Application Note 158/13 v02

Title:	**KNX Data Security**	
Status:		**Date:**
	Draft Proposal	2013.05.14
Transitional period:	Immediate effect after Final Voting.	
Date:	2013.05.14	

Subject: Specification of KNX S-AL, Secure Service, Secure configuration and security Resources.

Documents

Modified

[01] Chapter 3/1/2 “Glossary”
[02] Chapter 3/3/4 “Transport Layer”
[03] Chapter 3/3/7 “Application Layer” v1.3.00 AS of 2010.10.22
[04] Chapter 3/4/1 “Application Interface Layer””
[05] Chapter 3/5/1 “Resources”
[06] Chapter 3/5/2 “Management Procedures” V1.4 AS of 2009.01.14
[07] Chapter 3/5/3 “Configuration Procedures” v1.2.00 AS of 2010.12.22
[08] Chapter 3/7/2 “Datapoint Types”
[09] Part 4/4 “Installation Safety Requirements”
[10] Volume 5 “KNX Certification of Products”
[11] Volume 6 “Profiles”

Referred

[12] Chapter 3/1/3 “Communication Security” v0.00.01

Further, general reference is made to the following papers.

[13] Chapter 3/3/2 “Data Link Layer General” v1.2 AS of 2009.06.29.
[14] Chapter 3/2/2 “Twisted Pair 1 ” v1.1 AS of 2008.12.19
[15] Chapter 3/2/5 “Communication Medium RF” v1.4.00 DV of 2010.11.02
[16] Chapter 8/3/7 “Application (Interface) Layer Testing – Network Management Server/Client Testing”
[17] Part 10/1 “Logical Tag Extended”
[18] AN127 “Master Reset”
[19] AN132 “A_NetworkParameter_InfoReport
[20] AN145 “Private IOs Property encoding in LTE-services”
[21] ISO/IEC 24767-2 Home network security / Secure communication protocol middleware (SCPM)
[22] NIST SP 800-38A “Recommendation for Block Cipher modes of Operation” – “Methods and Techniques”
[23] NIST SP 800-38C “Recommendation for Block Cipher modes of Operation: The CCM Mode for Authentication and Confidentiality”
[24] FIBPS PUB 113 “Computer Data Authentication”
[25] RFC 3610 “Counter with CBC-MAC (CCM)”

Document updates

Version	Date	Modification
AN158 v01	2013.02.04	• Preparation of the Draft Proposal.
0.17.01	2013.03.18	• Accepted all modifications. • Proposal for resolution of comments from RfV.
0.17.02	2013.04.23	• Inclusion of resolution of comments from RfV by TF ComSec. • Added examples of Permissions Tables.
AN158 v02	2013.05.14	• Inclusion of feedback of online meeting of 2013.05.13. • Draft Proposal for publication.

Contents

1 Purpose, motivation and scope (informative)

1.1 Motivation

🗎 *This clause is not intended for integration in the KNX Specifications.*

This document specifies security for KNX communication.

It bases on ISO/IEC 24767-2 Home network security / Secure Communication Protocol Middleware (SCPM) ([21]).

Having a secure KNX solution has several advantages.

- It makes the KNX RF Communication Medium more secure

  KNX RF Radio Frames in plain communication can easily be traced (by sniffer for example).

- It allows for secure applications.

  Secure communication is interesting in shutter – and door control and anti-intrusion security, in order to prevent intrusive commands (burglars…).

  It is also interesting in metering to protect for example electrical consumption data.

  This document does not define any type of application. The configuration of the security keys however is specified, as this is an essential feature.

1.2 Scope

🗎 *This clause is not intended for integration in the KNX Specifications.*

1.2.1 Common overview of the KNX Security

Please refer to Chapter 3/1/3 “Communication Security” ([12]) for the general scope of KNX Security.

This document specifies the KNX Data Security, its Resources and Procedures.

1.2.2 Product types

This document especially considers KNX S-Mode end devices.

KNX Data Security is designed to be supported on all existing KNX Communication Media (KNX TP1, KNX PL110, KNX RF and KNX IP).

In the same way, E-Mode devices can support KNX Data Security, but the Resources, Management – and Configuration Procedures to this are not specified in this document.

This document version does not introduce specific requirements on KNX data interfaces.

KNX Couplers are considered as well. The KNX Secure Frame format (see Figure 8) is designed so that it can be handled by existing (state 2008) KNX Couplers and newer.

EXAMPLE 1 KNX TP1/RF Couplers

NOTE 1 Products using TP-UART cannot handle more than 64 octets.

1.2.3 Secure and plain communication in an installation

The end user [1] wants to be sure that no unauthorized person will be able to control his receivers (shutters, doors….).

The end user can have many receivers of a unique kind and he would like to have secure communication only with some of them.

EXAMPLE 2 He may require security for shutters in the lower part of the house, but not request security for shutters in the upper part of the house.

As secure design requires an extension of the KNX stack, it will be useful to have products

- that use secure communication and plain communication, and
- other products that use only secure communication.

1.2.4 Prerequisites

Prerequisite 1

Secure communication shall be supported during runtime (typically multicast communication) and during Configuration (typically point-to-point communication).

Prerequisite 2

The need of a secure communication is a requirement from the receiver.

[1] home owner, building owner, building user etc.

1.2.5 Product scheme examples

1.2.5.1 Example 1 A secure transmitter linked to a receiver that only requires Authentication

Figure 1 below deals with bidirectional devices.

NOTE 2 For a unidirectional device, there is no link with the Datapoint “Info Status” (IMUD).

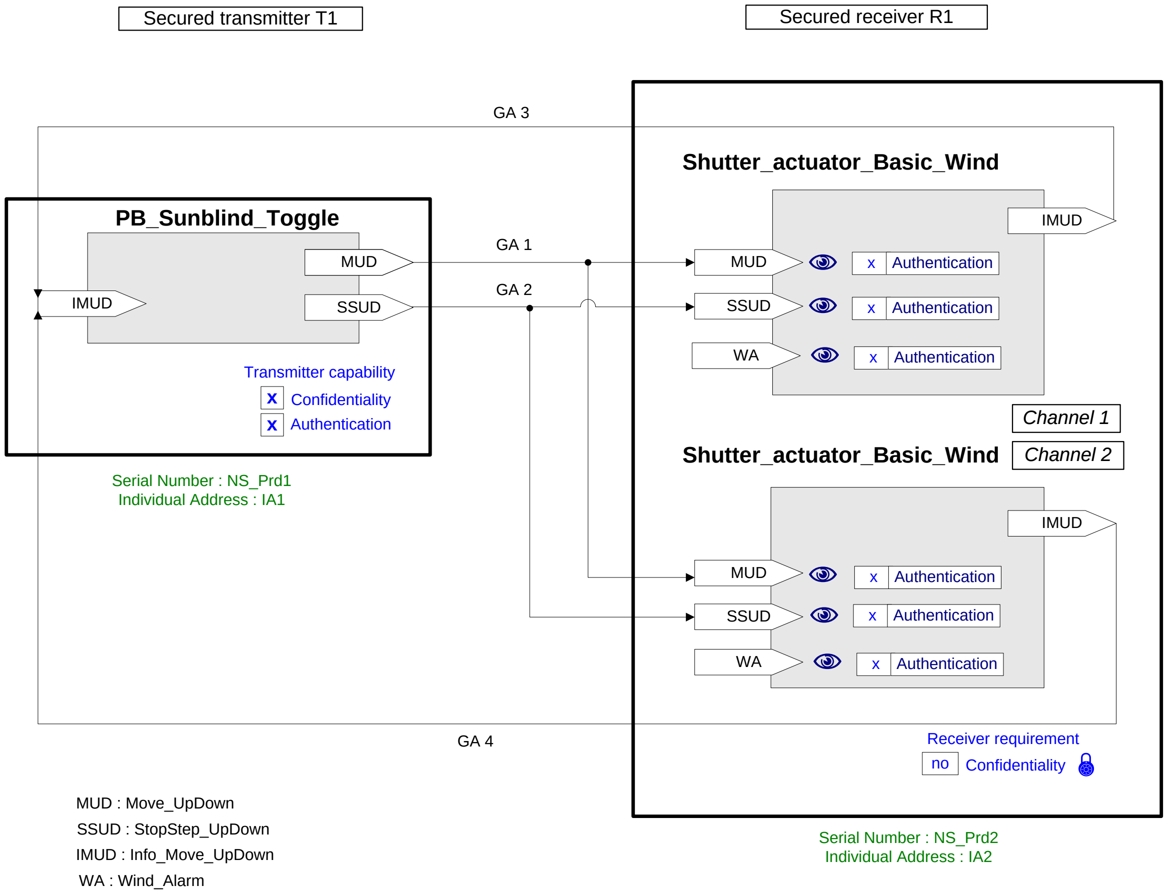

Figure 1 - Secure communication between a secure bidirectional transmitter and a bidirectional receiver that only requires authentication

The links are created during configuration. The security aspect can be defined by default in the product and in the ETS database or may be set by the configuration.

1.2.5.2 Example 2 A secure transmitter linked to a receiver that requires Authentication and Confidentiality.

Figure 2 below deals with bidirectional devices.

NOTE 3 For a unidirectional device, there is no link with the Datapoint “Info Status” (IMUD).

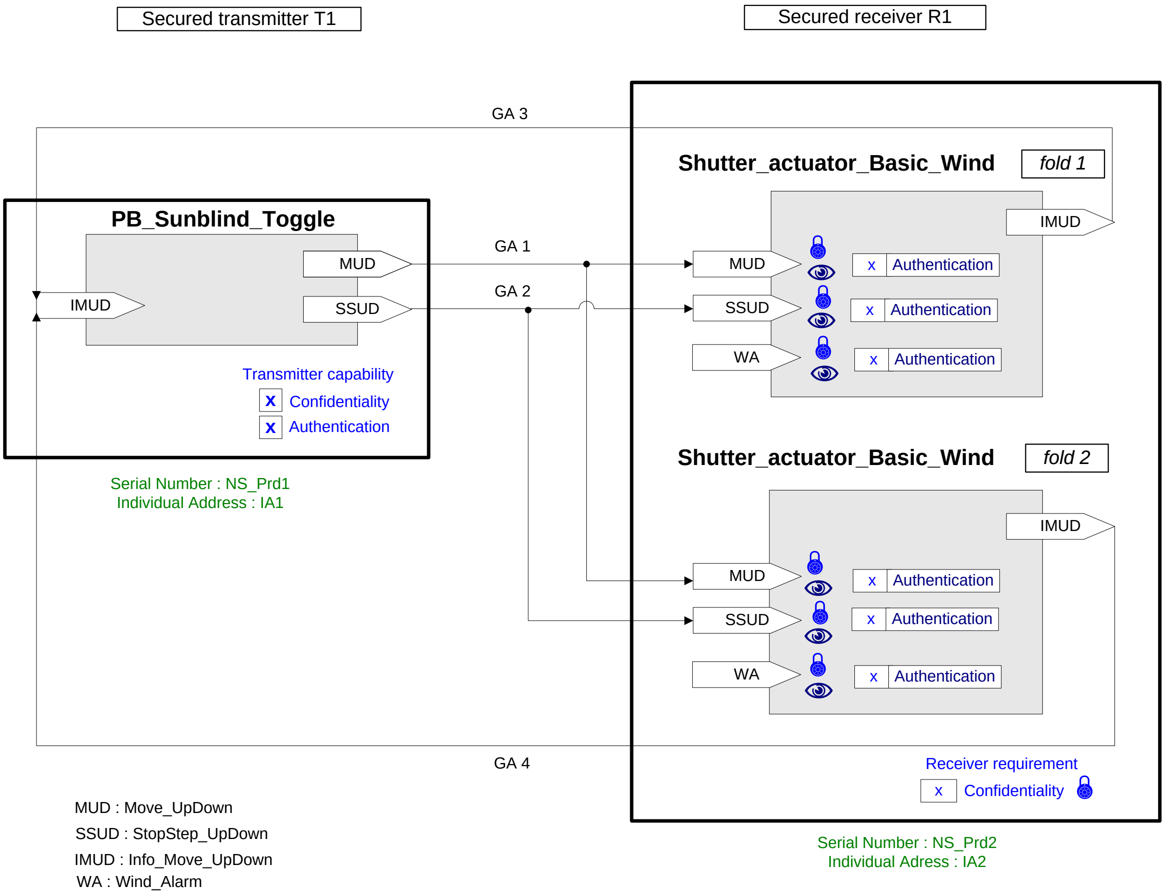

Figure 2 - Secure communication between a secure bidirectional transmitter and a bidirectional receiver that requires both authentication and confidentiality

1.2.6 Constraints

This document focusses on the KNX Security of end devices, for runtime and for configuration. For the full-fledged support of KNX Security in KNX installations, the following is additionally considered necessary, but not yet in full specified in this document version. The functionality listed below is outside the scope of this specification version.

- Visualisation and interfaces

  Visualisation requires that the visualisation device (e.g. touch panel) or tool obtains in a secure way the security keys used in the KNX installation.

  Additionally, the interface (KNXnet/IP Tunnelling, USB) needs either to be a commissioned communication partner in the installation, or needs to be synchronised with the Sequence Number (SeqNr) field of the secure devices to which it communicates (sending and receiving).

- Media Coupler between a Secure KNX RF Subnetwork and a Plain KNX Subnetwork

  USE CASE An existing KNX TP1 installation is extended with a KNX RF Subnetwork. Because of the open nature, it is chosen to have KNX Security communication on the KNX RF Subnetwork. Yet, it is not wanted that the existing KNX TP1 installation needs to be modified; the KNX TP1 installation is considered secure.

  This requires that the KNX TP1/RF Media Coupler acts as a "Secure Proxy" for the KNX TP1 communication on the RF Subnetwork and handles the KNX Data Security when transferring secure messages from KNX RF to KNX TP1.

  This KNX Secure Proxy and this KNX TP1/RF Media Coupler ("Security Gateway") are not modelled in this document version.

- KNX Data Security in a Subnetwork that exists the building and no KNX Data Security inside the building.

- If a communication partner is given the key for the verification of a secure message, then this participant is also capable of sending secure messages itself. The installed devices would also react if it uses an IA that is present in their Security Link Table.

  EXAMPLE 3 Suppose that the access of people to a building is controlled using KNX with KNX Data Security, using authentication. If the building manager gives the used security key to a visualisation tool, then this tool can rightfully verify the access control system as the proper sender of the messages. However, by having this security key, this visualisation tool can itself in turn also send secure messages with this key. If the visualisation tool would moreover use the IA of a legitimate device in the access control installation (and inherit its Sequence Number), then the receivers would also not be able to differentiate between messages from the "normal" senders and messages from this visualisation tool.

- The introduction of KNX Data Security makes that in case of a failure of group communication it is more difficult to diagnose the cause to be either due to the configuration of the KNX Data Security or to other group communication aspects. Extended Group Object Diagnostics are worked out by KNX Association.

EXAMPLE 4 If a secure sender sends a secure message to a secure receiver, and the secure receiver does not react, this can have many causes. Without security, there can be the following causes.

- The GA is not assigned to the receiver.
- The GA is no linked to the expected GO in the receiver.
- The flags for the GO in the receiver do not allow writing the GO-value.
- There are other causes in the receiver that make it not react to the message, like a locked state, a priority, etc.

Adding KNX Data Security, the following causes may be added.

- The sender and the receiver use different keys.
- The sender's IA is not in the Security Link Table of the receiver.
- There is something wrong with the Sequence Number of the sender.
- The sender does not have the permission (Role) to write the GO-value in the receiver.

1.3 Introduction

1.3.1 Features of KNX Data Security

- Authentication
- Confidentiality
- Access Control through Roles and Permissions

1.3.2 Situation of KNX Data Security in the KNX stack

KNX Data Security is handled by the Secure Application Layer (S-AL), the Application Layer and the Application Interface Layer, as indicated in Figure 3.

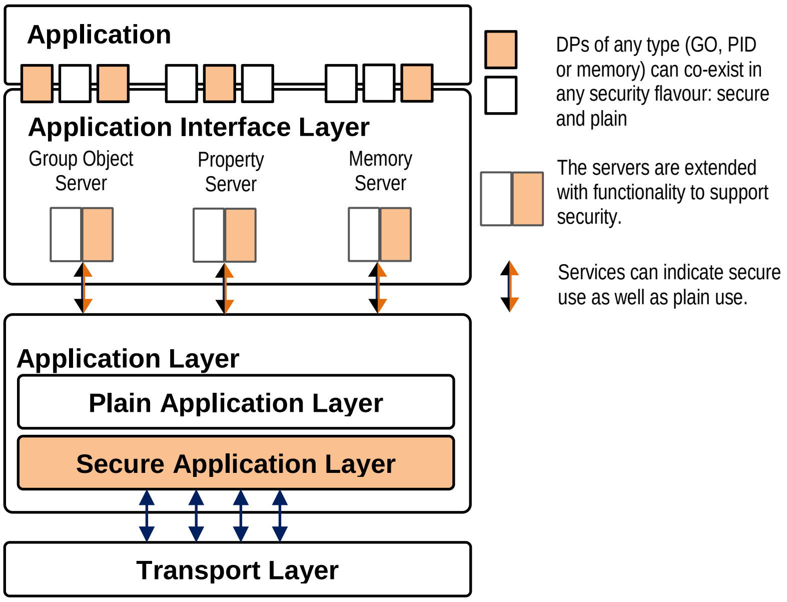
Application
Application Interface Layer
Group Object Server
Property Server
Memory Server
Application Layer
Plain Application Layer
Secure Application Layer
Transport Layer
DPs of any type (GO, PID or memory) can co-exist in any security flavour: secure and plain
The servers are extended with functionality to support security.
Services can indicate secure use as well as plain use.

Figure 3 – Location of KNX Data Security in the KNX stack

Table 1 gives a brief overview of how security is handled at these layers.

Table 1 – Security features of the KNX layers (informative)

Feature	Layer		
	S-AL	**P-AL**	**AIL**
Operation Level	Link Level		Service Level Datapoint Level
Functionality	• Encrypt and decrypt secure messages • Handle exceptions at link level.	• Forward the AL-service with the security features for sending and receiving and the indication of the link. - Authentication or Confidentiality - Link Index	• Access Control - Support Roles - Handle Permissions
Resource	• Security Link Table	None.	• Role Table

2 Specification

2.1 Terms and definitions

Terms

Access Control *The definition and evaluation of which communication partner has the right to access which data or call which services. This is solved by collecting communication partners with the same rights for all data and services in Roles and defining for each Role and for each piece of data or service the Permissions that this Role has.*

Security Black List *A standard list of services or DPs that shall exclusively be accepted using KNX Secure communication using confidentiality.*

Cipher text *Cipher text* is a generic term that denotes the encrypted data.
Cipher text opposes to *plain data*.

Permission The definition and conditions (plain, authentication, confidentiality) of the functionality that will be accepted from a Role, in accessing a DP in a device or in accepting services from a communication partner.

Plain data This is a generic term that denotes unencrypted data. The content of the plain data depends on the service and the user or not of confidentiality and authentication.
Plain data opposes to *cipher text*.

Secure DP *Datapoint that requires either authentication and/or confidentiality.*

Role *A Role is an identification of a group of links to a device (multicast, unicast and other) that have the same Permissions throughout the AIL.*

Secure Link *Link to a Secure DP.*

Group Address Security Flags *The indication in ETS whether for a Group Address, no secure communication will be used, or secure communication with authentication and/or confidentiality.*

Security White List *A standard list of services or DPs that shall always be accepted using plain communication.*

Abbreviations

🗎 *This shall be added to clause 2 “List of abbreviations” in [01].*

Abbreviation	Description
CFB	Cipher feedback
FDSK	Factory Default Setup Key
IV	Initialisation Vector
MaC	Management Client
MaS	Management Server
MAC	Message Authentication Code
MiM	Man-in-the-Middle
P-AL	Plain Application Layer
SAI	Security Algorithm Identifier
S-AL	Secure Application Layer
SCF	Security Control Field
SeqNr	Sequence Number

Abbreviation	Description
SFCC	Security Failure Common Counter
SFCL	Security Failure Counters on Links
SHD	Secure Header
SKI	Security Key Info

2.2 Stack and communication

2.2.1 Secure Application Layer

🗎 *This clause shall be integrated as new clause 5 “Secure Application Layer” at the end of [03].*

2.2.1.1 General requirements and overview

2.2.1.1.1 Embedding of the S-AL within the Application Layer and basic functionality

🗎 *This shall be clause 5.1, to be integrated in [03].*

The Secure Application Layer (S-AL) shall take care of the KNX Data Security at the level of the links.

The S-AL shall be part of the Application Layer (AL). This shall allow that the S-A_Data-service be close to the Application Layer services and shall allow security policies to be possibly adapted to each Application Layer service.

The use of the S-AL shall not influence the functionality of Plain Application Layer.

- In reception direction, after accepting the S-A_Data-service, the S-AL shall check the security according the Security Link Table, restore the Plain APDU and if successful forward the contained plain AL-service request internally to the Plain Application Layer.
- In transmission direction, according the Security Link Table for the S-A_Data-service, the S-AL shall secure the plain AL-service and shall transmit the secure APDU through the S-A_Data-service.

However, the Plain Application Layer shall transparently forward the security service parameters (par_auth, par_conf and link_index) between the S-AL and the AIL: see 2.2.2.

Figure 4 shows the location of the S-AL within the KNX communication stack. It shows the handling of secure messages. This is only a basic scheme. It does not show the handling of plain messages and the possible exceptions. The structure and priorities of the S-AL inside are not complete in this figure.

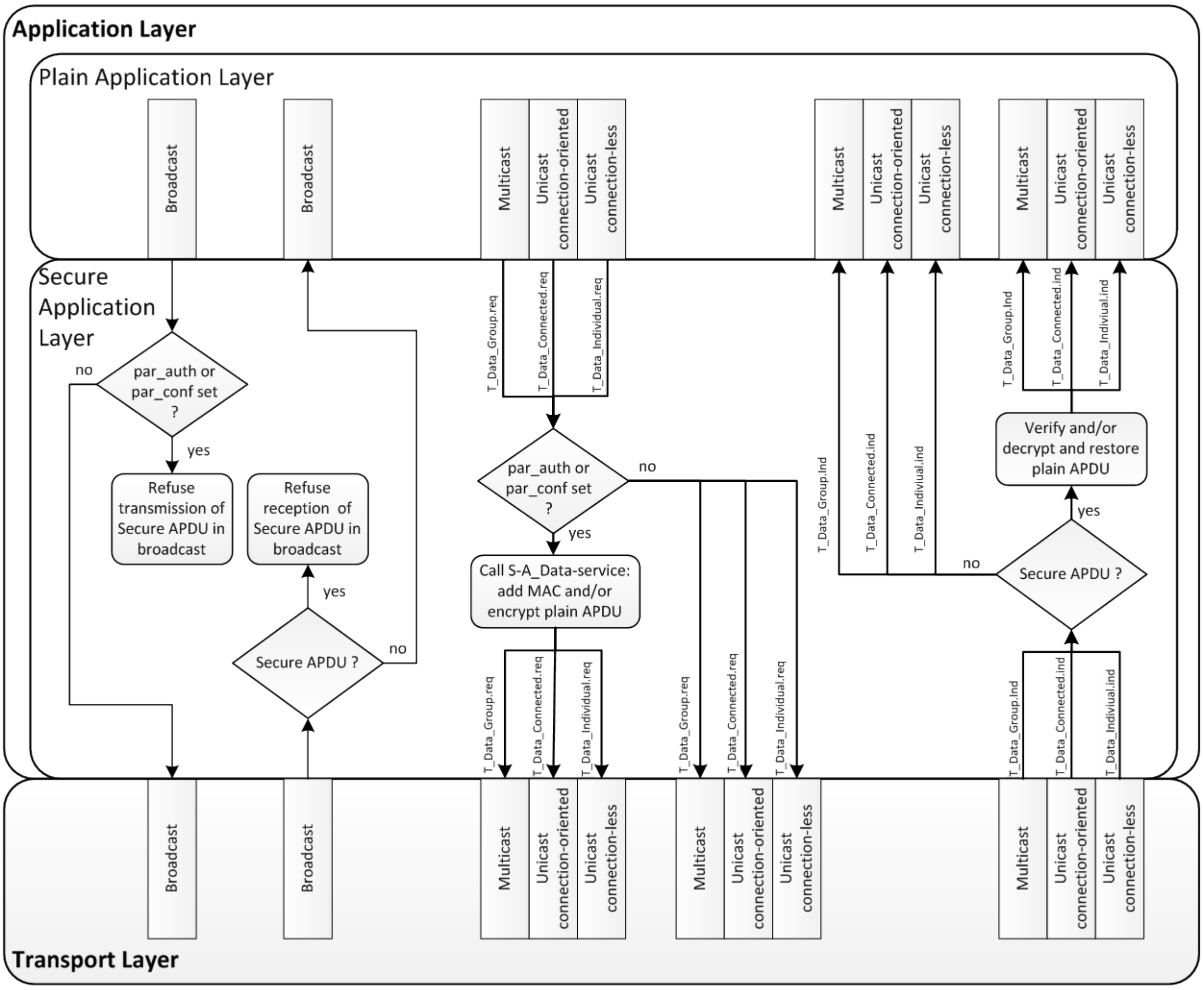

Figure 4 – Location of the S-AL within the Application Layer

2.2.1.1.2 S-AL – general requirements

The S-AL shall have the following functionality.

- Support the S-A_Data-service	see clause 2.2.1.2
- Handling of the Sequence Number:	see clause 2.2.1.3
- Handling of security failures	see clause 2.2.1.4
- Secure handling of the Transport Layer services	see clause 2.2.1.5

The following clauses specify these and other functions.

2.2.1.2 Support the S-A_Data-service

2.2.1.2.1 S-A_Data-service

The S-AL shall support the S-A_Data-service. In transmission direction, this service shall secure the plain AL-service; in reception direction it shall check the security of a received S-A_Data-PDU. The S-AL shall handle the Security Control Field within the secure ASDU.

NOTE 4 This clause solely specifies the S-A_Data-service. It does not specify when this service shall be applied. This depends on the value of the security service parameters par_auth and par_conf and this is explained in more detail in 2.2.1.5 “Secure Handling of the Transport Layer services” below.

The S-A_Data-service shall be an Application Layer internal service. It shall only be possible to call this service in the S-AL through the use of the security service parameters (par_auth, par_conf and link_index) of the plain AL-services; it shall not be possible to call this service explicitly. The S-A_Data-service shall not be available on any External Message Interface.

The S-A_Data-service shall replace the plain AL-service-PDU with the encrypted Secure APDU in transmission direction and vice versa in reception direction.

The local S-AL shall apply the S-A_Data-service to transmit an S-A_Data-PDU on the bus. The local S-AL shall secure the contained plain AL-service-PDU in function of all the following.

- the security service parameters par_auth and par_conf of the plain AL-service, and
- the Security Configuration Information in function of the plain AL-service (key, TSAP-type)
- the TSAP or ASAP

  EXAMPLE For a different destination GA or a different destination IA, the security key may be different.
- the layer parameters of the S-AL.

This is specified in further detail in the following clauses. The S-A_Data-PDU shall be formatted as given in Figure 5.

If the remote S-AL receives an S-A_Data-PDU then it shall check the security according to the Security Resources and reject or transmit internally to AL according to the security checking. This is specified in further detail in the following clauses.

octet 6								octet 7								octet 8							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						Secure Header																	
						APCISec										SCF							
transport control	"	"	"	"	"	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI								
						1	1	1	1	1	1	0	0	0	1								

octet 9									...		octet m							
7	6	5	4	3	2	1	0		...		7	6	5	4	3	2	1	0
Secure Data																		
									...									
									...									

Figure 5 - Format of the encrypted S-A_Data-PDU (informative)

The below paragraphs specify the format and use of this message format. For a worked out example, please refer to Annex C

The Secure APDU shall consist of the following fields.

- **Secure Header (SHD)**

  definition: The Secure Header shall always be composed of the APCISec and the SCF.

  - **APCISec**

    definition: This shall be the dedicated APCI that shall be used to indicate this APDU as a secure APDU.

    value: APCISec = $F31_{APCI}$ = 3F1h = 1111110001b

  - **Security Control Field (SCF)**

    definition: This field shall give the indications about Confidentiality, Authentication and the used security algorithm and its possible operation modes.

    encoding: The SCF shall be a 1 octet field formatted as specified in Figure 6.

	b7	b6	b5	b4	b3	b2	b1	b0	
	SAI		T	S-AL service type					
1.	SAI		T	0	0	0	A	C	S-A_Data-PDU
2.	SAI		T	0	0	1	1	1	S-A_Sync_Req-PDU
3.	SAI		T	0	1	0	1	1	S-A_Sync_Res-PDU

Figure 6 - Security Control Field

- **Tool Access (T)**

  definition: This bit shall be set if the security (authentication and/or confidentiality) is done using the specific tool entry in the PID_SKI_TOOL.

  encoding:

b5	Tool Access
0	Not using the specific tool entry
1	Using the specific tool entry

Figure 7 – Tool Access

- **Security Algorithm Identifier (SAI)**

  definition: This field shall identify the applied security algorithm and the operation mode if any shall be used.

  For more information about security algorithms, operation modes and Initialisation Vector, please refer to Annex B.

  encoding:

Table 2 – Security Algorithm Identifier

b7	b6	Security algorithm and operation mode
0	0	CCM (AES CTR for confidentiality and AES CBC for authentication)
0	1	Reserved. Shall not be used.
1	0	Reserved. Shall not be used.
1	1	Reserved. Shall not be used.

- **Authentication, Confidentiality**

  definition: Secure communication can be used for authentication and/or confidentiality. This leads to differences in the encrypted data. These fields Authentication and Confidentiality shall indicate whether the APDU is used for Authentication, for Confidentiality or for both.

  encoding:

Table 3 – Encoding of b1 and b0

b1	b0	Security
0	0	reserved
0	1	Reserved [a] for Frame with Confidentiality only
1	0	Frame with Authentication only
1	1	Frame with Authentication and Confidentiality

[a] In the use of the S-A_Data-service with CCM, the MAC shall always be present. Therefore, the use case “Frame with Confidentiality only” cannot exist in that case. This encoding of b1 and b0 in the SCF is however maintained to allow for possible future alternative algorithms.

The further fields in the Secure APDU shall depend on the Security Algorithm indicated in the field Algorithm in combination with the use of Authentication and Confidentiality. This is specified in the following clauses.

2.2.1.2.2 AES-128 with CTR operation mode and AES-CBC-MAC signature (CCM)

2.2.1.2.2.1 Secure Data

The common format for the Secure Data with this algorithm shall be as specified in Figure 8. The Secure Data shall contain the following fields. (These are specified in detail below.)

- Sequence Number (SeqNr), and
- Plain APDU (APCI + ASDU = Data), and
- Message Authentication Code (MAC).

The Secure APCI and the SCF are security algorithm independent and are specified in 2.2.1.2.1.

<table>
<tr><td colspan="8">octet 6</td><td colspan="8">octet 7</td><td colspan="8">octet 8</td><td colspan="8">octet 9</td><td>...</td><td colspan="8">octet m</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>...</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="35">Secure APDU</td></tr>
<tr><td colspan="6">TPCI</td><td colspan="10">Secure APCI</td><td colspan="25">Secure ASDU</td></tr>
<tr><td colspan="6"></td><td colspan="10"></td><td colspan="8">SCF</td><td colspan="17">Secure Data</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">12 bit</td><td colspan="8">8 bit</td><td colspan="17">n octets</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="17"></td></tr>
</table>

Legend

🟨 Plain text specific for the KNX S-A_Data-service.

Figure 8 – Secure APDU (example on TP1)

- **Sequence Number (SeqNr)**

  See 2.2.1.3.

- **Plain APDU (APCI + Data):**

  definition: This shall be the APDU of the Plain Application Layer service.

  format: The length, possible subfields and their encoding shall comply with the specification as given for the plain AL-service in [03].

- **Message Authentication Code (MAC)**
  See in the following clauses.

2.2.1.2.2.2 Common requirements

This clause only gives the KNX specific use of the CCM parameters. It is not the intention or the scope of this paper to define CCM. For the definition of CCM, please refer to [25].
In Annex A, an informative overview is given of the resulting use of CCM for KNX.

The below requirements shall apply both for the use with authentication only as well as with authentication and confidentiality.

B_0

The composition of the first block B_0 in the CCM algorithm shall be KNX specific. This is specified in Figure 9. These fields from the KNX Frame are included here, so that they cannot be altered in the communication path between sender and receiver without being detected.

octet nr												
15 MSB	…	10	9	8	7	6	5	4	3	2	1	0 LSB
SeqNr			SA		DA		FT	AT	TPCI	00h	00h	Q

Figure 9 – Format of B_0

- FT shall have the 8 bit value F0000000b where the msb F shall equal the Frame type of the KNX Control Field.
- AT shall have the 8 bit value A000EEEEb where A shall equal the Address Type of the KNX Frame.
  EEEEb shall equal the Extended Frame format (EFF) if L_Data_Extended Frames are used and 0000b otherwise.
- TPCI shall have the value TTTTTT00b, this is, the first 6 bits shall be the TPCI of the KNX Frame and the last 2 bits shall be zero.
- Q shall be the length of the payload P: see A.3.

Ctr_j

Also the format of the Block Counter Ctr_j (see [23]) is KNX specific. Ctr_j shall be composed as specified in Figure 10. The counter [j] for Ctr_0 shall be 00h.

octet nr												
15 MSB	…	10	9	8	7	6	5	4	3	2	1	0 LSB
SeqNr			SA		DA		00h	00h	00h	00h	01h	[j]

Figure 10 – Format of Ctr_j

NOTE 5 Octet 1 of Ctr_j shall be 01h. This shall guarantee that B_0 and Ctr_j always have different values, regardless of the values of FT, AT and TPCI in B_0.

Each counter value shall be calculated by incrementing the preceding counter value by 1.

$$Ctr_j = Ctr_{j-1} + 1 \quad \text{// For } j = 1 \text{ to } n;\ n \leq 255$$

2.2.1.2.2.3 Confidentiality only

This use case does not exist in the case of “AES-128 with CTR operation mode and AES-CBC-MAC signature (CCM)”, because the field MAC shall always be present.

2.2.1.2.2.4 Authentication only (Multicast)

2.2.1.2.2.4.1 Format of the Secure Data

The SCF, SeqNr and plain APDU shall not be encrypted. These shall be followed by the MAC.

<table>
<tr><td colspan="8">octet 6</td><td colspan="8">octet 7</td><td colspan="8">octet 8</td><td colspan="8">octet 9</td><td>...</td><td colspan="8">octet m</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>...</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="35">Secure APDU</td></tr>
<tr><td colspan="6">TPCI</td><td colspan="10">Secure APCI</td><td colspan="25">Secure ASDU</td></tr>
<tr><td colspan="6"></td><td colspan="10"></td><td colspan="8">SCF</td><td colspan="17">Secure Data</td></tr>
<tr><td colspan="6"></td><td colspan="10"></td><td colspan="8"></td><td colspan="8">SeqNr</td><td>Plain APDU
APCI + Data</td><td colspan="8">MAC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">12 bit</td><td colspan="8">8 bit</td><td colspan="8">6 octets</td><td>n octets</td><td colspan="8">4 octets</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8"></td><td></td><td colspan="8"></td></tr>
</table>

Legend

🟨 Plain text specific for the KNX S-A_Data-service.

⬜ Plain text from the original AL-service.

🟦 encrypted text

Figure 11 – Secure Data in case only authentication is used (example on TP1)

NOTE 6 The Sequence Number (SeqNr) is transmitted in plain text to allow the receiver to have a quick decision on the freshness of the message without having to decrypt the message.

2.2.1.2.2.4.2 Message processing (authentication only)

Message generation

The MAC shall be calculated according to the CCM algorithm as specified in [25].

1. In the SCF, clear the field “Confidentiality” and set the field “Authentication”; fill the field SAI with 00b, to indicate “AES-128 with CTR operation mode and AES-CBC-MAC signature (CCM)”.

2. Increment SeqNr by 1.

3. For Authentication only

   For “Authentication only” P [2] shall be empty and so the Plain APDU shall be included in A [2].

   A = Secure APCI | SCF | SeqNr | Plain APDU (=plain APCI + data)

   P = {empty}

4. The further calculation shall be according to Annex A.

   The Block B_0 shall be as specified in 2.2.1.2.2.2.

5. For KNX, the 32 most significant bits of Y_n ($MSB_{32}(Y_n)$) shall be used as MAC.

[2] For the definition of A and P, please refer to the CCM specification in [25]. ‘A’ is the “associated data” and ‘P’ is the payload.

Message verification

The receiver shall verify the message exactly as it is done in the sender and as it is specified in [23].

1. Verify SeqNr (see 2.2.1.3).
2. Compute MAC based on the received fields and check that it equals the received MAC-value.

If either one of the two verifications fails, the received secure message shall be ignored and the Security Failure shall be handled. (See 2.2.1.4.)

2.2.1.2.2.5 Authentication and Confidentiality (Multicast)

2.2.1.2.2.5.1 Format of the Secure Data

The Secure Data shall contain the SeqNr, the cipher APDU and the MAC. The cipher APDU shall be based on the plain APDU encrypted with the AES-128 algorithm with CTR operation mode.

<table>
<tr><td colspan="8">octet 6</td><td colspan="8">octet 7</td><td colspan="8">octet 8</td><td colspan="8">octet 9</td><td>...</td><td colspan="8">octet m</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>...</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="35">Secure APDU</td></tr>
<tr><td colspan="6">TPCI</td><td colspan="10">Secure APCI</td><td colspan="25">Secure ASDU</td></tr>
<tr><td colspan="6"></td><td colspan="10"></td><td colspan="8">SCF</td><td colspan="17">Secure Data</td></tr>
<tr><td colspan="6"></td><td colspan="10"></td><td colspan="8"></td><td colspan="8">SeqNr</td><td>cipher text of the Plain APDU</td><td colspan="8">MAC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">12 bit</td><td colspan="8">8 bit</td><td colspan="8">6 octets</td><td>n octets</td><td colspan="8">4 octets</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8"></td><td></td><td colspan="8"></td></tr>
</table>

Legend

🟨 Plain text specific for the KNX S-A_Data-service.

🟦 encrypted text

Figure 12 – Secure Data in case Authentication and Confidentiality are used

The cipher text of the plain APDU and the MAC shall be calculated as follows.

2.2.1.2.2.5.2 Message processing (authentication and confidentiality)

Message generation

The MAC shall be calculated according to the CCM algorithm and shall then be used in the CCM encryption as specified in [23].

The same key k shall be used for authentication as well as for confidentiality.

1. In the SCF, set the field “Confidentiality” and set the field “Authentication”; fill the field SAI with 00b, to indicate “AES-128 with CTR operation mode and AES-CBC-MAC signature (CCM)”.
2. Increment SeqNr by 1
3. For Authentication and Confidentiality

   For “Authentication and Confidentiality”, P [2)] shall consist of the Plain APDU and A [2)] shall not contain the Plain APDU since it is already included in P.

   A = Secure APCI | SCF | SeqNr

   P = Plain APDU (=plain APCI + data)

   NOTE 7 The Extended CTRL-field contains the hop count. In the calculation of A, this field shall be set to 0. (The value of the hop count changes if the Frame passes a Coupler.)

4. The further calculation shall be according to Annex A.2.

   The Block B_0 shall be as specified in 2.2.1.2.2.2.

   The blocks Ctr_0 to Ctr_n shall be as specified in 2.2.1.2.2.2.
5. Referring to Figure 37, it shall be the result C (cipher text) that shall be transmitted on KNX as cipher text after the SeqNr.

Message decryption and verification

The message decryption and verification in the receiver shall start exactly as is done in the sender and as it is specified in [23].

1. Verify SeqNr (see 2.2.1.3).
2. Decrypt the received message with the security key and reconstruct the payload.
3. Compute MAC based on the received fields and check if it equals the received MAC-value.

If either one of the two verifications fails, the received secure message shall be ignored and the Security Failure shall be handled. (See 2.2.1.4.)

2.2.1.2.3 Other security algorithms

Table 2 lists the specified KNX Security Algorithms.

Values for b7 and b6 that are not defined are reserved. These values shall not be used.

A received secure message with a reserved value for b_5b_4 shall be ignored.

Expectations for possible future algorithms

The algorithm should be selected according to the available hardware performance (processor speed, memory size). The memory size is the more important criterion because the computing time is only relevant during initial configuration of the device.

2.2.1.3 Handling of the Sequence Number

2.2.1.3.1 Runtime handling

Goal

The SeqNr shall provide Data Freshness and by this protect the system against replay attacks.

Requirements for sender and receiver

The Sender shall increment the value of the Sequence Number for each subsequent transmission on any of its secure DPs. There shall be only one single Sequence Number used for all secure communication; there shall be no different sequence numbers per secure link or per secure Datapoint.

The receiver shall evaluate the value of the SeqNr for each received secure message from the sender’s IA. On KNX open media (KNX RF, KNX PL110) the sender shall additionally be identified by its KNX Serial Number and Domain Address.

Format

The Sequence Number shall be an unsigned six octet (48 bit) counter as specified in Figure 13.

NOTE 8 In B.2 it is motivated why 6 octets are used.

Sequence Number					
octet 5 MSB	octet 4	octet 3	octet 2	octet 1	octet 0 LSB
6 octets					

Figure 13 – Format of the Sequence Number

Basic principle

The Sequence Number shall be a simple counter value.

The 6 octet format makes that it takes over 100 000 year until a counter overrun happens (for calculation of this value see Annex A.2 - Figure 40). Therefore, there are no requirements on the handling of the overrun of the Sequence Number.

For the specific requirements on the handling of the SeqNr for the various communication modes, please refer to the clauses 2.2.1.5.3, 2.2.1.5.4, 2.2.1.5.5 and 2.2.1.5.6.2.

Transmission

The sender shall maintain one single Sequence Number for all its outgoing communication. This Sequence Number shall be stored in the Property “Sequence Number Sending” as specified in 2.3.2.14. The sender shall use that Sequence Number value for when transmitting on any Secure Link.

- It shall contain that value in the SeqNr in the Secure APDU (see 2.2.1.2.2.1).
- It shall use the value of the Sequence Number in the calculation of the Initialisation Vector (see 2.2.1.2.2.4.2 and 2.2.1.2.2.5.2).

If the S-A_Data-PDU is forwarded to the Transport Layer, the sender shall always increment the Sequence Number in the Security Link Table.

NOTE 9 This shall always be done regardless of the further handling of the secure message by the lower layers. Even if the transmission fails, for instance because of a negative Layer-2 acknowledge, the Sequence Number shall be incremented.

Reception

If a receiver receives an S-A_Data-PDU then it shall compare the contained Sequence Number with the Sequence Number stored as “Last Valid SeqNr” in the Security Individual Address Table (see 2.3.2.8) for the IA of the sender of the received S-A_Data-PDU.

- If the received Sequence Number is **higher** than the “Last Valid SeqNr” then the S-AL shall accept the S-A_Data-PDU.

  If the further security checks are positive then the receiver shall write the contained Sequence Number value for that sender in the Security Individual Address Table (see 2.3.2.8) as new value for “Last Valid SeqNr”.

- In case the received Sequence Number **equals** the “Last Valid SeqNr” then the S-AL shall ignore the S-A_Data-PDU. It shall not increment the Security Failure Counter.

  NOTE 10 This allows that a second or further retransmission of a valid Frame by the Data Link Layer of the sender that would by the receiver not properly be recognised as a retransmission, will not lead to a false recognition as replayed Frame.

- In case the received Sequence Number is **lower** than the “Last Valid SeqNr” then the S-AL shall do the following.
  - The S-AL shall ignore this S-A_Data-PDU.
  - The S-AL shall not block further messages from this sender. Any subsequent message from this sender, with correct SeqNr, shall again be processed as normal.
  - It shall increment the Security Failure Common Counter by 1. If the Security Failure Counters on Links is implemented, then in the entry corresponding with the link on which the error happens, the S_AL Failure Counter shall also be incremented by 1 (see 2.3.2.9.3).

Resources

As for the storage of the Sequence Number, please refer to 2.3.2.8.

Both sender as well as receiver shall store for each communication partner to which they have a secure link, store the Last Valid Sequence Number in the appropriate entry in the Security Individual Address Table (see 2.3.2.8).

Initial value

This shall be the value:

- ex-factory and prior to any secure communication on a secure link, or
- the value after a Master Reset of the device.

This initial value shall be 000000000000h.

2.2.1.3.2 S-A_Sync-service

The local S-AL user shall apply the S-A_Sync-service to obtain from the remote S-AL the Sequence Number that the remote S-AL will use and the Sequence Number that the remote S-AL expects from the local S-AL.

The local S-AL user shall set the key_select parameter to indicate whether this synchronisation shall be protected using:

- the Tool Key, or

  NOTE 11 The use case for using the Tool Key is for when the common tool, ETS ™, intends to manage a secure device and is uncertain about the Sequence Number that it should use and the Sequence Number that the device will use. The local S-AL user in this case is ETS.

- the key that is stored in the *Security Individual Address Table* for that ASAP = IA_Index (see Figure 30).

  NOTE 12 The typical use case for using the key stored for point-to-point communication in the Security Link Table, is for synchronisation: if one secure device asks the sequence number from another secure device. This functionality is at first foreseen for E-Mode devices.

The remote S-AL shall be addressed with a local ASAP, which shall be the IA_Index in the Security Individual Address Table.

The local S-AL shall accept the service request and shall build the S-A_Sync_Req-PDU as specified in Figure 14.

Secure APCI	SCF	$SeqNr_{local}$	$SeqNr_{remote}$	Challenge	MAC
10 bit	1 octet	6 octets	6 octets	4 octets	4 octets
3F1h	26h		0	N_1	

Figure 14 – Format of the S-A_Sync_Req-PDUs

The local S-AL shall challenge the remote S-AL with a challenge N_1 and shall add the Sequence Number $\text{SeqNr}_{\text{local}}$ that it assumes that it will use to communicate with the remote S-AL and the field reserved for the Sequence Number $\text{SeqNr}_{\text{remote}}$ which shall be 0.

The challenge shall be a four octet unsigned integer value. This also means that if, in the below, N_1 equals 0, that N_1-1 shall be FFFFFFFFh.

NOTE 13 The value of $\text{SeqNr}_{\text{remote}}$ is of no further value in the synchronisation and shall not be interpreted by the remote S-AL; the remote S-AL shall only use it to verify the authentication of the local S-AL user.

The local S-AL shall secure the S-A_Sync_Req-PDU with authentication and confidentiality. In function of the parameter key_select, this “key_request” shall be

- the Tool Key [3], in which case the flag T in the SCF shall be set,

or

- the key linked to the IA_Index of the remote S-AL in the Security Link Table, in which case the flag T in the SCF shall be cleared.

The local S-AL shall protect the S-A_Sync_Req-PDU with authentication and confidentiality as specified for the S-A_Data-service as specified in 2.2.1.2.2.5.

A = Secure APCI | SCF | $\text{SeqNr}_{\text{local}}$ | $\text{SeqNr}_{\text{remote}}$

P = challenge only

Ctr_0 and B_0 shall be as in 2.2.1.2.2.2; the SeqNr shall be the SeqNr of the sender, in this case $\text{SeqNr}_{\text{local}}$.

The local S-AL shall forward the request with a T_Data_Individual.req to the local Transport Layer. The parameters ASAP and priority shall be mapped to the corresponding parameters TSAP and priority of the T_Data_Individual.req primitive, the TSDU shall be an S-A_Sync_Req-PDU.

S-A_Sync.req(ASAP, key_select)

ASAP: shall either indicate the IA_Index of the IA of the communication partner or be an indication that the Tool Key is used.

key_select: shall indicate whether the communication shall be protected using the Tool Key, or whether the key shall be used as stored for the ASAP (IA_Index of the communication partner) from the Security Link Table under TSAP-type 0000b.

The local S-AL shall start a time-out timer of 6 s when the message is forwarded to the local TL.

If the remote S-AL receives a T_Data_Individual.ind primitive with TSDU = S-A_Sync_-Req-PDU, then it shall verify the security. If the flag T is set in the S-A_Sync_Req-PDU then it shall use the Tool Key; otherwise, it shall use the key that is in the Security Link Table for the TSAP-type 0000b linked with the IA_Index on which the Source Address of the message is found in the *Security Individual Address Table*.

If the remote S-AL has any error, it shall not respond to the S-A_Sync.req.

- If the remote S-AL does not find the Source Address of the S-A_Sync.req message in its *Security Individual Address Table*, then it shall ignore the message.
- If the remote S-AL encounters an error in the verification of the security, then it shall ignore the message.

[3] If the Tool Key is not assigned yet, the MaC uses the FDSK.

If the remote S-AL can properly decrypt the S-A_Sync_Req-PDU, then it shall respond with an S-A_Sync_Res-PDU as specified in Figure 15.

Secure APCI	SCF	SeqNr$_{remote}$	SeqNr$_{local}$	Challenge	MAC
10 bit	1 octet	6 octets	6 octets	4 octets	4 octets
3F1h	2Ah			N_1-1	

Figure 15 – Format of the S-A_Sync_Res-PDU

NOTE 14 In the PDU, the SCF is always followed by the SeqNr of the sender of the PDU.

The contained challenge shall be the value of the challenge of the request, decremented by 1 (N_1-1).

NOTE 15 Decrementing the challenge by 1 makes that an attacker cannot replay a previous response and that a device cannot replay a response from a previous synchronisation process.

The SeqNr$_{local}$ shall be the SeqNr that the remote S-AL expects from the local S-AL in subsequent communication and the SeqNr$_{remote}$ shall be the SeqNr that the remote S-AL will itself use in further secure communication.

NOTE 16 The SeqNr$_{local}$ in the response is the SeqNr that the remote S-AL expects the local S-AL to use in subsequent communication; this shall thus be the value that the remote S-AL last received from the local S-AL, incremented by 1. It is the lowest SeqNr that the remote S-AL will accept from the local S-AL. This value typically differs from the value in the request.

The remote S-AL shall secure the response with authentication and confidentiality using the type of key that is used in the request; this “key_response” shall thus either be

- the Tool Key or
- the key that is configured in its Security Link Table with the TSAP-type 011b, this is “sending point-to-point” for the IA_Index of the IA of the requester.

The flag T in the SCF shall be set accordingly.

The local S-AL (requester) shall be addressed with an ASAP, which shall be the IA_Index of the local (requester) IA in the Security Individual Address Table of the remote S-AL, that shall be mapped to an Individual Address by the remote Transport Layer.

The remote S-AL shall protect the S-A_Sync_Res-PDU with authentication and confidentiality as specified for the S-AL_Data-service as specified in 2.2.1.2.2.5; the key shall be the key referred in the Security Link Table for ptp-communication for the IA of the local S-AL (requester.)

A = Secure APCI | SCF | SeqNr$_{remote}$ | SeqNr$_{local}$

P = challenge only

Ctr$_0$ and B$_0$ shall be as in 2.2.1.2.2.2; the SeqNr shall be the SeqNr of the sender, in this case SeqNr$_{remote}$.

The remote S-AL shall forward the request with a T_Data_Individual.req to the remote Transport Layer. The parameters ASAP (IA_Index) and priority shall be mapped to the corresponding parameters TSAP (IA) and priority of the T_Data_Individual.req primitive, the TSDU shall be an S-A_Sync_Res-PDU.

The message sequence shall be as follows.

local S-AL remote S-AL

The S-AL_SYNC_REQ-PDU shall be secured with a the key_request as specified here.

S-AL_SYNC_REQ-PDU
$(N_1 \mid SeqNr_{local} \mid SeqNr_{remote})_{key\ request}$

The S-AL_SYNC_RES-PDU shall be secured with the key_response as specified here.

S-AL_SYNC_RES-PDU
$(N_1 -1 \mid SeqNr_{remote} \mid SeqNr_{local})_{key\ response}$

If the local S-AL receives a T_Data_Individual.ind primitive with
TSDU = S-A_Sync_Res-PDU then it shall verify the security. If the flag T is set in the S-A_Sync_Res-PDU then it shall use the Tool Key; otherwise, it shall use the key that is through the Security Individual Address Table and the Security Link Table linked with the Source Address of the message for the TSAP-type 0011b.

If the local S-AL can properly decrypt the message, then it shall confirm the service positively to the S-AL user. If there is any problem, like a security error or a time-out, then the S-AL shall confirm the service negatively to the local S-AL user.

S-A_Sync.res(ASAP, result)

ASAP:		shall either indicate the IA_Index of the IA of the communication partner or be an indication that the Tool Key is used.
error:	ok:	the Sequence Number of the remote S-AL is properly received
	not_ok:	there is an error in the request of the Sequence Number of the remote S-AL.

Error – and exception handling

- The error handling for decryption failure is given in the specification above.
- If the S-AL receives any A_Secure-PDU with SCF encoding for an S-A_Sync_Req-PDU or S-A_Sync_Res-PDU that is not transferred using point-to-point connectionless or – connection-oriented communication, then it shall ignore the message.

2.2.1.4 Registration of Security Failures

The **handling** of security failures is mandatory and is contained within the specification of the various parts of the S-AL. This clause only considers the **registration** of security failures.

The Security Failure is specified to only have to do with the failure of the security check in the S-AL service. It is not related to any other exception in the S-AL.

A Security Failure is thus only considered in the operation “Check the Security” in the example flow-charts in Figure 18, Figure 19, Figure 20, Figure 21, Figure 22 and Figure 23.

The following events shall be interpreted as Security Failure.

These events shall lead to an increment of the SFCC (see 2.3.2.17) and may lead to an increment of the S-AL Failure counter in the SFCL (see 2.3.2.9) if it is implemented.

- Decryption failure
- Mismatch between the calculated MAC-value and the MAC-value received.

The following event may be interpreted as Security Failure.

These events may lead to an increment of the Security Failure Counters on Links (see 2.3.2.9).

- Reception of a $\text{SeqNr}_{\text{local}}$ or a $\text{SeqNr}_{\text{remote}}$ that is different from the stored $\text{SeqNr}_{\text{local}}$ or $\text{SeqNr}_{\text{remote}}$ if the local S-AL assumes having a valid value for this SeqNr.

  EXAMPLE 5 If the local S-AL assumes that it has a valid $\text{SeqNr}_{\text{local}}$, but it receives a higher value for the $\text{SeqNr}_{\text{local}}$ expected from the communication partner, then this may signal that the communication partner has been addressed with properly secured messages by another client than this S-AL user. This may signal a hack, or it may simply mean that the local S-AL does after all not have the valid $\text{SeqNr}_{\text{local}}$, for instance if it is a restored back up of a visualisation software. It thus depends on the local S-AL user whether SeqNr values different than the expected ones shall be interpreted as Security Failure or not.

2.2.1.5 Secure Handling of the Transport Layer services

2.2.1.5.1 Common requirements

The Secure AL-Service shall be invoked by the TL-service primitives request(req), indication (ind) and confirmation(con).

- In reception direction, for each TL-service, the *S-AL* shall handle the following situations caused by the remote communication partner, which may be hacks or attempts to bypass the S-AL.
  - It shall filter unsecure use of links that are contained in the Security Link Table.
  - If the received message is secured, it shall verify the security.
- In transmission direction, there are no such exceptions. The choice between the plain AL-Services or the S-A_Data-service shall be given by the Security Resources.

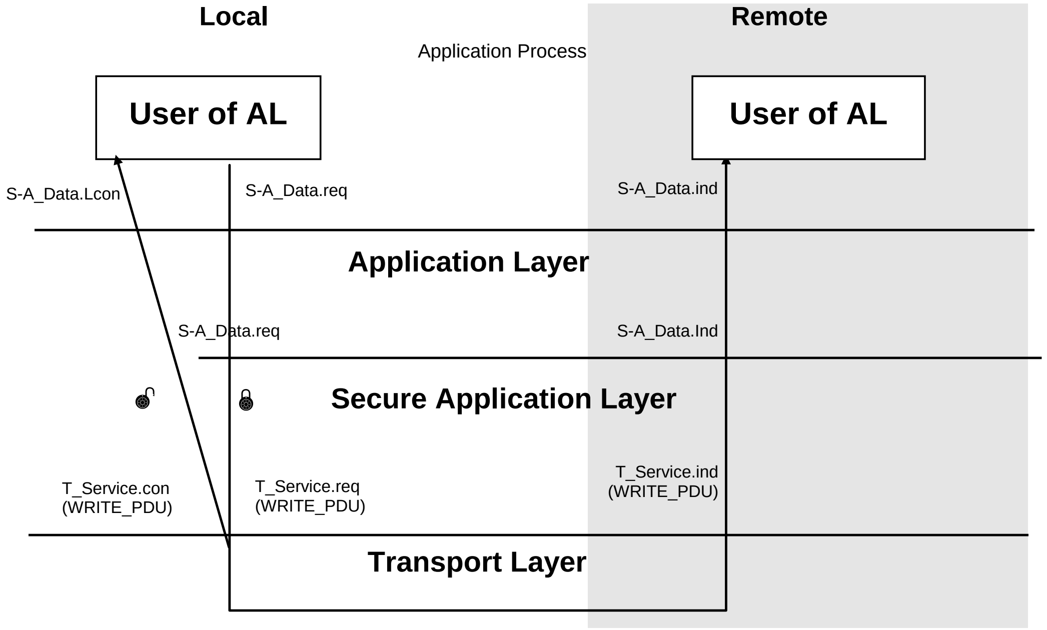

Figure 16 - Interactivity of the Application Layer for services that are not remotely confirmed

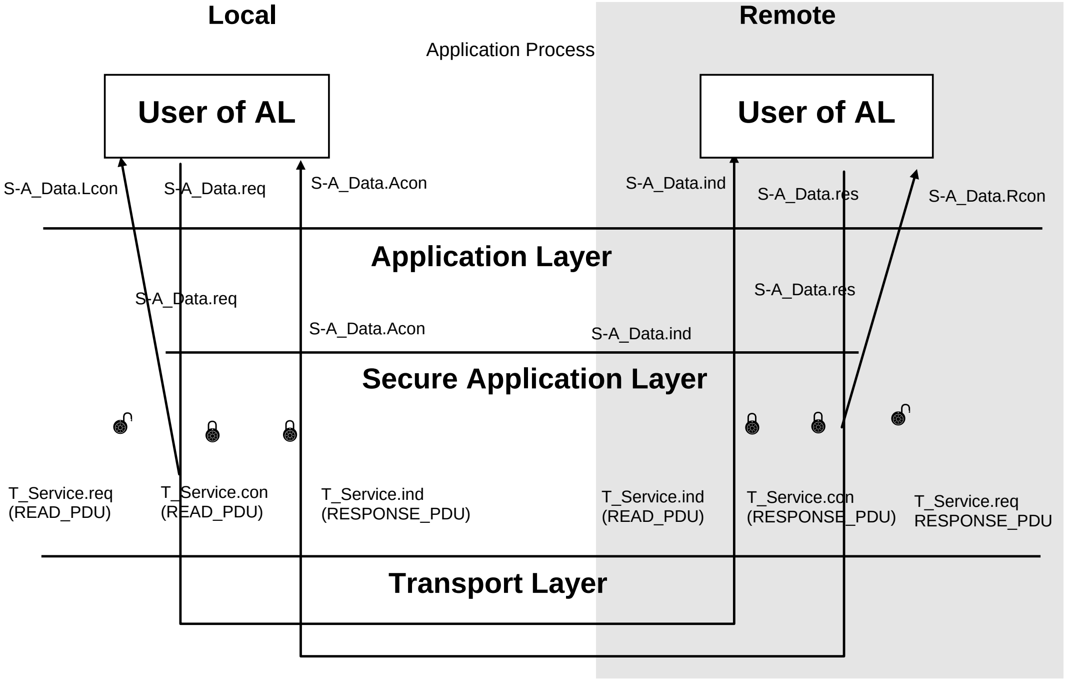

Figure 17 - Interactivity of the Application Layer for services that are remote confirmed

Any Transport Layer service indication - or confirmation primitive containing a Secure AL-Service indication or –confirmation shall in any case be handled by the *S-AL*. (This is not shown in Figure 4.)

2.2.1.5.2 Common error handling

1 In reception direction, if the security check fails AND only authentication is requested, then it would be possible to forward the Plain APDU to the S-AL, because it is available unencrypted in the Secure APDU. This means that it would be possible to access a secure DP that requires only authentication also via a badly protected APDU. This is not allowed. There are no exceptions to a failed security check.

2 In reception direction, if any of the needed SKI-Tables has any other Load State than “Loaded”, this shall also lead to the result “TSAP not found in the SKI-Tables”.

3 In transmission direction, if an AL-service is requested with any of the flags par_auth or par_conf set and if the *Security Link Table* has any other Load State than the state “Loaded” then this AL-service shall be confirmed negatively.

2.2.1.5.3 Secure AL-service on Multicast Communication Mode - T_Data_Group

2.2.1.5.3.1 References

For the specification of the T_Data_Group-service and the handling of TSAPs and ASAPs in the Group Object Association Table, please refer to clause 3.1.1 “The relation between TSAPs and ASAPs” in [03].

2.2.1.5.3.2 Reception - T_Data_Group.ind and T_Data_Group.con

The below sequence specifies a decision tree to come to a single conclusion on handling the T_Data_Group.ind.

1. If the T_Data_Group.ind contains an S-A_Data-PDU then the S-AL shall search the TSAP in the *Security Link Table*. The TSAP-type shall be “receiving Group Address”.

   1.1 If the TSAP is found in the *Security Link Table*, then the S-AL shall decrypt the contained S-A_Data-SDU according the security features indicated in the SCF and check the security of the message.

   1.1.1 If the security check results positive then the S-AL shall inform the P-AL according the contained A_GroupValue_Read.ind(/.Acon) or A_GroupValue_Write.ind(/.Acon) as decoded from S-A_Data-service. The security service parameters par_auth and par_conf shall be set as used in the S-A_Data-PDU and the index of the secure link (link_index) shall be forwarded as well.

   1.1.2 If the security check results negative then the S-AL shall consider this as an attack and shall handle this. (See also exception 1 below The message shall not be forwarded to the P-AL.

   1.2 If TSAP is not found in the *Security Link Table* then the S-A_Data-PDU shall be ignored. (See also exception 2). The message shall not be forwarded to the P-AL.

2. If the T_Data_Group.ind does not contain an S-A_Data-PDU then the S_AL shall forward the message unchanged to the P-AL. The security service parameters *par*_auth and *par*_conf shall both be cleared; the parameter link_index shall be “none”. The T_Data_Group.ind can at this point only contain an APCI_GroupValue_Write, APCI_GroupValue_Read or an APCI_GroupValue_Response.

Error - and exception handling

The common error handling applies, as specified in 2.2.1.5.2. There is no further error – and exception handling specific to the service primitives T_Data_Group.ind or T_Data_Group.con.

Flowchart example

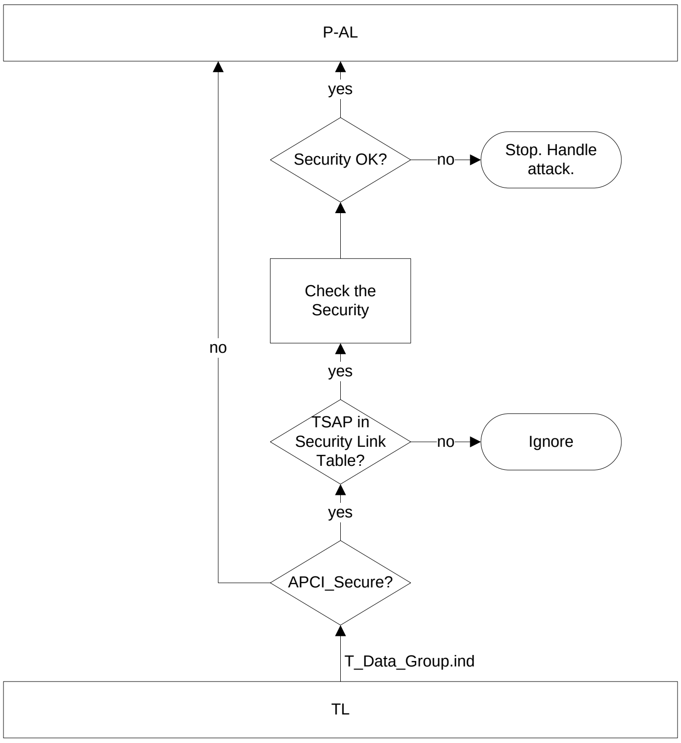

Figure 18 – Handling of T_Data_Group.ind by S-AL (informative)

2.2.1.5.3.3 Transmission - T_Data_Group.req

If the S-AL handles an APDU (APCI + Data) that shall be transmitted using a T_Data_-Group.req then the S-AL shall conclude on the handling based on the security service parameters *par_auth* and *par_conf* of the AL-service according the following decision tree.

1. The S-AL shall check the security service parameters *par_auth* and *par_conf* of the AL-service.

   1.1 If any of these parameters is set, then the S-AL shall check whether the TSAP used for the T_Data_Group.req is contained the *Security Link Table*. The TSAP-type shall be “sending Group Address”.

   1.1.1 If the TSAP is found in the *Security Link Table* then the S-AL shall apply the security (authentication and/or confidentiality) as requested in the service and build the Secure APDU (see 2.2.1.2.2.1), which shall be forwarded with a T_Data_Group.req to the TL. (See common error 3.)

   1.1.2 If the TSAP is not found in the *Security Link Table* then the S-AL shall confirm the AL-service negatively to the AIL.

   1.2 If none of the security service parameters is set then the S-AL shall forward the APDU unchanged with a T_Data_Group.req to the TL.

Error - and exception handling

The common error handling applies, as specified in 2.2.1.5.2. There is no further error – and exception handling specific to the service primitive T_Data_Group.req.

Flowchart example

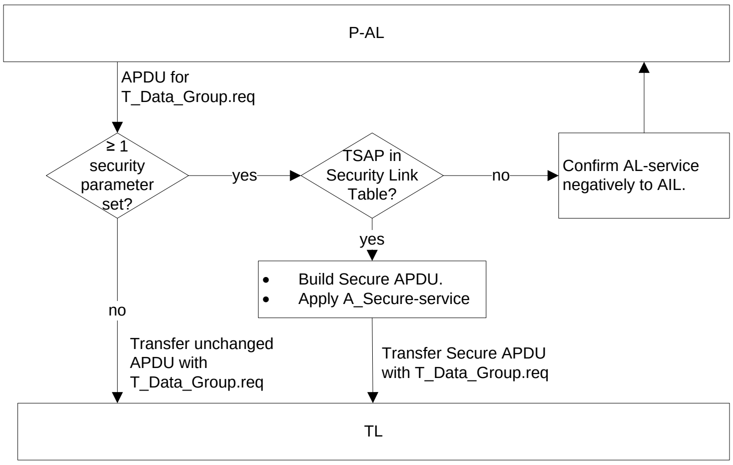
P-AL
APDU for T_Data_Group.req
≥ 1 security parameter set?
yes
TSAP in Security Link Table?
no
Confirm AL-service negatively to AIL.
yes
Build Secure APDU.
Apply A_Secure-service
no
Transfer unchanged APDU with T_Data_Group.req
Transfer Secure APDU with T_Data_Group.req
TL

Figure 19 - Handling of T_Data_Group.req by S-AL (informative)

2.2.1.5.4 Secure AL-service on Multicast Communication Mode - T_Data_Tag_Group

2.2.1.5.4.1 References

For the specification of the T_Data_Tag_Group-service, please refer to [02].

In the below, the Security Link Table is used. Only these records shall be used that have the field “Extended Frame Format” equal to the EFF-Type of the TL-service – or AL-service primitive being handled.

2.2.1.5.4.2 Reception - T_Data_Tag_Group.ind and T_Data_Tag_Group.con

The below sequence specifies a decision tree to come to a single conclusion on handling the T_Data_Tag_Group.ind.

1. If the T_Data_Tag_Group.ind contains an S-A_Data-PDU then the S-AL shall search the zone address (TSAP) in the *Security Link Table*. The TSAP-type shall be “receiving LTE Group Address”.

   1.1 If the zone address is found in the *Security Link Table* – possibly with the evaluation of wildcards -, then the S-AL shall decrypt the contained S-A_Data-SDU and check the security of the message.

      1.1.1 If the security check results positive then the S-AL shall inform the P-AL according the contained A_GroupPropValue_Read.ind, A_GroupPropValue_Response.ind, A_GroupPropValue_Write.ind or A_GroupPropValue_InfoReport.ind as decoded from the S-A_Data-service.

      The security service parameters *par_auth* and *par_conf* shall be set as used in the S-A_Data-PDU and the index of the secure link (link_index) shall be forwarded as well.

1.1.2 If the security check results negative then the S-AL shall consider this as an attack and shall handle this. The message shall not be forwarded to the P-AL.

1.2 If the zone address is not found in the Security Link Table then the S-A_Data-PDU shall be ignored. (See also 2). The message shall not be forwarded to the P-AL.

2. If the T_Data_Tag_Group.ind does not contain an S-A_Data-PDU then the S-AL shall forward the T_Data_Tag_Group.ind unmodified to the P-AL. The security service parameters *par*_auth and *par*_conf shall both be cleared; the parameter link_index shall be "none".

Error - and exception handling

The common error handling applies, as specified in 2.2.1.5.2. There is no further error – and exception handling specific to the service primitives T_Data_Tag_Group.ind and T_Data_Tag_Group.con.

Flowchart example

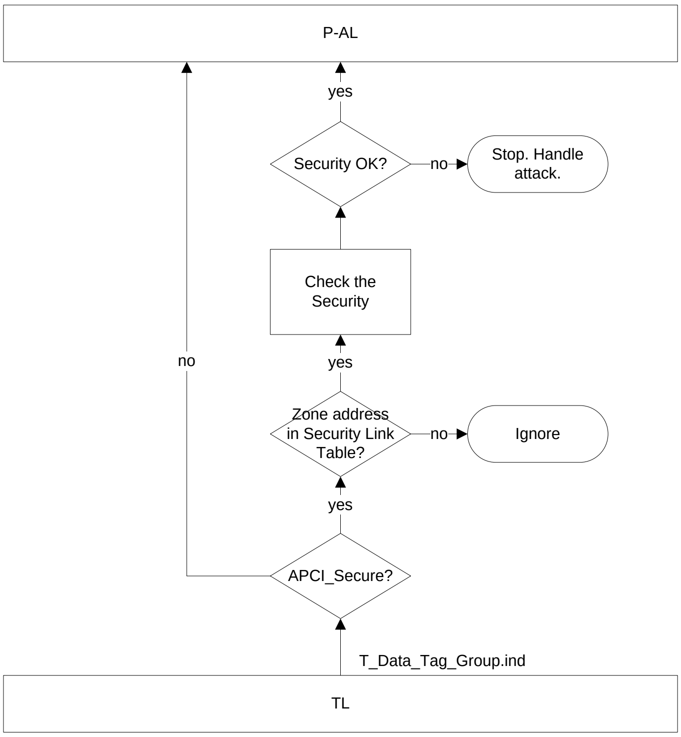

Figure 20 – Handling of T_Data_Tag_Group.ind by S-AL (informative)

2.2.1.5.4.3 Transmission – T_Data_Tag_Group.req

If the S-AL handles an APDU (APCI + Data) that shall be transmitted using a T_Data_Tag_Group.req then the S-AL shall conclude on the handling based on the security service parameters *par_auth* and *par_conf* of the AL-service according the following decision tree.

1. The S-AL shall check the security service parameters *par_auth* and *par_conf* of the AL-service.

1.1 If any of these parameters is set, then the S-AL shall search the zone address and the used Frame format of the used AL-service in the *Security Link Table.*

1.1.1 If the combination TSAP + frame format is found in the *Security Link Table* then the S-AL shall apply the security (authentication and/or confidentiality) as requested in the AL-service parameter and build the Secure APDU (see 2.2.1.2.2.1), which shall be forwarded with a T_Data_Tag_Group.req to the TL.

1.1.2 If the TSAP + frame format is not found in the *Security Link Table* then the S-AL shall confirm the AL-service negatively to the AIL.

1.2 If none of the security service parameters is set then the S-AL shall forward the APDU unchanged with a T_Data_Tag_Group.req to the TL.

Error – and exception handling

There is no specific error – or exception handling specified.

Flowchart example

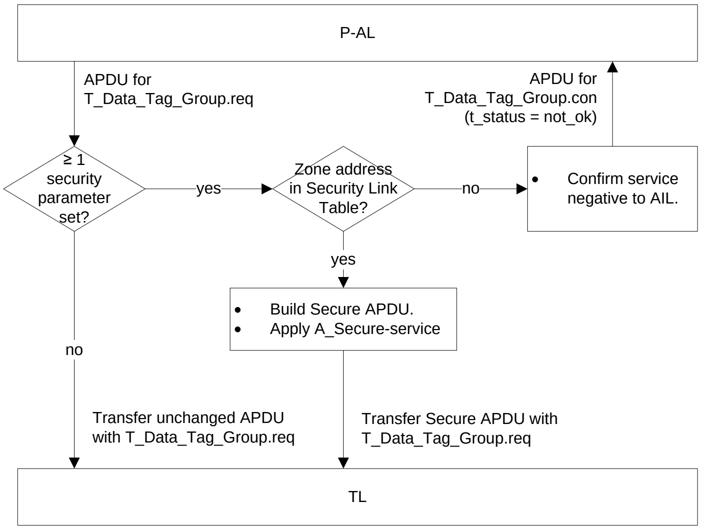
P-AL
APDU for T_Data_Tag_Group.req
APDU for T_Data_Tag_Group.con (t_status = not_ok)
≥ 1 security parameter set?
yes
Zone address in Security Link Table?
no
Confirm service negative to AIL.
yes
Build Secure APDU.
Apply A_Secure-service
no
Transfer unchanged APDU with T_Data_Tag_Group.req
Transfer Secure APDU with T_Data_Tag_Group.req
TL

Figure 21 - Handling of T_Data_Tag_Group.req by S-AL (informative)

2.2.1.5.5 Secure AL-Service on point-to-point communication mode

2.2.1.5.5.1 Introduction

This clause is valid for the TL-services T_Data_Individual as well as T_Data_Connected.

For the specification of these services, please refer to [02].

NOTE 17 The TL-services T_Connect and T_Disconnect are handled by the TL itself. These do not contain an APDU (APCI) and can thus not be handled by the S-AL. This means that there is no security on the opening or closing of a TL-connection to a secure device.

2.2.1.5.5.2 Reception – T_Data_Individual.ind/con and T_Data_Connected.ind/con

The below sequence specifies a decision tree to come to a single conclusion on handling the T_Data_Individual.ind, T_Data_Individual.con, T_Data_Connected.ind and T_Data_Connected.con.

1. If the TL-service primitive contains an S-A_Data-PDU then the S-AL shall search the Source Address in the *Security Link Table*. The TSAP-type shall be “receiving point-to-point”.

   On KNX RF, additionally the KNX RF Domain Address of the sender shall be compared and be equal.

   1.1 If the Source Address is found in the *Security Link Table* for this TSAP-type then the S-AL shall decrypt the contained S-A_Data-SDU according the security features indicated in the SCF and check the security of the message.

   1.1.1 If the security check results positive then the S-AL shall inform the P-AL according the contained AL-service primitive as decoded from the S-A_Data-service. The security service parameters *par_auth* and *par_conf* shall be set according the use of authentication respectively confidentiality in the S-A_Data-SDU and the index of the secure link (link_index) shall be forwarded as well.

   1.1.2 If the security check results negative then the S-AL shall consider this as an attack and shall handle this. The message shall not be forwarded to the P-AL.

   1.2 If the Source Address is not found in the SKI-Table for this TSAP-type then the S-A_Data-PDU shall be ignored. The message shall not be forwarded to the P-AL.

2. If the TL-service primitive does not contain an S-A_Data-PDU then the S_AL shall forward the message unchanged to the P-AL. The security service parameters *par*_auth and *par*_conf shall both be cleared; the parameter link_index shall be “none”.

Error – and exception handling

1. There is no relation between a hack attempt and the TL state machine. If a security check of a T_Data_Connected.ind fails, then this has no repercussions on the TL State Machine. Any open Transport Layer connection is not closed.

Flowchart example

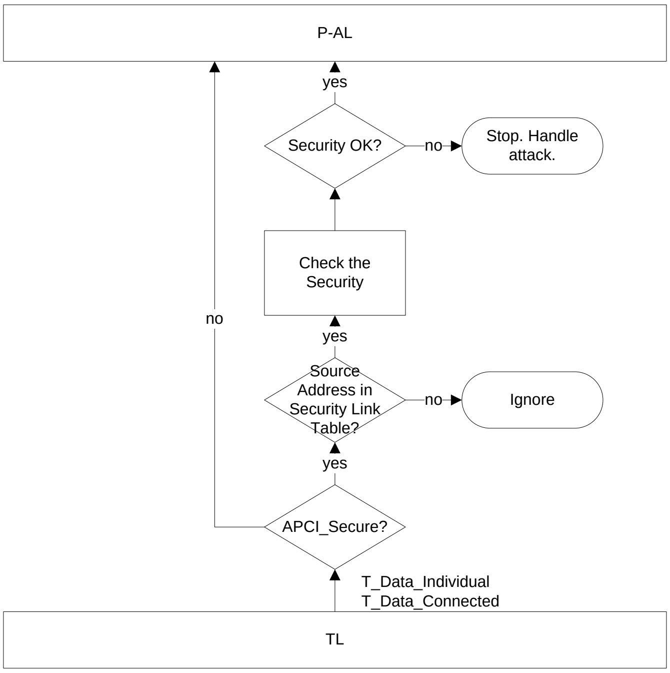

Figure 22 - Handling of T_Data_Individual.ind/.con and T_Data_Connected.ind/.con by S-AL (informative)

2.2.1.5.5.3 Transmission – T_Data_Individual.req and T_Data_Connected.req

If the S-AL handles an APDU (APCI + Data) that shall be transmitted using a T_Data_-Individual.req or T_Data_Connected.req then the S-AL shall conclude on the handling based on the security service parameters *par_auth* and *par_conf* of the AL-service in the following decision tree.

1. The S-AL shall check the security service parameters *par_auth* and *par_conf* of the AL-service.

    1.1 If any of these parameters is set, then the S-AL shall check whether the TSAP used for the T_Data_Individual.req or T_Data_Connected.req is contained in the Security Link Table with TSAP-type = “sending point-to-point”.

    1.1. 1 If the Destination Address is found in the *Security Link Table* for this TSAP-type then the S-AL shall apply the security (authentication and/or confidentiality) as requested in the service and build the Secure APDU (see 2.2.1.2.2.1), which shall be forwarded with a T_Data_Individual.req or T_Data_Connected.req to the TL.

    1.1.2 If the TSAP is not found in the *Security Link Table* for this TSAP-type then the S-AL shall confirm the AL-service negatively to the AIL.

    1.2 If none of the security service parameters is set then the S-AL shall forward the APDU unchanged with a T_Data_Group.req to the TL.

Error – and exception handling

The common error handling applies, as specified in 2.2.1.5.2. There is no further error – and exception handling specific to the service primitives T_Data_Individual.req or T_Data_-Connected.req.

Flowchart example

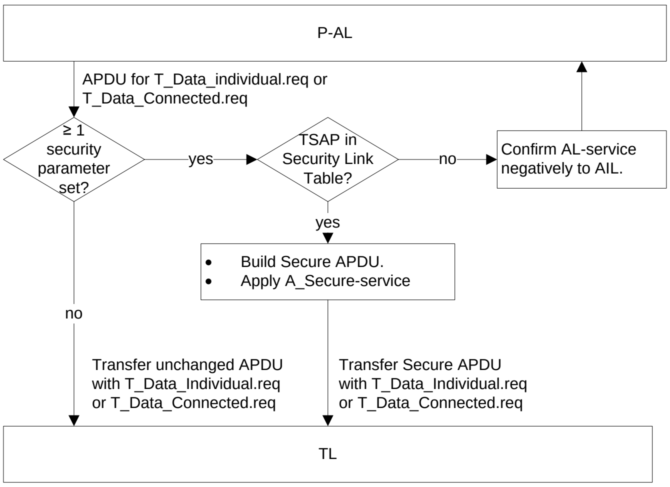
P-AL
APDU for T_Data_individual.req or T_Data_Connected.req
≥ 1 security parameter set?
yes
TSAP in Security Link Table?
no
Confirm AL-service negatively to AIL.
yes
Build Secure APDU.
Apply A_Secure-service
no
Transfer unchanged APDU with T_Data_Individual.req or T_Data_Connected.req
Transfer Secure APDU with T_Data_Individual.req or T_Data_Connected.req
TL

Figure 23 - Handling of T_Data_Individual.req and T_Data_Connected.req by S-AL (informative)

Notes to keys for point-to-point communication (informative)

- Receiving point-to-point and sending point-to-point are different TSAP-types in the Security Link Table. It is thus possible that a device uses in it secure point-to-point communication with a remote partner, a different keys for securing outgoing messages than for verifying incoming messages from that partner.
- Also, a secure device may use a different key per communication partner.

This depends on the configuration by the MaC.

2.2.1.5.6 Secure AL-Service on (system) broadcast communication mode

2.2.1.5.6.1 References

This clause is valid for the TL-services T_Data_Broadcast and T_Data_SystemBroadcast. For the specification of these services, please refer to [02].

The below specification is given for the T_Data_Broadcast, but the protocol shall be the same for the T_Data_SystemBroadcast.

KNX Data Security is not defined for (system) broadcast communication mode. Therefore, the below only defines the exception handling.

2.2.1.5.6.2 Reception – T_Data_Broadcast.ind, T_Data_Broadcast.con, T_Data_System-Broadcast.ind and T_Data_SystemBroadcast.con

1. If the T_Data_Broadcast.ind contains an S-A_Data-PDU then the S-AL shall discard this message. The message shall not be forwarded to the P-AL.

   NOTE 18 The Security Link Table does not contain a TSAP-type for (system) broadcast communication, so it is not possible to verify the security of this message.

2. If the T_Data_Broadcast.ind does not contain an S-A_Data-PDU then the S_AL shall forward the message unchanged to the P-AL.

Error – and exception handling

None.

2.2.1.5.6.3 Transmission - T_Data_Broadcast.req and T_Data_SystemBroadcast.req

If the S-AL handles an APDU (APCI + Data) that shall be transmitted using a T_Data_Broadcast.req or T_Data_SystemBroadcast.req then the S-AL shall conclude on the handling according the following decision tree.

1. The S-AL shall check the security service parameters *par_auth* and *par_conf* of the AL-service.

   1.1 If any of these parameters is set, then the S-AL shall confirm the AL-service negatively to the AIL.

   1.2 If none of the security service parameters is set then the S-AL shall forward the APDU unchanged with a T_Data_Broadcast.req to the TL.

Error – and exception handling

None.

2.2.2 Application Layer

🗎 *This document does not specify the modifications of the AL-services. Instead, it indicates how the existing AL-services shall be modified for AL to forward the security service parameters between the S-AL and the AIL.*
This clause is as such not intended to be integrated in the KNX Specifications, but does explain how the AL-services in [03] shall be modified.

2.2.2.1 A_GroupValue_Write service

🗎 *The specification in [03] shall be modified as follows (modifications in red).*
This kind of modification shall be done for each AL-service, unless we find a common way to specify this, without having to modify all services.

The local Application Layer shall accept the service request and map the ASAP to the TSAP (IA_Index).

If none of the security service parameters *par_auth* or *par_conf* is set, then the local Application Layer shall forward the APDU with a T_Data_Group.req to the local Transport Layer. The user decides during configuration about this mapping. The parameters TSAP and priority shall be mapped to the corresponding parameters of the T_Data_Group.req primitive, the TSDU shall be an A_GroupValue_Write-PDU.

If one or more of the security service parameters is set, then the local Application Layer shall forward the APDU with an S-A_Data.req to the Secure Application Layer. Please refer to the specifications of the S-AL for the handling of Secure AL-services (see 2.2.1.5.3.3) for the further handling by the S-AL.

🗎 *Additionally, it shall be added for confirmed AL-services (A_GroupValue_Read, A_PropertyValue_Read and other), the following.*

If the service is requested with the Tool Key, then the remote AL shall confirm it with the Tool Key. If the service is not requested with the Tool Key, then the response service primitive ASAP shall identify the security key that shall be used.

2.2.2.2 All service primitives

The specifications of all AL-services shall be extended with the following security service parameters.

par_auth: This parameter shall indicate whether the service is communicated using secure communication with authentication or not.

par_conf: This parameter shall indicate whether the service is communicated using secure communication with confidentiality or not.

🗎 *For the Ind-, Lcon- and Acon-service primitives, the following shall be added.*

link_index: This parameter shall contain the Link Index of Link over which this service is handled or be an indication that the Tool Key is used.
If the Link is not found in the Security Link Table, then this parameter shall have the value “none”.

🗎 *For the Req and Res service primitives, the following shall be added.*

link_index: This parameter shall be an indication if the Tool Key shall be used or not.

2.2.3 Application Interface Layer

🗎 *This clause is intended for integration in [04] as new clause 4.6 “KNX Data Security for Interface Objects”. The existing clause 4.6 “Interface Object Interworking” shall be shifted down.*

2.2.3.1 General requirements and overview

The AIL shall have the following functionality.

-	Support Permissions and Roles	see clause 2.2.3.2
-	Support the Security White Lists and the Security Black Lists	see clause 2.2.3.3
-	Support the Security Mode	see clause 2.2.3.4
-	Handle Security Failures in the AIL	see clause 2.2.3.5

The following clauses specify these and other functions.

Note to the realisation

The below clauses give common requirements as a structured specification, differentiating between acceptance for services and Datapoints, White – and Black List and Security Mode. However, within the S-AL, there are no interfaces (data interfaces, message interfaces, API etc.) and no Resources standardised. The below clauses thus give requirements to a black box behaviour: it is thus possible follow the same structure in the implementation, or to have a more monolithic solution in the implementation that solves all requirements in a single step or resource. Annex C gives examples of possible realisations.

2.2.3.2 Permissions and Roles

2.2.3.2.1 Introduction

2.2.3.2.1.1 Motivation (informative)

For group communication, the Security Link Table in the S-AL contains the SA and the GA and can thus control which sender has access to which GO. However, this does not allow differentiating between the multicast services and it does not allow further differentiation of any security conditions. The S-AL controls the secure communication of a GA by a SA, but forwards the plain use of that same GA by a sender (IA) that is not in the Security Link Table.

EXAMPLE 6 It may be possible that a GO shall from any sender only be readable using A_GroupValue_Read, but that it shall only be writeable with A_GroupValue_Write using authentication. This cannot be controlled through the Security Link Table.

For all other communication modes, the security is linked to the SA of the sender only. The Security Link Table does not allow differentiation in the authorization between two or more senders: any sender that can properly secure its messages has further full access to all the data and services in the AIL and Management.

EXAMPLE 7 The restart of a KNX S-Mode device is normally only executed by the MaC at the end of the configuration, or by the user, for diagnostic purposes. A restart excludes the MaS for some time from the runtime communication and may also reset certain variables in the MaS. The restart may thus be a weakness in the security. Hence, it is better if the restart can only be requested by an authorised communication partner.

EXAMPLE 8 If an automation client authenticates itself to get access to the schedules of an HVAC control system, then - using this authentication - the automation client also has access to the linking and all other parameter values of that device. Vice versa, the properly authenticated ETS user will have access to the HVAC application parameters.

EXAMPLE 9 In a meter device, a MaC 1 may set tariff information and may read the registers for billing, but may not read the current consumption information. That is only available to a MaC 2, e.g. a display, which then again can only read, but not modify, the tariff information.

To be able to control which sender has what kind of access to which data and which services in the receiver, it is necessary that Permissions are defined.

2.2.3.2.1.2 Permissions - introduction

A Permission defines for a communication partner for each piece of data (Group Object, Property, memory location or other) or service in the receiver whether it has access and what the conditions are.

2.2.3.2.1.3 Roles

There can be many Links using Secure Communication and there can be many Secure Datapoints in a secure application. A large amount of Permissions would have to be set if Permissions would have to be defined for each combination of Link and Datapoint or service.

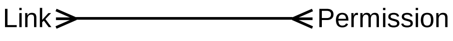

Figure 24 – Many Links with many different Permissions cause a lot of definition data

Instead, it is however assumed that the Permissions to a DP will not differ so much between the Links. It may thus be better to group Links with the same Permissions in all DPs together, and define the Permissions for a group of Links, instead of for each separate Link. Such group of Links that have the same Permissions in a device is called a *Role*.

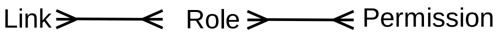

Figure 25 – The introduction of Roles reduces the amount of different Permission definitions for each DP.

NOTE 19 The use of Roles has the following benefits.

- It keeps the size of the Permissions Tables smaller.
- It reduces the complexity. Instead of having an n-to-m relationship between Links and Permissions, it introduces an intermediate Level of 1-to-1 relationships.
- This eases the management and the administration.
- It is easier to understand for the ETS user.

2.2.3.2.2 Permissions – requirements

Conceptual - Permissions Table

The Permissions Table is a functional extension of the description of the DP. One Permission shall for one Role give the access conditions to the DP or service.

It is a functional extension of the *Group Config Flags* of a GO

It is also a functional extension of the access levels and read-only flag of the Property description. The A_PropertyDescription_Response-PDU is not modified for this and shall contain the access conditions (write flag and access levels) of the “unlisted” access, this is, for the MaC that has no permissions for the Property, even if the MaC uses authentication in the request.

Resource

There is no standard KNX Resource for the storing of Permissions. The format is implementation specific. Annex C.2 shows some simple examples; the implementation may optimise in terms of size and evaluation speed and its relationship to the Role Table (see below).

For the ease of understanding, the definition of all Permissions for all Roles for one DP or service is called a *Permissions Table*. Note again that this is not a KNX standard Resource. Further, it is assumed that each DP or service has one Permissions Table. This may however be realized differently.

Evaluation and functioning

The AIL shall evaluate the Permissions Table to check whether a certain link and a certain service give access to a Datapoint or service or not. If this evaluation is negative, the normal error handling for the service shall apply.

EXAMPLES

10 An A_GroupValue_Write.ind will not lead to an update of the GO-value.

11 An A_GroupValue_Read.ind will not be responded. (The request Telegram will however be acknowledged at Data Link Layer level as the Group Address is present in the Group Address Table).

12 An A_PropertyValue_Write.req or an A_PropertyValue_Read.ind will be confirmed with an A_PropertyValue_Response-PDU with nr_of_elem = 0 and no data.

In the evaluation of the Permissions, it should be possible to conclude on the access rights for Roles that are not listed in the Permissions Table. The solution for this is implementation specific. This can for instance be solved through an internal Role “unlisted”. Great care shall be taken in this, as a faulty configuration may make the entire Permissions mechanism to be void.

NOTE 20 The Role “unlisted” is commonly better known as the Role “anonymous”. As however on KNX, the message always contains at least a Source Address, such communication is not really anonymous, but rather using a link that is not listed in the Security Link Table. Hence, the name “Unlisted”.

EXAMPLE 13 For a GO, it may be defined that any access using a Role that is not in the Permissions Table, only gives the read-access and does not allow writing the GO-value. For a Property, such access may only allow reading the Property description, but not allow reading or writing the Property Value.

Further on, the Permissions Table can be optimised by using wild cards or ranges for the Roles. This is not standardised.

Management and Configuration

There are no standard Management – and Configuration requirements for the handling of the Permissions Tables.

Yet, it is allowed that the Permission definitions are modified as part of the Configuration Procedure. In S-Mode, this can be part of the memory image (memory mapped parameterisation) or realised through MergeIDs and Load Controls.

It is recommended that the memory where the Permissions Table is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.2.3.2.3 Roles – requirements

Evaluation of Roles in the KNX devices

Roles can be used in KNX Data Security but may also be used outside that scope.

In the receiver, each Link shall be associated with exactly one Role through the Roles Table, which over the Permissions then finally controls the access to each separate DP.

The **sender** does not know what Role it is assigned in the Receiver. It consequently does not know its Permissions in the receiver. This is a matter of configuration in the receiver. The sender cannot control this. Different senders can in the same device have the same or different Roles.

ETS - Role

Roles are not standardised within KNX. However, every KNX device shall at least foresee the “ETS – role”. The MaC shall foresee at least one Link in the Security Link Table that will be related to the “ETS – role” in the Role Table.

EXAMPLES of Roles

- ETS - role
- Diagnostic Tool - role
- Visualisation Tool - role
- End User - role
- The heating technician - role
- The lighting controller - role
- A plain secure lighting sensor - role

NOTE 21 It is thus possible to make that the “Tool Key” cannot be used for group communication. (The application developer simply does not link this Role to any GO.)

Unlisted Role

Messages may be received on Links that are not contained in the Security Link Table. Obviously, these will be messages that are not protected using KNX Data Security. (S-AL will not forward these messages to P-AL). In order to grant or deny Permissions in such communication, this shall be assigned the Role “Unlisted”.

Links in the Security Link Table using Plain communication

An implementation specific extension of the Security Link Table may consist of entries that do not use KNX Data Security. This is an extension of the above Unlisted Role and allows relating different, non-authenticated senders to different Roles.

Definition and assignment of Roles

The application developer defines the Roles: the number of Roles, their “rights” within the device and their dependency.

The ETS user assigns the Roles: which sender gets which role.

For one Datapoint or service, there can be multiple Roles with different permissions. However, different Roles may have the same Permissions in one DP or service; these Roles may then have different Permissions in other DPs or services.

Roles are not “access levels”. There is no mandatory hierarchy between the Roles. As the Roles are chosen by the application developer, he can of course make them behave like access levels.

Roles are related to links in the Security Link Table, this is, can be defined for senders that at least authenticate themselves properly. In order to grant permissions to senders that are not in the Security Link Table, the implementation can use a role “Unlisted” (any sender that is not in the Security Link Table).

Role Table – general requirements

🗎 *This clause gives the general requirements for the Role Table, as parameter within the AIL, for integration in the KNX Chapter 3/4/1 “Application Interface Layer”.*
For the specification as a standard Resource (Property), please refer to 2.3.2.15, which will be integrated in Chapter 3/5/1 “Resources”.

The Role Table shall associate each link in the Security Link Table to one or more Roles.

The AIL shall evaluate the Role Table only in reception direction. In transmission direction, any message shall just be sent out with its SA (IA) and DA (GA) as usual.

There may be Links that do not appear in the Role Table. For these Links, it is thus not possible to specify Permissions in the Datapoints. These Links can thus possibly only use plain communication, or are in the Permissions evaluated as “Unlisted”.

- For group communication, a Link contains SA and DA (GA) and this means that a Sender can be assigned more than one Role in a receiver.

  multicast communication mode

  1 SA + 1 GA → 1 Role

- For point-to-point communication, the Security Link Table contains the TSAP-Type value 0000h. For a Sender to have different Roles in a Receiver, the sender may thus exhibit different Individual Addresses. However, the Link Index may as well appear multiple times in the Role Table. One sender may thus have multiple Roles in the receiver.

  point-to-point

  1 SA → ≥ 1 Role

In case the Role Table foresees more than one Role for a Link Index, the Permissions Table may have to be evaluated for all Roles assigned to this Link Index.

EXAMPLE 14 In a heating system, the Role "*HVAC Technician*" may set the minimal – and the maximal water temperature in function of the dimensions of the HVAC installation and the building. The Role "*building owner*" cannot change these values, but may set the minimal – and maximal room temperature, to guarantee minimal comfort and prevent from excess energy consumption. The Role "renter" in turn can modify neither of these values, but may only access the small temperature shifts for comfort. In a larger, commercial building, these Roles will be executed by three different persons, each assigned his own Role. In a small building, that is configured and inhabited by the same person; this person will have the three Roles at the same time.

- Two Links may be assigned the same Role.

Figure 26 shows an example of a Role Table.

Link Index	Role
1	A
2	A
4	B
5	C
7	B
7	D

Figure 26 – Example of a simple Role Table (informative)

NOTE 22

The Security Link Table is a Resource of the S-AL.

The Role Table is a Resource of the AIL.

The S-AL forwards the Link Information as AL-service parameter, over the AL, to the AIL.

2.2.3.3 Security White Lists and Security Black Lists

2.2.3.3.1 Introduction

The implementation of the Roles and the definition of the Permissions are implementation specific. Through these, it shall be possible for the KNX application – and stack developer to control for each DP and whether KNX Data Security will be needed, and how, or not.

Security White List

Yet, there shall be DPs and services that **shall always be accepted in plain communication, even regardless of the Security Mode**. These are mainly services that serve the support for network configuration and minimal MaC compatibility. These are listed in the Security White List. There shall be a Security White List for services and a Security White List for Datapoints.

Security Black List

There shall however also be DPs and services that **shall never be accepted in plain communication, regardless of the Security Mode** and that shall only be accessible using KNX Data Security with confidentiality (not with authentication only). These are mainly services and DPs that serve the configuration of the KNX Data Security parameters, as mainly stored in the Security Link Table, like the keys and the links. These are listed in the Security Black List. There shall be a Security Black List for services and a Security Black List for Datapoints.

Resources for Security White Lists and Security Black Lists

There are no Resources standardised for the White – or Security Black Lists. These lists instead impose requirements on the implementation specific Permissions. The KNX application – and stack developer thus shall take care in the definition of the Permissions, that these White – and Security Black Lists are respected.

2.2.3.3.2 Rules for Security White Lists and Security Black Lists

These Rules do not express requirements to the implementation, but express general conditions for the current and future contents of the Security White List and the Security Black List.

- A service or DP can only be in the Security Black List or in the Security White List; it cannot be in both lists.
- If a service or DP does not appear in the White – or Security Black List, then the KNX stack – and application developer can set the permissions for this service or DP implementation specific.
- If a service can access DPs that are not all in the Security White List or not all in the Security Black List, then this service cannot be in any list; instead, it has to be considered for each of the DPs that it may access whether it belongs in the Security White List for DPs, the Security Black List for DPs or whether the permissions for that DP are implementation specific.

2.2.3.3.3 Implementation specific extensions of the White –and Security Black List

Table 4 and Table 5 define the White – and Security Black Lists. An implementation may additionally have further services and DPs that shall always, respectively never be accepted using plain communication. This may be solved by an extension of the White - and Security Black Lists, or over the definition and interpretation of Roles and Permissions, which are anyhow implementation specific.

These implementation specific extensions shall however not contradict with the requirements of Table 4 and Table 5.

2.2.3.3.4 Security White List and Security Black List for services

This list is normative: it is not specified in the specification of each service if it is on the Security Black List, on the Security White List or on no list.

Legend

M	This service or DP shall be in this list.
X	This service or DP shall not be in this list.

Table 4 – White - and Security Black Lists for services

Service	Security White List	Security Black List
In this version of the KNX Data Security specifications, this table is empty.		
NOTE 1 This table does not contain (system) broadcast services, as it is not possible to apply KNX Data Security on these communication modes. These services will thus always be accepted.		
[a] The A_Link_Read-service is not contained in this list. This document version does not contain E-Mode requirements.		

2.2.3.3.5 Security White List and Security Black List for Datapoints

This list is normative and complementary to the DP (Property) specifications, as given for the KNX Data Security Properties of the Security Interface Object in clause 2.3.2.2 and further. These specifications are also exclusive: if a certain service is not specified for a Property, then that shall not give access to that Property.

EXAMPLE 15 For the Property Security Mode, only the “Function Property Service” is listed (A_FunctionProperty_Command and A_FunctionPropertyState_Read). This means that the Property shall in no way be accessible using the Data Property Services A_PropertyValue_Write and A_PropertyValue_Read.

The access to the Property may be subject to prior Authorization. It shall be noted that an authorization with the Role “ETS” does not automatically imply an authorization according a dedicated, appropriate access level! It may thus still be necessary that the MaC properly Authorizes with the A_Authorize-service.

Table 5 – White - and Security Black Lists for Datapoints

Datapoint	Security White List	Security Black List
Manufacturer Code	M	X
Any Interface Object Type.	M	X
Any Interface Object Index.	M	X
Security Interface Object		
- Interface Object Type	M	X
- Load Control	X	M
- Security Mode	X	M
- Security Link Table	X	M
- Security Key Table	X	M
- Security Individual Address Table	X	M
- SKI-Table Tool	X	M
- Security Report Control	X	M
- Sequence Number Sending	X	M
- Role Table	X	M
- Group Object Security Flags	X	M
- Group Object Diagnostics	X	M

2.2.3.4 Security Mode and Security Intermediate List

2.2.3.4.1 Definition

The Security Mode shall serve to support services and Datapoints that are neither on the Security Black List, nor on the Security White List, but of which the KNX Data Security requirements shall differ under control of the MaC.

EXAMPLE 16 The MaC may assign the IA in an ex-factory device using plain communication. It may then restart the device – to disable the Programming Mode in the device – and subsequently opens a TL-connection to the assigned IA to verify success. In this, the Security Mode is kept disabled. Therefore, the A_Restart-service shall be accepted.

During runtime however, a malicious client may put the device “out-of-service” by restarting it periodically. Hence, A_Restart shall during Runtime only be accepted by a properly authenticated MaC.

Security Mode shall have two states: disabled and enabled. The ex-factory state shall be “disabled”. The normal state after configuration by the MaC shall be “enabled”.

The services and DPs that are affected by the Security Mode are listed in the below Security Intermediate List. For each, the KNX Data Security requirements (accessibility, limitation to certain Roles, authentication and/or confidentiality) are listed.

- It may be possible that with disabled Security Mode the service or DP can simply be seen as part of the Security White List.
- It may be possible that with enabled Security Mode, the service or DP can simply be seen as part of the Security Black List.
- However, disabled Security Mode may not mean that the service or DP will be accepted with plain communication.

  EXAMPLE 17 It may be that with disabled Security Mode the service or DP can only be accessed using secure communication with the Role ETS and requiring authentication and confidentiality, and that with enabled Security Mode the service or DP will be ignored.

- Vice versa, enabled Security Mode may not mean that the service or DP will not be accepted even using secure communication and the proper Role and access level.

  EXAMPLE 18 It may be that with enabled Security Mode, the service or DP is only accepted using secure communication using the proper Role and access level.

Just like the Security White Lists and Security Black Lists, the support of the Security Mode can be realised entirely through a sufficiently powerful realisation of the Permissions and Roles.

The Security Mode shall not be a general switch to allow or deny the handling of AL-services using plain communication. It shall neither be seen as a common switch to enable or disable the evaluation of Roles and Permissions. In particular, the Security Mode shall not influence the evaluation of the Security White Lists and the Security Black Lists. The Security Mode shall only have effect on the handling of the services as listed in clause 2.2.3.4.2 “Security Intermediate List for services” and the DPs as listed in clause 2.2.3.4.3 "Security Intermediate List of Datapoints”. Services and DPs that are in the Security White Lists or in the Security Black Lists shall not be in the Security Intermediate List.

- The secure or plain access to services and DPs that are not in the Security Intermediate List, shall not be controlled by the Security Mode, but shall be given by the Security White Lists, the Security Black Lists and the further Roles and Permissions.
- It is however allowed to add additional implementation specific meaning to the Security Mode. This shall however not contradict with the Security White Lists, Security Black Lists or Security Intermediate Lists.

2.2.3.4.2 Security Intermediate List for services

Table 6 is the Security Intermediate List for services. This list is normative: it is not specified in the specification of each service whether it is on the Security Intermediate List or not.

NOTE 23 This list so far only contains services that are used for Network Management on (system) broadcast communication mode. It is not possible to apply KNX Data Security on these communication modes. These services shall be accepted if Security Mode is disabled and shall be ignored if Security Mode is enabled. The underlying use is that Security Mode is disabled ex-factory; the MaC then assigns the DoA and IA using plain communication, then enables the Security Mode and continues with the Device Configuration using secure communication. By enabling Security Mode, the MaC prevents a malicious modification of the DoA and IA.

Table 6 – Security Intermediate List for services

Service	Security Mode	Service shall be	Role [a]	Security
A_Restart (any Restart Type)	disabled:	accepted	n/a	n/a
	enabled:	accepted	ETS	A+C
See EXAMPLE 16.				
A_Authorize_Request A_Key_Write	disabled:	accepted	n/a	n/a
	enabled:	accepted	ETS	A+C
If Security Mode is disabled, then this service shall be accepted in plain communication. This shall allow that a MaC that does not support KNX Data Security can still configure the device and use the authorization. If Security Mode is enabled, then any MaC can attempt an authorization or set authorization keys. Yet, this shall be done using confidentiality, so that the keys are not transferred in plain text.				
[a] These Roles specify minimal requirements. Other Roles may be added implementation specific.				
[b] The A_Link_Write-service is not contained in this list. This document version does not contain E-Mode requirements.				

2.2.3.4.3 Security Intermediate List of Datapoints

Table 7 – Security Intermediate List for Datapoints

Service	Security Mode	DP shall be	Role	Security
Programming Mode	disabled:	accepted	n/a	n/a
	enabled:	accepted	ETS	A+C
This only concerns the Resource for controlling the Programming Mode (clause 4.19 in [05]). It does not concern the Domain Address or the Individual Address. In an ex-factory device, with Security mode disables, the MaC can assign the Domain Address and the Individual Address in plain communication. The MaC will then enable Security Mode and continue the further Configuration using secure communication. This prevents that a hacker can maliciously interfere with the address assignment of further devices by manipulating the Programming Mode of the already configured KNX Data Security devices.				
Domain Address (PID_DOMAIN_ADDRESS)	disabled:	accepted	n/a	n/a
	enabled:	accepted	ETS	A
Individual Address (PID_SUBNET_ADDR and PID_DEVICE_ADDRESS)	disabled:	accepted	n/a	n/a
	enabled:	accepted	ETS	A

2.2.3.5 Register security failures in the AIL

The AIL shall register failures in the evaluation of Roles and Permissions.

- Any failure shall increment the SFCC (see 2.3.2.17).
- If the attempt to access a service or DP is done using a link in the *Security Link Table*, than the error shall additionally be registered by incrementing the *AIL Failure Counter* in the SFCL (see 2.3.2.9) if implemented, in the array element corresponding with the link that is used.

2.3 Resource definition or used Resources

🗎 *This clause shall be integrated in [05].*

2.3.1 Security requirements of existing KNX Resources

🗎 *The above title is not intended for integration in the KNX Specifications. It only serves for structuring this document.*

2.3.1.1 Access rights to Resources

🗎 *This clause shall be added to clause 1.2.2 "Access rights" in Chapter 3/5/1 "Resources" ([05]).*

Next to the read-only or read-and-write access to a Resource, the access can be limited to secure communication, using at least authentication. For the KNX standard Resources, this is indicated further in this paper.

2.3.1.2 Access Control to the Group Address Table

🗎 *This clause, with title, shall be added to clause 4.9 "Group Address Table" (GrAT) in [05], as new clause 4.9.2. The existing clause 4.9.2 shall be shifted down.*

If the MaS supports KNX Secure Communication, then it shall support Access Control to the Group Address Table. The write access to the contents and any other parameter related to the Group Address Table shall be limited to the Role "ETS". Other Roles shall not have write access but may have read access.

EXAMPLE 19 Parameters related to the Group Address Table can be the write access to the Load State Machine (memory mapped or Property based), Group Address Table length, pointers to the memory mapped location etc.

NOTE 24 Read access is no problem, as the Group Addresses are at runtime anyhow visible unencrypted on the bus.

2.3.1.3 Access Control to the Group Object Association Table

🗎 *This clause, with title, shall be added to clause 4.10 "Group Object Association Table" (GrOAT) in [05], as new clause 4.10.2. The existing clause 4.10.2 shall be shifted down.*

If the MaS supports KNX Secure Communication, then it shall support Access Control to the Group Object Association Table. The write access to the contents and any other parameter related to the Group Object Association Table shall be limited to the Role "ETS". Other Roles shall not have write access but may have read access.

EXAMPLE 20 Parameters related to the Group Object Association Table can be the write access to the Load State Machine (memory mapped or Property based), Table length, pointers to the memory mapped location etc.

2.3.1.4 Access Control to the Group Object Table

🗎 *This clause, with title, shall be added to clause 4.11 “Group Object Table” in [05], as new clause 4.11.2. The existing clause 4.11.2 shall be shifted down.*

If the MaS supports KNX Secure Communication, then it shall support Access Control to the Group Object Table. The write access to the contents and any other parameter related to the Group Object Table shall be limited to the Role “ETS”. Other Roles shall not have write access but may have read access.

EXAMPLE 21 Parameters related to the Group Object Table can be the write access to the Load State Machine (memory mapped or Property based), Table length, pointers to the memory mapped location etc.

2.3.2 Security Interface Object

2.3.2.1 Introduction and overview

The Security Interface Object shall be used for managing the KNX Data Security. This concerns configuration information (e.g. encryption keys) and runtime information (e.g. window counter).

The Security Interface Object (Interface Object Type = 17) shall hold the Properties as listed in Table 8. (This table only gives an overview. For the mandatory and optional Properties, please refer to 2.9).

The security conditions for accessing these Properties are specified in the detailed Property descriptions in clause 2.3.2.2 and further.

Therefore, please refer to the detailed Property definitions below for the requirements concerning access.

Table 8 – Properties in the Security Interface Object

Property Name	Property Identifier	Property Datatype
Interface Object Type	1 = PID_OBJECT_TYPE	PDT_UNSIGNED_INT
Interface Object Name	2 = PID_OBJECT_NAME	PDT_UNSIGNED_CHAR[]
Load Control	5 = PID_LOAD_STATE_CONTROL	PDT_CONTROL
Security Mode	51 = PID_SECURITY_MODE	PDT_FUNCTION
Security Link Table	52 = PID_SKI_TABLE_LINK	PDT_GENERIC_14[]
Security Key Table	53 = PID_SKI_TABLE_KEY	PDT_GENERIC_16[]
Security Failure Counters on Links (SFCL)	55 = PID_SECURITY_FAILURE_COUNTERS_-ON_LINKS	PDT_GENERIC_04[]
Security Individual Address Table	54 = PID_SECURITY_INDIVIDUAL_-ADDRESS_TABLE	PDT_GENERIC_08[]
SKI-Table Tool	56 = PID_SKI_TOOL	PDT_GENERIC_08
Security Report	57 = PID_SECURITY_REPORT	PDT_BITSET8
Security Report Control	58 = PID_SECURITY_REPORT_CONTROL	PDT_BINARY_INFORMATION
Sequence Number Sending	59 = PID_SEQUENCE_NUMBER_SENDING	PDT_GENERIC_06
Role Table	60 = PID_ROLE_TABLE	PDT_GENERIC_03[]
Group Object Security Flags	61 = PID_GO_SECURITY_FLAGS	PDT_GENERIC_01[]
Security Failure Common Counter (SFCC)	63 = PID_SECURITY_FAILURE_COMMON_CTR	PDT_UNSIGNED_INT

An additional Resource is the Factory Default Setup Key (FDSK). This is not available as Property. See 2.3.2.13.

2.3.2.2 PID_OBJECT_TYPE (PID: 1)

- **Property name:** Interface Object Type
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_PropDataType (DPT_ID = 7.010)
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Data:	**Read:**	any [4]	plain	3
	Write:	none	n/a	n/a

List: This Property shall be on the Security White List.

Please refer to clause 4.2.1 in [05] for the general requirements of PID_OBJECT_TYPE.

This Property shall contain the Object Type of the Interface Object. The Security Interface Object shall have Object Type 17.

2.3.2.3 PID_OBJECT_NAME (PID: 2)

- **Property name:** Interface Object Name
- **Property Datatype:** PDT_UNSIGNED_CHAR[]
- **Datapoint Type:** None.
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Data:	**Read:**	any	plain	3
	Write:	none	n/a	n/a

List: It is not required that this Property be on the Security White List or on the Security Black List.

Please refer to clause 4.2.2 in [05] for the general requirements for PID_OBJECT_NAME.

This Property shall contain the name of the Security Interface Object.

2.3.2.4 PID_LOAD_STATE_CONTROL (PID: 5)

- **Property name:** Load Control
- **Property Datatype:** PDT_CONTROL
- **Datapoint Type:** None.
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

This Property *Load Control* shall control the load state machine of all configuration data of the Security Interface Object. The load state machine shall comply with the “Realisation Type 1” as specified in clause 4.16.2 in [05].

The *Load Control* shall influence the evaluation of the Security Resources as specified for each Resource.

[4] “Any” is not a specific Role, but means that there is no differentiation between the Roles for the given security and access conditions. This indication may exclude Roles if any explicit is given. See 2.3.2.14 “PID_SEQUENCE_NUMBER_SENDING (PID: 59)”.

NOTE 25 This considers the following Resources.

- Security Link Table
- Security Key Table

The Security Tool Key and the Security Mode however shall not depend on the Load State. Please consult the formal Resource specifications for the correct and full formal requirements. This note is only informative.

The Load Control shall only be accessible using the Security Key listed in the *Security Tool Key* (see 2.3.2.10).

NOTE 26 The Load Control is designed for S-Mode Configuration. Access to the Security Resources in PB-Mode will be specified separately.

2.3.2.5 PID_SECURITY_MODE (PID: 51)

- **Property name:** Security Mode
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** This Property is a Function Property. The coding of the data depends on whether data is written to the function or responded by the function. No single DPT can be given.
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Function:	**Command:**	ETS	A+C	2
	State:	ETS	A+C	2

List: This Property shall be on the Security Black List.

This Property shall be used to enable and disable the Security Mode.

This Property shall be a Function Property.

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

This Property shall only be accessible using Secure Communication, regardless of its value. This means that it shall only be possible to request the command enable and disable Security Mode using secure communication. Even if Security Mode is disabled, it shall only be possible to enable it by using secure communication: this Property shall be on the Security Black List.

The Security Mode does not depend on the Load St ate of the Security Interface Object.

a) Write (A_FunctionPropertyCommand-PDU)

octet 10
Security Mode

Codes:	00h :	Security Mode disabled
	01h :	Security Mode enabled

The response shall respond the Return Code of the operation.

Response (A_FunctionPropertyState_Response-PDU)

octet 10
Return Code

Codes:	00h :	SUCCESS
	01h :	The Security Tool Key is not yet assigned and the MaS does not have a FDSK. The Security Mode shall be disabled.
	02h:	The Security Tool Key is assigned but SKI_TABLE_LINK is Unloaded. The Security Mode shall be **enabled**.
	FFh:	ERR_SKI_TABLE_ERROR (Empty, error between SKI_Table_Link and SKI_Table_Key). The Security Mode shall be disabled.

The request to change the Security Mode shall be processed prior to sending the response.

- If the Security Mode changes from disabled to enabled, then the response shall be transmitted using the S-A_Data-service with confidentiality.
- If the Security Mode changes from enabled to disabled, then the response shall be transmitted using the plain A_FunctionPropertyState_Response.req.

b) Read (A_FunctionPropertyState_Read-PDU)

octet 10
00h

Response (A_FunctionPropertyState_Response-PDU)

octet 10
Security Mode

Security Mode and Master Reset

Erase Code		Effect on Security Mode
01h	**Confirmed Restart**	Security Mode shall not change.
02h	**Factory Reset**	Security Mode shall be disabled.
03h	**ResetIA**	Security Mode shall not change.
04h	**ResetAP**	Security Mode shall not change.
05h	**ResetParam**	Security Mode shall not change.
06h	**ResetLinks**	Security Mode shall not change.
07h	**Factory Reset without IA**	Security Mode shall be disabled.

2.3.2.6 PID_SKI_TABLE_LINK (PID: 52)

- **Property name:** Security Link Table
- **Property Datatype:** PDT_GENERIC_14[]
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.6.1 Abstract Resource definition

The elements of this Property Value shall contain the Security Configuration Link information, which shall define the Security information for each communication link.

2.3.2.6.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

It is recommended that the memory where the security key resource is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.6.3 Format

2.3.2.6.3.1 General

The Security Link Table shall be an array Property formatted as described in Figure 27.

Array index = Link Index	**Security Configuration Link Record**						
	Extended Frame Format	**TSAP Type**	**TSAP or ASAP**	**Source Address**	**Key Index**	**KNX Serial Number or Domain Address**	**Security Parameters for Links**
	(4 bits)	(4 bits)	(2 octets)	(2 octets)	(2 octets)	(6 octets)	(1 octet)
1							
…							
N							

Figure 27 – Security Link Table

The Security Link Table shall be sorted firstly according the Source Address (IA_Index) and then according the TSAP or ASAP.

Relation to other KNX Resources

- The *Security Link Table* shall not refer to a communication partner explicitly by its IA: instead, reference to an IA shall be done by the *IA_Index*, which shall be the entry in the *Security Individual Address Table* that holds the IA. This is specified in closer detail below and in Table 9.
- Likewise, the *Security Link Table* shall not contain GAs, but refer to them using the *GA-index* in the Group Address Table. This is specified in closer detail below and in Table 9.
- This *Security Link Table* shall have the same number of elements as the *Security Failure Counters on Links*; see 2.3.2.9). Elements in both tables at the same Property Value array element index shall refer to the same security key.
- For secure communication, in reception direction, the S-AL shall forward the array index of the Security Configuration Link on which the secure message is received, as parameter link_index of the AL-service.

Error - and exception handling

- If the *Security Link Table* has any other Load State than “Loaded”, then its evaluation shall return negative. This is, the result shall be as if the searched link is not present in the Security Link Table.

2.3.2.6.3.2 Dependencies between the TSAP, KNX Serial Number, TSAP Type, Extended Frame Format and Source address

These fields shall be interpreted in function of the TSAP Type as specified in Table 9.

NOTE 27 The IA_Index of a communication partner can appear multiple times in the Security Link Table.

- As Source Address for incoming group communication.
- As Source Address for incoming point-to-point communication.
- As Destination Address for outgoing point-to-point communication.

These can and typically will relate to different security keys.

Table 9 – Interpretation of the Security Configuration Link Record in function of the contained TSAP Type

TSAP Type	Definition and interpretation of the fields		
0000b	**Receiving point-to-point** This TSAP-type shall be used to classify the Security Configuration Link Records used for incoming point-to-point communication (connectionless and connection-oriented).		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	IA_Index of the IA of the sender	TSAP:0000h	• TP1, KNX IP: 000000000000h • RF, PL110: DoA of Sender
0001b	**Receiving Group Address** This TSAP-type shall be used to classify the Security Configuration Link Records used for incoming T_Data_Group-services.		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	IA_Index of the IA of the sender	TSAP (GA-index)	• TP1, PL110, KNX IP: 000000000000h • RF: KNX Serial Number of sender (Extended Group Address)

TSAP Type	Definition and interpretation of the fields		
0010b	**Sending Group Address** This TSAP-type shall be used to classify the Security Configuration Link Records used for outgoing T_Data_Group-services. There shall be one record for each sending ASAP (Group Object).		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	0000h	ASAP (GO number)	000000000000h
0011b	**Sending point-to-point** This TSAP-type shall be used to define the Security Configuration Link Record for the type “sending point-to-point” (connectionless and connection-oriented).		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	0000h	IA_Index of the IA of the sender	000000000000h
0100b	**Receiving LTE Group Address** This TSAP-type shall be used to classify the Security Configuration Link Records used for incoming T_Data_Tag_Group-services.		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	• 0000h: wildcard: all senders in this zone • IA of the sender: Link Record only valid for this IA.	zone address (with wildcards possibility)	• TP1, PL110, KNX IP: 000000000000h • RF: KNX Serial Number of sender (Extended Group Address)
0101b	**Sending LTE Group Address** This TSAP-type shall be used to classify the Security Configuration Link Records used for outgoing T_Data_Tag_Group-services.		
	SA	**TSAP or ASAP**	**KNX Serial Number or DoA**
	0000h	zone address	000000000000h

2.3.2.6.3.3 Key Index

This field shall refer to the Security Key that shall be used for confidentiality or authentication for this communication with the TSAP identified in this Configuration Link Record. The Security Key shall be referred by its index Security Key Table (PID_SKI_TABLE_KEY).

2.3.2.6.3.4 Security Parameters for Links

The field *Security Parameters* for Links shall contain additional security configuration information for the secure link.

Format

This shall be a single octet field encoded as specified in Figure 28.

bit number	7	6	5	4	3	2	1	0
field name	r	r	r	r	r	r	algorithm	
value	0	0	0	0	0	0		

Figure 28 – Security Parameters for Links

The value “algorithm” shall indicate the security algorithm and possibly its operation mode that shall be used for the reception and transmission of messages on this secure link. This value shall also be indicated in the field “Algorithm” in the Secure APDU, as specified in 2.2.1.2.2.1.

These *Parameters* are foreseen for future extensions. The value shall always be 00b, as specified for “Algorithm” in Table 2.

Bits 7 to 2 are reserved and shall always be 0.

2.3.2.6.4 Security Link Table and Master Reset

Erase Code		Effect on the Security Link Table
01h	**Confirmed Restart**	Security Link Table shall not change.
02h	**Factory Reset**	The Security Link Table shall be cleared [a]. Its length shall become 0.
03h	**ResetIA**	The Security Link Table shall not change [b].
04h	**ResetAP**	The Security Link Table shall not change.
05h	**ResetParam**	If this involves parameters that relate to Security Links then these links shall be cleared. EXAMPLE 22 A parameter that controls the activation of a Secure Group Object.
06h	**ResetLinks**	The Security Link Table shall be cleared [a]. Its length shall become 0.
07h	**Factory Reset without IA**	The Security Link Table shall be cleared [a]. Its length shall become 0.

[a] Because this Erase Code will also clear the project specific Security Key Table, the references in this Security Link Table will become void and therefore the Security Link Table shall be cleared as well.

[b] As the secure communication partners of this device evaluate its original IA in their security check, after resetting the IA of this device its communication partners will reject its secure messages.

2.3.2.6.5 Usage by the MaC

The MaC shall write the entries of the Security Link Table, from the lower indexes to the higher, sorted firstly according the field Source Address (IA_Index) and then according the field “TSAP or ASAP”.

2.3.2.7 PID_SKI_TABLE_KEY (PID: 53)

- **Property name:** Security Key Table
- **Property Datatype:** PDT_GENERIC_16[]
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.7.1 Abstract Resource definition

The security Key Table shall contains all security keys, both for confidentiality as well as for authentication.

2.3.2.7.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

The configuration and evaluation of the Security Key Table shall depend on the load state of the Security Interface Object (PID_LOAD_STATE_CONTROL as specified in 2.3.2.5).

It is recommended that the memory where the security key resource is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.7.3 Format

The Security Key Table shall be an array Property as described in Figure 29.

There are no requirements concerning the sorting of this table and concerning the positioning of the keys solely used for authentication and the keys used for authentication and confidentiality.

If a key is shorter than 16 octets then key data shall be aligned to left; unused octets shall be filled with 00h.

NOTE 28 The currently used AES-128 block cipher uses the full 16 octet size.

Array index	Security Key
	16 octets
1	
2	
…	
N	

Figure 29 – Security Key Table

2.3.2.8 PID_SECURITY_INDIVIDUAL_ADDRESS_TABLE (PID: 54)

- **Property name:** Security Individual Address Table
- **Property Datatype:** PDT_GENERIC_08[]
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	Read:	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.8.1 Abstract Resource definition

The *Security Individual Address Table* shall store the IAs of the communication partners to which the MaS has secure links. The table shall additionally for each IA store the *Last Valid SeqNr* that is accepted from any message from that IA.

2.3.2.8.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

It is recommended that the memory where the *Security Individual Address Table* is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.8.3 Format

The *Security Individual Address Table* shall be an array Property as shown in Figure 30. Each element shall be composed of the two octets IA and the six octets *Last Valid* SeqNr that is accepted from this IA.

The MaC shall sort the Individual Address Table from the lower indexes to the higher, according the numerical value of the Individual Address.

Array index = IA_Index	**Individual Address**	**Last Valid SeqNr**
	(2 octets)	**(6 octets)**
1	**IA 1**	Last Valid SeqNr from IA 1
2	**IA 2**	Last Valid SeqNr from IA 2
…	**…**	…
M	**IA N**	Last Valid SeqNr from IA M

Figure 30 – Security Individual Address Table

The *Security Individual Address Table* shall be related to the Security Link Table by the usage of the IA_Index in the Security Link Table.

NOTE 29 In a typical situation, a MaS will have multiple Links from a remote partner, typically multiple GAs, so, the number of entries in the Security Link Table will typically be a factor larger than the number of entries in the Security Link Table.

2.3.2.8.4 Usage by the MaS

If the MaS needs an IA for reception or transmission, then it shall check the IA in the *Security Individual Address Table* and if found, use the IA_Index to find the further Security Parameters in the *Security Link Table*.

If the IA is not found in the *Security Individual Address Table*, then the MaS shall assume that it does not have a KNX Data Security link with the KNX device on that IA. If Roles are used, this mean the Role shall be “Unlisted”.

Last Valid SeqNr

If the MaS receives an S-A_Data-PDU from a given SA, it shall compare the contained SeqNr with the value stored in this table (see 2.2.1.3).

If the S-AL can successfully decrypt an S-A_Data-PDU and forward the contained AL-service to the P-AL, then it shall update the field *Last Valid SeqNr* for that IA in the *Security Individual Address Table* with the received value, regardless of the TSAP-type.

All values of the Last Valid SeqNr in all entries of the *Security Individual Address Table* shall be saved in full at power-down and be restored in full at power-up.

2.3.2.8.5 Usage by the MaC (ETS)

The MaC shall during Configuration fill the Security Individual Address Table with the IAs as assigned for group- and point-to-point communication to the MaS.

The MaC shall sort the entries according the numerical values of the Individual Addresses from low to high. Consequently, if the MaC removes or changes an IA, the entire Security Link Table and Security Individual Address Table shall be rewritten.

- If the MaS knows the Last Valid SeqNr of the communication partner, then it shall write this value in the field Last Valid SeqNr.

  EXAMPLE 23 If the MaS has firstly read out the Last Valid SeqNr from the communication partner, prior to configuring this MaS.

- If the MaS does not know the Last Valid SeqNr of the communication partner, then it shall write the value 000000000000h.

2.3.2.9 PID_SECURITY_FAILURE_COUNTERS_ON_LINKS (PID: 55)

- **Property name:** Security Failure Counters on Links (SFCL)
- **Property Datatype:** PDT_GENERIC_04[]
- **Datapoint Type:** none
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	any	none	3
	Write:	ETS	A+C	2

List: This Property shall not be on the Security White List, Security Black List or Intermittent List.

2.3.2.9.1 Abstract Resource definition

This Property shall be a table that shall for each Link contain the total number of errors that has occurred in the S-AL and in the AIL for that Link. Both S-AL as well as AIL register security failures in this counter.

The SFCL shall allow for a more detailed security failure diagnostics than the Security Failure Common Counter, as specified in 2.3.2.17, up to the level of the separate Links.

2.3.2.9.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

This Property Value may be stored in unencrypted memory.

2.3.2.9.3 Format

The *SFCL* shall be an array Property and shall have the same number of elements as the *Security Link Table* (PID_TABLE_LINK; see 2.3.2.6). Elements in both tables at the same Property Value array element index shall refer to the same Link.

NOTE 30 As the SFCL registered security errors per Link, it is not possible to register in this security errors caused by devices that send to this MaS and that are not in the Security Link Table. Security errors caused by such “Unlisted” senders cannot be registered in this Resource. These shall however be counted in the SFCC (see 2.3.2.17).

Each element of the SFCL shall be formatted as specified in Figure 31.

S-AL Failure Counter	AIL Failure Counter
(2 octets)	**(2 octets)**

Figure 31 – Security Failure Counters on Links

The *SFCL* may be stored in volatile memory, but shall be saved at power down and shall be restored at power up.

The ex-factory value for both fields shall be 0000h.

Counter overflow

If any field of the SFCL reaches its maximal value of 65 535, it shall not wrap around.

There is no dedicated handling required concerning the counter overflow: the last increment of the SFCL-field may cause a transmission of a *Security Report* (see 2.3.2.11). There are no further requirements to the MaS. Other additional, implementation specific behaviour is allowed.

2.3.2.9.4 Usage by the MaS (device)

The MaS (device) shall increment the SFCL-entry of a Link if a secure message is received with a security failure other than the SeqNr checking.

The field *S-AL Failure Counter* shall be incremented explicitly on the following security failures in the **S-AL**.

- any authentication failure, and
- any confidentiality failure

for the entry corresponding with the Link on which the error occurs.

The field *AIL Failure Counter* shall be incremented explicitly on the following security failures in the **AIL**.

- any failure in Roles or Permissions

for the entry corresponding with the Link on which the error occurs.

As the error is linked to an entry in the Security Link Table, the SFCL can thus only register Authorization (Roles, Permissions) of authenticated communication partner. These are thus partners attempting access to services or Datapoints that they do not have the right to. This may rather point to a configuration mistake than to a hack. Authorization problems with senders that are not present in the Security Link Table, “Unlisted” senders, are only commonly registered in the Security Failure Common Counter (SFCC): see 2.3.2.17.

Master Reset

The MaS shall reset the SFCL to it ex-factory value with the Erase Codes “Factory Reset”, and “Factory Reset without IA”; the MaS shall not modify the value of the SFCC for any other Erase Code.

2.3.2.9.5 Usage by the MaC

The MaC may read out the SFCL for security diagnostic purposes on the level of the individual Link. The reading does not require KNX Data Security and can be done by any Role.

The MaC may reset the SFCL to 0. This shall only be possible with the Role ETS, using Authentication and Confidentiality and access level 2. The MaC may also reset the SFCL as part of a Master Reset.

2.3.2.10 PID_SKI_TOOL (PID: 56)

- **Property name:** Security Tool Key
- **Property Datatype:** PDT_GENERIC_16
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.10.1 Abstract Resource definition

The Security Tool Key shall be used to store the security information for the central MaC in KNX S-Mode (ETS®) and KNX Ctrl-Mode.

NOTE 31 The Security Tool Key is specified as a stand-alone Resource, outside the Security Link Table, to allow a MaC to clear all Security Configuration Link data without clearing its own Tool Key and hence possibly locking itself out.

2.3.2.10.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

At runtime, the security key resources shall not be readable; this Property shall be on the Security Black List for Datapoints.

It is recommended that the memory where the security key resource is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.10.3 Format

The Security Tool Key shall be a single element Property formatted as described in Figure 32.

The Security Tool Key shall be a stand-alone Resource that shall not be part of the Security Link Table. This shall allow the MaC to easily clear and unload this Resource.

Security Tool Key
(16 octets)

Figure 32 – Security Tool Key

The Security Tool Key is stored outside the Security Link Table. This shall mean the following.

- It is not linked to any Individual Address or TSAP. This means that communication using the Security Tool Key shall be possible from any Source Address and in any communication mode.
- There is no Resource defined for accessing the SeqNr for the Tool Key. This can only be read using the S-A_Sync-service. The MaS shall store the last valid SeqNr from the MaC internally.
- There is no differentiation between incoming – and outgoing messages. The MaC and the MaS shall use the same Tool Key.

2.3.2.10.4 Usage by the MaS (device)

Configuration

The Security Tool Key shall become active immediately after it is written by the MaC, this is, as soon as the Property has a length of 1. If this Property has the length 0, then the FDSK shall be active. The length of this Property is thus the criterion for the MaC to know whether it shall use the Tool Key or the FDSK.

The A_PropertyValue_Response-PDU to confirm the A_PropertyValue_Write-PDU shall immediately be protected using the newly written Security Tool Key.

The Security Tool Key shall not be influenced by the Load State of the Security Interface Object! Even if the Security Link Table is unloaded, the Security Tool Key shall remain valid.

The Security Tool Key shall only be used with “authentication and confidentiality”; it shall not be used with “authentication only”.

Runtime - transmission of a Secure APDU

If a secure APDU is transmitted and encrypted according the Security Tool Key then the field *Tool Access* in the SCF shall be set; otherwise, it shall be cleared.

Runtime - reception of a Secure APDU

If a secure APDU is received for which the field *Tool Access* is set, then it shall be decoded according the parameters contained in the Security Tool Key; otherwise, it shall be decoded using the Security Link Table.

2.3.2.10.5 Usage by the MaC

The MaC shall use the Security Tool Key to authorize itself for accessing any Resource that it intends to access as the Role “ETS” (the Security Link Table, the Security Key Table, the Security Failure Counters on Links, the Security Tool Key, etc. This list is not complete.)

EXAMPLE 24 The MaC may through this reset the Security Failure Counter.

If the MaS is in ex-factory state, the MaC shall set the Security Tool Key after it assigns the IA of the MaS and prior to managing any further Resource in the MaS. The MaC shall to this secure the communication using the Factory Default Setup Key.

If the MaS is not in ex-factory state, then the MaC shall use the already configured Security Tool Key for any next access to this Security Tool Key.

The MaS shall properly handle the change or the activation of the Security Tool Key: if the Security Tool Key is assigned (after the FDSK is used) or modified (after another Security Tool Key value is used), then the MaS will confirm that modification with the preceding Security Tool Key, but from then on only accept the newly Configured Tool Key. The MaC shall protect the next message to the MaS with the newly assigned Security Tool Key. The assignment – or change of the Security Tool Key shall not lead to loss of communication between MaC and MaS or to error messages.

The FDSK and the Tool Key shall have the same function, but shall not be active at the same time.

2.3.2.11 PID_SECURITY_REPORT (PID: 57)

- **Property name:** Security Report
- **Property Datatype:** PDT_BITSET8
- **Datapoint Type:** DPT_Security_Report (see 2.7.1)
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Data:	**Read:**	ETS:	none	3
		Unlisted	none	3
	Write:	ETS:	A+C	2
NwPar:	IR	n/a	n/a	n/a

List: This Property shall be on the Security Black List.

2.3.2.11.1 Abstract Resource Definition

This Property shall be used to announce KNX Data Security related status - and diagnostic information.

This shall only serve for a basic security reporting. This specification does not foresee extended reporting (adding time of events, building logs…).

2.3.2.11.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

This Property Value may be stored in unencrypted memory.

2.3.2.11.3 Format

The Security Report shall be encoded according DPT_Security_Report as specified in 2.7.1.

2.3.2.11.4 Usage by the MaS

If any security failure is detected then MaS shall set the field *Security Failure*. If the Parameter Property Security Report Control is set to “Enabled” then additionally this Property *Security Report* shall be communicated using the Management Procedure DMP_InterfaceObjectInfoReport_RCl (see [19]. This shall be done regardless of whether the field *Security Failure* is set prior to this security failure or not. This is, the MaS shall report any security failure, even if a security failure is reported before or not.

```
        /* Report a KNX Data Security failure. */
DMP_InterfaceObjectInfoReport_RCl (mpp_ASAP = void, mpp_comm_mode = broadcast,
        mpp_hop_count_type = 6, mpp_object_type = Security Interface Object,
        mpp_PID = PID_SECURITY_REPORT, mpp_priority = urgent, mpp_test_info = 00h,
        mpp_test_result = SecurityReport)
```

The MaS shall not automatically reset the Security Failure. This shall only be possible by the MaC, in a secure communication.

In order to support also receivers that do not support the DMP_InterfaceObjectInfoReport_-RCl or do not support Property Reading, the MaS may also communicate the security report via a Group Object. The DPT_Security_Report shall be used to this purpose.

2.3.2.11.5 Usage by the MaC

The MaC may receive this Property value communicated with A_NetworkParameter_-InfoReport.

The MaC shall read this Property to find out if there is any security failure inside the MaS. If any security failure is indicated, then the MaC may find more detailed information in the Security Failure Common Counter (see 2.3.2.17) and in the Security Failure Counters on Links (see 2.3.2.9).

The MaC may write this Property to reset it, after it has properly authenticated itself using secure communication.

2.3.2.12 PID_SECURITY_REPORT_CONTROL (PID: 58)

- **Property name:** Security Report Control
- **Property Datatype:** PDT_BINARY_INFORMATION
- **Datapoint Type:** DPT_Enable (DPT_ID: 1.003)
- **Access:**

Service:		Role:	Security:	Access level:
Data:	Read:	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.12.1 Abstract Resource Definition

The *Security Report Control* shall be a Property Parameter that shall control the spontaneous communication of the *Security Report* through the procedure DMP_InterfaceObject-InfoReport_RCl. This parameter shall not influence the access to the *Security Report* through Data Property Services (A_PropertyValue_Read and A_PropertyValue_Write), neither the possible group communication of the *Security Report*.

2.3.2.12.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

This Property shall be on the Security Black List for Datapoints.

2.3.2.12.3 Format

This shall be a single bit, Boolean Property, encoded according DPT_Enable
(DPT_ID: 1.003).

The default value shall be “Disabled”.

2.3.2.12.4 Usage by the MaS

If the *Security Report Control* has the value “Enabled” then the MaS shall communicate the *Security Report* with DMP_InterfaceObjectInfoReport_RCl as specified in 2.3.2.11.

If the *Security Report* Control has the value “Disabled” then the MaS shall not communicate the *Security Report* with DMP_InterfaceObjectInfoReport_RCl.

The *Security Report Control* shall only influence the communication to the *Security Report* through the DMP_InterfaceObjectInfoReport_RCl; it shall not influence the access to the *Security Control* through data Property services.

2.3.2.12.5 Usage by the MaC

The MaC shall clear or set this Property to disable or respectively enable the security failure reporting by the MaS.

The MaC shall use the Data Property services (A_PropertyValue_Read, A_PropertyValue_Write) to this purpose. The

2.3.2.13 Factory Default Setup Key (FDSK)

2.3.2.13.1 Abstract Resource definition

The Factory Default Setup Key (FDSK) shall be a default tool key for the authentication and confidentiality to be used by the MaC when the KNX secure device is firstly configured fresh from factory.

The FDSK may be printed on the device’s wrapping or on a paper note that is shipped with the device. Other indications that guarantee that this FDSK cannot be obtained by any unauthorized installer or MaC are possible as well.

2.3.2.13.2 Security

It is recommended that the memory where the FDSK is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.13.3 Location and format

The FDSK shall not be accessible over the KNX medium or through any local interface to the device. There is no KNX access method (Group Object, memory mapped parameter, Property Value or other) standardised to access the FDSK.

The FDSK shall not be included in the Security Link Table specified in 2.3.2.6.

The FDSK shall have the format in function of the Security Algorithm that is used for confidentiality. As there is no storage standardised, there are no further requirements on how to store the FDSK.

2.3.2.13.4 Usage by the MaS

The MaS shall allow secure communication using the FDSK if no Security Tool Key is programmed.

The MaS shall make the FDSK inactive (can no longer be used) as soon as the Security Tool Key for confidentiality is configured by the MaC. The FDSK shall however remain in memory of the MaS.

The FDSK and the Tool Key shall have the same function, but shall not be active at the same time.

The FDSK shall only become active again after the MaS performs a Master Reset with Erase Code “Factory Reset” (see [17]).

2.3.2.13.5 Usage by the MaC

The MaC **shall** activate the Security Mode (if needed for the links that are made) and prior to any other Device Management [5] operation of the MaS, using the FDSK in KNX Secure Communication, assign the Security Tool Key.

After the MaC has assigned the Security Tool Key, the MaC shall no longer use the FDSK, until possibly after it has executed a Master Reset of the MaS.

The MaC shall store the FDSK in a secure way in its database, so that when perform any procedure to the MaS that clears the Tool Key (e.g. via Master Reset), it will again get access to the MaS via the Tool Key.

2.3.2.14 PID_SEQUENCE_NUMBER_SENDING (PID: 59)

- **Property name:** Sequence Number Sending
- **Property Datatype:** PDT_GENERIC_06
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
		any [6]	A+C	3
	Write:	ETS	A+C	2
Sync:		ETS	A+C	n/a
		any	A+C	n/a

List: This Property shall be on the Security Black List.

2.3.2.14.1 Abstract Resource definition

This Property shall store the Sequence Number that shall be used by the MaS for its outgoing secure communication.

[5] This explicitly considers Device Management only. Network Management (elf. assignment of Domain Addresses and Individual Addresses) is done unsecured. See 2.6.

[6] This indication means that the Role “ETS” shall authorize with access level 2. “Any” other Role than ETS shall authorize with access level 3.

For the common specification of the Sequence Number, please refer to 2.2.1.3.

2.3.2.14.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

2.3.2.14.3 Usage by the MaS (device)

The MaS shall handle the *Sequence Number Sending* as specified for the sender in 2.2.1.3. If the S-AL forwards an S-A_Data-PDU to the TL, then it shall increment the *Sequence Number Sending* by 1.

This shall be done regardless of the further result of the TL-service, this is, regardless of the t_status of the TL-service confirmation primitive.

NOTE 32 This is special the case on TP1: the Sequence Number Sending shall be incremented even if the transmission of the TP1 frame is on the medium not confirmed or negatively confirmed.

The ex-factory value of the *Sequence Number Sending* for the MaS shall not be zero, but shall be a small, random value in the range from 0 to 255.

If the MaC writes the SeqNr with a new value, then the MaS shall use this new value for the immediate next secure message that it sends. The MaS shall does this use the new value of the SeqNr already to protect the A_PropertyValue_Response-PDU that confirms the A_PropertyValue_Write-PDU from the MaC that changes the SeqNr.

Power down behaviour

The *Sequence Number Sending* shall be saved in full at power-down and be restored at power-up.

2.3.2.14.4 Usage by the MaC

Procedure:

1. The MaC gets its own SeqNr to use and the SeqNr it has to expect from the MaS through the synchronisation mechanism.
2. Then, it can set, using secure communication, the new value of the Sequence Number Sending in the MaS.

NOTE 33 Any new SeqNr written by the MaC applies immediately. Therefore, the MaC shall support that the confirmation of this setting is protected by a different SeqNr.

2.3.2.15 PID_ROLE_TABLE (PID: 60)

- **Property name:** Role Table
- **Property Datatype:** PDT_GENERIC_03[]
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

2.3.2.15.1 Abstract Resource definition

For the common requirements concerning the Role Table, please refer to 2.2.3.

2.3.2.15.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

It is recommended that the memory where the Role Table is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.15.3 Format

The Role Table shall be an array Property of which each element shall contain the relation between one Link, through its Link Index, and one Role.

Link Index (2 octets)	**Role** (1 octet)
…	…

Figure 33 – Role Table

The Role Table is not sorted.

Every Link shall be at maximum once in the Role Table. Two Links may have the same Role.

2.3.2.15.4 Usage by the MaS (device)

The MaS shall through the Link Index search the Role that is assigned to a Link. The MaC shall use this Role to find the Permissions assigned to this Role in the Permissions Tables of the Datapoints and services. If a Link is not found in the Role Table then the MaS shall assign it the Role “Unlisted”.

The MaS shall only allow access to the Role Table from a Client that authenticates properly using the FDSK or the Tool Key.

2.3.2.15.5 Usage by the MaC (ETS)

The MaC (ETS) shall fill the Role Table according the Role assignment to the links given by the end user.

NOTE 34 There is no further standard Resource in the device that contains the Roles. It is thus important that the Roles that are used by the application in the Permissions, with the exact same values are known to ETS, so that ETS can fill in these values in the Roles Table: see ETS requirement 1.

The MaC shall secure the write access to the Role Table using at least authentication with the FDSK or Tool Key. Possibly, this access can be protected using confidentiality.

The MaC (ETS) shall foresee at least one Link in the Role Table that is assigned the “ETS – role”.

2.3.2.16 PID_GO_SECURITY_FLAGS (PID: 61)

- **Property name:** Group Object Security Flags
- **Property Datatype:** PDT_GENERIC_01[]
- **Datapoint Type:** None
- **Access:**

Service:		Role:	Security:	Access level:
Data:	**Read:**	ETS	A+C	2
	Write:	ETS	A+C	2

List: This Property shall be on the Security Black List.

🗎 *These flags are stored in a separate Resource. These are not specified as an extension of the Group Object Config flags, which are Profile dependent and would thus also require the update of multiple Resources. (This note is not intended for integration in the KNX Specifications.)*

2.3.2.16.1 Abstract Resource definition

The *Group Object Security Flags* shall for each Group Object define the minimal required security requirements. If a Group Object is accessed in a way that does not respect these *Security Config Flags*, then the Group Object server shall not give access (read or write) to this Group Object.

2.3.2.16.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

It is recommended that the memory where the *Group Object Security Flags* is stored be encrypted as well, so that it cannot be interpreted by any other interface to this memory, bypassing the KNX S-AL, such as a JTAG-interface.

2.3.2.16.3 Format

The Group Object Security Flags shall be an array Property of which each element shall contain the Security Config Flags for one Group Object in the Group Object Table with the same index.

The Security Config Flags of one entry, for one Group Object, shall be formatted as defined in Figure 34.

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
0	0	0	0	read		write	
				conf	auth	conf	auth

Figure 34 – Group Object Security Flags

Each entry shall be an 8 bit value. Only the bits b_3 to b_0 are defined. The bits b_7 to b_4 are reserved for future extensions and the MaC shall always write these bit values with 0. The MaS shall ignore these reserved bits.

2.3.2.16.4 Usage by the MaS (device)

Bits b_3b_2 shall specify the KNX Security features for reading the value of the Group Object [7]. If the Group Object Value is read with security flags of the AL-service matching these security requirements, then the Group Object Server shall accept and answer this service; if not, it shall ignore this service. Each cell shall be encoded as follows.

b_3	b_2	
authentication	**confidentiality**	**Description, remarks**
0	0	This GO can be read without confidentiality or authentication.
0	1	This GO can be read without authentication but shall only be readable with confidentiality.
1	0	Read requests to this GO shall only be accepted if the request uses at least authentication.
1	1	Read requests to this GO shall only be accepted if the request uses authentication and confidentiality.

Bits b_1b_0 shall specify the KNX Security features for writing the value of the Group Object [8]. If the Group Object Value is written with security flags of the AL-service matching these security requirements, then the Group Object Server shall accept this service and update the Group Object value; if not, it shall ignore this service. Each cell shall be encoded as follows.

b_1	b_0	
authentication	**confidentiality**	**Description, remarks**
0	0	This GO can be written without confidentiality or authentication.
0	1	This GO can be written without authentication but shall only be written with confidentiality.
1	0	Write requests to this GO shall only be accepted if the request uses at least authentication.
1	1	Write requests to this GO shall only be accepted if the request uses authentication and confidentiality.

2.3.2.16.5 Usage by the MaC (ETS)

The MaC shall set the Group Object Security Flags in each GO that is associated with a GA for which KNX Data Security is used.

[7] Mainly considered here is the A_GroupValue_Read-service. The permissions shall however also apply for alternative accesses to the Group Object Value, like the Group Object Indirection.

[8] Mainly considered here is the A_GroupValue_Write-service. The permissions shall however also apply for alternative accesses to the Group Object Value, like the Group Object Indirection.

2.3.2.17 PID_SECURITY_FAILURE_COMMON_CTR (PID: 63)

- **Property name:** Security Failure Common Counter (SFCC)
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_Value_2_Ucount
- **Access:**

Service:		**Role:**	**Security:**	**Access level:**
Data:	**Read:**	any	none	3
	Write:	ETS	A+C	2

List: This Property shall neither be on the Security Black List, on the Security White List or on the Security Intermediate List.

2.3.2.17.1 Abstract Resource definition

The *Security Failure Common Counter* shall increment with each KNX Data Security Failure that occurs in any KNX Data Security part in the device. It shall be sum of the following.

- From the S-AL
  - any authentication failure, and
  - any confidentiality failure

  for any link
- and from the AIL
  - any failure in Roles or Permissions

  for any Role

The ex-factory value of the *SFCC* shall be 0000h.

This counter shall allow for basic security diagnostic also in simple devices. Extended security diagnostics in more enhanced devices can be realised through the Property *Security Failure Counters on Links* as specified in 2.3.2.9.

2.3.2.17.2 Security

Please refer to the above access requirements. These requirements are exclusive: other Roles, security features or services shall not have access to this Property.

2.3.2.17.3 Format

The *SFCC* shall be a 16 bit unsigned integer as show in Figure 35

Security Failure Common Counter
(2 octets)

Figure 35 – Format of the SFCC

2.3.2.17.4 Usage by the MaS (device)

The MaS shall increment the SFCC with each occurrence of any security failure for any Link, service or DP.

If the SFCC value reaches its maximum of 65 535, then it shall not reset to 0 and it shall stay at this maximum.

The MaS shall reset the SFCC to it ex-factory value with the Erase Codes “Factory Reset”, and “Factory Reset without IA”; the MaS shall not modify the value of the SFCC for any other Erase Code.

2.3.2.17.5 Usage by the MaC (ETS)

The MaC may read out the SFCC for basic security diagnostic purposes. The reading does not require KNX Data Security and can be done by any Role.

The MaC may reset the SFCC to 0. This shall only be possible with the Role ETS, using Authentication and Confidentiality and access level 2. The MaC may also reset the SFCC as part of a Master Reset.

2.4 Management Procedures

🗎 *This document does not specify neither modify any Management Procedures.*

2.5 Installer Procedures

🗎 *This clause is put firstly, before the clause on the Configuration Procedures, because it introduces the concepts that are further to be supported by the Configuration Procedures.*

2.5.1 Installation recommendations

🗎 *It is not yet clear where this clause shall be integrated in the KNX Specifications. This can for instance be done in [09]. This has to be discussed.*

The access to the installation should be prevented as much as possible. Parts of the installation that are located outside the building should be connected to the rest of the installation only over a Coupler.

2.5.2 Configuration environment

2.5.2.1 Recommendations to the installation procedure

These are recommendations for the installer to take, in order to increase the security of the configuration

1. **The configuration can be done in a “safe environment”.**

   The configuration can be done in a location other than where the devices that are configured will finally be mounted. This can for instance be done in the premises of the electrical installer instead of at the construction site.

2. **The configuration can be done in a geographically very limited space**

   The devices to be configured and the MaC (ETS, E-Mode Controller…) can be moved temporarily to a very small setup with short communication paths between the participants, e.g. limited to a desk or table. This way, the probability that the configuration procedure is recorded by any third party is reduced.

   For KNX TP1 this means that very short cabling is used.

3. **The time-span for the exchange of the keys shall be limited to the minimum**

   Installer procedures shall last as short as possible. Any Programming Mode or Learning Mode shall only be active as long as necessary.

2.6 Configuration Procedures

2.6.1 Introduction

🗎 *This clause shall not be integrated in the KNX Specifications.*

This clause does not specify full Configuration Procedures for any KNX Profile, but it does specify how KNX Data Security shall be integrated in the existing Configuration Procedures.

The Configuration Procedures for E-Mode will be specified in the frame of PB FEC (Flexible E-Mode Channels). It is the intention that the KNX Data Security for PB-Mode will be natively but optionally supported in PB FEC.

2.6.2 Common requirements for the Configuration Procedure

🗎 *This clause shall be integrated in the KNX Specifications in [07], in a dedicated clause on KNX Data Security.*

2.6.2.1 Scope

These are requirements for the MaC (ETS, E-Mode Controller…).

2.6.2.2 Key exchange

The key exchange mechanism will only happen once during the configuration of each device. After configuration, during runtime, the symmetric encryption algorithm shall be used.

Any exchange of Security Keys shall happen through KNX Secure Communication using a Factory Default Setup Key.

2.6.2.3 Recommendations

1.

Minimum one key is required for each sender. A unique key may be used for all Group Addresses of one sender.

2.6.3 S-Mode

2.6.3.1 Basic rules

Rule 1 Once secure configuration is configured in a MaS that MaS shall further on always be configured using secure communication.

Some Resources do not allow differentiating between Secure and Plain data.

EXAMPLE 25 It is not possible to have Group Addresses in the Group Address Table that can only be configured using secure communication, and other GAs that can be configured without secure communication.

Therefore, in order to prevent unsecure access (hacker) from modifying the secure configuration, the entire configuration shall further be secure, once any security setting is used in the MaS.

Rule 2 The initial IA assignment is done unsecure.

Any next IA Management shall be done in point-to-point communication accessing PID_INDIVIDUAL_ADDRESS in the MaS using secure communication.

This way, it shall be prevented that a hacker modifies the IA of the MaS and in that way excludes a rightful MaC to manage the MaS.

Rule 3 The MaC shall use secure communication encrypted for authentication and confidentiality with the FDSK to set the Security Tool Key.

The MaC shall then activate the Security Mode in the MaS using secure communication accessing PID_SECURITY_MODE.

2.6.3.2 Support of the 16 octet FDSK

The initial key exchange shall be protected using the FDSK. This is the only protection mechanism for key exchange for S-Mode devices.

The FDSK may be printed on the device in some way. Other means to provide the FDSK to the ETS user are possible as well.

Each shipped KNX Secure device shall also have a different key. Having the same FDSK for all devices will make the key worthless.

It may be an ETS option to use the same runtime key for multiple devices for all runtime communication or not.

The runtime keys will be randomly generated by ETS.

2.6.3.3 Description and handling of security features in the KNX product database

The ETS MT shall provide the possibility per Datapoint (Group Object or Property) to indicate whether it requires authentication or authentication and confidentiality.

It shall be possible that an application or a channel has a combination of secure DPs and insecure DPs. The effective use of the security shall be the same for all secure DPs of a channel or an application. It shall not be possible that of the secure DPs of a channel or application, part of them is handled insecure and the remaining is handled secure. This is not allowed as it may compromise the security at application level.

EXAMPLE 26 If the input GOs to an application are handled confidential, but the state report is handled unsecure, then the hacker may conclude or guess well on the assumed input data and by this easier hack the security.

If secured DPs of a secure channel or secure application are already linked and the ETS user wants to link a further secure DP but not in a secure way, then ETS shall warn the ETS user and ask him to use security for this link as well, or abandon the security of the already established links.

Likewise, if an unsecure DP is attributed a GA that is used for secure communication, ETS shall warn the ETS user and propose to either abandon the link, or use the security also for the newly added DP (if possible) or abandon the security of the other already linked GOs; it may be noted to this last case that this may mean that already programmed devices have to be reprogrammed.

If a product or application uses security, then its entire Device Configuration Procedures shall be done using secure communication.

NOTE 27 This is necessary because the Resources (Group Address Table, Group Object Association Table…) are shared between the secure DPs and the unsecure DPs.

2.6.3.4 Integration in the S-Mode Configuration Procedures

2.6.3.4.1 General

Security Mode ex-factory

Security Mode shall be disabled ex-factory. S-Mode Configuration Procedures do not support devices of which Security Mode is enabled ex-factory.

If there would be a need that Security Mode is enabled ex-factory, then this shall be seen as the use of pre-configured devices in the MaC and shall be solved through solutions to be provided by the MaC (ETS).

More information is given in B.4.

Diffie-Hellman

The S-Mode Configuration Procedures do not support Diffie-Hellman key exchange, but shall use the FDSK. This leaves no option open for a man-in-the-middle attack.

2.6.3.4.2 Overview

The following steps shall be done.

1.	Network Configuration	see 2.6.3.4.3
2.	Synchronisation of the Sequence Numbers	see 2.6.3.4.4
3.	Activating the Security Mode	see 2.6.3.4.5
4.	Set the Tool Key	see 2.6.3.4.6
5.	Device Configuration	see 2.6.3.4.7

2.6.3.4.3 Network Configuration Procedure (assignment of DoA and IA)

2.6.3.4.3.1 Initial assignment

The initial assignment of DoAs and IAs, this is, if the device is in ex-factory state, shall be done unsecure. The Configuration Procedures are not specific for Secure Devices; they shall be identical as for Unsecure Devices.

Some background information is given in B.4.

2.6.3.4.3.2 Modification of existing DoA or IA

If the MaS is not in ex-factory state and the Security Mode is enabled in the MaS, and the MaC wants to modify the DoA or the IA, then the MaC shall do this using KNX Data Security, with the Tool Key, using Authentication and Confidentiality, accessing the relevant Property PID_DOMAIN_ADDRESS or PID_SUBNET_ADDR and PID_DEVICE_ADDRESS in the Device Object.

The MaC shall not disable Security Mode in the MaS to this.

2.6.3.4.4 Synchronisation of the sequence numbers

This shall prevent that every activation of the Security Mode will use the same Sequence Number. This actually authenticates the MaC to the MaS.

The MaC and MaS shall initialise their Sequence Numbers with a small random value. Please refer to clause 2.3.2.14.3.

2.6.3.4.5 Enable the Security Mode in the MaS

This shall be done using unsecure communication. This uses a Function Property.

```
        /* Procedure to enable Security Mode. */
DM_FunctionProperty_Write_R(object_type = Security Interface Object, property = PID_SECURITY_MODE,
        start_index = 1, noElements = 1, command = Enable, error)
```

The Sequence Numbers used shall be the one resulting from the preceding step. After this, Security Mode shall be enabled. The Device Management shall from this step onwards use secure communication and shall be protected with the FDSK.

Security Mode can only be disabled using Secure Communication.

2.6.3.4.6 Set the Tool Key

This step is mandatory: the MaC shall replace the FDSK with the Tool Key that shall have a different value.

This shall be a normal Property access, using secure communication with confidentiality with the FDSK as key.

```
        /* Procedure to set a link */
DMP_InterfaceObject_Write_R(object_type = Security Interface Object, property = PID_SKI_TOOL,
                start_index = 1, noElements = 1, data = Tool Key)
```

2.6.3.4.7 Device Configuration Procedures

The further full device Configuration shall use secure communication, with confidentiality, using the Tool Key.

This can use point-to-point connectionless or - connection-oriented communication. The Configuration Procedures shall further be identical as specified in [07], but the AL-services shall be encrypted.

NOTE 35 Possible T_Connect- and T_Disconnect-PDUs during the Configuration Procedure only base on TL-services and are not forwarded to AL. These can thus not be secured.

Group Object Security Flags

As part of the download of the Group Address and Group Object configuration, ETS shall for the used Group Objects also set the appropriate bits in the Group Object Security Config Table.

2.7 Runtime Interworking

2.7.1 Datapoint Type DPT_Security_Report

This clause shall be integrated in [08].

Format:	1 octet: B_8							
octet nr.	1							
field names	Security Report							
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
encoding	b	b	b	b	b	b	b	b
PDT:	PDT_BITSET8 (alt: PDT_GENERIC_01)							

Datapoint Types

ID:	Name:	Encoding, range:	Use:
21.1002	DPT_Security_Report	See below	System

Bit	Description	Encoding	Unit	Range
b_0	Security Failure This field shall indicate whether there has been a security failure since the previous transmission or not.	0 = There is no Security Failure. 1 = There is a Security Failure	none	{0,1}
b_1 to b_7	reserved	0	none	0

2.8 Usage and context

2.9 Profile definition

🗎 *The following clause shall be added to the Profiles ([11]).*
It shall be added as new clause A.2.11 "Security Interface Object" (S-Mode Profiles) and as new clause A.4.22 "Security Interface Object" (E-Mode Profiles).

2.9.1 Profile Module "S-AL"

🗎 *The "Profiles" known so far from Volume 6 ([11]) rather specify "Device Profiles", with all optional and mandatory features for a full device, from Physical Layer to Configuration Mode. It is difficult in such approach to specify mandatory and optional features for functionality that can be used (optionally) by more than one "Device Profiles"' (e.g. on more than one mask), on more than one medium. This clause tends to specify one or more standard combinations of mandatory, conditional and optional features related to S-AL.*
There have been similar "Profile Modules" in the past. This is functionality that can be combined with any existing Device Profile. Example are: TP1 DPSU Profile, KNXnet/IP Tunnelling Server, KNX USB interface etc.

2.9.1.1 Medium dependent features

🗎 *The symbols in this clause are inherited from [11].*
"M": mandatory
"n/a": not applicable

	Feature	**All media**	**KNX TP1**	**KNX PL110**	**KNX PL110+**	**KNX RF**	**KNXnet/IP**	**KNX IP**
1	KNX Data Security	M	M	M	M	M	M	M
2	KNXnet/IP Secure	n/a	n/a	n/a	n/a	n/a	M	M

2.9.1.2 KNX Data Security

2.9.1.2.1 Secure Application Layer

🗎 *S-A_Data-service and CCM are separated here, so that, if later other algorithms than CCM would be introduced, that for this 1st generation of devices, the next algorithm can be optional.*

	Feature		KNX Secure	KNXnet/IP Security	KNX Data Security
1.	S-A_Data-service	2.2.1.2.1	M	M	M
2.	CCM	2.2.1.2.2	M	M	M
3.	Handle the Sequence Number	2.2.1.3	M	M	M
4.	Register Security Failures	2.2.1.4	M	M	M
5.	Secure handling of Transport Layer services	2.2.1.5	M	M	M

2.9.1.2.2 Support of KNX Data Security services in the plain Application Layer

	Feature		KNX Secure	KNXnet/IP Security	KNX Data Security
1.	par_auth, par_conf and link_index	2.2.2	M	M	M

2.9.1.2.3 Application Interface Layer

	Feature		KNX Secure	KNXnet/IP Security	KNX Data Security
1.	Support Permissions and Roles	2.2.3.2	M	M	M
2.	Support the Security White Lists and the Security Black Lists	2.2.3.3	M	M	M
3.	Support the Security Mode	2.2.3.4	M	M	M
4.	Security Intermediate List for services	2.2.3.4.2	M	M	M
5.	Security Intermediate List for Datapoints	2.2.3.4.3	M	M	M
6.	Register Security Failures in the AIL	2.2.3.5	O	O	O

2.9.1.2.4 Interface Objects and Properties

🗎 *In [11] clause A.1.2.1, the legend for the notation style for data Properties shall be extended as follows.*

	Interface Object	KNX Secure	KNXnet/IP Security	KNX Data Security
0	Device Object	M	M	M
9	Group Object Table Object	O	O	O
17	Security Interface Object	M	M	M

2.9.1.2.5 Device Object

Property Identifier	Support	Property Service type		All Profiles		
				Role	Security	Access level
12 PID_MANUFACTURER_ID	M	Data	Read	any	plain	3
			Write	manuf	plain	0
				OEM	plain	1
				other	none	X
54 PID_PROG_MODE	M	Data	Read	any	plain	3
			Write	ETS	A+C	2
62 PID_OBJECT_VALUE	X	Function	Command	none	n/a	n/a
			Read	none	n/a	n/a

2.9.1.2.6 Group Object Table Object

Property Identifier	Support	Property Service type		All Profiles		
				Role	Security	Access level
1 PID_OBJECT_TYPE	M	Data	Read	any	plain	3
			Write	none	n/a	n/a
2 PID_OBJECT_NAME	O	Data	Read	any	plain	3
			Write	none	n/a	n/a

2.9.1.2.7 Security Interface Object

The below Profile solely gives an indication about whether or not a Property shall be implemented, how it shall be accessed and the possible read- and write possibilities. Further dependencies, e.g. accessibility in function of the Security Mode or accessibility through plain AL-services or the secure AL-services are not specified here. For these, please refer to the specification of these Properties.

Property Identifier	Support	Property Service type		All Profiles		
				Role	Security	Access level
1 PID_OBJECT_TYPE	M	Data	Read	any	plain	3
			Write	none	n/a	n/a
2 PID_OBJECT_NAME	O	Data	Read	any	plain	3
			Write	none	n/a	n/a
5 PID_LOAD_STATE_CONTROL	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
51 PID_SECURITY_MODE	M	Function:	Command:	ETS	A+C	2
			State:	ETS	A+C	2
52 PID_SKI_TABLE_LINK	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
53 PID_SKI_TABLE_KEY	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
54 PID_SECURITY_INDIVIDUAL_-ADDRESS_TABLE	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
55 PID_SECURITY_FAILURE_COUNTERS_ON_LINKS	O	Data:	Read:	any	none	3
			Write:	ETS	A+C	2
56 PID_SKI_TOOL	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
57 PID_SECURITY_REPORT	M	Data:	Read:	ETS:	none	3
				Unlisted	none	3
			Write:	ETS:	A+C	2
		NwPar:	IR	n/a	n/a	n/a
58 PID_SECURITY_REPORT_CONTROL	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
59 PID_SEQUENCE_NUMBER_SENDING	M	Data:	Read:	ETS	A+C	2
				any	A+C	3
			Write:	none	n/a	n/a
	M	Sync:	n/a	ETS	A+C	n/a
			n/a	any	A+C	n/a
60 PID_ROLE_TABLE	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
61 PID_GO_SECURITY_FLAGS	M	Data:	Read:	ETS	A+C	2
			Write:	ETS	A+C	2
63 PID_SECURITY_FAILURE_COMMON_CTR	M	Data:	Read:	any	none	3
			Write:	ETS	A+C	2

2.9.1.3 Security features

	Feature	All S-Mode Profiles
1	Security Mode	O
	Enabled ex-factory	X
	Disabled ex-factory	M
2	Secure key transfer	M
	FDSK	M
	Diffie-Hellman	X

2.10 Identifiers and discovery

🗎 *This clause is not intended for integration in the KNX Specifications.*

The support of KNX Data Security is not related to the S-Mode device Profile. It cannot be concluded upon based on the value of the Device Descriptor Type 0 (Mask Version). For an unknown device, the support of KNX Data Security can be concluded through the presence of an Interface Object of type “Security Interface Object”.

This shall be done in point-to-point communication mode, using A_PropertyValue_Read for the Property Object Type in each Interface Object in the device, this is, using the procedure DMP_InterfaceObjectScan_R as specified in [06]. The Object Type in the Security Interface Object shall be accessible using unsecure communication, but shall not be accessible using Network Parameter service (A_NetworkParameter_Read) (see 2.9.1.2.4).

For a known product in an ETS project, ETS shall however base on the product description for this discovery. See also 3.5.

The discovery in E-Mode is not yet standardised (state October 2012). This will at first be integrated in PB FEC; this specification is under work.

3 Impact and dependencies

3.1 System specification (“Handbook”) dependencies

🗎 *This paragraph shall not be integrated in the KNX Specifications.*

The preceding specifications in this document give indications about the integration in the KNX Specifications.

3.2 Configuration interworking

3.2.1 General

This clause considers compatibility during Configuration with devices that do not use KNX Data Security and with legacy versions of ETS.

The Network Configuration Procedures are done unsecure using standard KNX Frames and – messages. There is no conflict thus with existing, plain devices.

Through the Security White Lists, it is also required that the KNX Data Security devices support basic Network Configuration, like the checking of whether an IA is free or not. This prevents conflicts with legacy ETS versions and hence with other devices.

The configuration of the KNX Data Specific Resources is done in point-to-point communication and does not disturb plain devices.

3.2.2 Access control and authorization

Access Control and Authorization do not exclude each other and can be combined in a device using KNX Data Security.

ETS will typically authorize authorise twice to the device: one with the free access key and once with the key given in the project. ETS will then use the key for which the device gives the lowest access level (higher permissions).

ETS is able to differentiate if either one of both mechanisms fails.

- If there is a failure in the KNX Data Security, then the MaS will not react.

  The MaS will not react to confirmed AL-services, like A_Authorize_Request, A_PropertyValue_Write and even A_Memory_Write (with active Verify Mode). The MaS does not respond to A_Memory_Read either.

If the KNX Data Security is correct but there is any failure in the Authorization, then the MaS will act as follows.

> The A_Authorize_Request.res will give the lowest access level. This is the normal behaviour in case of faulty authorization.
>
> Confirmed services will be confirmed negatively: the A_PropertyValue_Write.req will contain 0 as nr_of_elem and will contain no data; the A_Memory_Write.res will have a number of data 0 and will contain no data either with active Verify Mode. If Verify Mode is inactive, ETS is not informed about an Authorization failure. This again is the situation as today.

Summary

For confirmed services, ETS can differentiate between a single error in either the KNX Data Security or in the Authorization. ETS cannot differentiate between a failure in both mechanisms at the same time and a failure of the KNX Data Security.

For unconfirmed services, ETS cannot differentiate between a failure in the KNX Data Security and an Authorization failure.

3.3 Runtime Interworking

In 2.6.3 it is specified that it shall be possible to indicate the security needs and features of the application in the ETS MT.

It may be possible that in the specification of an application, by a KNX Application Specification Group or by KNX Working Group Interworking, mandatory security features are imposed and specified in Volume 7 of the KNX Specifications. It is then up to the manufacturer to properly implement these requirements and – set the indications properly in ETS MT.

EXAMPLE 28 It may be possible that, when standard Functional Blocks are established for anti-intrusion applications on KNX, that the detailed DP specifications in their contain requirements about the use of KNX Data Security.

3.4 Registration and certification

3.4.1 KNX label

In [10], labelling requirements shall be foreseen for KNX products that support secure communication.

3.5 Integration and common tool impact

🗎 *This clause is not intended for integration in the KNX Specifications.*

3.5.1 Introduction

Differentiation shall be made between the *Security Link Parameters* and the *Roles and Permissions*. The *Security Link Parameters* are selected for a GA in ETS, are downloaded in the Security Link Table and in the Security Group Object Config Flags of the GOs. The Security Link Parameters thus define the type of security that will be used for a Group Address.

The Permissions are however defined in the application in a fix way. This defines the security features that are minimal required by the Group Object.

If the Security Link Parameters and the Permissions do not match, this is, if what is required and what is used, does not match, then a Telegram on that GA will by the receiver for that GO simply be ignored.

3.5.2 Discovery

The support of KNX Data Security shall not be related to any Device Descriptor Type 0 (“mask version”).

ETS shall base the discovery of the support of KNX Data Security on the product description in the database. It shall be possible in the ETS MT to indicate the security that is required for a DP (GO)

. If this is not available, this is, for an unknown product and application, then this may be based on online discovery through searching for an Interface Object of the type Security Interface Object in the device.

3.5.3 Group Address creation – GA Security Flags

If a new GA is created in ETS, it shall be possible for the ETS user to indicate whether this GA shall be used for plain communication or for secure communication with authentication or with confidentiality. These setting are called the “Group Address Security Flags” and can have the values: “plain”, “authentication only” or “authentication and confidentiality”.

NOTE 36 Throughout this paper, there are several variables that store information about the use of plain communication, authentication only or authentication and confidentiality. Obviously, these are related to each other and the one is derived from the other. However, it is important not to confuse these.

Security Link Parameter	This is contained in the Security Link Table in the device for each link and used by the AIL.
Group Address Security Flags	These are the ETS user's indication for one GA about the security used for GOs using this GA. These are not stored in the device.
Group Object Security Flags	These are the flags, resulting from the above GA Security Flags, set by ETS for each GO in the Property PID_GO_SECURITY_FLAGS in the device.

3.5.4 Permissions

Permissions and Roles are fixed in the application. There is no standard way for defining Permissions or Roles. However, it is allowed the Permissions can be modified through implementation specific Parameters in the ETS application description.

As these Permissions influence the links that can be made between a GO and a GA with certain Security Link Parameters, it is possible that the modification of such Parameter value makes that an already assigned GA does no longer match the Permissions of the GO and that the link shall be broken.

EXAMPLE 29 The ETS user creates a GA for use with authentication only and links this to a GO that has a Role that only requires authentication. If now the ETS user modifies such implementation specific Parameter that changes the Permissions for that GO in such a way that confidentiality is required, then the following can be done.

a. The ETS user can be informed and the link can be broken.
b. Or, the Security Link Parameters for the GA in the ETS project are modified to "authentication". This may mean that the (other) senders get a different, possibly unwanted Role in this or in other devices.

3.5.5 Linking a GO to a GA

- If the ETS user links a new GO to this GA, then ETS shall show the Role(s) that are attributed to this GO with the chosen Security Link Parameter (none, authentication or confidentiality) in the application, so that it is clear for whom the GO is thought by the application developer. This shall help the ETS user to make the appropriate links and avoid creating links with unwanted possibilities.

  NOTE 37 The ETS user chooses the Security Link Parameters (none, authentication, confidentiality) for the GA and ETS will set this in the Group Object Security Flags. The Permissions and Roles are however implementation specific. So, it is possible that no Role matches with the chosen Security Link Parameter and that ETS cannot create any link.

  EXAMPLE 30 If the application developer has decided that a GO requires confidentiality, then he will not implement any Role that allows access to this GO using a Link that supports no KNX Data Security or only authentication.

- If the ETS user links a new GO to this GA, ETS shall set the security features (plain, authentication or confidentiality) from this GA in the *Group Object Security Flags* of that Group Object and set these parameters accordingly during a download.

3.5.6 Configuration Procedures

3.5.6.1 Extension of the initial Configuration Procedures

ETS shall extend the existing Configuration Procedures with the handling of the FDSK and the assignment of the Tool Key as specified in clause 2.6.2 "Common requirements for the Configuration Procedure" and clause 2.6.3 "S-Mode".

ETS shall replace the FDSK with the Tool Key; this shall be done without asking the ETS user.

3.5.6.2 Device replacement

If a secure device is replaced by another secure device, then it is recommended that ETS firstly reads out the old device: its own Sequence Number that it uses for sending (PID_SEQUENCE_NUMBER_SENDING) and the Sequence Numbers of the devices from which the old device receives messages (contained in PID_SKI_TABLE_LINK (PID: 52)). This information shall be written in the according Properties in the new device.

If the old device is no longer accessible, e.g. because it is broken, then ETS shall – from its project information, read out the PID_SEQUENCE_NUMBER_SENDING of all the communication partners and store this in the PID_SKI_TABLE_LINK (PID: 52) of the new device. To learn the value for the PID_SEQUENCE_NUMBER_SENDING in the new device, it is sufficient that one link in PID_SKI_TABLE_LINK (PID: 52) in one communication partner is read out.

3.5.6.3 Adding or removing secure links to a secure device

It is required that the Security Individual Address Table and the Security Link Table be sorted. Consequently, if a secure communication partner is removed or changes Individual Address, these tables have to be rewritten entirely. Also, it is needed that, when the remaining IAs of the remaining communication partners are written back to the device, that the Sequence Numbers of these devices be written as well. Hence, ETS should read out the Sequence Number of each linked Individual Address prior to clearing these tables, and write these Sequence Numbers correctly back when reprogramming the device.

3.5.7 Database

Passwords

A Role in a KNX application may be associated with at most one password. Two different Roles will typically have different passwords, but also may share the same password.

Product description

ETS shall foresee in the product description the possibility to indicate whether the device support KNX Data Security or not.

Application description

ETS requirement 1

It shall be possible to list and describe (in text) the Roles that the application developer foresees in the application. The identifiers used for these Roles shall be the values that shall be contained in the Role Table. (These will also be the values that the device uses internally in its Permissions Tables.) It shall be possible in the linking to classify a Link to none or at maximum one Role. Specifically, in the list of Group Addresses assigned to an Input Group Object or a Group Object that can be read, it shall for each Group Address be possible to indicate what its Role shall be.

It shall be possible in the application description to describe the Permissions that are implemented in an application.

Project store

ETS shall for each device store the FDSK, the Tool Key and the runtime keys and the device KNX Serial Number in an encrypted way in the database and in encrypted parts of the import- and export file formats (XML and other). It shall not be possible to read any key by any other access to the database than through the means that will be foreseen in ETS (e.g. binary file access, using other database access than ETS or any other.)

ETS may use one common Tool Key for all devices in a project; it may also have multiple Tool Keys and it may have a Tool Key for each separate device.

3.6 Risks and compatibility issues

3.6.1 Programming Mode in a configured KNX Data Security device

This clause is about the following ETS functionalities.

- Locate KNX devices by flashing the Programming LED.
- Switch Programming LED on or off.

As these Resources are in a configured KNX Secure device only accessible using KNX Data Security with the correct Tool Key, it will not be possible to do this function without the availability of the correct ETS project database with the correct Tool Key.

This also means that, WITH the availability of this information, the procedure (services) does not have to be modified, but the services shall be secured using the Tool Key.

This functionality will also fail with legacy ETS versions.

Annex A
(informative)

Use of CCM

A.1 Goal

CCM is specified in [23] and [25]. The goal of this annex is not to redefine CCM, but to describe in an informative way how it is used in KNX. The KNX specific use of CCM, the contents of the variable, is given in the normative part of this specification in clause 2.

A.2 Definitions

Firstly, the following operations and functions need to be defined.

Operations and functions

Symbol	Description
X | Y	The concatenation of the two strings X and Y.
$X \oplus Y$	The bitwise exclusive-OR of the two bit strings X and Y of the same length.
$AES_K(X)$	The encryption of X according the AES algorithm using the key K.
$MSB_n(X)$	The most n significant bits of X.
X_{padded}	X padded with 0-bits, zero or mode; on the least significant positions to come to a length of X_{padded} that is equal to the block length of 16 octets.

Fields

CCM protects a payload P and additional data A. The content of these variables is KNX specific and is therefore not given in this informative annex. Please refer to the clauses 2.2.1.2.2.4 “Authentication only (Multicast)” and 2.2.1.2.2.5 “Authentication and Confidentiality (Multicast)” for the definitions of A and P.

A_{len}	The length of A in bit.
a	The length of A in octets. (size of a = 16 bit)
P_{len}	The length of P in bit.
Q	The length of P in octets.

A.3 CCM operation

1. The composition of the first Block Counter B_0 is KNX specific. This is specified in clause 2.2.1.2.2.2 and in Figure 9.
2. Build the blocks B_1 to B_n as follows

   The blocks B_1 to B_n shall be built by concatenating a with A_{padded} and then with P_{padded}.

```
pseudo code:
    $B_1$ to $B_n$    =    a | $A_{padded}$ | $P_{padded}$
```

NOTE 38 B_1 to B_n are thus always built in this way - a | A_{padded} | P_{padded} – regardless of whether “authentication only” is used or “authentication and confidentiality” is used; however, the contents of A and P is different: see clause 2.2.1.2.2.4.2 and clause 2.2.1.2.2.5.2 respectively.

3. Calculate the first output block Y_0 as AES128 encryption of the Block B_0 with the encryption key k.

```
pseudo code:
    $Y_0$    =    $AES_K(B_0)$
```

4. Calculate the further output blocks Y_1 to Y_n as AES128 encryption with the key k of the exclusive-OR function of the block Y_0 to Y_{n-1} respectively

```
pseudo code:
    $Y_i$    =    $AES_K(Y_{i-1} \oplus B_i)$    for i = 1 … n
```

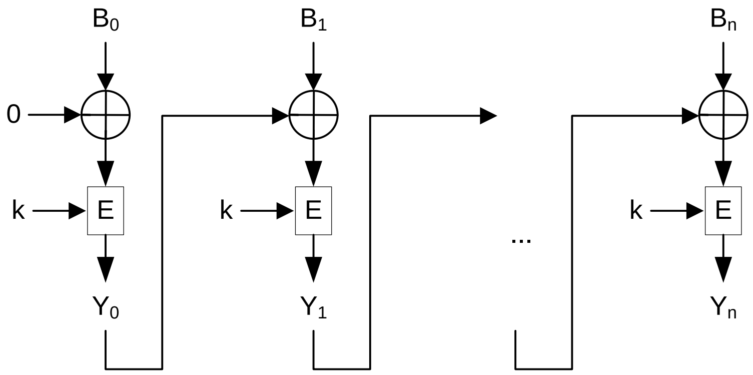

Figure 36 – Block diagram for AES-128 with CBC-MAC

5. The composition of the Block Counter Ctr_0 is KNX specific. This is specified in Figure 10.

```
pseudo code:
    $Ctr_j$    =    SeqNr | SA | DA | 0000000001h | j            // for j = 0 to n; $n \leq 255$
                  // This is, $CTR_j = CTR_{j-1} +1$: the Ctr is incremented by 1 for each block.
```

6. Calculate the output blocks S_0 to S_n, which shall be the successively calculated cipher texts of the AES encryption of the Block Counters Ctr_0 to Ctr_n with the key k.

```
pseudo code:
    $S_i$    =    $AES_K(Ctr_j)$            // for j = 0 to n; $n \leq 255$
```

7. The stream block S is built by concatenating the output blocks S_1 to S_n.

NOTE 39 S_0 is not used in this case!

```
pseudo code:
    S    =    $S_1 | S_2 | … | S_n$
```

8. The encryption is done by bitwise XOR-ing the stream block S with the bits of the payload P, starting from the most significant bits and concatenating the result with the MAC (this is $MSB_{32}(Y_n) \oplus S_0$)

```
pseudo code:
    C    =    $(P \oplus MSB_{Plen}(S)) \mid (MSB_{32}(Y_n) \oplus MSB_{32}(S_0))$
```

NOTE 40 If “authentication only” is used, P is empty and only $(MSB_{32}(Y_n) \oplus MSB_{32}(S_0))$ remains.

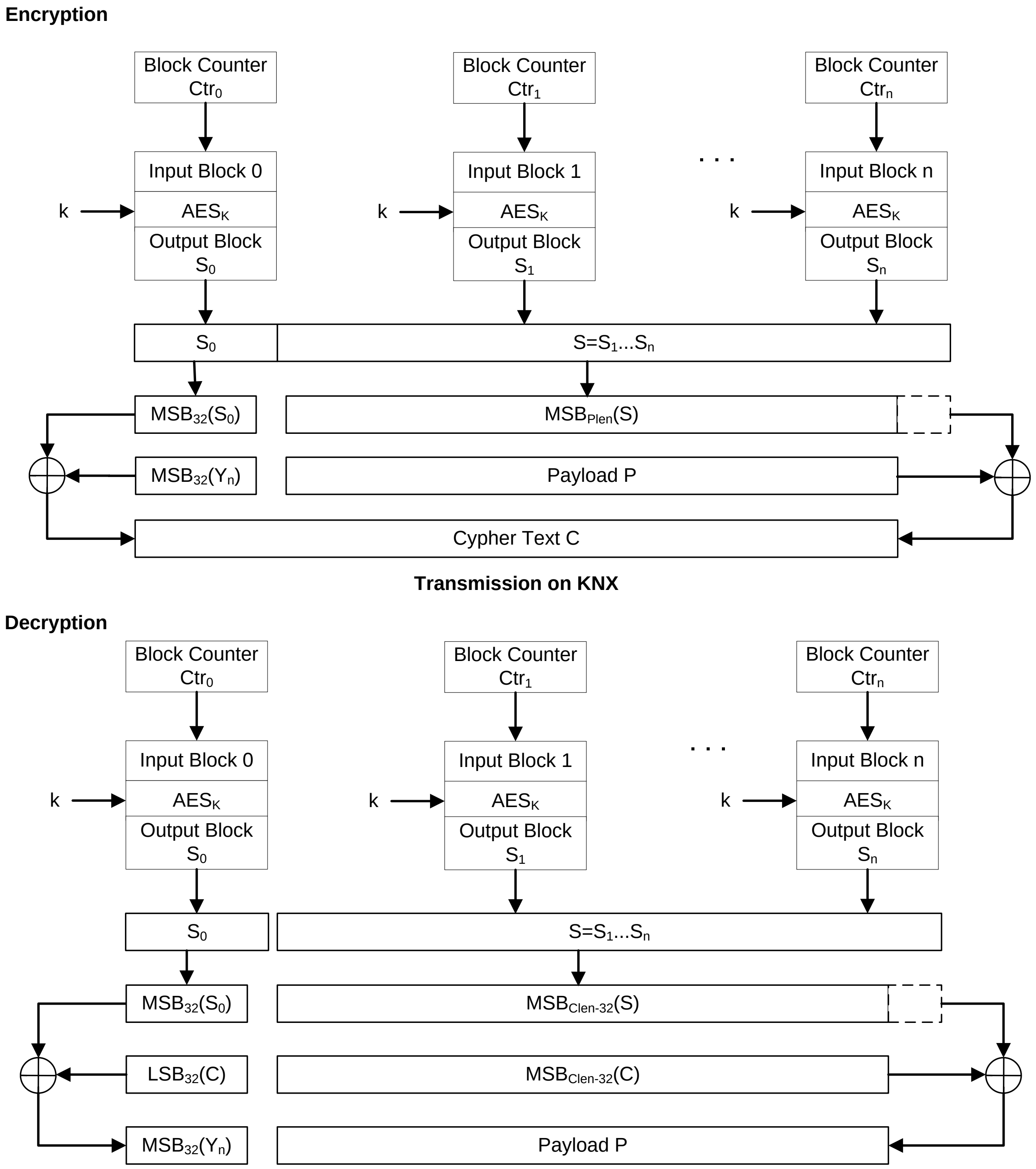

Figure 37 – Block diagram for AES-CTR Mode

Message verification

1. Decrypt the cipher text to obtain the Secure Data.

```
pseudo code:
    Ctr_j  =  SeqNr | SA | DA | 0000000001h | j        // for j = 0 to n; n ≤ 255
                  // This is, Ctr_j = Ctr_{j-1} +1: the CTR is incremented by 1 for each block.

    S_j    =  AES_K(Ctr_j)        // for j = 0 to n; n ≤ 255

    S      =  S_1 | S_2 | … | S_n
```

The received payload is then reconstructed as follows:
C_{len} is the length of the received cipher text.

```
pseudo code:
    P      =  $MSB_{Clen-32}(C) \oplus MSB_{Clen-32}(S)$
```

2. The received MAC (T_R) shall be decrypted as follows.

```
pseudo code:
    T_R    =  $LSB_{32}(C) \oplus MSB_{32}(S_0)$
```

3. The composition of the first block B_0 is KNX specific. This is specified in Figure 9.

```
pseudo code:
    B_0    =  SeqNr | SA | DA | 0000000000h | Q
           Q  =  The length of the payload. Q shall be one octet long.
```

4. Build the blocks B_1 to B_n as follows

The blocks B_1 to B_n shall be built by concatenating A_{len} with A_{padded} and then with P_{padded}.

```
pseudo code:
    B_1 to B_n   =  a | A_padded | P_padded
```

5. Calculate the first output block Y_0 as AES128 encryption of the Block B_0 with the encryption key k.

```
pseudo code:
    Y_0    =  AES_K(B_0)
```

6. Calculate the further output blocks Y_1 to Y_n as AES128 encryption with the key k of the exclusive-OR function of the block Y_0 to Y_{n-1} respectively

```
pseudo code:
        Y_i     =     $AES_K(Y_{i-1} \oplus B_i)$        for i = 1 … n
```

7. The receiver shall then compare Y_n with the decrypted T_R and accept the message solely of both equal.

Annex B
(informative)

Background and motivation

B.1 Runtime - symmetric security algorithm

B.1.1 Algorithm

KNX security algorithm for runtime

Apart from the initial configuration, for runtime communication a symmetric key encryption algorithm shall be used. This allows sender and receiver to have the same key, for encryption as well as for decryption.

For encryption AES 128 shall be used. This requires 10^{28} MIPS years to attack. Hence it gives a protection lifetime of minimal 20 years.

Timings indications

Free sources in C for AES 128 have been implemented on a MSP430F2370 at 4MHz.

(http://www.progressive-coding.com/tutorial.php?id=0)

- 16 bytes encryption → 38 ms
- 16 bytes decryption → 38 ms

Other implementations exist.

A commercialized solution mentions for AES-128 for MSP430 (asm + C interfaces) mentions:

- Encryption : 5342 cycles (thus 1,4 ms at 4 MHz)
- Decryption : 8802 cycles (thus 2,2 ms at 4 MHz)

http://jce.iaik.tugraz.at/sic/products/crypto_software_for_microcontrollers/texas_instruments_msp430_microcontrollers

B.1.2 Operation Mode

Introduction

Any security algorithm may have several operation modes.

EXAMPLE 31 AES has 5 operation modes: ECB, CBC, CFB, OFB, CTR

AES operation modes are used if the data size to encrypt is greater than the block length.

NOTE 41 AES-128 uses a block length of 16 octets

CFB Mode

KNX runtime communication shall use CFB mode.

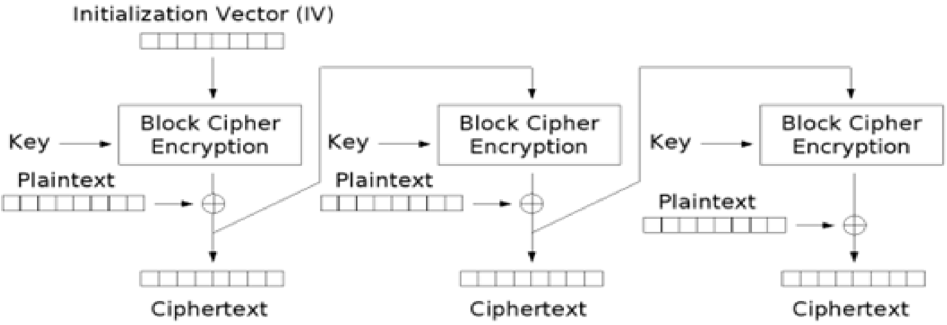

Figure 38 – Cipher Feedback (CFB) mode encryption

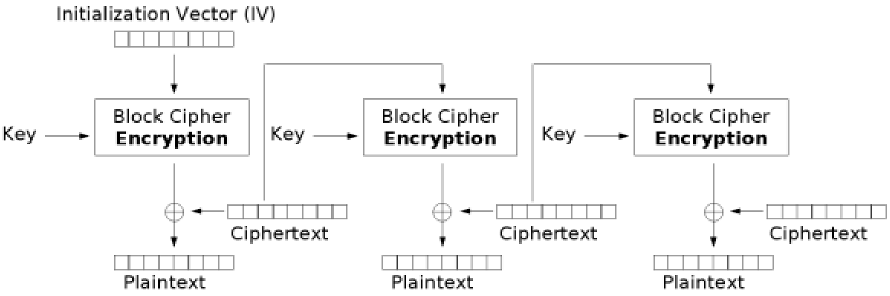

Figure 39 – Cipher Feedback (CFB) mode decryption

Initialisation Vector (IV)

To make each message unique, an initialization vector must be used in the first block. This is shown in Figure 38.

For AES-128 CFB the Initialization Vector shall have a size of 16 octets. The IV shall precede the protected payload. Including the IV in each message ensures that decryption of each received message can be performed, even if some messages are dropped.

B.2 Sequence Number field – calculation of size

This calculation is based on the communication possibilities of the KNX RF Physical Layer.

KNX RF uses a signalling speed of 32 768 cps. This allows to encode 16 384 bits per second [9].

A typical KNX Frame size counts 23 octets. These are 184 bits.

From this, the maximal number of KNX RF messages that can be sent per second can be calculated.

$$\frac{16\ 384\ bits/second}{184\ bits/message} = 89\ messages/second$$

[9] This is the current data rate (April 2011). The KNX RF Multi is not used.

According to the size of the sequence number, a wrap around occurs approximately after the duration below

size random value (bits)	128	64	48	32	16
counter values possibilites	3,40282E+38	1,84467E+19	2,81475E+14	4294967296	65536
max Knx messages sent / sec	89	89	89	89	89
years number	1,21239E+29	6572386712	100286,6625	1,530253028	2,335E-05

Figure 40 - Calculation of the suitable size of the Sequence Number field

The above shows that a wraparound of counter happens after 1,5 year for a size of 32 bits. Therefore, for KNX Secure Communication, the size of the Sequence Number shall be 32 bit.

B.3 Timing calculation of KNX secure communication

Figure 41 shows the resulting timing of a secure communication. The calculation is done for the KNX RF communication medium.

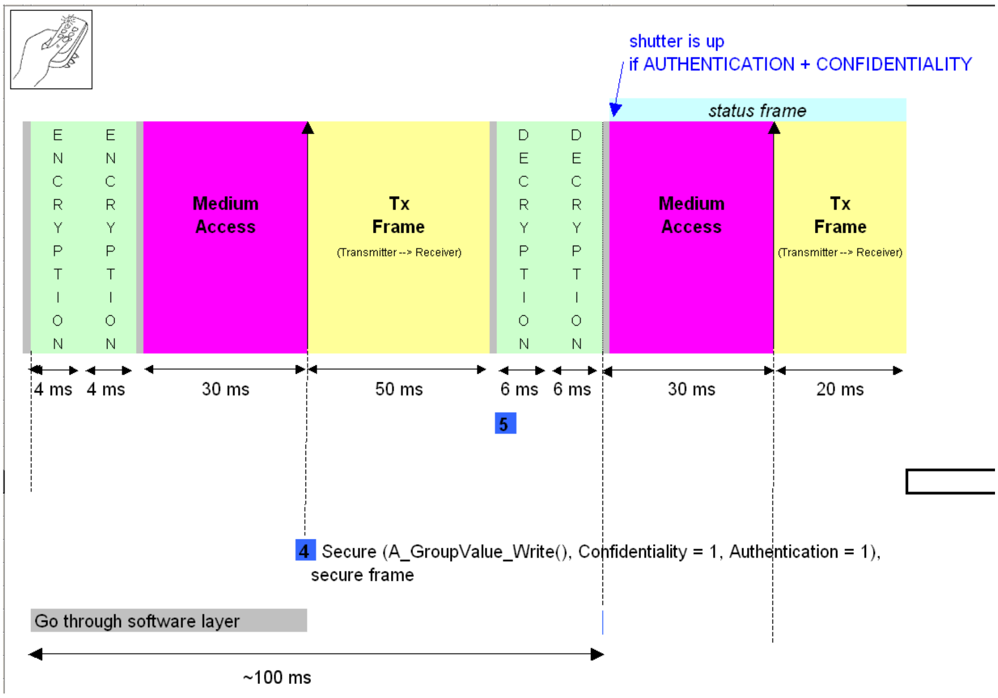
shutter is up
if AUTHENTICATION + CONFIDENTIALITY
status frame
ENCRYPTION
ENCRYPTION
Medium Access
Tx Frame
(Transmitter --> Receiver)
DECRYPTION
DECRYPTION
Medium Access
Tx Frame
(Transmitter --> Receiver)
4 ms
4 ms
30 ms
50 ms
6 ms
6 ms
30 ms
20 ms
5
4 Secure (A_GroupValue_Write(), Confidentiality = 1, Authentication = 1), secure frame
Go through software layer
~100 ms

Figure 41 – Timing of secure communication in KNX RF

B.4 Risks of unsecured Network Configuration.

B.4.1 Risk of using Programming Mode

B.4.2 Risk

Figure 42 shows how a MiM can intervene during an IA assignment procedure between MaC and MaS using the Programming Mode. The MiM can interrupt the broadcast communication, read out the standard MaS identification (Device Descriptor Type 0, manufacturer code ...) from the MaS and act towards the MaC as if it were the legitimate MaS. This will go unnoticed to the ETS user, as he only observes the deactivation of the Programming Mode (e.g. programming mode switched off).

NOTE 42 This requires that the attacker is an active attacker, this is, who has not only the possibility to intercept the network traffic, but who also has the possibility to modify the network traffic. The configuration is safe for a passive attacker, who does not have the possibility to modify the network traffic.

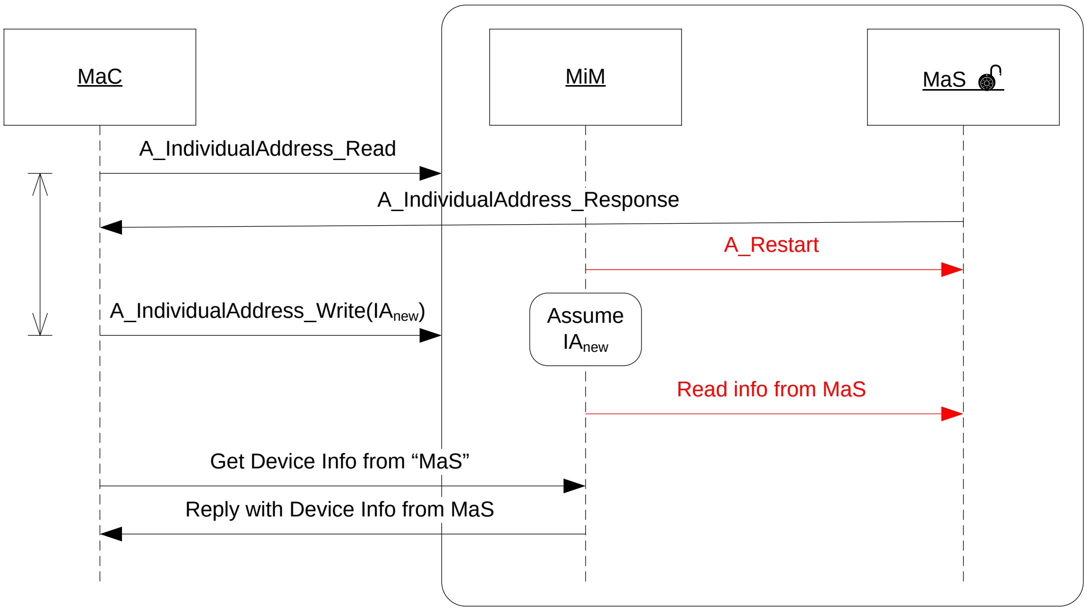
MaC
MiM
MaS
A_IndividualAddress_Read
A_IndividualAddress_Response
A_Restart
A_IndividualAddress_Write(IA$_{new}$)
Assume IA$_{new}$
Read info from MaS
Get Device Info from "MaS"
Reply with Device Info from MaS

Figure 42 – MiM acting as MaS during IA assignment

Alternatively, the IA can be assigned using the KNX Serial Number of the MaS (NM_IndividualAddress_SerialNumber_Write, see [06]). This would leave no option for the MiM to prevent the MaS from assuming the IA$_{new}$. Yet, this reveals the MaS's KNX Serial Number in plain text to the MiM. The MiM may after this action, and before the MaC addresses the MaS again, assign a different IA (IA$_{faulty}$) to the MaS and further on assume the MaS's identity to the MaC.

B.4.3 Solution

The initial Network Configuration shall be performed insecure. This is the initial assignment of Domain Addresses and Individual Addresses shall not be done using secured communication. Secure communication shall only be started after the synchronisation of the Sequence Number between MaC and MaS as described in 2.6.3.4.1.

The MaC shall only enable Security Mode after the Network Configuration is completed.

NOTE 43 The ex-factory delivery of devices with disabled Security Mode also allows for compatibility with legacy ETS versions.

The Network Configuration is thus not protected. This allows to the MiM solely to hamper the Network Configuration. The subsequent Device Configuration shall be protected using the FDSK, which is unknown to the MiM. The MiM will thus not be able to learn the full configuration (keys, links) of the device. The hacker cannot mimic a device during a Device Configuration.

Annex C
(informative)

Examples

C.1 Full encoding of a KNX Secure APDU

A full example of a properly encrypted KNX Frame will be provided here in a next version.

C.2 Permissions Tables

C.2.1 Simple Permission Tables

In a simple and straightforward implementation, every Group Object and every service has its own Permission Table, which can look as shown in Figure C.1.

EXAMPLE C.1

Role	R+W	A	C
	2 bit	1 bit	1 bit
A	01b	1b	0b
B	10b	1b	1b
D	00b	0b	0b

Figure C.1 – Simple Permissions Table implementation basic model

Legend

R+W: This gives the type of access – none, reading or reading and writing – for this Role entry for this Group Object.

00b: no reading and no writing

01b: only reading is allowed

10b: reading and writing is allowed

A: indicates whether Authentication is required

0b: authentication is not required

1b: authentication is required

C: indicates whether Confidentiality is required

0b: confidentiality is not required

1b: confidentiality is required

Role A can read the GO-value, but only with Authentication. Role B can read and write the GO-value, but requires authentication and confidentiality for both actions. Role D has explicitly no permissions for this GO.

This format obviously allows for nonsense combinations and contents. It can for instance indicate that a certain Role may allow writing the GO-value in plain text, without security, but only read the GO-value with Confidentiality, which in most cases would not be meaningful.

What happens with Roles that are not in this Permissions Table?

These are communication partners of this device, which are linked to other DPs, but not to this DP. Such devices will then normally not access this GO; they will not have the proper GA in their GA-table. As also the Security Link Table in the receiver will only accept and pass the incoming message if the GA is contained, this situation can only happen if this sender with this GA is also linked to another GO in this receiver.
One possible answer can be that the standard GO Config flags apply in this case.
Another example may be that the Permissions Table for this GO is extended with a Role standing for “Unlisted” (see 2.2.3.2).

EXAMPLE C.2

Role	R+W	A	C
	2 bit	1 bit	1 bit
A	01b	1b	0b
B	10b	1b	1b
D	00b	0b	0b
Unlisted	*01b*	*1b*	*0b*

Figure C.2 – Simple Permissions Table implementation extension with “Unlisted”

Figure C.2 shows how any other Role than A, B or D, can only read the GO-value with authentication.

What happens with senders that are not in the Security Link Table and hence are not assigned any Role?

This case is more likely than the preceding one: a device that is not linked to this device, attempts to access this GO.
The solutions may be the same as above, but as the Source Address is not contained in the Security Link Table, it will not be possible to relate a Role to this Source Address.

NOTE C.1 This situation is foreseen through the extension of the AL-service parameters (see 2.2.1.5) with the security service parameters par_auth, par_conf and link_index. In this case, the service parameter link_index will be “none” (see 2.2.2.2).

EXAMPLE C.3

Role	R+W	A	C
	2 bit	1 bit	1 bit
A	01b	1b	0b
B	10b	1b	1b
D	00b	0b	0b
Unlisted	01b	1b	0b
none	*00b*	*0b*	*0b*

Figure C.3 – Simple Permissions Table implementation extension with anonymous access

Obviously, in this case, it is not possible to mark authentication or confidentiality, as the sender’s IA is not contained in the Security Link Table and these features can thus not be supported.

C.2.2 Combined Permission Tables

Multiple Simple Permission Tables may be efficient if each communication partner to a device has its own set of services and DPs in the receiver and there is little overlap (see Figure C.4).

If a receiver however has a multitude of communication partners with overlapping links, then it may be more efficient to have Combined Permission Tables (see Figure C.5). A Combined Permission Table collects the Permissions for more than one, possibly all, Datapoints and services in a single Resource.

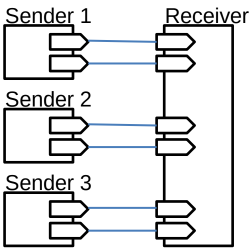

Figure C.4 – Efficient Simple Permission Tables

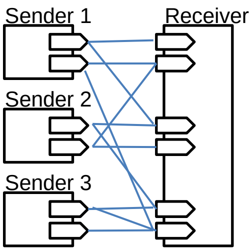

Figure C.5 – Efficient Combined Permission Tables

Such Combined Permissions Table could look like as shown in Example C.4.

EXAMPLE C.4

	Role				
	A	B	C	Unlisted	None
	RWAC	RWAC	RWAC	RWAC	RWAC
GO	4 bit	4 bit	4 bit	4 bit	4 bit
GO 1	0110b	1011b	0100b	0100b	0100b
GO 2	1011b	0110b	0100b	0100b	0100b
GO 3	1011b	0110b	0110b	0000b	0000b
GO 4	1010b	0110b	0110b	0000b	0000b
GO 5	0100b	0100b	0100b	0100b	0100b
GO 6	1011b	1000b	1000b	1000b	1000b

Figure C.6 – Combined Permissions Table for Group Objects

Legend

Every field is composed of 4 bits as follows.

R+W	A	C
2 bit	1 bit	1 bit

The fields have the same encoding as in Example C.1.

Role A is the most powerful Role and can, using authentication and confidentiality, write GO 2 and GO 4.

Role B is can access less GOs, but is the only one that can write GO 1.

Role C is the least powerful Role, but, as it uses Authentication, it can read GO 4, which is not possible with “Unlisted” or “None”.

C.2.3 Permissions for services (primitives)

The same model for Permissions Tables can be used to control the acceptance of service primitives from communication partners.

NOTE C.1 The below Figure C.7 only shows service primitives that do not access data that is stored as Group Object or Property; the access to these can be controlled through one of the preceding Permissions Tables.

NOTE C.2 For point-to-point (connectionless and connection-oriented) communication mode, one sender, through one IA, may have multiple Roles (see 2.2.3.2.3). If this is the case, the Permission Table has to be evaluated for each Role. The Permissions Table is not standardised and implementation specific optimisations are thus possible.

EXAMPLE C.5

	Role				
	A	**B**	**C**	**Unlisted**	**None**
	AC	**AC**	**AC**	**AC**	**AC**
Service primitive	2 bit	2 bit	2 bit	2 bit	2 bit
A_DeviceDescriptor_Read.ind	01b	01b	01b	01b	01b
A_Restart.ind	10b	00b	00b	00b	00b
A_IndividualAddress_Read.ind	01b	01b	01b	01b	01b
A_IndividualAddress_Write.ind	10b	00b	00b	00b	00b
A_Memory_Write.ind	10b	00b	00b	00b	00b
A_Authorize_Request.ind	11b	11b	00b	00b	00b
A_Key_Write.req	11b	11b	00b	00b	00b

Figure C.7 – Permissions for services

The field AC is in this case encoded differently. (This is now an enumeration and not a bit field!)

00b The service primitive is not accepted.

01b The service primitive is accepted without further conditions.

10b: The service primitive is only accepted with authentication.

11b: The service primitive is only accepted with authentication and confidentiality.

Role A can actually modify the entire device. For writing values, authentication is required. Role A can for instance be ETS.

Role B cannot change the device, but, with authentication, it may be able to set keys and authorize itself, so that it can for instance set further Properties in the device. Role B can for instance be a Management Station.

No communication partner except Role A can restart the device, but all communication partners can read its Device Descriptor.

Permissions for services used to control permissions for Group Objects

Basically, the read- and write access to a GO is nothing else than the permission to access the GO with a certain primitive of a Group Object service: A_GroupValue_-Write.ind and A_GroupValue_Read.ind.

EXAMPLE C.6

This is how the permissions could look like for a Group Object. (The table is not complete.)

		Role				
		A	B	C	Unlisted	None
		AC	AC	AC	AC	AC
GO	Service primitive	2 bit	2 bit	2 bit	2 bit	2 bit
GO 1	A_GroupValue_Write.ind	**11b**	00b	00b	00b	00b
	A_GroupValue_Read.ind	**11b**	00b	00b	00b	00b
GO 2	A_GroupValue_Write.ind	10b	00b	00b	00b	00b
	A_GroupValue_Read.ind	10b	01b	01b	01b	00b
GO 3	A_GroupValue_Write.ind	00b	00b	00b	00b	00b
	A_GroupValue_Read.ind	01b	01b	01b	01b	01b
GO 4	A_GroupValue_Write.ind	00b	00b	**10b**	00b	00b
	A_GroupValue_Read.ind	01b	01b	01b	01b	01b

Figure C.8 – Service Permissions to control access to Group Objects

GO 1 is a very secure GO. It can only be written and read with authentication and confidentiality, and only by Role A.
GO 2 can also only be written by Role A, but can be read by any other Role, this is any communication partner that is linked to the device: the “role” “None” is not served.
GO 3 can be read by any device, but cannot be written. This is a full “read-only” GO.
GO 4 is can as well be read without restrictions, but it can only be written by Role C, which has to authenticate. (Think for instance of a diagnostic value, that can only be reset (written) by authorized service personnel.)

Permissions for services used to control permissions for Group Objects

The same approach can be used to control the access to Properties. However, this needs the Permission definitions for more service primitives.

EXAMPLE C.7

			Role				
			A	B	C	Unlisted	None
			AC	AC	AC	AC	AC
IO index	PID	Service primitive	2 bit	2 bit	2 bit	2 bit	2 bit
7	51	A_PropertyDescription_Read.ind	01b	01b	01b	01b	01b
		A_PropertyValue_Write.ind	10b	10b	00b	00b	00b
		A_PropertyValue_Read.ind	01b	01b	01b	01b	00b
		A_NetworkParameter_Read.ind	00b	00b	11b	00b	00b
		A_NetworkParameter_Write.ind	00b	00b	00b	00b	00b

Figure C.9 – Service Permissions to control access to Properties

The Property Description can be read by anybody (01b).

The Property Value can only be written by Roles A and B, with authentication (10b). The other Roles cannot write the Property Value.
All Roles can read the Property Value, but senders that are not linked (Role "None") cannot read the Property Value.
The Property cannot be accessed through A_NetworkParameter_Write.
Only Role C can access the Property through A_NetworkParameter_Read, with authentication and confidentiality (11b).

NOTE C.3 The current KNX Data Security specification does not allow for secure communication on broadcast communication mode. The above definition does counts for the use of A_NetworkParameter_Read and - _Write in point-to-point connectionless or connection-oriented communication mode.

C.2.4 Permissions supporting the White List and the Black List

The Security White List and the Security Black List are not stored in any KNX standard Resource. In the above examples, it is shown that it is possible to control the access to Datapoints and services in a data driven way. By including the services and Datapoints of the Security White List and the – Black List, and by foreseeing implementation specific, internal Roles like "any" or "all", it is possible to solve the requirements of these lists through the mechanism of the Permissions Table.

EXAMPLE C.8

	Role					
	ETS	A	B	C	Unlisted	None
	AC	AC	AC	AC	AC	AC
Service primitive	2 bit	2 bit	2 bit	2 bit	2 bit	2 bit
White List services						
some white list service	01b	01b	01b	01b	01b	01b
…	01b	01b	01b	01b	01b	01b
Black List services						
some black list service	11b	11b	11b	11b	11b	11b
…	11b	11b	11b	11b	11b	11b
White List Datapoints						
Manufacturer Code	01b	01b	01b	01b	01b	01b
Any Interface Object Type	01b	01b	01b	01b	01b	01b
Any Interface Object Index	01b	01b	01b	01b	01b	01b
…	01b	01b	01b	01b	01b	01b
Black List Datapoints						
Security Interface Object						
- Load Control	11b	11b	11b	11b	11b	11b
- Security Mode	11b	11b	11b	11b	11b	11b
- Security Link Table	11b	11b	11b	11b	11b	11b
- Security Key Table	11b	11b	11b	11b	11b	11b
- …	11b	11b	11b	11b	11b	11b
Intermediate List services						
A_Restart – security mode disabled	01b	01b	01b	01b	01b	01b
A_Restart – security mode enabled	11b	00b	00b	00b	00b	00b

Service primitive	Role					
	ETS	A	B	C	Unlisted	None
	AC	AC	AC	AC	AC	AC
	2 bit	2 bit	2 bit	2 bit	2 bit	2 bit
A_Authorize – security mode disabled	01b	01b	01b	01b	01b	01b
A_Authorize – security mode enabled	11b	00b	00b	00b	00b	00b
A_Key_Write – security mode disabled	01b	01b	01b	01b	01b	01b
A_Key_Write – security mode enabled	11b	00b	00b	00b	00b	00b
Implementation - and application specific Permissions						
A_DeviceDescriptor_Read.ind	01b	01b	01b	01b	01b	01b
A_IndividualAddress_Read.ind	01b	01b	01b	01b	01b	01b
A_IndividualAddress_Write.ind	10b	10b	00b	00b	00b	00b
A_Memory_Write.ind	10b	10b	00b	00b	00b	00b

Figure C.10 – Permissions Table to support White – and Black Lists