BASIC GUIDE TO ENOCEAN SECURE COMMUNICATION

22 July 2026

Observe precautions! Electrostatic sensitive devices!

BASIC GUIDE TO ENOCEAN SECURE COMMUNICATION

REVISION HISTORY

The following major modifications and improvements have been made to this document:

Version	Author	Reviewer	Date	Major Changes
1.0	AZ	SA, JB		Original version

Published by EnOcean GmbH, St.-Martin-Straße 76, 81541 Munich, Germany
www.enocean.com, info@enocean.com, phone +49 (89) 6734 689-0

Important!
This information describes the type of component and shall not be considered as assured characteristics. No responsibility is assumed for possible omissions or inaccuracies. Circuitry and specifications are subject to change without notice. For the latest product specifications, refer to the EnOcean website: http://www.enocean.com.
As far as patents or other rights of third parties are concerned, liability is only assumed for modules, not for the described applications, processes and circuits.
EnOcean does not assume responsibility for use of modules described and limits its liability to the replacement of modules determined to be defective due to workmanship. Devices or systems containing RF components must meet the essential requirements of the local legal authorities.
The modules must not be used in any relation with equipment that supports, directly or indirectly, human health or life or with applications that can result in danger for people, animals or real value.
Components of the modules are considered and should be disposed of as hazardous waste. Local government regulations are to be observed.
Packing: Please use the recycling operators known to you.

TABLE OF CONTENTS

LIST OF FIGURES

1 Introduction

This document provides a basic overview of EnOcean security features and explains how to use them with selected products. It is intended to give fundamental guidance and provide a simple and clear overview.
Security is a key requirement in modern building automation systems. Wireless communication in such systems must protect data against unauthorized access, manipulation, and replay attacks. EnOcean radio networks support secure communication using standardized encryption and authentication mechanisms. These mechanisms ensure that only authorized devices can exchange data and that transmitted telegrams cannot be easily intercepted or modified. Secure EnOcean devices use established cryptographic methods such as AES-128 encryption and CMAC authentication, which are widely accepted industry standards

2 References

Familiarity with the following will be useful for this test procedure:

[1] For details, always refer to the latest version of the Security of EnOcean Radio Networks specification, available on the https://www.enocean-alliance.org/security

[2] DolphinView: https://www.enocean.com/en/product/dolphinview/

[3] USB300 https://www.enocean.com/en/product/usb-300/

[4] EnOcean Link https://www.enocean.com/en/product/enocean-link/

[5] USER MANUAL TCM 615 / TCM 615U / TCM 615J
https://www.enocean.com/wp-content/uploads/downloads-produkte/en/products/enocean_modules_902mhz/tcm-615/TCM-615-User-Manual.pdf

[6] EnOcean Serial Protocol Version 3 (ESP3)
https://www.enocean.com/wp-content/uploads/Knowledge-Base/EnOceanSerialProtocol3-1.pdf

[7] NFC configurator https://www.enocean.com/en/product/enocean-nfc-configurator

[8] EnOcean click board https://www.mikroe.com/enocean-5-click

3 Overview of EnOcean Security

The security mechanisms used in EnOcean systems are defined in detail by the EnOcean Alliance in the specification: Security of EnOcean Radio Networks Specification v3.02 https://www.enocean-alliance.org/security/

This specification describes how secure communication is implemented in EnOcean radio networks, including:

- Encryption of radio telegrams
- Authentication of transmitted data
- Protection against replay attacks using rolling codes (RLC)
- Secure teach-in procedures for device pairing

During a security teach-in process, transmitter and receiver exchange and store the following parameters:

- Device ID (unique identification)
- Security Key (used for encryption and authentication)
- Security Level Format (SLF), defining the applied security mode.

These mechanisms ensure secure and reliable communication for building automation, IoT sensors, and industrial applications.

4 Location of Decryption

In EnOcean systems, decryption of secure telegrams can take place in different locations depending on the system architecture.

4.1 Decryption inside the receiver (e.g. TCM615)

In this case, the receiver module handles all security-related tasks internally. This simplifies the system design, because the host application does not need to implement any security processing. The receiver module (e.g. TCM615) performs:

- Storage of security parameters in the internal security link table
- Decryption of incoming encrypted telegrams (AES-based)
- Verification of data integrity and authenticity (CMAC)
- Management of rolling code (RLC) synchronization

As a result, the host system receives already decoded and validated payload data.

Secure teach-in and secure link table management use ESP3. When secure communication is set up via the host interface, the host sends ESP3 commands to the receiver, for example to TCM615. These commands are used to create, read, update, or delete entries in the secure link table. The secure link table contains the parameters needed by the receiver to process secure telegrams, such as the security key, rolling code (RLC), security level format (SLF), and Teach-in Info (TII). Once these parameters are available, the TCM615 handles encryption, decryption, authentication, and RLC management internally. In the TCM615 concept, secure teach-in can be done in two ways:

- by receiving a secure teach-in telegram while Teach-in Mode is active, or
- by writing the required parameters directly from the host via ESP3 using CO_WR_SECUREDEVICEV2_ADD.

This ESP3-based method is always used for setting up the local parameters for transmission from TCM615 to a remote device. It may also be used for reception if the host already knows the remote device security data, for example from a QR code.

Main ESP3 commands for secure teach-in:

CO_WR_LEARNMODE – enable automatic secure teach-in
This command enables Teach-in Mode for a defined time. During this period, TCM615 can accept secure teach-in telegrams and store their parameters in the secure link table. Successful teach-in is reported by CO_EVENT_SECUREDEVICES, and the end of Teach-in Mode by CO_LRN_MODE_DISABLED. Unknown secure teach-in telegrams are ignored when Teach-in Mode is not active.

CO_WR_SECUREDEVICEV2_ADD – add a secure device to the link table
This command is used for host-controlled secure teach-in. It stores the security parameters of a device in the secure link table so that secure communication can be processed correctly. It writes all required security parameters into the secure link table: SLF, Device ID or EURID, Private Key, Rolling Code, and Teach-in Info.
It can be used for both received and transmitted secure telegrams, depending on the selected direction.

CO_WR_SECUREDEVICE_SENDTEACHIN – transmit a secure teach-in telegram
This command is used to send a secure teach-in telegram to another device. It is typically used after the required security parameters have already been stored in the secure link table.

CO_WR_SECUREDEVICE_DEL – delete a link table entry
This command removes a secure device from the secure link table. It can be used to delete stored security information before a new secure teach-in or system reconfiguration.

CO_RD_NUMSECUREDEVICES – read the number of stored devices
This command is used to check how many secure devices are currently stored in the secure link table. It helps to verify the current status of the secure device configuration.

CO_RD_SECUREDEVICE_BY_ID – find an entry by EURID
This command is used to find a secure device in the link table by its EURID. It helps identify whether a device is already stored and can be used before reading or deleting an entry.

CO_RD_SECUREDEVICEV2_BY_INDEX – read an entry by index
This command is used to read a secure device entry from the link table by its index. It is useful for checking stored security information during configuration or diagnostics.

These ESP3 commands are used to manage the secure link table in the receiver. They allow the host to enable automatic secure teach-in, add or delete secure devices, send secure teach-in telegrams, and read back stored security information such as EURID, SLF, RLC, and Teach-in Info. Once the parameters are available, e.g. TCM615 performs secure telegram processing internally.

4.2 Decryption in an external application

In this architecture, the receiver module, such as TCM615, mainly acts as a radio interface. It receives secure radio telegrams and forwards them to the host system without performing internal security processing. The host application handles secure teach-in, decryption, authentication, and rolling code management. This gives the customer application full control over secure communication.
This approach is typically used in gateway or controller applications where advanced host-side processing is required. For such use cases, the module can operate in Transparent Mode. In this mode, internal security functions such as secure teach-in, encryption, and decryption are disabled, while basic receive and transmit functions remain active. The host receives the secure telegrams and processes them with its own software.
A typical example is a PC-based analysis tool such as DolphinView, where secure telegrams are received through the gateway and decoded in the host application for analysis or debugging. The same principle can also be used in a customer controller, for example a microcontroller or PC application, which receives raw ESP3 telegrams and performs decryption using a security library or custom software.
For successful decoding, the host application must manage the relevant security data of the learned devices. This includes the security key, rolling code, and selected security settings. The host must identify the sender, match the telegram to the correct stored security information, and perform decryption and authentication checks.
The rolling code must be updated correctly after each valid secure telegram. Otherwise, later telegrams may no longer be accepted because sender and receiver are no longer synchronised. For reliable operation, the security data should be stored in non-volatile memory so that it is not lost during reset or power interruption.
A related implementation option is gateway-based decryption. In this case, the gateway performs the security processing internally instead of forwarding all security handling to the external host application. The gateway stores the security profiles and performs decryption, authentication, and rolling code handling. The host system mainly handles configuration, data interpretation, and application logic. This can reduce the software effort in the host application and is well suited for line-powered gateway or receiver solutions.
The key difference is where the security processing takes place. In external application decryption, the host application performs decryption and authentication. In gateway-based decryption, these tasks are handled inside the gateway.
The main advantage of external application decryption is flexibility. The host application can fully control how telegrams are interpreted, how security data is stored, and how secure communication is integrated into the system. This is useful for custom gateways, diagnostic tools, and controller platforms that need direct access to encoded telegram data.
The disadvantage is the higher implementation effort. The host application must provide complete security handling in software, including secure key storage, rolling code handling, teach-in processing, telegram authentication, and robust application logic.

Examples:

- DolphinView: encrypted telegrams are received and decoded externally for analysis.
- Customer controller, for example µC or PC: receives raw ESP3 telegrams and performs decryption using security libraries or a custom implementation.
- Decoding gateway: performs decryption and authentication internally, while the host application mainly handles configuration and application logic.

This approach provides high flexibility but requires:

- implementation of the security stack, unless handled by the gateway,
- secure storage of keys,
- handling of rolling code synchronisation,
- correct teach-in and authentication processing.

Examples of host-side implementation can be found in EnOcean Link [4]. In this architecture, encoded telegrams are forwarded by the gateway and processed in the customer application, making EnOcean Link a useful reference for external security handling.

4.3 Internal vs External Decryption (Comparison)

The following table summarizes the main differences between internal and external decryption approaches:

Feature	Internal Decryption (example TCM615)	External Decryption (Host / Gateway)
Implementation effort	Low	High
Security handling	Inside module	In application
Key storage	In receiver module	In host system
Flexibility	Limited	High
Required expertise	Low	High
Typical use case	Standard gateway, simple system	Advanced system, custom application

4.4 Summary

Internal decryption:
Recommended for most applications
Provides robust and validated implementation
Reduces development effort

External decryption:
Suitable for advanced or custom solutions
Allows full control over security processing
Requires careful implementation and validation

5 Example: PTM215 Secure Communication

This example shows a secure communication between a PTM215 pushbutton module and a receiver.

5.1 NFC Configuration

Using the EnOcean NFC Configurator tool [7], it is possible to access and read the PTM 215.

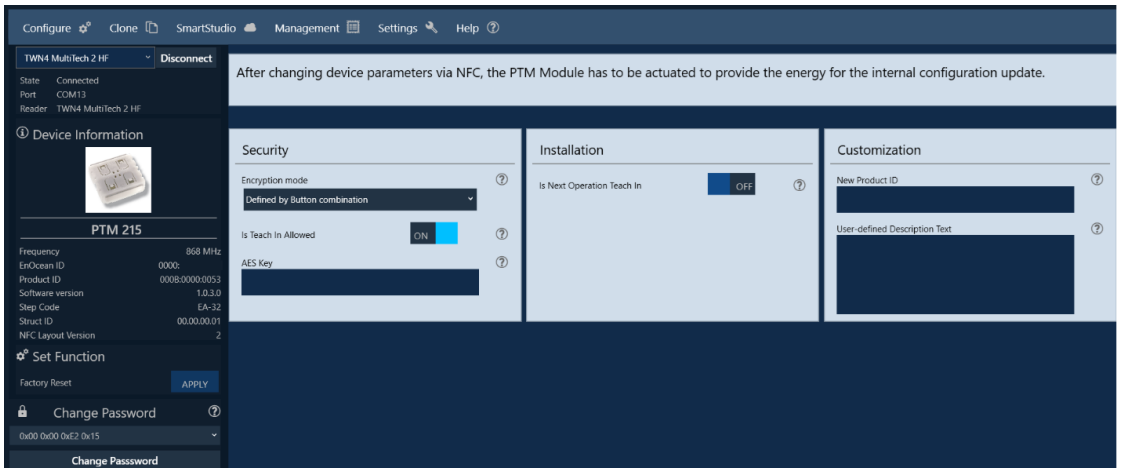


Figure 1 NFC configurator PTM215 access.

Enable security mode
Define security key
Set security level parameters

The user can then enable encrypted mode and generate a teach-in telegram.

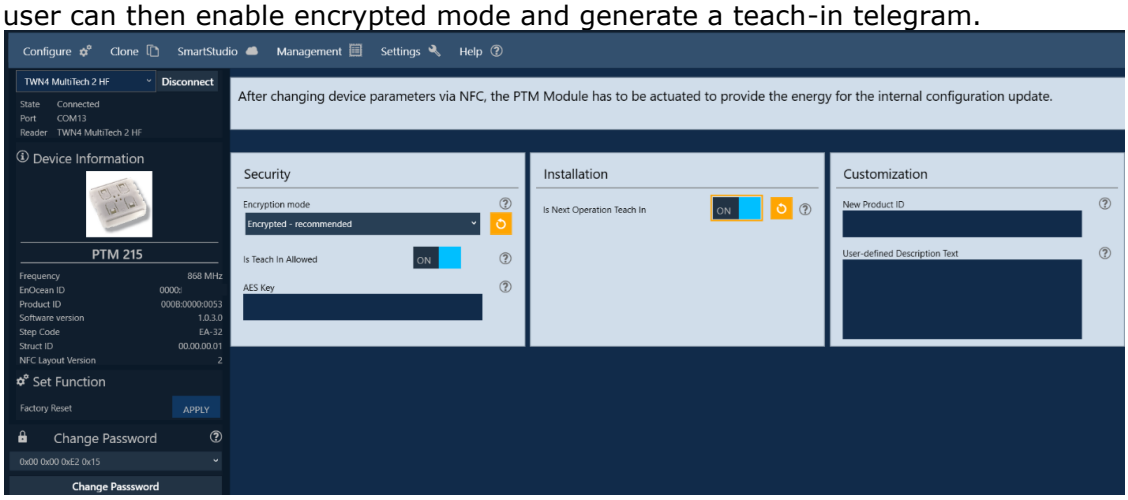


Figure 2 PTM215 security configuration in the NFC Configurator

By default, the AES key is not visible via NFC access. However, a user-defined key can be configured, and the teach-in procedure can be repeated for testing. In this example, a very

simple key was used to make the process easier to see and understand. In a real application, such a key should not be used.

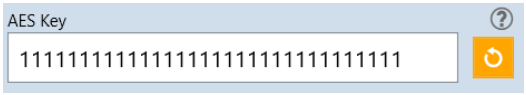


Figure 3 Simple AES key example for debugging purpose.

This allows easy commissioning without wired interfaces.

5.2 Transmission of encrypted telegrams

After configuration:

PTM215 transmits encrypted radio telegrams
Each telegram includes:

Encrypted payload
Rolling code (RLC)
Authentication data (CMAC)

To test the security concept, a receiver such as USB300 with DolphinView is required:
DolphinView [1][2]
USB300 [3]

The default, unencrypted RSP telegram received from the PTM 215 looks like this:

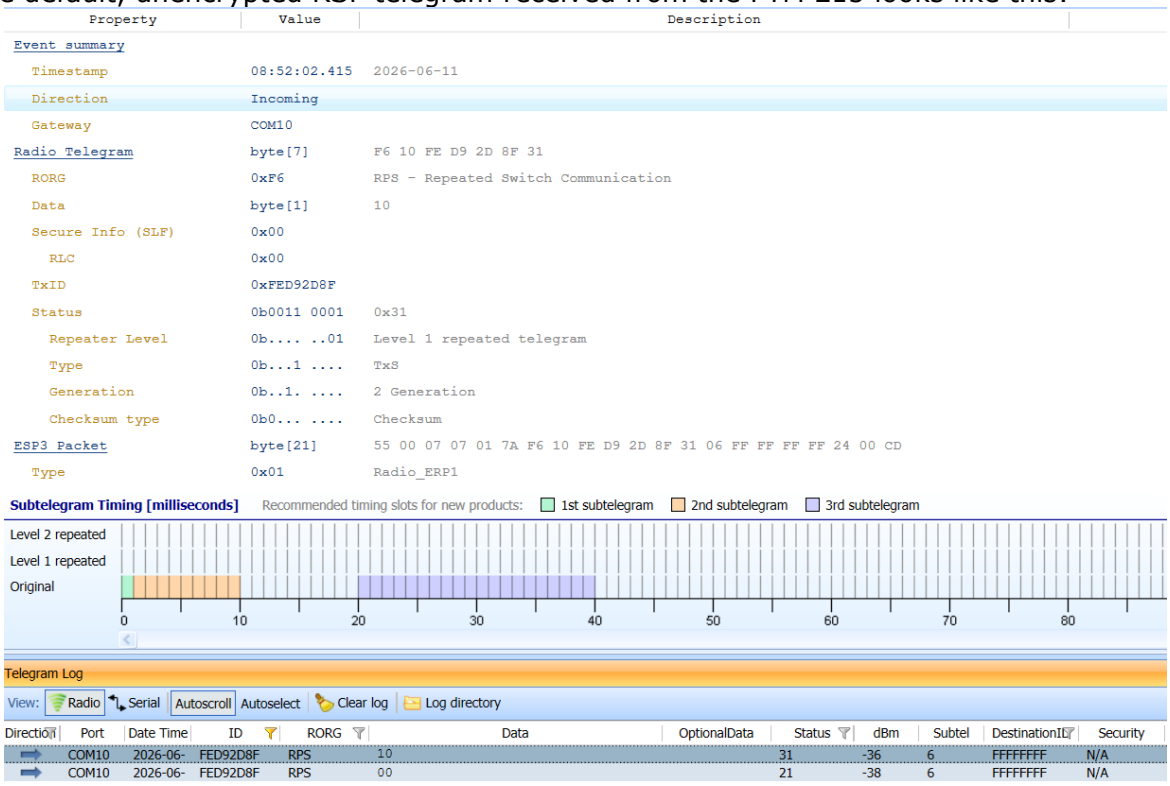


Figure 4 PTM215 unencrypted RSP telegram

Enabling encryption on the PTM215 as described in section NFC Configuration, triggers secure transmission, making the data unreadable in DolphinView.

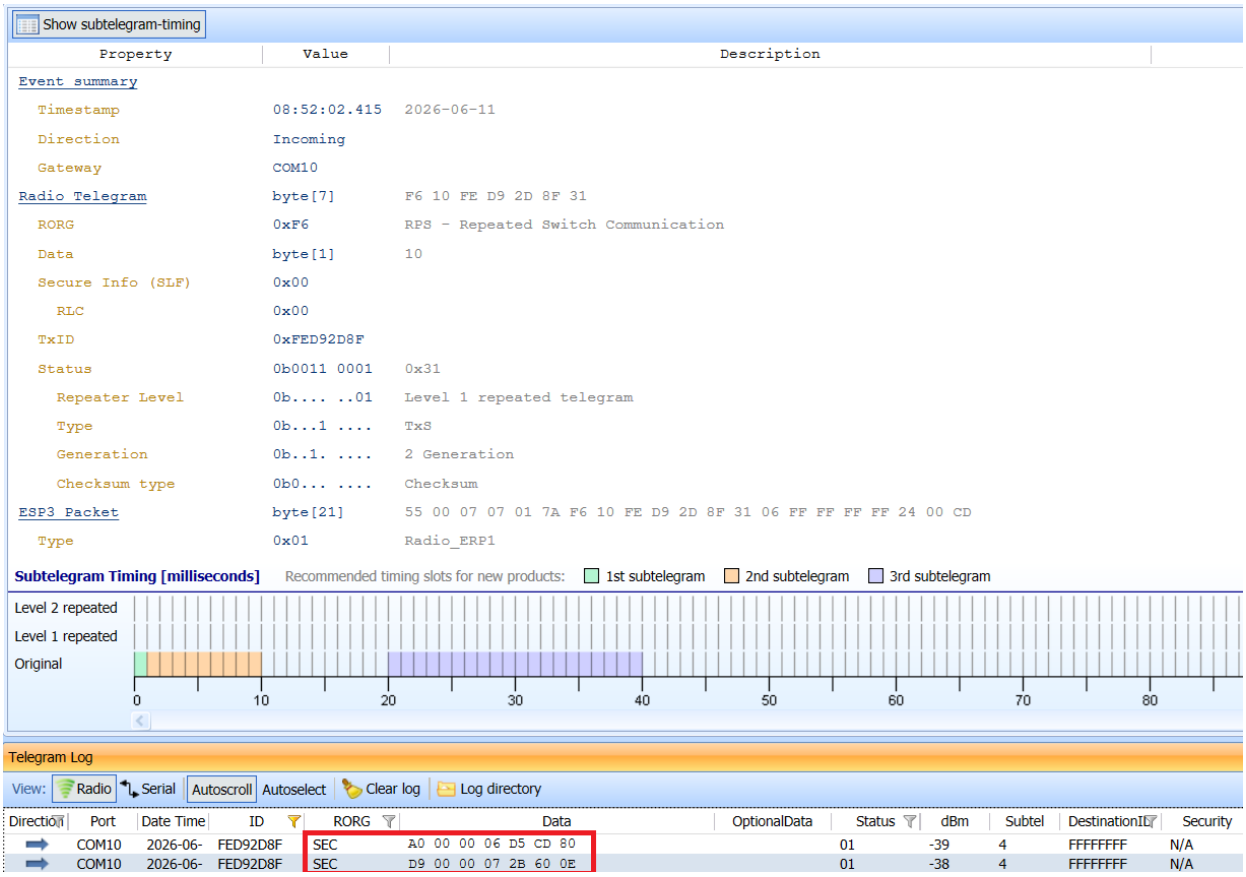


Figure 5 Encrypted, undecoded RPS telegram from PTM215

5.3 Decoding in DolphinView

Encrypted telegrams are received via USB gateway
DolphinView displays:

- raw telegram data
- the decrypted payload, if the required keys are available

This allows the security teach-in procedure to be observed and analysed.
Decryption takes place in the DolphinView application. The raw ESP3 serial data can still be observed and remains unchanged, while the telegram payload is decrypted and the decoded data is displayed separately.

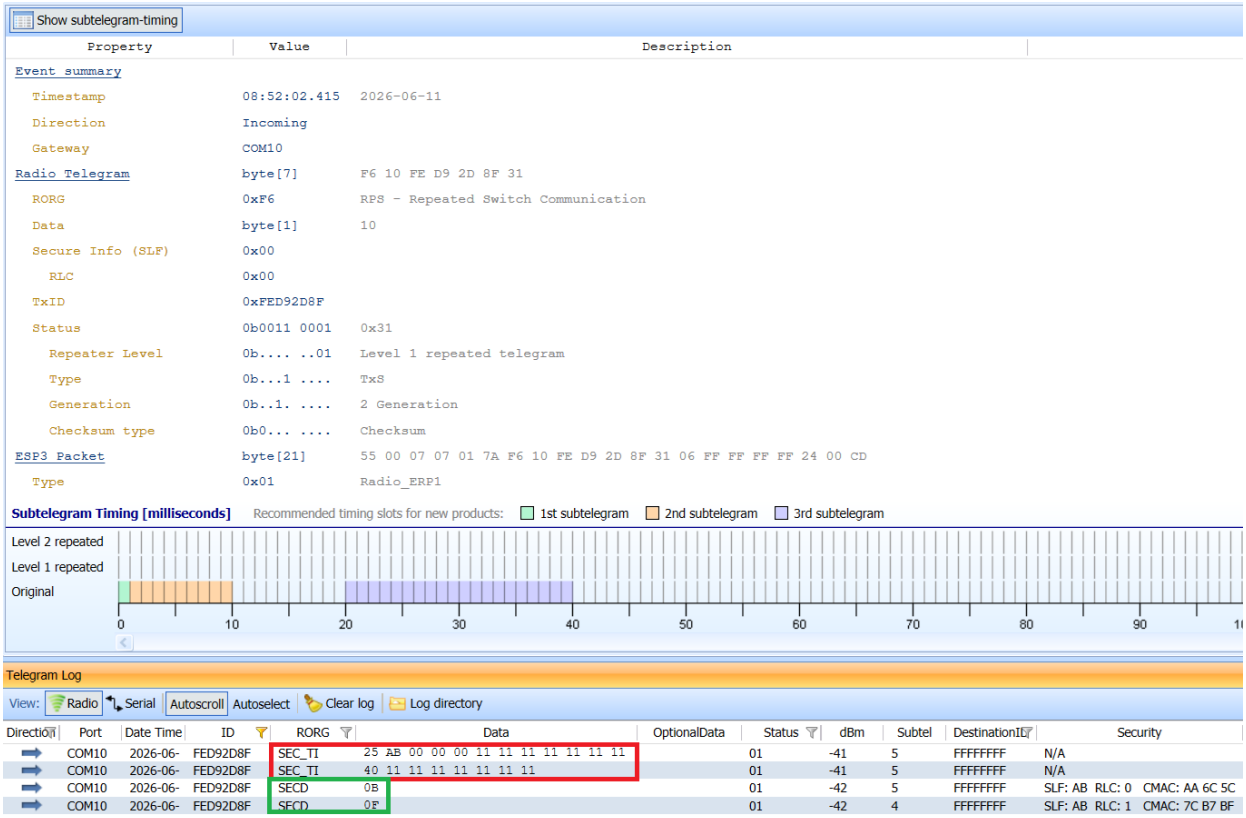


Figure 6 Secure teach-in telegram reception visualised in DolphinView

Helpful for debugging and validation, the security decoding feature in DolphinView is shown below. It allows the decoding process to be tested manually and helps the user better understand the basic security features.

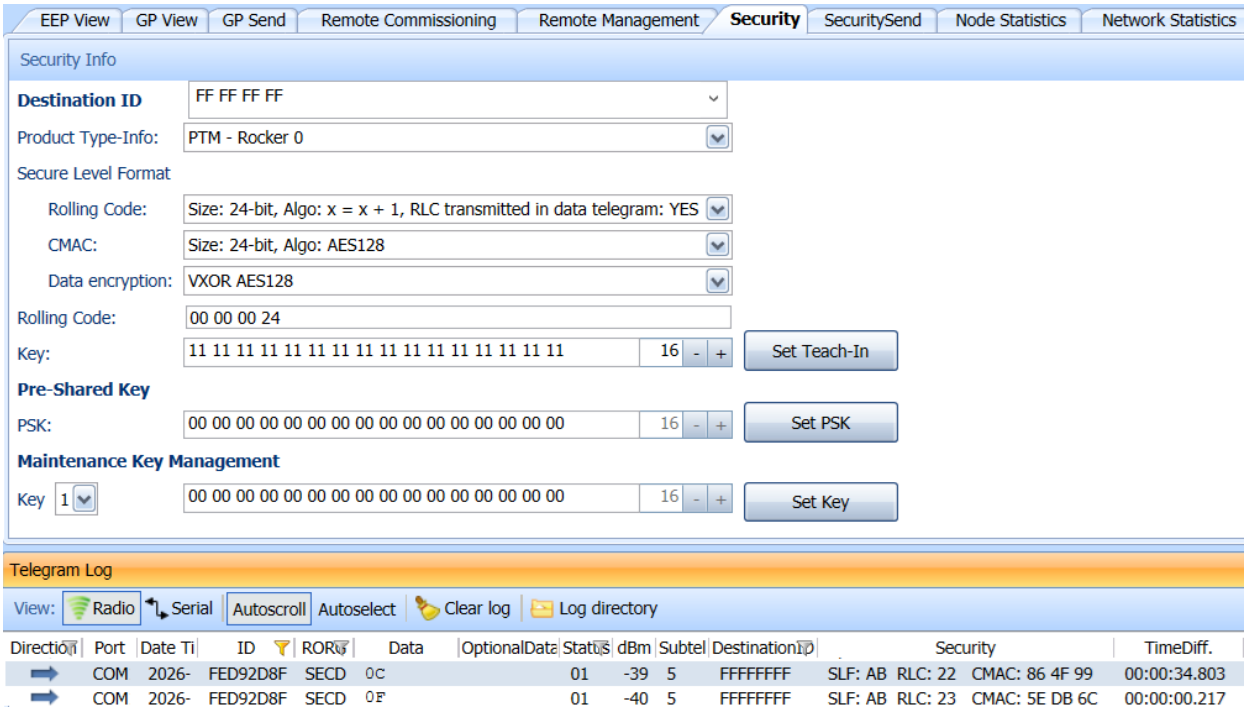


Figure 7 Security decoding feature in DolphinView

5.4 Decoding in TCM615

If security is handled internally:

TCM615 decrypts and validates telegrams
Application receives already decoded data
No additional processing required on host side
The module can process multiple secure devices without external support
Example, use CO_WR_SECUREDEVICEV2_ADD

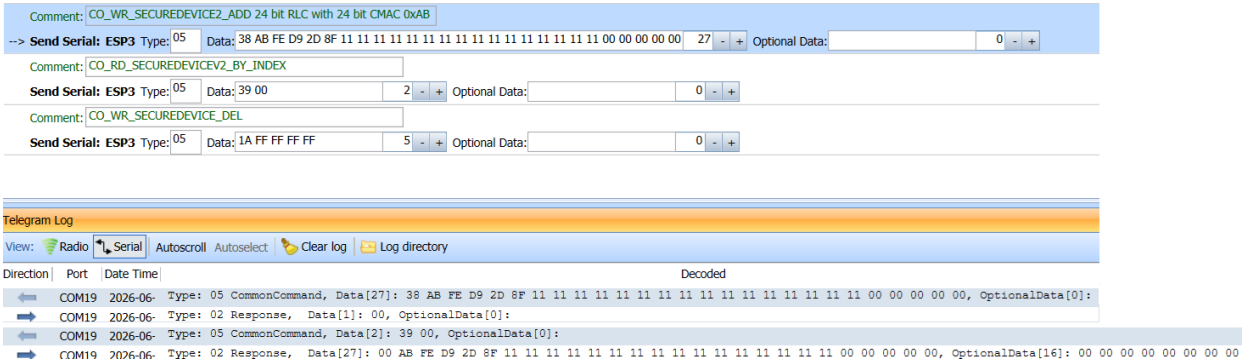


Figure 8 Example, use CO_WR_SECUREDEVICEV2_ADD handled security internally

Raw data CO_WR_SECUREDEVICEV2_ADD where important parameters are highlighted.

```
55 00 1B 00 05 55 38 AB FE D9 2D 8F 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00 00
00 C7
```

CO_RD_SECUREDEVICEV2_BY_INDEX

```
55 00 02 00 05 CD 39 00 44
55 00 1B 10 02 17 00 AB FE D9 2D 8F 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 18
```

`Security Level Format:` refer to e.g. TCM615 user manual 7.4.3 Security level format.
`Device ID`
`Private Key`
`Rolling Code`
`PSK`
`Teach-In info`: for details check Security of EnOcean Radio Networks, Teach-in Info format

As a result, already decrypted payload is received.

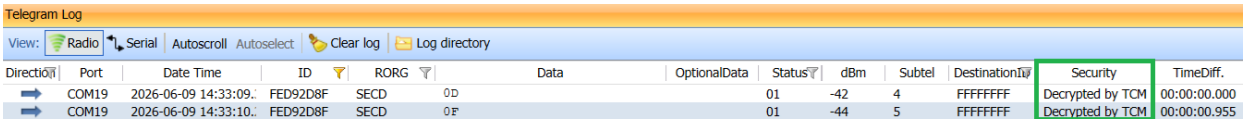


Figure 9 Example of decrypted payload received by the DolphinView.

6 Example: Secure communication from TCM615 to TCM615

This example illustrates secure communication between two TCM615 modules. In this setup, one TCM615 transmits secure telegrams and the other TCM615 receives and processes them. Secure communication is established by configuring the required security parameters and performing a secure teach-in procedure.

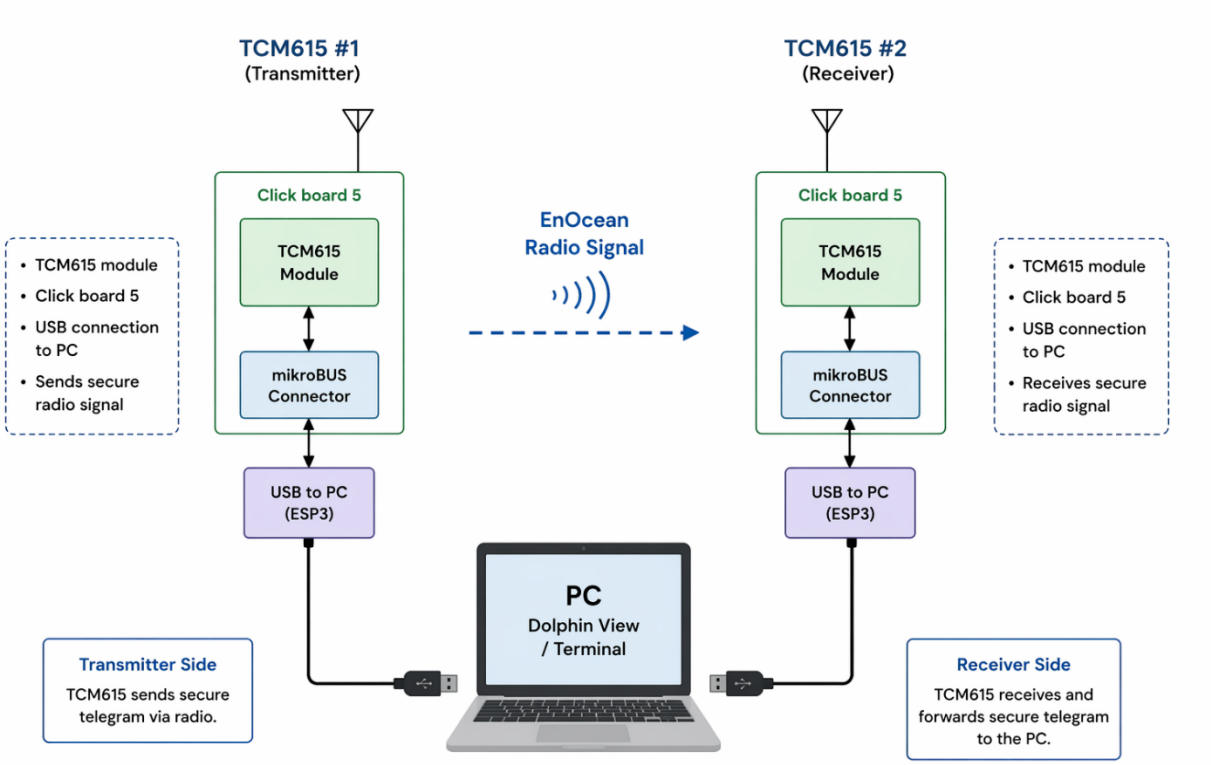
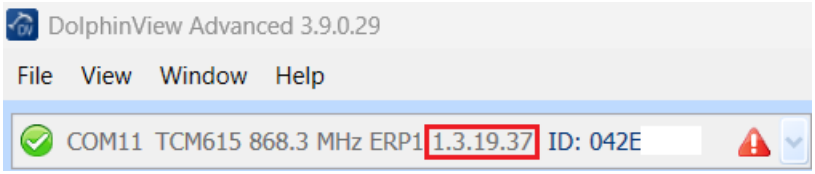


Figure 10 TCM615 test setup example [8]

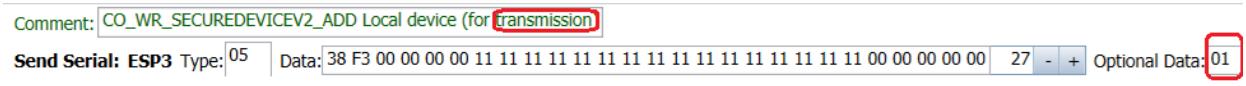
The examples were collected using the TCM615 DB-11 version, which can be identified in DolphinView. Older versions should not be used.



6.1 Configuration of secure communication

Before secure communication can start, the required security parameters must be stored in the secure link table. On the **receiver** side, this can be done by the host via the ESP3 interface using the command CO_WR_SECUREDEVICEV2_ADD. These stored parameters define how secure communication between the two devices is handled.

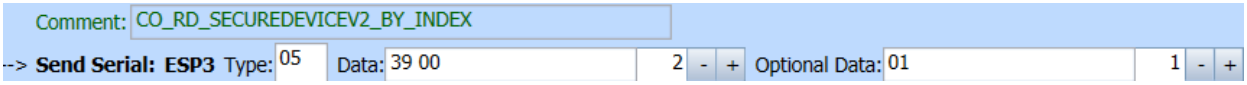
On the **transmitter** side, secure communication is also configured by writing the required parameters to the outbound secure link table using CO_WR_SECUREDEVICEV2_ADD. If a matching destination is present in the outbound secure link table, TCM615 automatically applies the configured security to transmitted telegrams. For secure broadcast transmission, the ID can be set to the module's own EURID or to 0x00000000.
Example ESP3 configuration message for the transmitter: CO_WR_SECUREDEVICEV2_ADD, with Security Level Format (SLF) 0xF3 as an example for 32-bit RLC and 32-bit CMAC.



55 00 1B 01 05 40 38 F3 00 00 00 00 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00 00 00 01 F0

Device ID: Broadcast telegram.
Private Key
Rolling Code
PSK
Teach-In info: for details check Security of EnOcean Radio Networks, Teach-in Info format
Direction

Double-check the transmitter configuration using the CO_RD_SECUREDEVICEV2_BY_INDEX command.

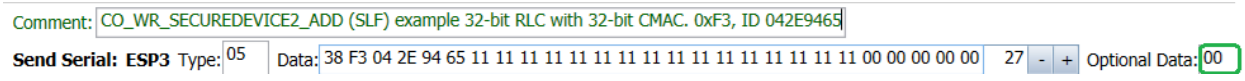


Expected response:
55 00 1B 10 02 17 00 F3 FF FF FF FF 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4E

6.2 Secure transmission and decoding in TCM615

After the required security parameters have been configured, the transmitting TCM615 can send a secure teach-in telegram to the receiving TCM615. This is typically done using the ESP3 command CO_WR_SECUREDEVICE_SENDTEACHIN. The secure teach-in telegram transfers the information required to establish secure communication between both devices. The receiving device must be in Teach-in Mode, which is started using the CO_WR_LEARNMODE command.

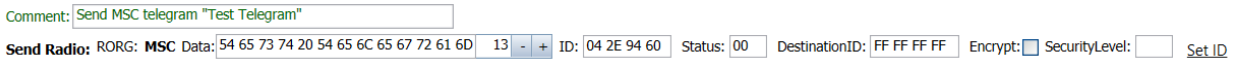
As an alternative, the **receiving** device can also be configured directly by the host using CO_WR_SECUREDEVICEV2_ADD, without relying on a received secure teach-in telegram.
In this example, the receiver link table is configured to expect the Chip ID of the transmitting device, like the procedure described in section Decoding in TCM615.



55 00 1B 01 05 40 38 F3 04 2E 94 65 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00 00 00 00 10

Device ID for known receiver ID.
Private Key
Rolling Code
Teach-In info
Direction

Send test telegram on the transmitter side:



Example of the secure telegram received by the second device before decoding; three sub-telegrams are shown in this example:

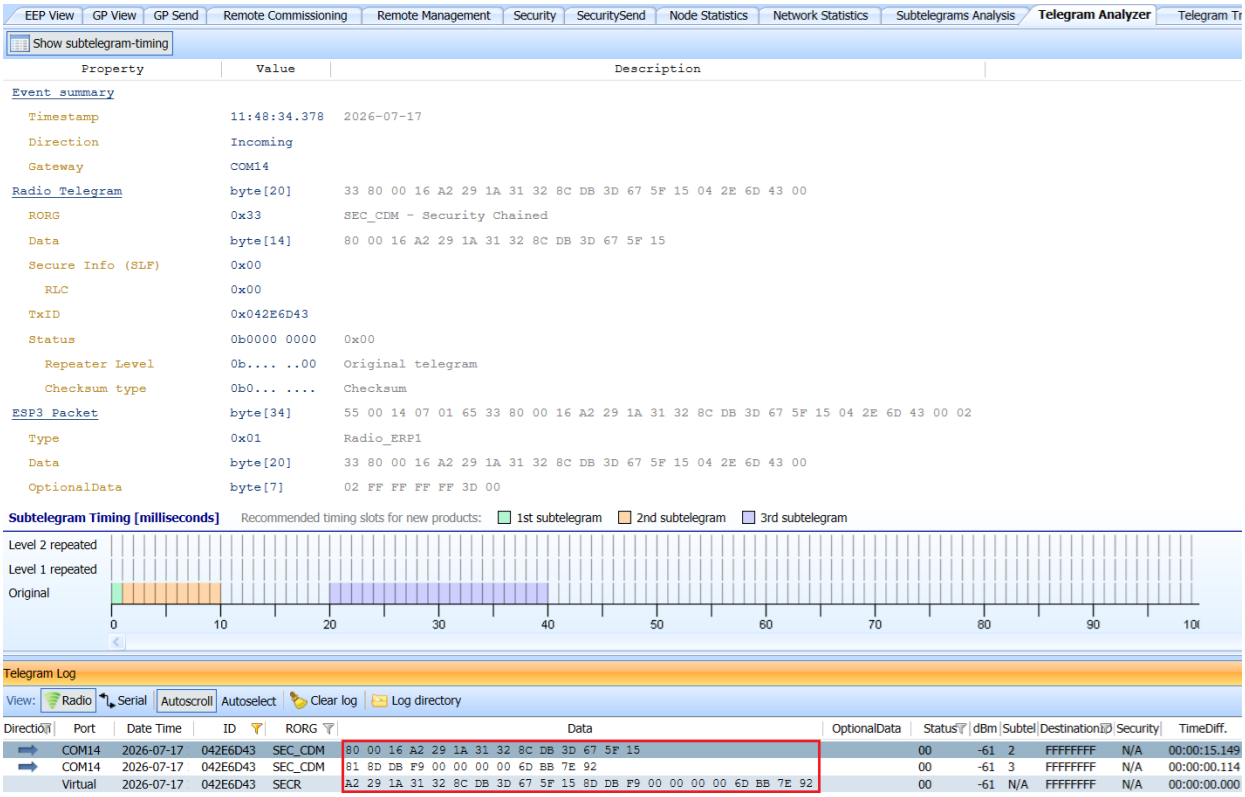


Figure 11 Telegram received by the second device before decoding

If on the receiver side, the command CO_WR_SECUREDEVICEV2_ADD is enabled.

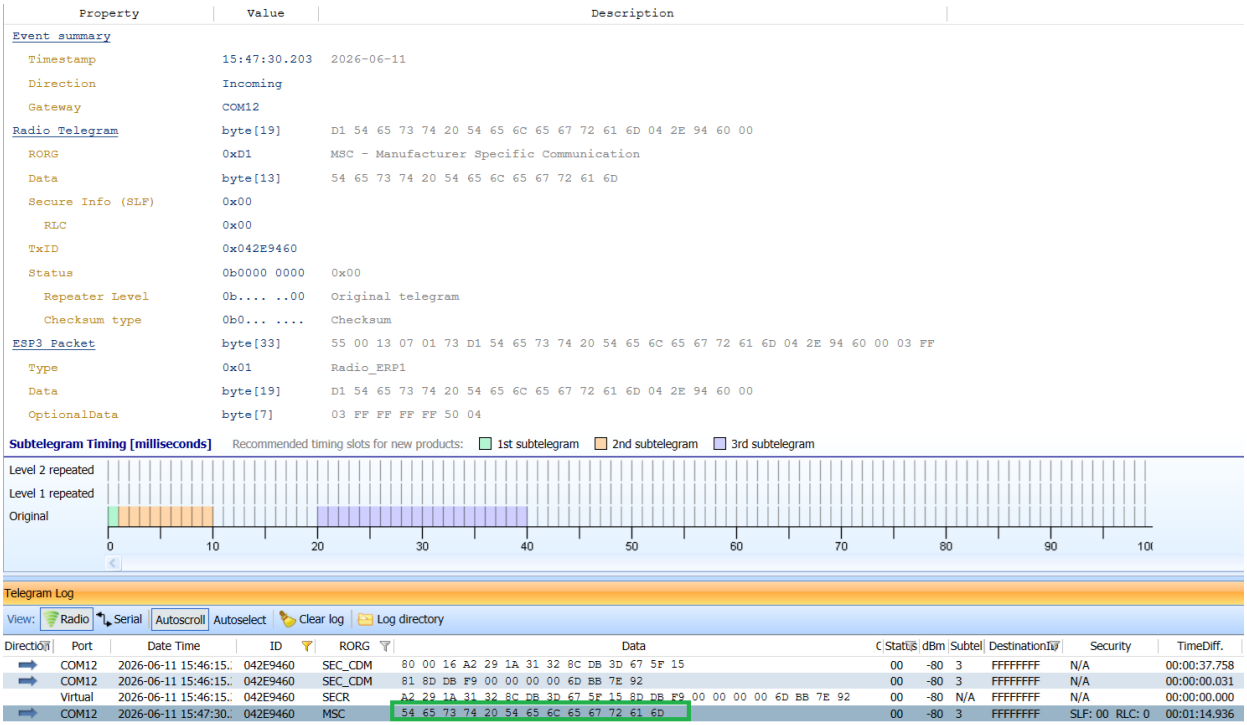


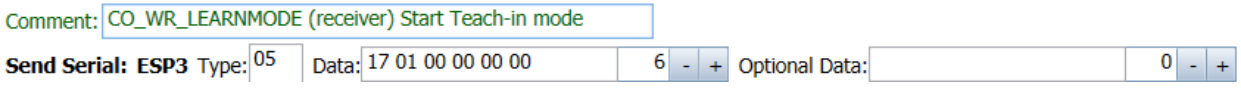
Figure 12 Telegram received by the second device after decoding

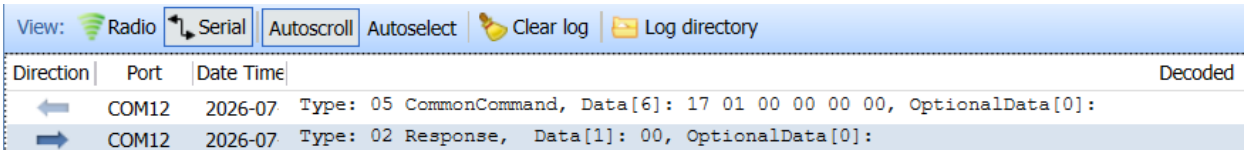
On the receiver side, the TCM615 can process secure telegrams internally if the required security parameters are available. In this case, the TCM615 decrypts the received telegram, verifies authentication and rolling code, and forwards the processed data to the host system. This simplifies the host application because no external decryption is required.

6.3 Example of decoding in TCM615 with Teach-in Mode enabled

To receive a secure teach-in telegram, the receiving TCM615 must first be put into Teach-in Mode using CO_WR_LEARNMODE. The transmitting TCM615 then starts the secure teach-in procedure with CO_WR_SECUREDEVICE_SENDTEACHIN. In this example, the receiver link table is initially empty. The link table entry is then created automatically when the receiver is in Teach-in Mode.

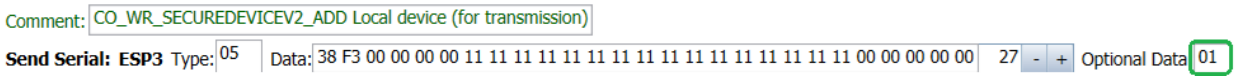
Put the receiver into Teach-in Mode
Start Teach-in Mode on the receiving TCM615 using CO_WR_LEARNMODE. In this mode, the receiver can accept a secure teach-in telegram and automatically create a new entry in the secure link table. If no custom timeout is given, the default Teach-in Mode duration is 60 seconds. The host is informed about the result by the corresponding event messages.





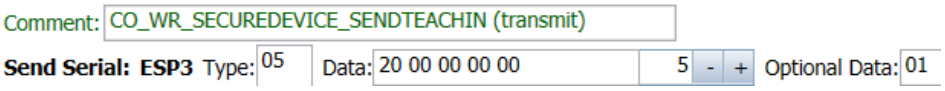
Configure the transmitter security entry

Before transmission, add the required security parameters to the transmitter link table using CO_WR_SECUREDEVICEV2_ADD. In this example, Security Level Format 0xF3 is used, representing 32-bit RLC and 32-bit CMAC. This entry defines how the transmitter will apply secure communication. As described in section 6.1.



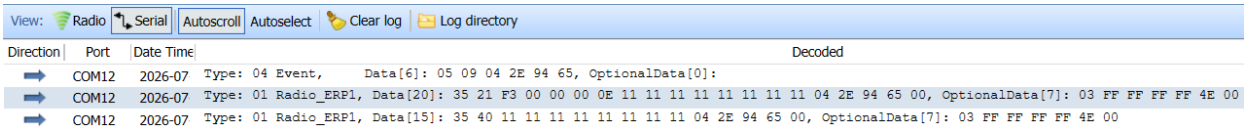
Start secure teach-in from the transmitter

On the transmitting TCM615, initiate CO_WR_SECUREDEVICE_SENDTEACHIN. The secure teach-in telegram is sent using the security settings already stored in the link table entry. In this example, no optional data is used.



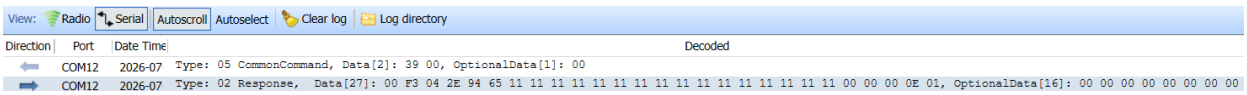
Receive the secure teach-in telegram

The receiving TCM615 receives the secure teach-in telegram while Teach-in Mode is active. This telegram contains the information needed to establish secure communication between the two devices.



Verify automatic link table creation

After successful reception, the first link table entry is added automatically on the receiver side. This can be checked using CO_RD_SECUREDEVICEV2_BY_INDEX, which reads back the stored entry from the secure link table.



7 Security Best Practices

To ensure a reliable and secure EnOcean system, the following best practices should be applied:

Use secure teach-in procedures
Always use secure teach-in instead of standard teach-in. Avoid operation with unsecured devices in production systems. Ensure that all devices are properly authenticated during commissioning.

Protect security keys
Store security keys in a secure location. Avoid exposing keys in firmware or debugging interfaces. Use secure memory (internal or external) when possible.

Ensure correct rolling code (RLC) handling.
Maintain synchronisation between transmitter and receiver. Avoid loss of telegrams during commissioning. Re-teach device if synchronisation is lost.

Rolling code protection prevents replay attacks and is an essential part of EnOcean security.

Select appropriate system architecture.
Use internal decryption (e.g. TCM615) for simple and robust systems. Use external decryption only if flexibility is required. Ensure the host system can securely handle keys and processing.

Follow secure system design principles.
Minimise access to security configuration interfaces. Use encryption not only on radio level but also in backend communication (e.g. IP, cloud). Keep firmware and software updated to address vulnerabilities.

8 Troubleshooting

This section describes common issues when working with EnOcean security and how to resolve them.

Device not accepted during teach-in
Possible causes:

Incorrect security key
Wrong teach-in method used
Device already learned in another mode
Solution:
Verify correct teach-in command (e.g. secure teach-in)
Reset device and repeat teach-in
Ensure matching security configuration

No valid data after reception
Possible causes:
Decryption not configured
Missing key in security link table
Wrong processing mode (internal vs external)
Solution:
Check if decryption takes place in correct component
Verify key storage and configuration
Use DolphinView for debugging

Rolling code (RLC) mismatch
Possible causes:
Loss of telegrams
Device reset
Receiver out of synchronisation
Solution:
Re-teach device
Ensure stable communication during operation
Avoid filtering of telegrams during commissioning

Telegram visible but not decoded
Possible causes:
Telegram is encrypted but not processed
Missing security configuration in receiver
Incorrect interpretation in external application
Solution:
Check if security mode is enabled
Verify whether key is available
Confirm decoding method (internal vs external)

DISCLAIMER

The information provided in this document describes typical features of the EnOcean radio transmission system and should not be misunderstood as specified operating characteristics. No liability is assumed for errors and / or omissions. We reserve the right to make changes without prior notice. For the latest documentation visit the EnOcean website at **www.enocean.com**.