International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

J.1002

(03/2013)

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

Conditional access and protection

Pairing protocol specification for renewable conditional access system

Recommendation ITU-T J.1002

ITU-T

International Telecommunication Union

Recommendation ITU-T J.1002

Pairing protocol specification for renewable conditional access system

Summary

Recommendation ITU-T J.1002 specifies the pairing protocol that supports the conditional access module (CAM) and descrambler (DSC) pairing function, which is specified in Recommendation ITU-T J.1001.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T J.1002	2013-03-01	9

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

Table of Contents

Introduction

Recommendation ITU-T J.1001 specifies the requirements for renewable conditional access system (RCAS), and it identifies the pairing protocol that is one of the functional requirements.

The RCAS is a new paradigm technology for renewing conditional access (CA) client software by securely downloading the new version of software through the digital cable two-way environment. The benefit of RCAS is that no additional budget is required for issuing a new security hardware module when the multiple systems operator (MSO) wants to upgrade the old CA client software to a new one.

The pairing protocol is an authentication protocol between the conditional access module (CAM) and descrambler (DSC). The authentication process between the CAM and DSC is one of the most important security requirements for the RCAS. If the pairing is not performed properly, it may cause a control word (CW) disclosure problem. For example, a hacked DSC could intercept CWs transferred from the CAM through impersonation attack. As a result, a hacker could watch pay broadcasting programs without proper entitlement by taking advantage of the intercepted CW.

If the pairing is not performed properly, this may cause another problem – of managing paid-viewers. For example, a malicious user could remove the physically-implemented CAM from one set-top box that stores entitlement information, and connect the removed CAM to another set-top box. Then a malicious user could watch pay broadcasting programs on multiple set-top boxes with one CAM. As a result, MSO cannot properly manage pay subscribers, and undergoes unwanted business losses.

To prevent the above drawbacks, a pairing protocol is specified in this Recommendation, which can provide a mutual authentication and security channel establishment between the CAM and the DSC. Using the pairing protocol can efficiently prevent a hacked DSC from eavesdropping CWs, which are transferred from the CAM to DSC, as well as unwanted usage of one CAM to multiple set-top boxes.

Recommendation ITU-T J.1002

Pairing protocol specification for renewable conditional access system

1 Scope

This Recommendation specifies the pairing protocol that provides the conditional access module (CAM) and descrambler (DSC) pairing function of renewable conditional access system (RCAS), which is specified in [ITU-T J.1001].

2 References

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

[ITU-T J.1001] Recommendation ITU-T J.1001 (2012), *Requirements for renewable conditional access system*.

[ITU-T X.509] Recommendation ITU-T X.509 (2008) | ISO/IEC 9594-8:2008, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 conditional access (CA) [b-ITU-T J.193]: The conditional granting of access to cable services and content based upon what service suite has been purchased by the customer.

3.1.2 descrambling [b-ITU-T J.93]: The processes of reversing the scrambling function (see "scrambling") to yield usable pictures, sound, and data services.

3.1.3 entitlement control messages (ECMs) [b-ITU-T J.290]: An ECM is an encrypted message that contains access criteria to various service tiers and a control word (CW).

3.1.4 entitlement management messages (EMMs) [b-ITU-T J.290]: The EMM contains the actual authorization data and shall be sent in a secure method to each CPE device.

3.1.5 scrambling [b-ITU-T J.93]: The process of using an encryption function to render television and data signals unusable to unauthorized parties.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 authorization centre (AC): An entity which issues identification information of CAM and performs authentication process when CAM requests renewing of CACS.

3.2.2 conditional access module (CAM): A cryptographic functional module which is located in set-top boxes, whose main function is entitlement validation, key management and authentication. Set-top boxes can have one chip of secure hardware that includes the functions of CAM and

descrambler, or physically separated CAM in the form of a secure hardware IC or smart-card. The form of CAM can be determined by the policy of the MSO or CAS vendor.

3.2.3 conditional access client software (CACS): An image of conditional access client software code downloaded onto the CRS CAM.

3.2.4 control word (CW): The value which is used to scramble and descramble transport streams; it is refreshed frequently during the service operation to enhance security.

3.3 Security symbols

Security symbols	Descriptions
Pub(X)	RSA public key of 'X'
Prv(X)	RSA private key of 'X'
E(k,m)	Encryption of a message 'm' with key 'k'. RSAES-OAEP is used to encrypt a message when the encryption key is a public key. AES-ECB is used to encrypt a message when the encryption key is a symmetric key
S(k,m)	Digital signature for a message 'm' with signing key 'k'. RSASSA-PSS is used for message signing
H(m)	SHA-256 hashing for a message 'm'
HMAC(k,m)	HMAC-SHA1 for a message 'm' with key 'k'
X||Y	Concatenation of 'X' and 'Y'
Cert(X)	ITU-T X.509 certificate of 'X'
PRF(X)	Pseudo random function having a seed value of 'X'
$X_{msb(Y)}$	'Y' bits from MSB of 'X'

3.4 Parameter definitions

Parameter names	Descriptions
DSC_ID	The value of identification of DSC having a size of 40 bytes
CAM_ID	The value of identification of CAM having a size of 8 bytes
KeyPairingID	The value of concatenation with CAM_ID and DSC_ID, i.e., CAM_ID||DSC_ID
CWEK	The abbreviation of control words encryption key, and used to encrypt *control words* The CWEK generation method is $CWEK = H(CWEK\|\|CAM\_ID\|\|DSC\_ID)_{msb(128)}$
KPK	The abbreviation of key pairing key. The AC generates the KPK if KeyPairingID is valid
HMAC_KEY	An HMAC secret key. The CAM uses HMAC_KEY to generate an HMAC value for the message including *control words* The HMAC_KEY generation method is $HMAC\_KEY = H(RAND_{HMAC}\|\|CAM\_ID\|\|DSC\_ID)_{msb(160)}$, Here $RAND_{HMAC}$ is achieved by $PRF(X)_{msb(320)}$
RAND	A random number with 320 bits
K_i	The pre-shared key having the size of 128 bits. AC uniquely assigns three K_i to each CAM

3.5 Security function definitions

Security functions	Requirements
RSA digital signature (RSASSA-PSS)	• Modulus (n): 1024 bits • Exponent: F4 (65537) • Message Encoding: RSASSA-PSS – Hash algorithm (default): SHA-1 – MGF (default): MGF1 with SHA-1 – Trailer field: 1 (corresponds to '0xbc') – Salt length: 160/8 = 20 bytes
RSA encryption (RSAES-OAEP)	• Modules (n): 1024 bits • Exponent: F4 (65537) • MGF1 with SHA-1 for the mask generation function • The empty string for the encoding parameter string
AES encryption	• Block cipher mode: AES 128 ECB

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AC Authorization Centre

AES Advanced Encryption Standard

CACS Conditional Access Client Software

CAM Conditional Access Module

CASS CAM Authentication Sub-System

CW Control Word

CWEK Control Words Encryption Key

DSC Descrambler

ECB Electric Code Block

HMAC Hashed Message Authentication Code

KPK Key Pairing Key

MSO Multiple Systems Operator

PSI Pairing Status Information

RCAS Renewable Conditional Access System

5 Conventions

In this Recommendation:

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

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

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

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

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

6 Overview of RCAS pairing protocol

The components of RCAS that participate in the pairing protocol are the 'CAM authentication sub-system', 'authorization centre', 'CAM' and 'descrambler' of RCAS, as shown in Figure 1.

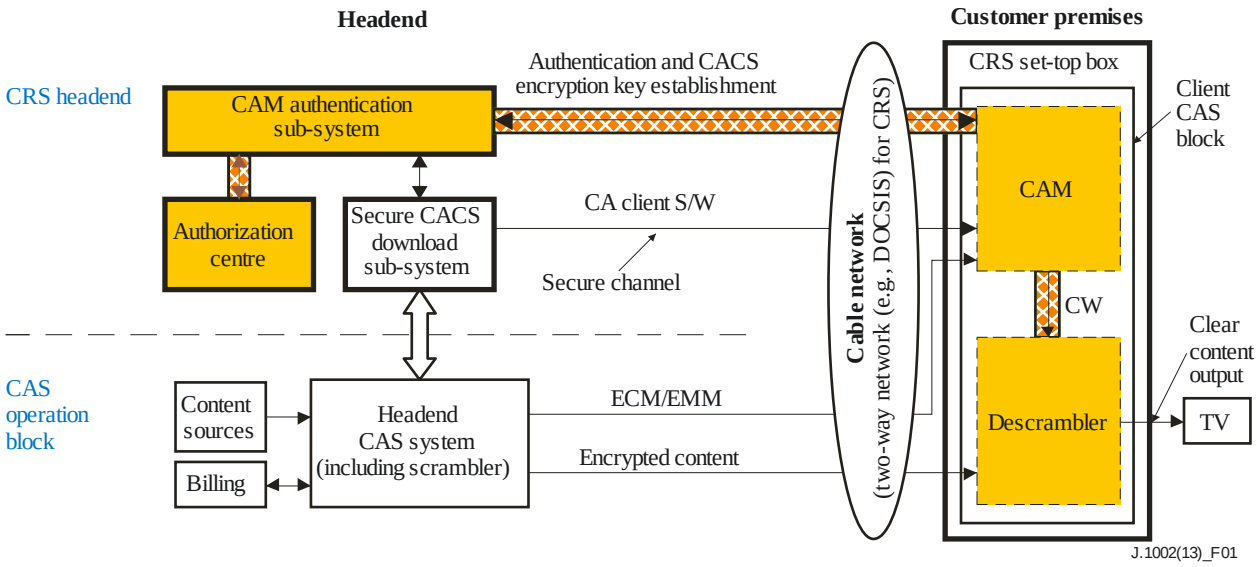

Figure 1 – Reference architecture of the RCAS and RCAS pairing protocol components

The specification of RCAS pairing function includes:

- **A pairing protocol that supports CAM and DSC pairing**: The participants of the protocol should be authorization centre (AC), CAM authentication sub-system (CASS), CAM and DSC.
- **A control words encryption key (CWEK) establishment protocol**: If the control words are delivered in plaintext from CAM to DSC, a malicious user could possibly watch pay programmes by using the disclosed control words for decrypting the scrambled video streams. Therefore, the CAM must provide confidentiality for the control words by encrypting them with the CWEK.
- **AC participation in CWEK establishment**: Since a successful CWEK establishment between CAM and DSC means that the CAM believes the DSC as its correct pair, or vice versa, CWEK establishment must not be performed before the AC confirms that the CAM and DSC are correctly paired. For this reason, the RCAS CAM and DSC pairing protocol should make sure that the CAM and DSC establish a CWEK after they have received confirmation that the CAM and DSC are correctly paired from the AC.

The RCAS pairing protocol consists of three sequences of phases: initialization, pairing and CWEK generation. The brief descriptions of each phase are as follows.

- **Phase I**: Phase I is the 'Initialization'. At the initialization phase, the pairing protocol between the CAM and DSC is initiated. The CAM monitors the triggering conditions of the protocol, and starts the protocol procedures if it meets one of the triggering conditions. Then, the CAM and DSC exchange their own ITU-T X.509 certificates [ITU-T X.509] with each other.
- **Phase II**: Phase II is the 'Pairing'. At the pairing phase, when the CAM transmits identification information of itself (i.e., CAM_ID) and the DSC (i.e., DSC_ID) to the AC via CASS, the AC verifies validation of the identification information. If the identification information of the CAM and the DSC are validated, the CAM and the DSC are managed in pair. Then, AC transmits a validation verification message containing encryption seed key (i.e., KPK) to the CAM via CASS.
- **Phase III**: Phase III is the 'CWEK generation'. At the CWEK generation phase, the CAM and the DSC generates the control word encryption key (CWEK) and encrypts mutual traffic with the generated CWEKs. If the CAM is paired with the DSC, a CWEK of the CAM is the same as a CWEK of the DSC.

In the following clause, each procedure of the security authentication method using the pairing protocol will be described in detail.

7 Details of RCAS pairing protocol

The detail descriptions of each RCAS pairing protocol steps are defined in the following clauses.

7.1 Initialization

Figure 2 illustrates a flowchart of the initialization for security authentication between the CAM and DSC. Referring to Figure 2, a pairing protocol between the CAM and the DSC is initiated if present initialization conditions are satisfied. Here, the initialization may be conducted when the CAM is newly booted due to newly supplied power or due to the reset, when the CAM in a virgin state receives a security announce message from the CASS or when the CAM in a non-virgin state receives a client update request from the CASS through a RCAS download message.

When any of the initialization conditions are satisfied, the CAM generates a certification request message (hereinafter referred to as DscCertReq message) including certificate information (CAM ITU-T X.509 CERTIFICATE) of the CAM and transmits the DscCertReq message to the DSC. In response to the receipt of the DscCertReq message from the CAM, the DSC verifies certificate signature using an AC root certificate. If the verification is successful, the DSC stores the certificate information of the CAM in a non-volatile memory of the DSC.

Then, the DSC generates a certification response message (hereinafter referred to as DscCertRsp message) and transmits the DscCertRsp message to the CAM. Then, in response to the receipt of the DscCertRsp message, the CAM verifies a certificate signature using the AC root certificate. If the verification is successful, the CAM stores certificate information (DSC ITU-T X.509 CERTIFICATE) of the DSC, which is included in the DscCertRsp message, in a non-volatile memory of the CAM.

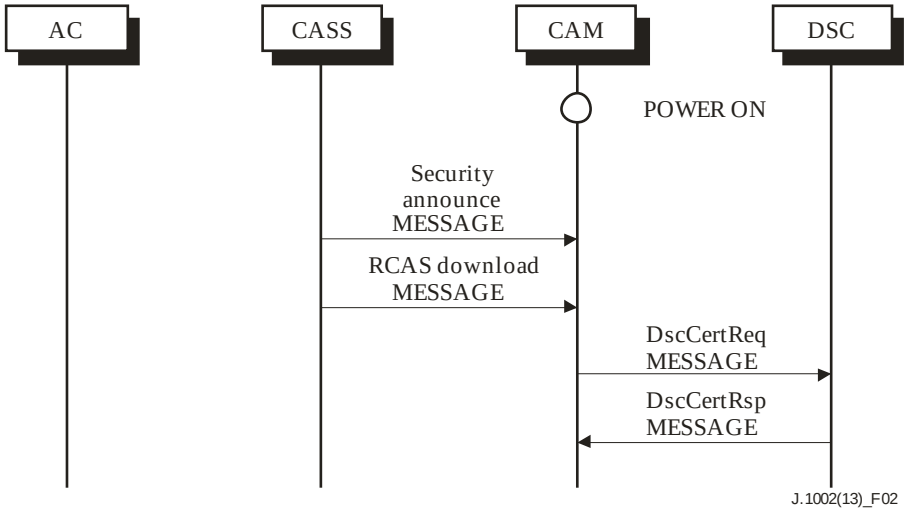

Figure 2 – Flowchart of the initialization phase

The summarized procedures of the initialization phase are as follows.

- Step 1: If the CAM meets one of the below conditions, the CAM sends its ITU-T X.509 certificate to the DSC through a DscCertReq message. Note that a CAM can achieve the CASS ITU-T X.509 certificate from the SecurityAnnounce message that is supposed to be delivered from the CASS.
  - * Condition 1: when CAM is powered up or reset,
  - * Condition 2: when a virgin CAM receives a SecurityAnnounce message from a CASS,
  - * Condition 3: when a CAM receives a SecurityAnnounce message from a CASS right after the CAM moves to another MSO network, or
  - * Condition 4: when a non-virgin CAM is requested to update CA client images from a CASS via a RCASDownload message.
- Step 2: Right after the DSC receives a DscCertReq message from the CAM, the DSC verifies the signature of the CAM ITU-T X.509 certificate using the public key of the AC. Only if the DSC can successfully verify the CAM ITU-T X.509 certificate, does the DSC store the CAM_ID and RSA public key of the CAM extracted from the CAM ITU-T X.509 certificate in the secure area of non-volatile memory. Otherwise, the DSC terminates this protocol. Finally, the DSC sends a DscCertRsp message including its ITU-T X.509 certificate to the CAM.
- Step 3: Right after the CAM receives the DscCertRsp message from the DSC, the CAM verifies the signature of the DSC ITU-T X.509 certificate using the public key of the AC. Only if the CAM can successfully verify the DSC ITU-T X.509 certificate, does the CAM store the DSC_ID and RSA public key of the DSC extracted from the DSC ITU-T X.509 certificate in the secure area of non-volatile memory, and goes to the next phase. Otherwise, the CAM terminates this protocol.

7.2 Pairing

Figure 3 illustrates a flowchart of an example of a method of pairing a CAM and a DSC for security authentication. Referring to Figure 3, the CAM transmits a CAM identifier (CAM_ID) and DSC identifier (DSC_ID) to a CASS, and an AC verifies the validation of the respective identification information. In detail, the CAM encrypts a key request message including a pair of identifiers (hereinafter referred to as KeyPairingID) of the CAM and the DSC, and transmits the KeyPairingID

to the AC via the CASS. The KeyPairingID is a concatenated value of the CAM identifier and the DSC identifier.

The AC which has received the key request message verifies the validation of the KeyPairingID. For the validation verification, the AC compares originally issued identifier (ID) values of the CAM and the DSC with ID values of the CAM and the DSC, which are received through the key request message. Only when the originally-issued ID values are identical with the ID values received through the key request message, does the AC verify that the KeyPairingID is validated.

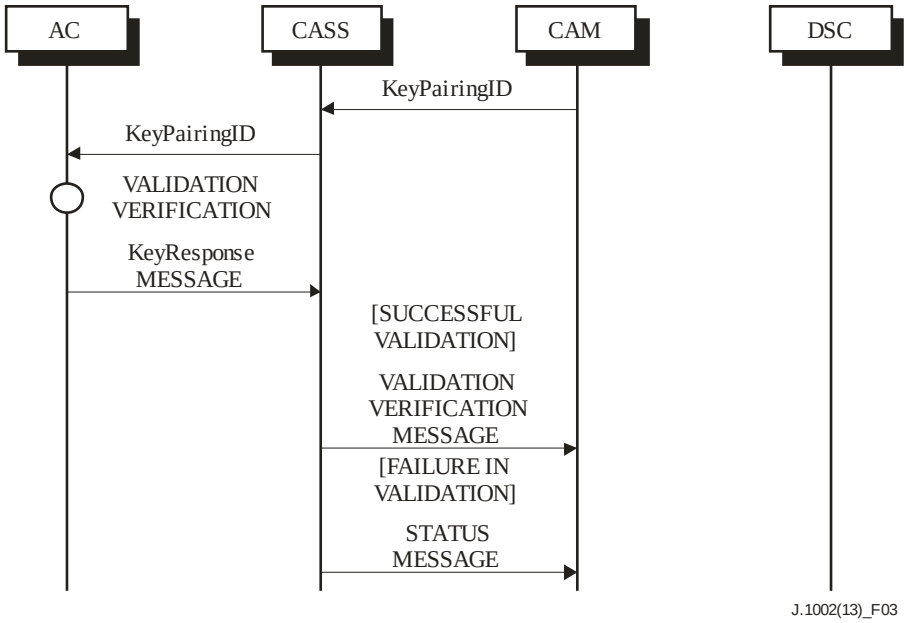

Figure 3 – Flowchart of the pairing phase

The AC generates a key response message based on the validation verification result of the KeyPairingID. If the KeyPairingID is validated, the AC generates a key pairing key (KPK) that is a seed encryption key, and transmits a validation verification message indicating the validation of the KeyPairingID to the CAM via the AC. If the KeyPairingID is invalid, the AC transmits a status message indicating that the KeyPairingID is invalid. In this case, a status message that sets all bytes of the KPK to '0xff' is transmitted to the CAM via the CASS.

The KeyPairingID validation process at AC utilizes pairing status information (PSI) as shown in Table 1. The PSI is maintained by the AC based on the pairing state diagram shown in Figure 4. As the pairing state diagram shows, PSI is classified into three types. The first type is Virgin('0x00'). The AC sets the PSI type as Virgin('0x00') when it issues identification information of the CAM and DSC and there have been no CAM-DSC pairing validation check requests from the MSO RCAS headend for them. The second type is Auth/Paired('0x01'). The AC changes the PSI type from Virgin('0x00') or Paired_Only('0x10') to Auth/Paired('0x01') when the RCAS host devices in either a Virgin('0x00') state or Paired_Only('0x10') state are connected to the MSO network and have passed the CAM-DSC pairing validation check in the AC. The third type is Paired_Only('0x10'). The AC sets the PSI type to Paired_Only('0x10') when the RCAS host devices in an Auth/Paired('0x01') state leave the MSO network.

Table 1 – Pairing state information

CAM state	DSC state	Pairing state information
0x00	0x00	Virgin
0x01	0x01	Auth/Paired
0x10	0x10	Paired Only

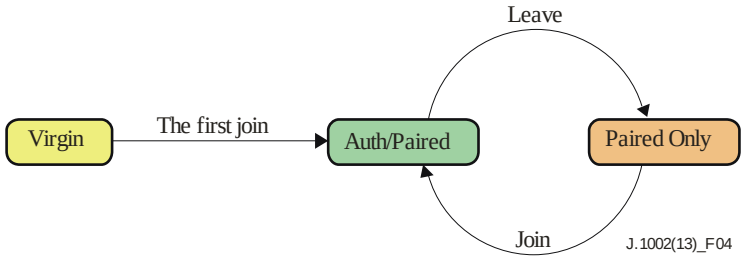

Figure 4 – Pairing state diagram

Table 2 – Generation method of the KeyPairingID and KPK

Parameters	Generation method
KeyPairingID	CAM_ID || DSC_ID
KPK	PRF(H (Ki_1||Ki_2||Ki_3||CAM_ID||DSC_ID||RAND))$_{msb(160)}$

The generation method of the security parameters is defined in Table 2. The following are detailed descriptions of the pairing phase.

- Step 1: The CAM sends a KeyPairingID and CAM ITU-T X.509 certificate through the KeyRequest message to the CASS. As shown in Figure 2, the KeyPairingID is encrypted with the public key of the CASS, and the content is signed with the private key of the CAM. Note that a CAM ITU-T X.509 certificate is added to the tail of this message without encryption.
- Step 2: The CASS receives the KeyRequest message from the CAM, and verifies the digital signature of the message. The CASS also stores the CAM ITU-T X.509 certificate for future communication with the CAM. If the CASS fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the CASS decrypts the KeyPairingID and generates a KeyRequest message including E(Pub(AC), KeyPairingID) instead of E(Pub(CASS), KeyPairingID). After that, the CASS sends this KeyRequest message to the AC. At this time, a CAM ITU-T X.509 certificate is not attached to this message.

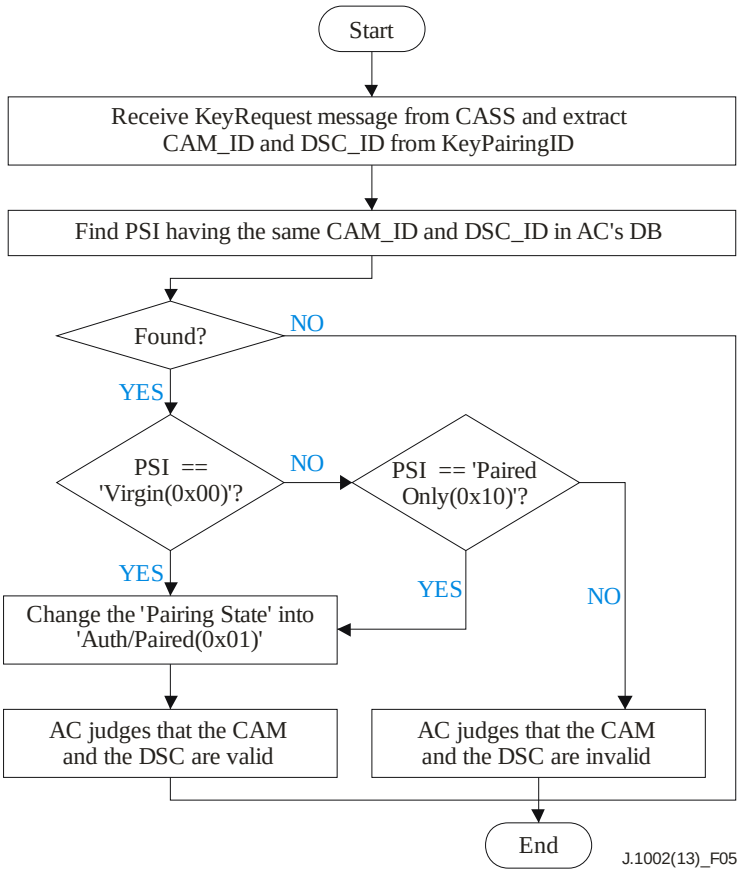

Figure 5 – Identification validation procedures at the AC

- Step 3: The AC receives the KeyRequest message from the CASS, and verifies the digital signature of the message. If the AC fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the AC decrypts the KeyPairingID and starts to validate the KeyPairingID based on the PSI as shown in Figure 5. The descriptions of Figure 5 are as follows.
  - After the AC receives a KeyRequest message from the CASS, it extracts the CAM_ID and DSC_ID from the KeyPairingID.
  - Then, the AC searches the PSI regarding the CAM_ID and DSC_ID from its own database.
  - If the AC fails to find the record in its database regarding the CAM_ID and DSC_ID, it terminates the protocol.
  - If the PSI for the CAM_ID and DSC_ID is equal to Virgin('0x00'), the AC changes the PSI from Virgin('0x00') to Auth/Paired('0x01') and judges that the CAM_ID and DSC_ID have successfully passed the CAM-DSC pairing validation check.
  - If the PSI for the CAM_ID and DSC_ID is not equal to Virgin('0x00') but the same as Paired_Only('0x10'), the AC changes the PSI from Paired_Only('0x10') to Auth/Paired('0x01') and judges that the CAM_ID and DSC_ID have successfully passed the CAM-DSC pairing validation check.
  - For all other cases, the AC judges that the CAM_ID and DSC_ID have failed to pass the CAM-DSC pairing validation check.

  After finishing the CAM-DSC pairing validation check, the AC generates a KeyResponse message including the encrypted KPK and signed KPK. At this time, the AC generates a KPK, which is uniquely assigned to the CAM, using the generation method shown in

Table 2. Otherwise, the AC sets all bytes of the KPK as '0xff' to indicate that the CAM_ID and DSC_ID pairing validation result is a failure. Finally, the AC sends the KeyResponse message to the CASS.

- Step 4: The CASS receives the KeyResponse message from the DSC, and verifies the digital signature of the message. If the CASS fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the CASS generates the KeyResponse message including E(Pub(CAM), KPK) instead of E(Pub(CASS), KPK), and sends this message to the CAM. At this time, the signed value of the KPK, i.e., S(Prv(AC),KPK), is inserted into the message content as it is received from the AC.

7.3 CWEK generation

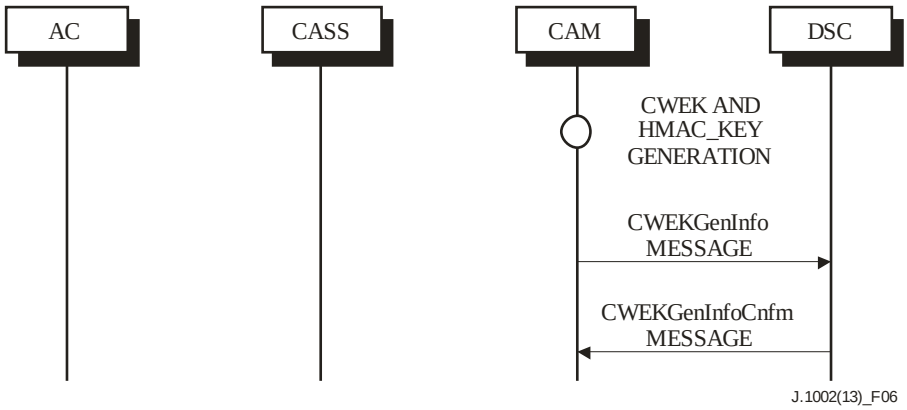

Figure 6 – Flowchart of the CWEK generation phase

Figure 6 illustrates a flowchart of a key generation method for security authentication between a CAM and a DSC. Referring to Figure 6, a control word encryption key (CWEK) for encrypting traffic between the CAM and the DSC is generated. Since the CWEK is generated using the above-described KPK, it is possible for the CAM and the DSC to generate the same CWEKs only when the pairing of the CAM and the DSC is normally performed. The CWEK is formed by the equation in Table 3.

More specifically, in response to the receipt of the validation verification message from the CASS, the CAM generates the CWEK and a hashed message authentication code key (hereinafter referred to as HMAC_KEY). Then, the CAM generates a CWEK message (hereinafter referred to as CWEKGenInfo message) and transmits the generated CWEKGenInfo message to the DSC. The CWEKGenInfo message is encrypted with a public key of the DSC, and electronically signed with a private key of the CAM. Alternatively, if the CAM receives a key response message that indicates the failure of the validation from the CASS, a key response message which has all bytes of the KPK set to '0xff', the CAM transmits to the DSC a CWEKGenInfo message that includes the KPK having all bytes set to '0xff'. Note that HMAC_KEY of the CAM is obtained by applying SHA1 to a concatenated value of the CAM_ID, the DSC_ID and a random number produced by a RAND function as shown in Table 3.

Table 3 – Generation method of the CWEK and HMAC_KEY

Parameters	Generation method
CWEK	H(KPK ‖CAM_ID‖ TP_ID)msb(128)
HMAC_KEY	H($RAND_{HMAC}$ ‖ CAM_ID‖TP_ID)msb(160)

In response to the receipt of the KPK and HMAC_KEY through the CWEK message from the CAM, the DSC generates a CWEK. The CWEK generated by the DSC is the same as the CWEK generated by the CAM if the DSC has been paired with the CAM.

Then, the DSC transmits a CWEK confirmation message (hereinafter referred to as CWEKGenInfoCnfm message) including the generated CWEK and the HMAC_KEY to the CAM. If all bytes of the KPK received from the CAM are set as '0xff', the DSC terminates the pairing protocol. The CWEKGenInfoCnfm message is encrypted with a public key of the CAM and electronically signed with a private key of the DSC.

Subsequently, in response to the receipt of the CWEKGenInfoCnfm message the CAM checks whether the CWEK generated by the CAM and the HMAC_KEY are the same as those included in the CWEKGenInfoCnfm message. If the CWEK and the HMAC_KEY of the CAM are identical with those included in the CWEKGenInfoCnfm message, the CAM shares the CWEK and the HMC_KEY with the DSC. Then, the CAM encrypts control words and transmits them to the DSC.

The summarized procedures of the CWEK generation phase are as follows.

- Step 1: The CAM receives the KeyResponse message from the CASS, and verifies the digital signature of the message. If the CAM fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the CAM decrypts the KPK, and verifies S(Prv(AC), KPK) with the decrypted value of the KPK using the public key of the AC. Note that the DSC already has the AC root certificate in its memory. If the CAM fails to verify the digital signature of the KPK, it also discards the KeyResponse message and terminates the protocol. Otherwise, the CAM generates the CWEK and HMAC_KEY as shown in Table 3. Then, the CAM also generates a CWEKGenInfo message including a KPK and $RAND_{HMAC}$ except when all bytes of the KPK are '0xff'. If all bytes of KPK are '0xff', it terminates this RCAS CAM and DSC pairing protocol since the value of '0xff' means that the CAM_ID and DSC_ID pairing validation result is a failure. Finally, the CAM sends the CWEKGenInfo message to the DSC. Note that the value of S(Prv(AC), KPK) is inserted into the message content as it is received from the CASS through a KeyResponse message.
- Step 2: The DSC receives the CWEKGenInfo message from the CAM, and verifies the digital signature of the message. If the DSC fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the DSC decrypts the KPK and $RAND_{HMAC}$. After that, the DSC verifies S(Prv(AC), KPK) with the decrypted value of KPK using the public key of the AC. Note that the DSC already has the AC root certificate in its memory. If the DSC fails to verify the digital signature of the KPK, it also discards the CWEKGenInfo message and terminates the protocol. Otherwise, the DSC generates a CWEK and HMAC_KEY with the KPK and $RAND_{HMAC}$ as shown in Table 3. Finally, the DSC generates a CWEKGenInfoCnfm message including H(CWEK)||H(HMAC_KEY) and sends this message to the CAM.
- Step 3: The CAM receives the CWEKGenInfoCnfm message from the DSC, and verifies the digital signature of the message. If the CAM fails to verify the digital signature of the message, it discards this message and terminates the protocol. Otherwise, the CAM decrypts H(CWEK)||H(HMAC_KEY) from the CWEKGenInfoCnfm message, and generates the hashed value with the CWEK and HMAC_KEY that were generated in Step 1. Finally, the CAM compares the hashed values received from the DSC with those generated by the CAM itself. If the two hashed values are mismatched, the CAM terminates the protocol.
- Step 4: After the CAM and DSC share the same CWEK and HMAC_KEY, the CWEK is used for encrypting the control words with the symmetric encryption algorithm, and HMAC_KEY is used for applying the HMAC algorithm to the messages, which includes control words, for the purpose of message authentication. Note that the CAS headend sends

updated control words very frequently, e.g., 1~20 seconds, to the CAM. Therefore, CAM also has to deliver control words from the CAS headend to DSC whenever CAM receives the updated control words from headend. In this circumstance, the primary decision criteria for a message authentication algorithm should be a computational overload, not a security concern. As a result, the HMAC algorithm is selected instead of a digital signature algorithm for the practical reason of reducing computational overload.

8 CAM and DSC interface message format and encryption

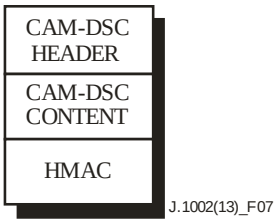

Figure 7 – CAM and DSC interface message format

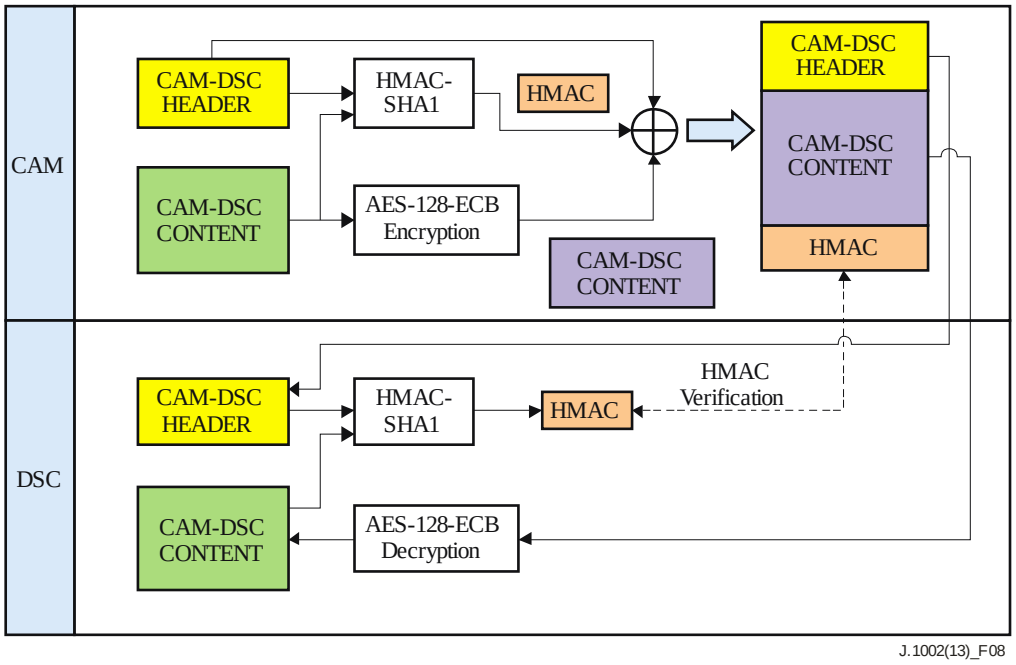

Figure 8 – CAM-DSC CONTENT encryption and HMAC processes

The encryption is performed on the CAM-DSC CONTENT field of a message, and HMAC authentication is performed on both a CAM-DSC HEADER field and the CAM-DSC CONTENT field. RSA encryption and RSA electronic signature verification is performed on the CAM-DSC CONTENT field of the CWEK message. For example, the RSA encryption is performed using an RSAES_OAEP scheme, and RSA electronic signature verification is performed using an RSASSA-PSS scheme.

Figure 7 illustrates a configuration of a message to be transmitted between the CAM and DSC. Referring to Figure 7, the message includes a CAM-DSC HEADER field, a CAM-DSC CONTENT field and a hashed message authentication code (HMAC) field.

Advanced encryption standard (AES) encryption and HMAC authentication are performed on all messages described with reference to Figures 2, 3 and 6, except the DscCertReq message, the DscCertRsp message and the CWEKGenInfo message.

In detail, the CAM and the DSC perform AES encryption selectively on such important fields as a control word in the CAM-DSC CONTENT field using the CWEK as an encryption key. The CAM and the DSC encrypt data to be transmitted to each other using the CWEK. The advanced encryption standard 128 electric code block (AES 128 ECB) scheme is used for the encryption. The AES 128 ECB scheme encrypts elements of a message, which requires encryption and is communicated between the CAM and the DSC, using the SWEK as an encryption key. For HMAC authentication, a 160-bit value produced by HMAC-SHA1 scheme with respect to the CAM-DSC HEADER and the CAM-DSC CONTENT is concatenated with the CAM-DSC CONTENT as shown in Figure 8.

8.1 DscCertReq message

The format of DscCertReq message is defined in Table 4.

Table 4 – The format of DscCertReq message

Field description	Type(Length) (ASN.1 notation)	Note
CAM-DSC_header{		
Message_Type	OCTET STRING (SIZE(2))	Value: 0x0011
Message_Length	OCTET STRING (SIZE(4))	The length of CAM-DSC_header and CAM-DSC_content
Message_Nounce	OCTET STRING (SIZE(8))	
Protocol_Version	OCTET STRING (SIZE(1))	
Reserved	OCTET STRING (SIZE(1))	Value: 0x00
}		
CAM-DSC content{		
Cert(CAM)	BIT STRING	
}		

8.2 DscCertRsp message

The format of DscCertRsp message is defined in Table 5.

Table 5 – The format of DscCertRsp message

Field description	Type(Length) (ASN.1 notation)	Note
CAM-DSC_header{		
Message_Type	OCTET STRING (SIZE(2))	Value: 0x0012
Message_Length	OCTET STRING (SIZE(4))	The length of CAM-DSC_header and CAM-DSC_content
Message_Nounce	OCTET STRING (SIZE(8))	
Protocol_Version	OCTET STRING (SIZE(1))	
Reserved	OCTET STRING (SIZE(1))	Value: 0x00
}		
CAM-DSC content{		

Table 5 – The format of DscCertRsp message

Field description	Type(Length) (ASN.1 notation)	Note
Cert(DSC)	BIT STRING	
}		

8.3 CWEKGenInfo message

The format of CWEKGenInfo message is defined in Table 6.

Table 6 – The format of CWEKGenInfo message

Field description	Type(Length) (ASN.1 notation)	Note
CAM-DSC_header{		
Message_Type	OCTET STRING (SIZE(2))	Value: 0x0013
Message_Length	OCTET STRING (SIZE(4))	The length of CAM-DSC_header and CAM-DSC_content
Message_Nounce	OCTET STRING (SIZE(8))	
Protocol_Version	OCTET STRING (SIZE(1))	
Reserved	OCTET STRING (SIZE(1))	Value: 0x00
}		
CAM-DSC content{		
E(Pub(DSC), KPK || $RAND_{HMAC}$) || S(Prv(DSC), KPK)	OCTET STRING (SIZE(128))	
}		
S(Prv(CAM), (CAM-DSC_header ||CAM-DSC_content)	OCTET STRING (SIZE(128))	

8.4 CWEKGenInfoCnfm message

The format of CWEKGenInfoCnfm message is defined in Table 7.

Table 7 – The format of CWEKGenInfoCnfm message

Field description	Type(Length) (ASN.1 notation)	Note
CAM-DSC_header{		
Message_Type	OCTET STRING (SIZE(2))	Value: 0x0014
Message_Length	OCTET STRING (SIZE(4))	The length of CAM-DSC_header and CAM-DSC_content
Message_Nounce	OCTET STRING (SIZE(8))	
Protocol_Version	OCTET STRING (SIZE(1))	
Reserved	OCTET STRING (SIZE(1))	Value: 0x00

Table 7 – The format of CWEKGenInfoCnfm message

Field description	Type(Length) (ASN.1 notation)	Note
}		
CAM-DSC content{		
H(CWEK) || H(HMAC_KEY)	OCTET STRING (SIZE(128))	
}		
S(Prv(DSC), (CAM-DSC_header ||CAM-DSC_content)	OCTET STRING (SIZE(128))	

Appendix I

The functional structures for the CAM and DSC

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

I.1 Functional structure for CAM

Figure I.1 illustrates a functional structure of a CAM. Referring to Figure I.1, the CAM includes a CAM pairing unit, a CAM key generating unit, a CAM encrypting unit and a CAM control unit.

When a preset initialization condition is satisfied, the CAM pairing unit transmits to the DSC a certification request message (DscCertReq message) including the certificate information (CAM ITU-T X.509 CERTIFICATE) of the CAM, and receives a certification response message (DscCertRsp message) including the certificate information (DSC ITU-T X.509 CERTIFICATE) of the DSC from the DSC.

In response to the receipt of the DscCertRsp message, the CAM pairing unit encrypts a key request message including an identifier pair consisting of a CAM identifier (CAM_ID) and a DSC identifier (DSC_ID), and transmits the encrypted key request message to a headend. In response, the CAM pairing unit receives a key response message including a KPK, which is a seed key for the identifier pair from the headend. The KPK produces a pseudo-random number sequence using a KeyPairingID value, which is obtained by concatenating the CAM_ID and the DSC_ID as a seed value when the KeyPairingID value is validated.

According to the result of verifying validation of the KeyPairingID value by the headend, the key response message to be received by the CAM includes the KPK when the KeyPairingID is validated, and values of all bytes of the KPK included in the key response message are set to '0xff' when the KeyPairingID is invalid.

The CAM key generating unit generates a CWEK message (CWEKGenInfo message), which includes a CWEK and an HMAC_KEY, based on the KPK, and transmits the generated CWEKGenInfo message to the DSC. The CWEKGenInfo message is encrypted with a public key of the DSC and electronically signed with a private key of the CAM.

Thereafter, when the DSC generates a CWEK and an HMAC_KEY, the CAM key generating unit receives a CWEK confirmation message (CWEKGenInfoCnfm message) including the generated CWEK and HMAC_KEY from the DSC. The CWEKGenInfoCnfm message is encrypted with a public key of the CAM, and electronically signed with a private key of the DSC. When the values of all bytes of the KPK transmitted from the CAM are set as '0xff', the DSC terminates the pairing protocol.

When the CAM key generating unit receives the CWEKGenInfoCnfm message, the CAM encrypting unit checks whether the CWEK and the HMAC_KEY, which are included in the CWEKGenInfoCnfm message, are identical with the CWEK and the HMAC_KEY that are generated by the CAM key generating unit. If the CWEK and the HMAC_KEY are the same as those of the CWEKGenInfoCnfm, the CAM shares the generated CWEK and the HMAC_KEY with the DSC, and the CAM encrypting unit encrypts a control word using CWEK and transmits the encrypted control word and keys to the DSC.

The CAM control unit controls the CAM pairing unit, the CAM key generating unit and the CAM encrypting unit.

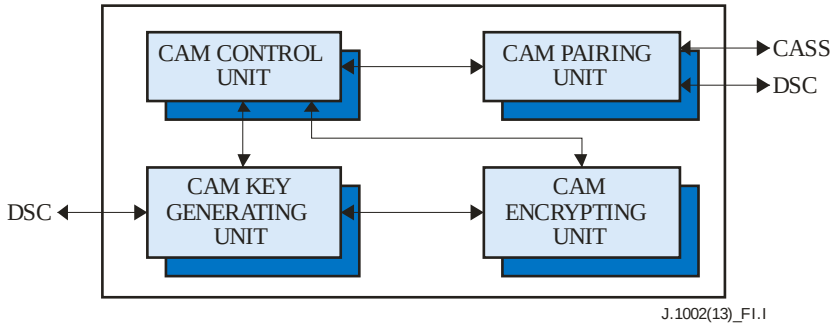

Figure I.1 – CAM functional structure

I.2 Functional structure for DSC

Figure I.2 illustrates a functional structure of a DSC. Referring to Figure I.2, the DSC includes a DSC pairing unit, a DSC key generating unit, a DSC encrypting unit and a DSC control unit.

The DSC pairing unit receives a DscCertReq message including CAM ITU-T X.509 CERTIFICATE of the CAM, and transmits a DscCertRsp including CAM ITU-T X.509 CERTIFICATE of the DSC to the CAM.

When a headend verifies the validation of a KeyPairingID, which is an identifier pair consisting of a CAM identifier and a DSC identifier and is received from the CAM, the DSC pairing unit receives a CWEKGenInfo message including a KPK and an HMAC_KEY for the KeyPairingID from the CAM. The KPK produces a pseudo-random number sequence using a KeyPairingID value as a seed value when the KeyPairingID value is validated.

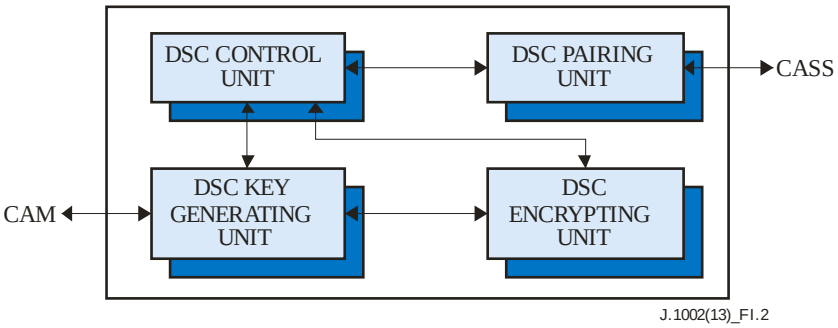

Figure I.2 – DSC functional structure

The DSC key generation unit generates a CKEK using the KPK and the HMAC_KEY of the received CWEKGenInfo message, then, the DSC key generating unit transmits a CWEKGenInfoCnfm message including the generated CWEK and HMAC_KEY to the CAM. If all bytes of the KPK received from the CAM are set as '0xff', the DSC key generating unit terminates the pairing protocol. The CWEKGenInfoCnfm message may be encrypted with a public key of the CAM and electronically signed with a private key of the DSC.

The DSC encrypting unit shares the CWEK and the HMAC_KEY, which are transmitted through the CWEKGenInfoCnfm message to the CAM, with the CAM if the transmitted CWEK and HMAC_KEY are identical with those of the CAM. In addition, the DSC encrypting unit encrypts data to be transmitted to the CAM, and performs hashed message authentication on a message to be transmitted.

The DSC control unit controls the DSC pairing unit, the DSC key generating unit and the DSC encrypting unit.

Bibliography

[b-ITU-T J.93] Recommendation ITU-T J.93 (1998), *Requirements for conditional access in the secondary distribution of digital television on cable television systems.*

[b-ITU-T J.122] Recommendation ITU-T J.122 (2007), *Second-generation transmission systems for interactive cable television services – IP cable modems.*

[b-ITU-T J.128] Recommendation ITU-T J.128 (2008), *Set-top gateway specification for transmission systems for interactive cable television services.*

[b-ITU-T J.193] Recommendation ITU-T J.193 (2004), *Requirements for the next generation of set-top-boxes*.

[b-ITU-T J.290] Recommendation ITU-T J.290 (2006), *Next generation set-top box core architecture*.

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2013