Using the DVB CSA algorithm* (licencing arrangement).
http://www.etsi.org/about/what-we-do/security-algorithms-and-codes/csa-licences

[b-ETSI DVB CSA3] ETSI: *Using the DVB CSA3 algorithm* (licensing conditions).
http://www.etsi.org/about/what-we-do/security-algorithms-and-codes/csa3-licences

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

سلاسل التوصيات الصادرة عن قطاع تقييس الاتصالات

السلسلة A	تنظيم العمل في قطاع تقييس الاتصالات
السلسلة D	مبادئ التعريفة والمحاسبة والقضايا الاقتصادية والسياساتية المتصلة بالاتصالات/تكنولوجيا المعلومات والاتصالات على الصعيد الدولي
السلسلة E	التشغيل العام للشبكة والخدمة الهاتفية وتشغيل الخدمات والعوامل البشرية
السلسلة F	خدمات الاتصالات غير الهاتفية
السلسلة G	أنظمة الإرسال ووسائطه والأنظمة والشبكات الرقمية
السلسلة H	الأنظمة السمعية المرئية والأنظمة متعددة الوسائط
السلسلة I	الشبكة الرقمية متكاملة الخدمات
السلسلة J	**الشبكات الكبلية وإرسال إشارات تلفزيونية وبرامج صوتية وإشارات أخرى متعددة الوسائط**
السلسلة K	الحماية من التداخلات
السلسلة L	البيئة وتكنولوجيا المعلومات والاتصالات، وتغير المناخ، والمخلفات الإلكترونية، وكفاءة استخدام الطاقة، وإنشاء الكبلات وغيرها من عناصر المنشآت الخارجية وتركيبها وحمايتها
السلسلة M	إدارة الاتصالات بما في ذلك شبكة إدارة الاتصالات وصيانة الشبكات
السلسلة N	الصيانة: الدارات الدولية لإرسال البرامج الإذاعية الصوتية والتلفزيونية
السلسلة O	مواصفات تجهيزات القياس
السلسلة P	نوعية الإرسال الهاتفي والمنشآت الهاتفية وشبكات الخطوط المحلية
السلسلة Q	التبديل والتشوير، والقياسات والاختبارات المرتبطة بهما
السلسلة R	الإرسال البرقي
السلسلة S	التجهيزات المطرافية للخدمات البرقية
السلسلة T	المطاريف الخاصة بالخدمات التليماتية
السلسلة U	التبديل البرقي
السلسلة V	اتصالات البيانات على الشبكة الهاتفية
السلسلة X	شبكات البيانات والاتصالات بين الأنظمة المفتوحة ومسائل الأمن
السلسلة Y	البنية التحتية العالمية للمعلومات، والجوانب الخاصة ببروتوكول الإنترنت وشبكات الجيل التالي وإنترنت الأشياء والمدن الذكية
السلسلة Z	اللغات والجوانب العامة للبرمجيات في أنظمة الاتصالات

طُبعت في سويسرا
جنيف، 2020