



# User Manual CANoe Security Extension

Version 1.1.1  
English

## **Imprint**

Vector Informatik GmbH  
Ingersheimer Straße 24  
D-70499 Stuttgart

Vector reserves the right to modify any information and/or data in this user documentation without notice. This documentation nor any of its parts may be reproduced in any form or by any means without the prior written consent of Vector. To the maximum extent permitted under law, all technical data, texts, graphics, images and their design are protected by copyright law, various international treaties and other applicable law. Any unauthorized use may violate copyright and other applicable laws or regulations.

© Copyright 2017, Vector Informatik GmbH. Printed in Germany.  
All rights reserved.

# Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
1.1	Overview Security Support	4
1.1.1	Features	4
1.1.2	Concept	5
1.1.3	Terms	6
1.1.4	History	6
1.2	About This User Manual	7
1.2.1	Certification	8
1.2.2	Warranty	8
1.2.3	Trademarks	8
<b>2</b>	<b>Installation</b>	<b>9</b>
2.1	Preconditions	10
2.2	Installation	10
<b>3</b>	<b>Vector Security Manager</b>	<b>11</b>
3.1	Overview	12
<b>4</b>	<b>Security Source</b>	<b>13</b>
4.1	AUTOSAR SecOC Security Source	14
4.1.1	AUTOSAR SecOC with Trip Based Freshness Management	14
4.1.2	AUTOSAR SecOC with Time Based Freshness	14
<b>5</b>	<b>CANoe Security Support</b>	<b>15</b>
5.1	Database Attributes	16
5.2	Automatic validation of Secured PDUs	16
5.3	Access to SecOC Data	17
5.4	CAPL Extensions	17
5.5	Vector Security Manager Client CANoe	18
5.5.1	CAPL Interface	18
<b>6</b>	<b>Quick Start</b>	<b>25</b>
6.1	Security Configuration	26
6.2	Setup of CANoe Simulation Model	26
6.3	VN8912 / RT System Usage	27
<b>7</b>	<b>Index</b>	<b>28</b>



# 1 Overview

In this chapter you find the following information:

---

1.1	Overview Security Support	page 4
	Features	
	Concept	
	Terms	
	History	
1.2	About This User Manual	page 7
	Certification	
	Warranty	
	Trademarks	

---

## 1.1 Overview Security Support

### Overview

This manual describes the features of the security support in **CANoe**.

Basically it is a matter of:

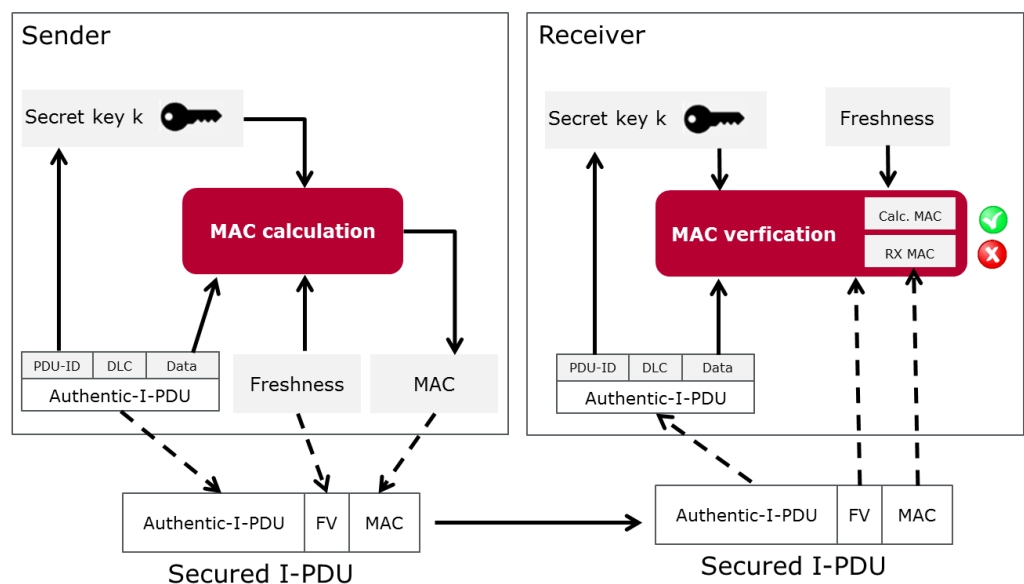
- > Creation of a Message Authentication Code
- > Verification of a Message Authentication Code

The features bases on AUTOSAR 4.2.2.

### Message Authentication Code

To protect important data related to authenticity and integrity secured PDUs have a Message Authentication Code (MAC), which can be verified by a receiver.

A symmetric key will be needed to calculate the authenticator (MAC). The key must be known at sender and receiver side.



**Freshness simulation** The security extension requires additional to the key a valid freshness value.

The freshness value can be transmitted with a sync message. This message will be evaluated and updated by the Vector Security Manager. The transmitted value is used for MAC calculation and MAC verification.

Additionally to this the CAPL Callback `QueryLocalFreshness` provides the possibility to set the freshness in a CAPL node.

### 1.1.1 Features

#### Feature set

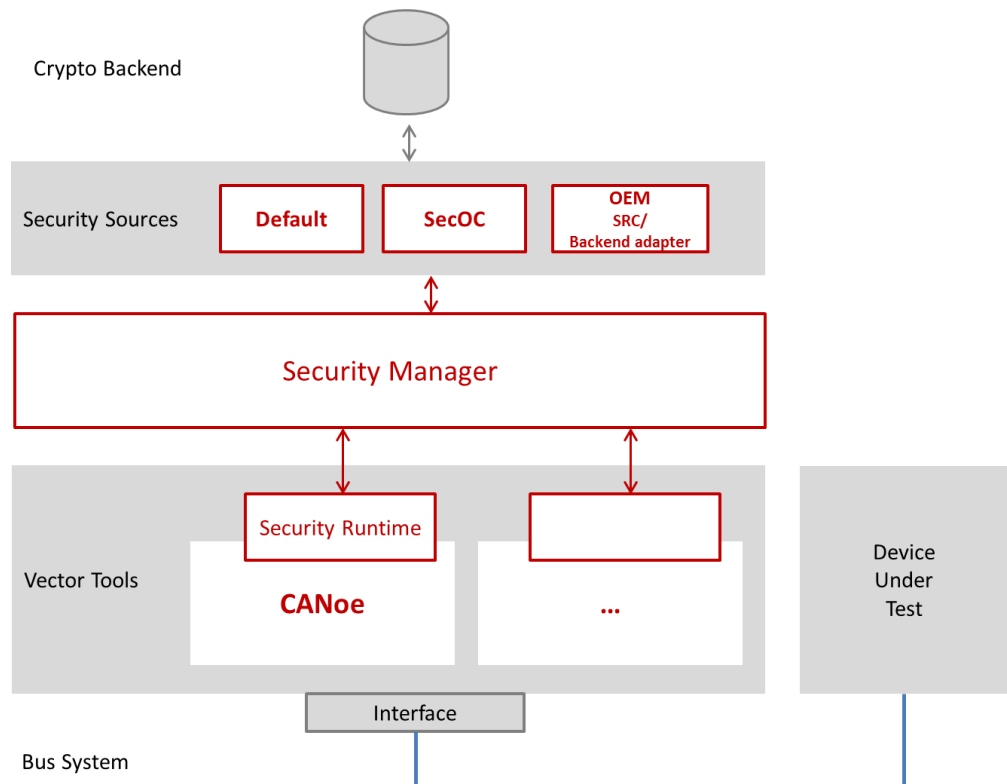
- > Automatic calculation of a Message Authentication Code of a secured PDU
- > Verification of the received Message Authentication Code of a Secured PDU
- > Management of received Freshness counter

#### Configuration

- > Secured PDUs identification and configuration from AUTOSAR database
- > Secured PDUs are defined in the AUTOSAR description as Secured-I-PDU
- > Pairs of Data ID and key are managed in the **Vector Security Manager**

## 1.1.2 Concept

### Security Concept



- OEM security source**
- > Implementation of OEM specific algorithm (SecOC, Authentication...)
  - > Management of keys and certificates
  - > Optional: Communication with a Crypto Backend
  - > Freshness Manager

- Vector Security Manager**
- > Homogeneous management of several OEM security sources
  - > Display and configuration of the data provided by the sources
  - > Communication with the tool clients of the supported tools

- CANOE**
- Node layer:
- > Callback Handler, called before transmission starts and after reception of a secured PDU.
  - > CAPL functions for MAC creation and verification
  - > Callback handler to set the freshness

### 1.1.3 Terms

Security source	Container for one or more security profile templates.
Security profile template	Defines a security set with all required parameters. A set describes the supported functions like SecOC and Authentication. Further it contains the specific cryptographic algorithms.
Security profile	A security profile is a concrete instance of a security profile template. The profile contains concrete parameters like keys and certificates. It is possible to define several profiles of one template in the <b>Vector Security Manager</b> .
SecOC	Secured on board communication
MAC	Message Authentication Code
CMAC	Cipher-based Message Authentication Code

### 1.1.4 History

Version	Changes
1.1.1.0	Initial version for <b>CANoe</b> 10.0 SP3



## 1.2 About This User Manual

### To Find information quickly









This user manual provides you with the following access help:





- > At the beginning of each chapter you will find a summary of the contents
- > The header shows in which chapter of the manual you are
- > The footer shows the version of the manual
- > At the end of the user manual you will find a glossary to look-up used technical terms
- > At the end of the user manual an index will help you to find information quickly

### Conventions

In the two tables below you will find the notation and icon conventions used throughout the manual.

Style	Utilization
<b>Bold</b>	Fields/blocks, user/surface interface elements, window- and dialog names of the software, special emphasis of terms. <b>[OK]</b> Push buttons in square brackets <b>File Save</b> Notation for menus and menu entries
<b>MICROSAR</b>	Legally protected proper names and marginal notes.
<code>Source Code</code>	File and directory names, source code, class and object names, object attributes and values
<u>Hyperlink</u>	Hyperlinks and references.
<Ctrl>+<S>	Notation for shortcuts.

Symbol	Utilization
	This icon indicates notes and tips that facilitate your work.
	This icon warns of dangers that could lead to damage.
	This icon indicates more detailed information.
	This icon indicates examples.
	This icon indicates step-by-step instructions.
	This icon indicates text areas where changes of the currently described file are allowed or necessary.
	This icon indicates files you must not change.
	This icon indicates multimedia files like e.g. video clips.

Symbol	Utilization
	This icon indicates an introduction into a specific topic.
	This icon indicates text areas containing basic knowledge.
	This icon indicates text areas containing expert knowledge.
	This icon indicates that something has changed.

### 1.2.1 Certification

**Quality Management System** Vector Informatik GmbH has ISO 9001:2008 certification.  
The ISO standard is a globally recognized standard.

### 1.2.2 Warranty

**Restriction of warranty** We reserve the right to modify the contents of the documentation or the software without notice. Vector disclaims all liabilities for the completeness or correctness of the contents and for damages which may result from the use of this documentation.

### 1.2.3 Trademarks

**Protected trademarks** All brand names in this documentation are either registered or non registered trademarks of their respective owners.

## 2 Installation

In this chapter you find the following information:

---

2.1	Preconditions	page 10
2.2	Installation	page 10

---

## 2.1 Preconditions

**CANoe** Minimal **CANoe** version is **10.0 SP3**.

## 2.2 Installation

**Vector Security Manager** With the installation of **CANoe** the **Vector Security Manager** will be installed automatically.

**CANoe security support** The security support is part of **CANoe** standard feature set and therefore the functions are still available.

**AUTOSAR SecOC source** With the installation of the **Vector Security Manager** the AUTOSAR SecOC source will be installed automatically.

**OEM security sources** The OEM security sources must be installed separately. Typically they are distributed by the OEM or by Vector on demand. After installation of the OEM security source the OEM security profiles are in the **Vector Security Manager** available.

## 3 Vector Security Manager

In this chapter you find the following information:

---

3.1 Overview

page 12

---

### 3.1 Overview

The **Vector Security Manager** manages the installed security sources and provides the configured profiles for the tool usage.

#### Tool access

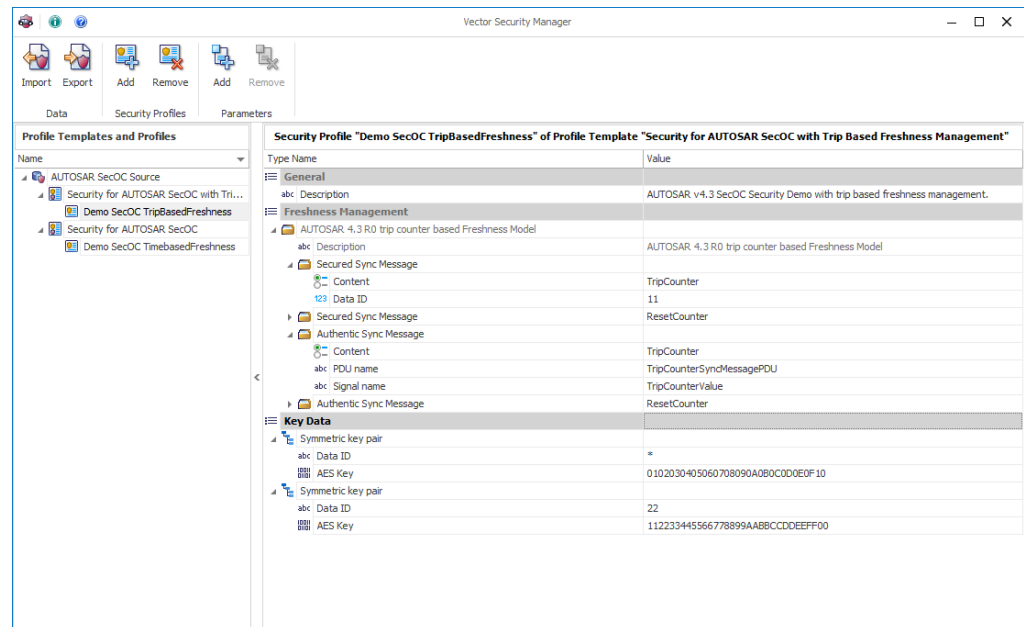
- > The **Vector Security Manager** can be started from the start menu. There is a separate folder **Vector Security Manager**.
- > **CANoe Tools** ribbon tab

The screenshot shows the **Vector Security Manager** with the AUTOSAR SecOC source.

In the tree on the left side the available sources with its profile templates and the configured security profiles are listed.

On the right the details/parameters of the selected security profile are displayed.

The ribbon provides the functions which can be executed on the selected element.



**Reference:** Further details are described in the online help of the **Vector Security Manager**.

## 4 Security Source

In this chapter you find the following information:

---

4.1	AUTOSAR SecOC Security Source	page 14
	AUTOSAR SecOC with Trip Based Freshness Management	
	AUTOSAR SecOC with Time Based Freshness	

---

## 4.1 AUTOSAR SecOC Security Source

The AUTOSAR SecOC source realizes the message authentication for secured PDUs.

With this source two security profile templates are provided:

- AUTOSAR SecOC with trip based freshness management
- AUTOSAR SecOC with time based freshness

### Key handling

The keys are stored in the security source and are used only there. The tool cannot access the keys.

Keys can be imported with the [Vector Security Manager](#).

### 4.1.1 AUTOSAR SecOC with Trip Based Freshness Management

	Parameter	Description
Freshness management	Secured Sync Message	Defines a secured sync message. <b>Content:</b> TripCounter or ResetCounter <b>Data ID:</b> Data ID for key lookup, a PDU with this data ID must be contained in the database.
	Authentic Sync Message	Defines a sync message without additional CMAC. The length of the signal is used as length of the trip or reset counter. <b>Content:</b> TripCounter or ResetCounter <b>PDU name:</b> Name of the PDU in the database. <b>Signal name:</b> Name of the signal in the database.
Key data	Symmetric key pair	<b>Data ID:</b> Data ID for key lookup, * represents a wildcard. <b>AES Key:</b> Key used for message authentication.
Global parameters	Rx Acceptance Window	Has no effect in Trip Based Freshness Management environment.

### 4.1.2 AUTOSAR SecOC with Time Based Freshness

	Parameter	Description
Freshness management Key data	-	
	Symmetric key pair	<b>Data ID:</b> Data ID for key lookup, * represents a wildcard. <b>AES Key:</b> Key used for message authentication.
Global parameters	Rx Acceptance Window	Range within a PDU can be validated.



## 5 CANoe Security Support

In this chapter you find the following information:

---

5.1	Database Attributes	page 16
5.2	Automatic validation of Secured PDUs	page 16
5.3	Access to SecOC Data	page 17
5.4	CAPL Extensions	page 17
5.5	Vector Security Manager Client CANoe CAPL Interface	page 18

---

## 5.1 Database Attributes



**Note:** The minimal AUTOSAR database version is AUTOSAR 4.2.2.

Only PDUs of the type **Secured-I-PDU** are relevant for the security support.

Each Secured-I-PDU has a set of attributes. Currently the following attributes are evaluated:

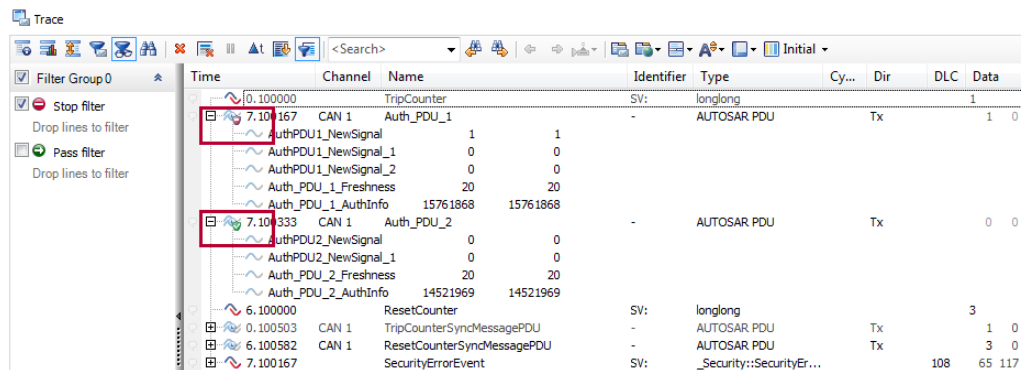
### Secured-I-PDU

Attribute	Description
AuthInfo Tx Length	Length of the CMACs in bit
DataID	ID for key assignment
Freshness Value Tx Length	Length of the transmitted Freshness in bit

## 5.2 Automatic validation of Secured PDUs

### Validation

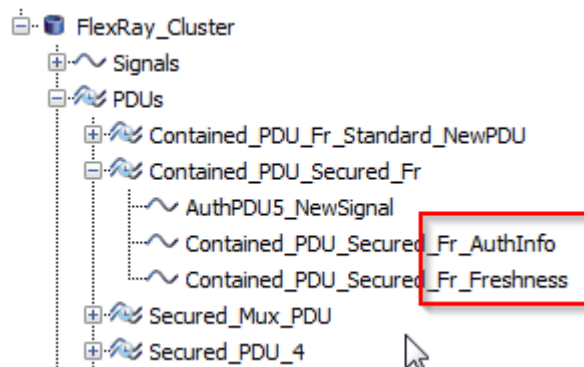
With the reception of secured PDUs the CMAC is automatically validated. The result is displayed in the Trace Window with overlay icons. Additionally the state can be checked in CAPL.



## 5.3 Access to SecOC Data

### SecOC signals

Additional data fields like **AuthInfo** and **Freshness** of a secured PDU are offered as separate signals:



So the secured info can be used in all signal based windows. The signals have the pattern **<PDUName>\_AuthInfo** and **<PDUName>\_Freshness**.

## 5.4 CAPL Extensions

### PDU selector

The AUTOSAR PDU object provides a new selector **ValidationFlags**. This selector contains the state of the validation after reception.

### Validation flags

Validation Flags	Description
Bit 0	0 = not validated – State unknown 1 = Validation executed, Result in Bit 1
Bit 1	0 = Not valid 1 = Valid
Bit 8	1 = Validation Error



### Example:

Validation executed, MAC is valid: Validation Flags = 0x3  
 Validation executed, Mac is not valid: Validation Flags = 0x1  
 Internal module error: Validation Flags = 0x100

## 5.5 Vector Security Manager Client CANoe

### Node layer

With the **CANoe** installation the node layer **SecMgrCANoeClient.dll** is available. This module provides the connection to the security setup of the **Vector Security Manager**. The security configuration of **CANoe** is used to initialize the module.

### 5.5.1 CAPL Interface

#### 5.5.1.1 SecOC

#### Authenticator calculation

<b>Syntax</b>	<code>long LocalSecurityCalculateAuthenticator (dword DATAID, byte[] Payload, dword PayloadLength, qword TruncatedAuthenticator, dword TruncatedAuthenticatorBitLength, qword Freshness, dword TruncatedFreshnessBitLength, dword FreshnessValueBitLength)</code>
<b>Function</b>	Calculates the authenticator (CMAC) for the given payload and freshness. This function can be used for testing.
<b>Parameter</b>	<ul style="list-style-type: none"> <li>&gt; <b>DATAID</b>: Id to lookup the desired key in the security source.</li> <li>&gt; <b>Payload</b>: Data array</li> <li>&gt; <b>PayloadLength</b>: Length of the data array in byte</li> <li>&gt; <b>TruncatedAuthenticator</b>: Contains the calculated truncated CMAC</li> <li>&gt; <b>TruncatedAuthenticatorBitLength</b>: Length of the truncated CMAC in bit.</li> <li>&gt; <b>Freshness [In/Out]</b>: In: Contains the freshness value for MAC calculation Out: Truncated freshness</li> <li>&gt; <b>TruncatedFreshnessBitLength</b>: Length of the truncated freshness in bit</li> <li>&gt; <b>FreshnessValueBitLength</b>: Length of the freshness in bit</li> </ul>
<b>Return</b>	<b>Vector Security Manager</b> status (refer to chapter 5.5.1.4) > <b>1</b> : Success > <b>&lt;=0</b> : Error occurred

Authentication  
verification

<b>Syntax</b>	<code>long LocalSecurityVerifyAuthenticationInformation (dword DATAID, byte[] Payload, dword PayloadLength, qword TruncatedAuthenticator, dword TruncatedAuthenticatorBitLength, qword RxFreshness, dword RxFreshnessBitLength, qword currentFreshness, dword FreshnessValueBitLength, dword ValidationResult)</code>
<b>Function</b>	Verifies the specified authentication information (CMAC and freshness) of the PDU. The payload is also an input parameter to the verification logic.
<b>Parameter</b>	<ul style="list-style-type: none"> <li>&gt; <b>DATAID</b>: Id to lookup the desired key in the security source.</li> <li>&gt; <b>Payload</b>: Data array</li> <li>&gt; <b>PayloadLength</b>: Length of the payload in byte</li> <li>&gt; <b>TruncatedAuthenticator</b>: Received truncated MAC</li> <li>&gt; <b>TruncatedAuthenticatorLength</b>: Length of the truncated CMAC in bits.</li> <li>&gt; <b>RxFreshness</b>: Received freshness value.</li> <li>&gt; <b>RxFreshnessBitLength</b>: Length of the received (typically truncated) freshness in bits</li> <li>&gt; <b>currentFreshness</b>: Freshness value to be used for calculation</li> <li>&gt; <b>FreshnessValueBitLength</b>: Length of the current freshness in bit</li> <li>&gt; <b>ValidationResult</b>: 1=valid CMAC, 0= invalid CMAC</li> </ul>
<b>Return</b>	<p><b>Vector Security Manager</b> status (refer to chapter 5.5.1.4).</p> <ul style="list-style-type: none"> <li>&gt; <b>1</b>: Success</li> <li>&gt; <b>&lt;=0</b>: Error occurred</li> </ul>

### 5.5.1.2 Utility

<b>Syntax</b>	<code>void LocalSecuritySetVerbosity(dword Level)</code>
<b>Function</b>	Configures the notification level.
<b>Parameter</b>	<b>Level:</b> Desired notification level. Range: 0-5 > 0: Only critical errors are printed. > 1: All errors are printed > 2: Additionally warnings are printed > 3: Information messages are printed too. > 4-5: Debug messages are active.

<b>Syntax</b>	<code>dword LocalSecurityActivateTxPDUs(char[] NodeName)</code>
<b>Function</b>	Allows a node (e.g. gateway) to do a CMAC calculation for Tx Secured-I-PDUs of another node.
<b>Parameter</b>	<b>NodeName:</b> Name of the other node
<b>Return</b>	> 1: Activation successful > 0: Specified node is already activated > -1: Specified node is null or empty > -10: Security is not usable

<b>Syntax</b>	<code>dword LocalSecurityActivateRxPDUs(char[] NodeName)</code>
<b>Function</b>	Allows a node (e.g. gateway) to validate Rx Secured-I-PDUs of another node.
<b>Parameter</b>	<b>NodeName:</b> Name of the other node
<b>Return</b>	> 1: Activation successful > 0: Specified node is already activated > -1: Specified node is null or empty > -10: Security is not usable

### 5.5.1.3 Callback Handler

<b>Syntax</b>	<pre>void OnLocalSecurityPDUPreTx(char pduName[], dword dataId, byte payload[], dword payloadLength, qword&amp; authInfo, dword authInfoBitLength, qword&amp; freshness, dword freshnessBitLength)</pre>
<b>Function</b>	This callback handler is called, after all data updates and the automatic Authenticator (CMAC) calculation has been done. Payload, AuthInfo and freshness can be modified in this handler before the transmission starts. Fault injection and evaluation of new algorithm are typical examples.
<b>Parameter</b>	<ul style="list-style-type: none"> <li>&gt; <b>pduName</b>: PDU name</li> <li>&gt; <b>dataId</b>: Used DataID</li> <li>&gt; <b>payload</b>: Byte array with the data to be transmitted</li> <li>&gt; <b>payloadLength</b>: Length of the payload array in byte</li> <li>&gt; <b>authInfo</b>: Authenticator (CMAC)</li> <li>&gt; <b>authInfoLength</b>: Length of the AuthInfo in bit</li> <li>&gt; <b>freshness</b>: Used freshness value</li> <li>&gt; <b>freshnessLength</b>: Length of the freshness in bit</li> </ul>
<b>Syntax</b>	<pre>void OnLocalSecurityPDUValidated(char pduName[], dword dataId, byte payload[], long payloadLength, qword authInfo, dword authInfoBitLength, qword freshness, dword freshnessBitLength, dword verificationResult)</pre>
<b>Function</b>	This callback handler is called, when a secured PDU is received in the node. Besides to the verification result all values which have been used for verification can be analyzed.
<b>Parameter</b>	<ul style="list-style-type: none"> <li>&gt; <b>duName</b>: Name der PDU</li> <li>&gt; <b>dataId</b>: Used DataID</li> <li>&gt; <b>payload</b>: Byte array with the received data</li> <li>&gt; <b>payloadLength</b>: Length of the payload array in byte</li> <li>&gt; <b>authInfo</b>: Received authentication (CMAC).</li> <li>&gt; <b>authInfoBitLength</b>: Length of the authentication in bit</li> <li>&gt; <b>freshness</b>: Received freshness value</li> <li>&gt; <b>freshnessBitLength</b>: Length of the freshness in bit</li> <li>&gt; <b>verificationResult</b>: Verification result. 1= verified, 0= not verified</li> </ul>

<b>Syntax</b>	<pre> dword OnLocalSecurityQueryLocalFreshness(dword context, char pduName[], dword dataId, dword freshnessValueId, dword attemptNr, byte payload[], dword payloadLength, qword&amp; freshness, dword&amp; freshnessLength, qword truncatedRxFreshness, dword truncatedRxFreshnessBitLength) </pre>
<b>Function</b>	<p>This callback handler is called, when a node a secured PDU receives and when a secured PDU is transmitted.</p> <p>Than this callback is triggered. It provides the possibility to build a CAPL based freshness. This freshness value is preferred against the internal Freshness Manager in the <b>Vector Security Manager</b>.</p> <p>Is the callback not implemented or returns an invalid value the internal freshness is used.</p>
<b>Parameter</b>	<ul style="list-style-type: none"> <li>&gt; <b>context</b>: 0: Transmitt (CMAC creation); 1: Receive ( validation)</li> <li>&gt; <b>pduName</b>: PDU name</li> <li>&gt; <b>dataId</b>: Used DataID</li> <li>&gt; <b>freshnessValueId</b>: AUTOSAR Freshness Value ID</li> <li>&gt; <b>attemptNr</b>: Number of validation attempt</li> <li>&gt; <b>payload</b>: Byte array with the payload to be transmitted</li> <li>&gt; <b>payloadLength</b>: Length of the payload array in byte</li> <li>&gt; <b>freshness</b>: Freshness to be used</li> <li>&gt; <b>freshnessLength</b>: Length of the Freshness in bit</li> <li>&gt; <b>truncatedRxFreshness</b>: truncated Rx freshness value</li> <li>&gt; <b>truncatedRxFreshnessBitLength</b>: truncated Rx Length of the truncated Rx freshness</li> </ul>
<b>Return Value</b>	<ul style="list-style-type: none"> <li>&gt; <b>&lt;=0</b>: The query call is not successful; the value of the Freshness Manager is used.</li> <li>&gt; <b>&gt;0</b>: The query call is successful and the arguments <b>freshness</b> and <b>FreshnessLength</b> are used for the MAC calculation / validation.</li> </ul>



### 5.5.1.4 Return Value Ranges

Vector Security  
Manager result

<b>Value Range</b>	<p>A value <math>\leq 0</math> means error Value 1 means action was successful.</p> <p>1: Success 0: Error, no details -1: Invalid handle -2: data incomplete -3: signal length does not fit -4: Security source error, no details -6: Not supported -10: Security is not usable. Reasons can be: Security Manager version is too old. Tool Version is too old. Security Profile is invalid.</p>
--------------------	--

Broadcast message  
status

<b>Value Range</b>	<p>A value <math>\leq 0</math> means error Value = 1 action was successful.</p> <p>1: Broadcast message has been successfully transmitted or received. 0: Error, no details -1: PDU could not be transmitted -2: Invalid broadcast message -3: Invalid length of the Tx data -4: MAC could not be calculated -5: Sequence error -6: Not supported function</p>
--------------------	--



## 6 Quick Start

In this chapter you find the following information:

---

6.1	Security Configuration	page 26
6.2	Setup of CANoe Simulation Model	page 26
6.3	VN8912 / RT System Usage	page 27

---

## Security setup

The chapters 6.1 and 6.2 describe the necessary steps to configure the security in CANoe.

## 6.1 Security Configuration



**Note:** A corresponding AUTOSAR database is needed. At least Secured-I-PDUs must be available. For details see chapter 5.1.

In the **Vector Security Manager** the desired profile template must be selected and a concrete security profile must be created. The profile has to be configured by entering the parameter of the profile, e.g. Data ID key pairs.

## 6.2 Setup of CANoe Simulation Model

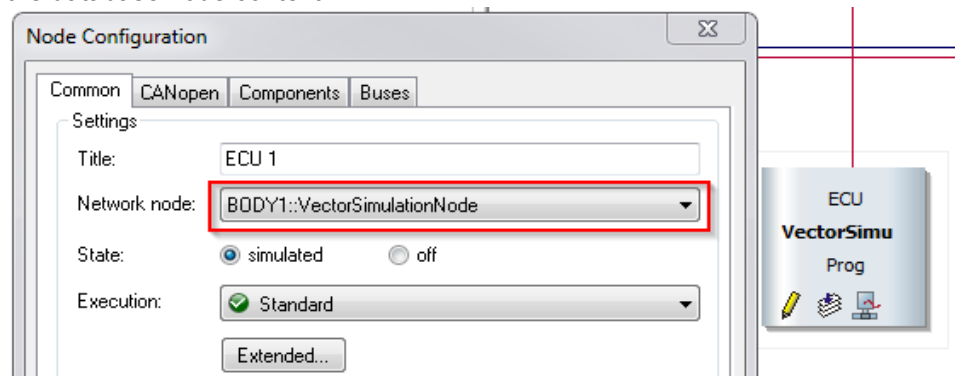
## Security profile mapping

CANoe needs a mapping between the network and the security profile to be applied on this channel. This mapping can be done in the **Security Configuration** dialog.

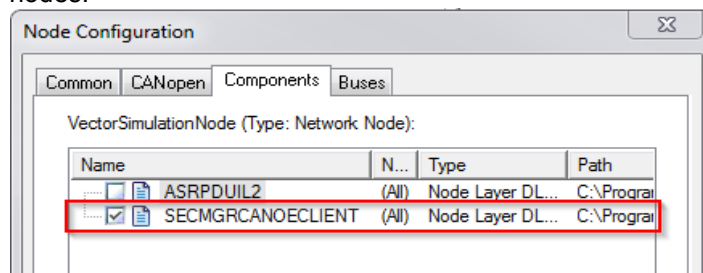
The dialog can be accessed from the **Simulation** ribbon tab.

## Simulation Setup

- > In the Simulation Setup all nodes to be simulated must be inserted and must have the database node context:



- > The node layer **SecMgrCANoeClient.dll** must be added to relevant simulation nodes:

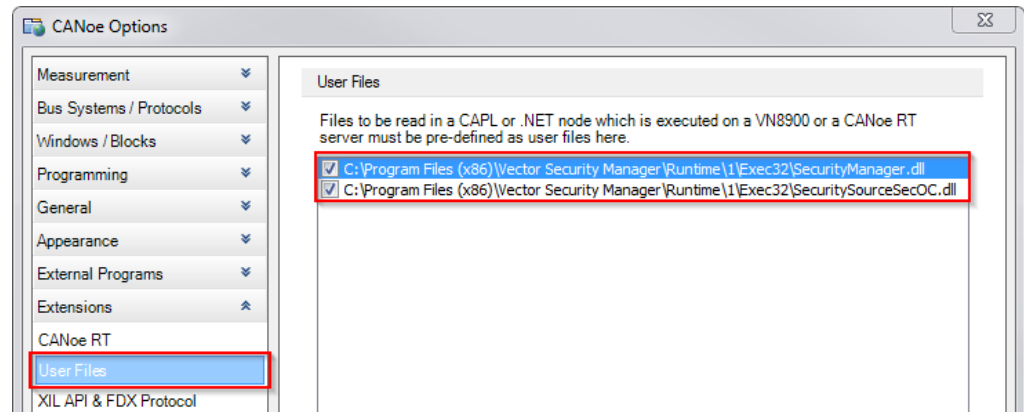


## 6.3 VN8912 / RT System Usage

### VN8912

To use the security extension on **VN8912** or a RT system some specific settings must be done.

The **Vector Security Manager** modules must be transferred to the interface. This exchange is not done automatically. In **CANoe** the modules must be configured to be transferred:



## 7 Index

### B

Broadcast Message Status .....23

### C

CMAC .....4

### D

Database Attributes .....16

### F

Freshness Simulation .....4

### K

Key Import .....14

### L

LocalSecurityActivateRxPDUs .....20

LocalSecurityActivateTxPDUs .....20

LocalSecurityCalculateAuthenticator .....18

LocalSecuritySetVerbosity .....20

LocalSecurityVerifyAuthenticationInformation ... 19

### N

Nodelayer ..... 18

### O

OnLocalSecurityPDUPreTx ..... 21

OnLocalSecurityPDUValidated ..... 21

OnLocalSecurityQueryLocalFreshness ..... 22

### P

PDU selector ..... 17

### S

Security Concept ..... 5

Security Manager Result ..... 23

### V

Validation Flags ..... 17

VN8912 ..... 27





## More Information

- > News
- > Products
- > Demo Software
- > Support
- > Training Classes
- > Addresses

**[www.vector.com](http://www.vector.com)**