APPLICATION NOTE 509

Overview of EnOcean Security features
EXPLANATION OF ENCEOAN SECURITY IN APPLICATIONS

Table of contents

1. INTRODUCTION

Modern technology has transformed all areas of our daily life, our way of working, travelling and communicating. Technology increasingly also enters the area of home automation, bringing greater convenience whilst saving costs.

Examples of such technologies include:

- Lighting control
- Heating control
- Automatic door openers (garage doors etc)
- Temperature sensors
- Illumination sensors
- Door / window contacts
- Occupancy sensors

In combining these diverse technologies we can for instance control room lighting depending on occupancy and ambient light or regulate the entire heating system depending on the outside temperature, time of the day and weather forecast.

It is this intelligent combination of different inputs (sensors) and outputs (actuators) by means of a smart central unit (gateway) that transforms our homes into smart homes. Using wireless communication between the different devices allows an easy upgrade of existing homes without requiring intrusive wiring work.

This approach however raises valid security concerns since – unlike with wired control systems - information and control commands now flow freely over the air and are subject to external monitoring and potentially even malicious external commands. Strong security mechanisms are therefore required to mitigate these threats in sensitive applications.

The EnOcean product portfolio uniquely meets the challenges of modern smart homes by providing an extensive portfolio of wireless products offering strong security and 100% maintenance free operation.

The following chapters will outline general security tasks in applications. It will provide an overlook on how security is applied in a bidirectional application and help the viewer to better understand how security is realized in EnOcean Radio networks.

This document provides additional information to the existing Security Specification of EnOcean radio networks. For detailed specification please refer to the Specification document [1].

1.1. Definitions

Term / Abbr.	Description
µC	Microcontroller (external)
AES	Advanced Encryption Standard
API	Application Programming Interface
APP	Application
ASK	Amplitude Shift Keying
CBC	Cipher Block Chaining
CMAC	Cipher Based Message Authentication Code
CRC	Cyclic Redundancy Codes
DATA	Payload of a radio telegram

Device	Customer end-device with an integrated EnOcean radio module
EEP	EnOcean Equipment Profile
EHW	Energy Harvested Wireless protocol
ERP	EnOcean Radio Protocol (ERP1 = Version 1, ERP2 = Version 2)
ESP3	EnOcean Serial Protocol V3
FSK	Frequency Shift Keying
Gateway	Module with a bidirectional serial communication connected to a HOST
GP	Generic Profiles
ID	Unique module identification number
KEY	Specific parameter used to encrypt / decrypt / transform DATA
MAC	Message Authentication Code
MSB	Most Significant Byte
PSK	Pre-shared Key
PTM	Pushbutton Transmitter Module
RLC	Rolling Code
R-ORG	Message parameter identifying the message type
SLF	Security Level Format specifying which security parameters are used
TXID	ID of a transmitter
VAES	Variable AES

1.2. References

[1] Security of EnOcean radio networks (System Specification) - http://www.enocean.com/en/security-specification/

[2] http://www.kotfu.net/2011/08/what-does-it-take-to-hack-aes/

[3] EEP Specification - http://www.enocean-alliance.org/eep/

[4] GP Specification - http://www.enocean-alliance.org/

[5] EnOcean Radio Protocol 1 - http://www.enocean.com/fileadmin/redaktion/pdf/tec_docs/EnOceanRadioProtocol.pdf

[6] Smart Acknowledge - http://www.enocean.com/fileadmin/redaktion/pdf/tec_docs/SmartAcknowledgement.pdf

[7] Remote Management - http://www.enocean.com/fileadmin/redaktion/pdf/tec_docs/RemoteManagement.pdf

1.3. Revision History

No	Major Changes
1.0.	First version

2. FUNDAMENTALS OF WIRELESS NETWORK SECURITY - OVERVIEW

As soon as wireless components control and monitor important aspects of our lives, security often becomes a major concern. Unlike with traditional "wired" networks, information now flows freely over the air and is not anymore restricted to the confines of the home.

This raises two significant concerns:

- Unauthorized interception (reception and correct interpretation) of transmitted data – "**Eavesdropping"**
- Unauthorized transmission of correct control commands – including "**Replay Attacks"**

Somewhat loosely speaking, the goal of security has to prevent an unauthorized person (often referred to as an "*Attacker*") both from learning about the current state of a system and from actively changing it. These two attacking scenarios and the countermeasures:

- content protection
- content authentication
- dynamic content modification

are addressed by the existing Security specification [1]. In the next chapters we will describe the character of these attacking scenarios in detail and the correct countermeasure.

2.1. Content protection addressing Eavesdropping

The goal of content protection is to prevent unauthorized receivers from correctly interpreting the content of a message. In the context of home automation this is for instance important for data allowing an unauthorized person to determine if somebody is at home or not. Simple scenario is shown in Figure 1.

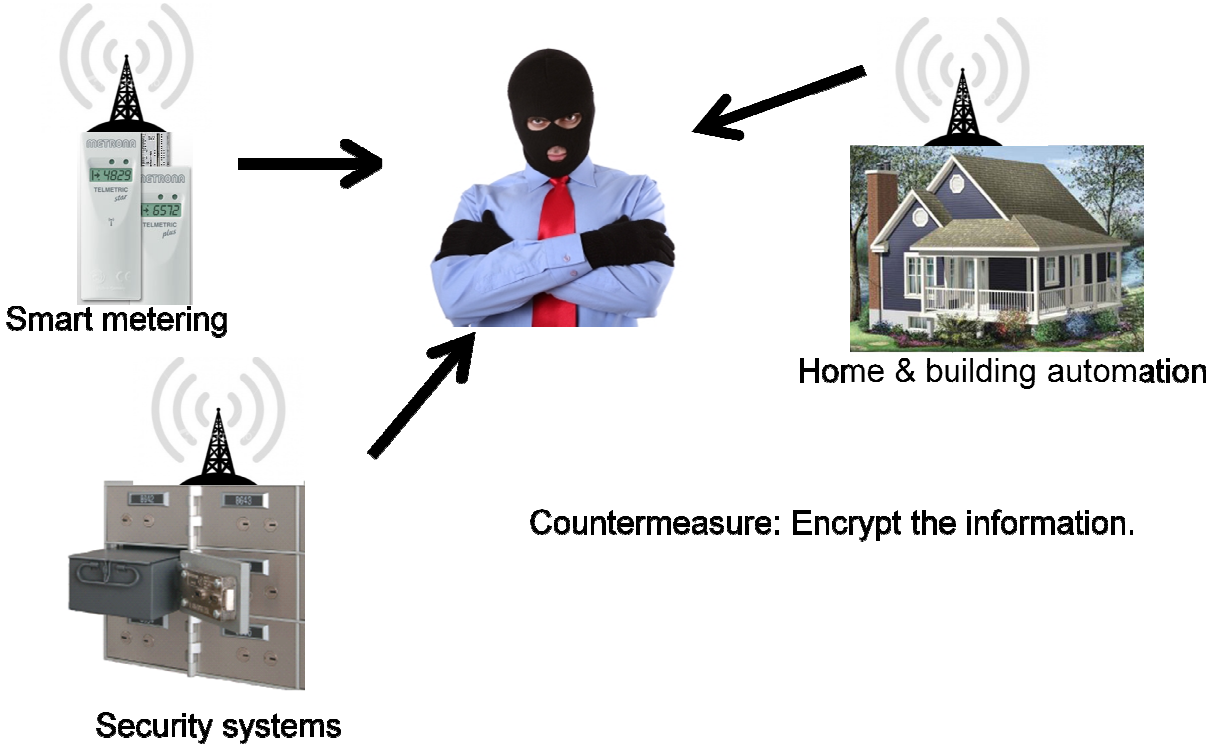

Figure 1 Eavesdropping scenario

Examples for sensitive data here include status information from occupancy sensors, actions of access control devices or devices showing the state of a building (vacation mode).

It is important to protect both information about the content of a message and the type of the message itself.

Taking the example of a data telegram originating from a occupancy sensor, it is not sufficient to merely protect the information about the command itself (occupied / unoccupied) but also the information that this command originated from a occupancy sensor. Otherwise an attacker could determine if a home is occupied or not based on the presence or absence of commands originating from occupancy sensors.

Content protection is achieved by means of *encrypting* the original (plain text) data with a *key* thus transforming it into encrypted, unreadable data. Only when the correct key is known it is possible to transform – *decrypt* - the encrypted data into readable data again.

Figure 2 below shows the concept of secure data transmission from a high-level perspective.

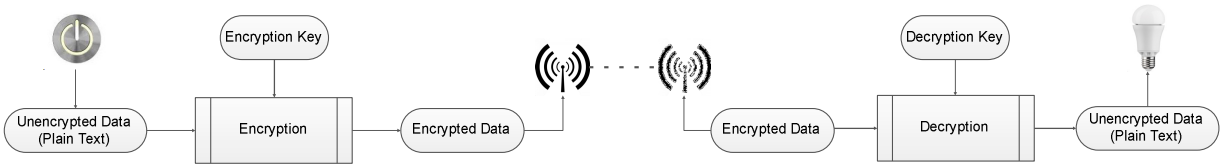

Figure 2 Secure data transmission

2.2. Content authentication and dynamic key modification addressing Replay attacks

2.2.1. Content authentication

The goal of content authentication is to prevent unauthorized transmitters to transmit apparently valid commands causing the receiver to perform unauthorized actions.

Content authentication works by creating a *message signature* (often referred to as *Message Authentication Code* or *MAC* in short) based on the content of the telegram and the secret key. Essentially, the telegram data is transformed via a defined algorithm using the secret key into a unique, fixed size signature. The signature is unique for every message and therefore can be used for content authentication.

Conceptually the correspondence between message and signature is similar to the one between a person and a finger print:

- Each person has a unique fingerprint
- Based on a given person one can easily determine his or her fingerprint
- Based on a given fingerprint one can easily check if it originated from a given person
- Based on the fingerprint one cannot determine any other properties of the person (height, gender, hair colour, eye colour, etc).

So both message signature and fingerprint uniquely identify their owners without revealing any additional properties.

For an ideal signature algorithm, the likelihood of two different messages creating the same message signature would be inversely proportional to the signature size, so for instance for 24 Bit signatures the likelihood would be one in 16 million.

For message authentication purposes, the message signature (MAC) is typically appended to the message itself and transmitted together with it.

When the receiver receives such a message, it will itself calculate the MAC with the defined algorithm based on the secret key and the content of the received message. The receiver then compares the MAC it calculated with the MAC it received as part of the message.

If both MAC are the same then the receiver can establish two important facts:

- The message originates from an owner of the secret key
- The content of the message has not been modified

Figure 3 below illustrates the content authorization via a MAC signature.

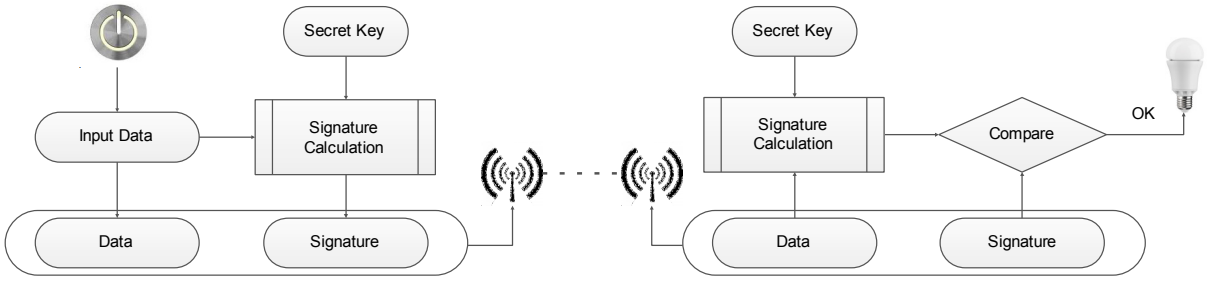

Figure 3 MAC signature-based message authentication

2.2.2. Dynamic key modification

One fundamental problem with both content protection and content authentication is that using the same input data (plain text) with the same key always yields the same encrypted data and same signature.

This enables attacks based on monitoring previous system behaviour. If an attacker has observed that a certain data telegram results in a certain light being turned on then he could use this information to identify - or even actively send - similar telegrams in the future. This type of attack is often called *Replay Attack* since it works by reusing (replaying) previously used data telegrams. An example scenario is listed in figure below.

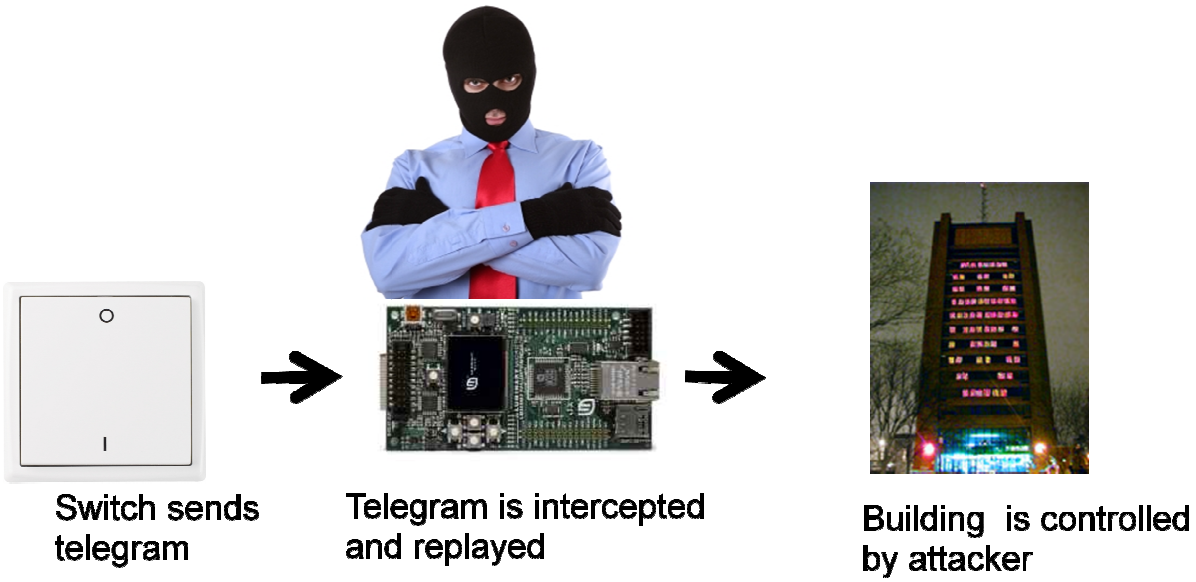

Figure 4 Replay Attack scenario

In order to prevent this type of attack, either the data or the key must continuously change to ensure that identical input data does not create identical encrypted radio telegrams.

The mechanism used by the transmitter to change data or key must be known to the receiver in order to correctly decrypt and authenticate received data telegrams. One common approach is to use the secret key together with an incrementing counter (e.g. rolling code) to generate a dynamic key.

For this scheme it is important that the counter on the transmitter and on the receiver side remain synchronized, i.e. will always have the same values. Both counters will therefore have to start based on the same value and both have to be incremented whenever a data telegram is exchanged.

In order to guard against the case of telegrams being lost (not correctly received), the receiver needs to check if the received telegram has been decrypted correctly based on the counter value being used.

This check can be done for instance via the message signature (MAC) or by other message integrity checks, e.g. based on *Cyclic Redundancy Codes* (or *CRC* in short) or more simply via Parity Bits. If a message is detected as non-valid based on the current counter value, the receiver can retry using the next counter value and so on. This results in following definition:

- We define that dynamic key modification must be used also with message authentication. Practically it means that, whenever the rolling code is used the MAC muss be used too.

Typically, a maximum number of future counter values to be tried will be defined. This parameter is often referred to as the *Rolling Code Window Size*. If message decryption based on a future counter value is successful then the counter will be set to this value, thereby re-synchronizing the transmitter and receiver counters.

Figure 5 below shows this mechanism.

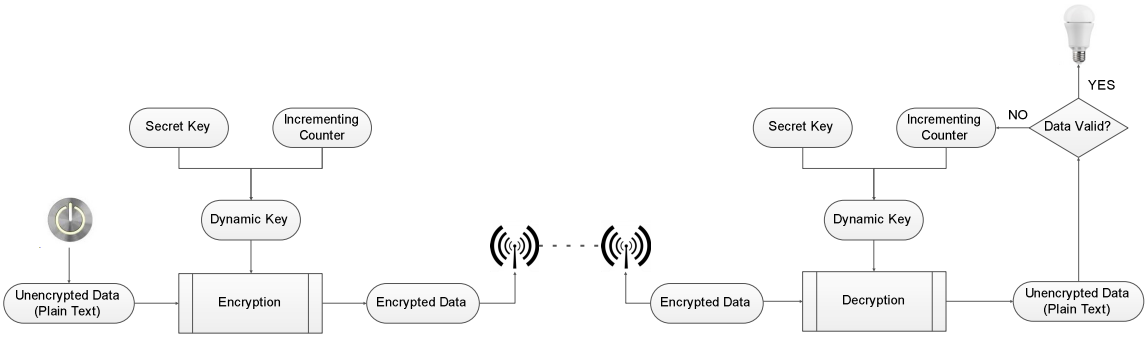

Figure 5 Secure transmission based on dynamic keys

2.3. Typical use cases

In the previous chapters we have outlined the fundamental techniques of message encryption, message authentication with dynamic key modification.

Message authentication can be used without dynamic key change, but as pointed out earlier, this bears risk of replay attacks.

Different use cases will require different combinations of these (or other) techniques to be employed. For instance:

- In many systems there is information that is not security critical. For instance, the transmission of the outside temperature as measured by a temperature sensor will usually not need to be secured, authenticated or prevented from being reused.
- Information allowing determining critical parameters (e.g. the thermostat state) could be encrypted to prevent it from being correctly interpreted by unauthorized persons but does not necessarily need to be authenticated or protected from being sent again.
- Commands causing actions such as a door being opened need to be both authenticated (to ensure that they originate from an authorized source) and prevented against reuse (replay).
- Finally, very sensitive information - such as metering information used for billing purposes - needs to be both encrypted, authenticated and being prevented against reuse.

Many other use cases are possible as well. Table 2 below summarizes these examples and their security requirements.

Typical Use Cases	Sender Identification (Unique Sender ID)	Eaves Dropping (Data Encryption)	Replay Attack (CMAC incl. Rolling Code)	Telegram Structure
Standard Wall Switch or Sensor	✔	–	–	ORG | Data | ID.ST.CKS
Automated Meter Reading	✔	✔	–	ORG-S | Data | ID.ST.CKS
Locker Control, Wall Switch, Simple Car Fob (Data length: 1–15 Byte)	✔	✔	✔	ORG-S | Data | RC | CMAC | ID.ST.CKS
Advanced Smart Metering (Data length 16+ Byte)	✔	✔	✔	ORG-CH | ORG-S | Data | ID.ST.CKS ORG-CH | Data | CMAC | ID.ST.CKS

Legend:

CMAC	Cipher Message Authentification Code (incl. Rolling Code)	ORG	R-ORG EnOcean Telegram	ID.ST.CKS	Transmitter Identifier, Statusbyte, Checksum
encrypted	Field is encrypted AES 128	ORG-S	R-ORG Security Telegram	Data	Telegram Payload
		ORG-CH	R-ORG Chained Telegram		

Table 1 Typical security use cases

Please note that Sender Identification as shown in Table 1 is still present also when no advanced security measures are added. Sender Identification trough unique Sender ID is present in EnOcean Radio networks from the very beginning. This protection ensures that an EnOcean based device can transmit telegrams only with its in production given chip ID. The receiver can rely on it, since EnOcean enforced it within its products. With standardizing the EnOcean Radio protocol as ISO/IEC 14543-3-10 protocol this features is not longer sufficient as security measure since other radio manufactures than EnOcean can produce modules / devices.

3. IMPLEMENTATION OF SECURITY FEATURES IN ENOCEAN NETWORKS

In chapter 2 we summarized the security features which were introduced to EnOcean Radio networks. Please see EnOcean Security Specification for Radio networks [1] for detailed explanation on how the security features are realized within the telegram structure. In this chapter we will focus on the features realisation from a more abstract standing point giving the reader a more detailed explanation.

In chapter 2 we defined three new advanced security features. Please see this summary on the implementation in EnOcean Radio network.

1. Feature: content protection – *implemented with*: data **encryption**
2. Feature: content authentication – *implemented with*: **CMAC** – cipher based message authentication
3. Feature: dynamic content modification – *implemented with*: **RLC –** rolling code

3.1. Used algorithms

Encryption of data can be realized in many different ways considering the benefits and given conditions. In an EnOcean network consisting also from autarkic devices, we use for encryption these algorithms:

- VAES – encryption of variable length of plain text data (e.g. 1 b, 2 b, 3 b, 4 b, ...)
- AES-CBC – encryption of 16 bytes chunks of plain text data (e.g. 16 b, 32 b, ...)

The CMAC algorithm and both data encryption algorithms (VAES, AES-CBC) exploit the standardized AES 128 algorithm [1], please see chapter 4.3.

In total we are focusing on these security features implementations:

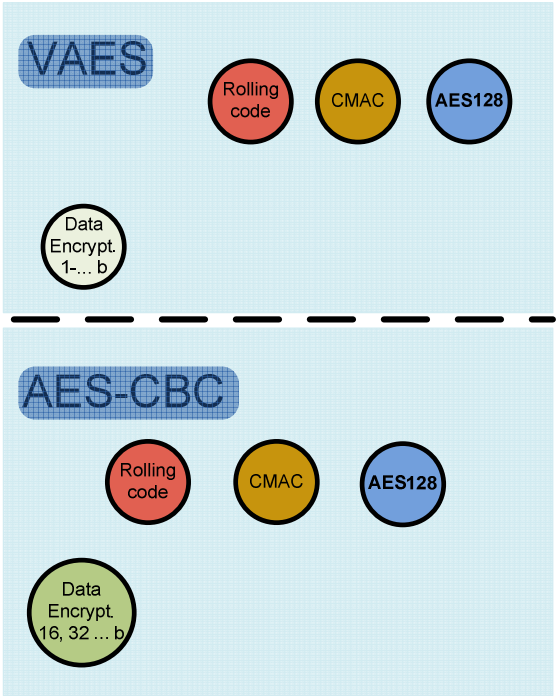

Figure 6 Security features implementation

What combination of the security features is used is defined by the Security Level Format – SLF [1]. Not every combination is allowed. Here is a more detailed explanation of every feature with focus on meaningful combinations.

3.1.1. Dependencies between algorithms

The RLC can be carried within the encrypted message or not. This means it can be **explicitly** part of the radio telegram and directly used in the validation algorithm or be **implicit** and thus not part of the radio telegram. Exact meaning of these two approaches will be explained in chapter.....

The security features implementations complement to each other and some prerequisite the usage of another. This is valid for RLC and CMAC usage.

Please see a summary of the used algorithms and their dependencies in Figure 7.

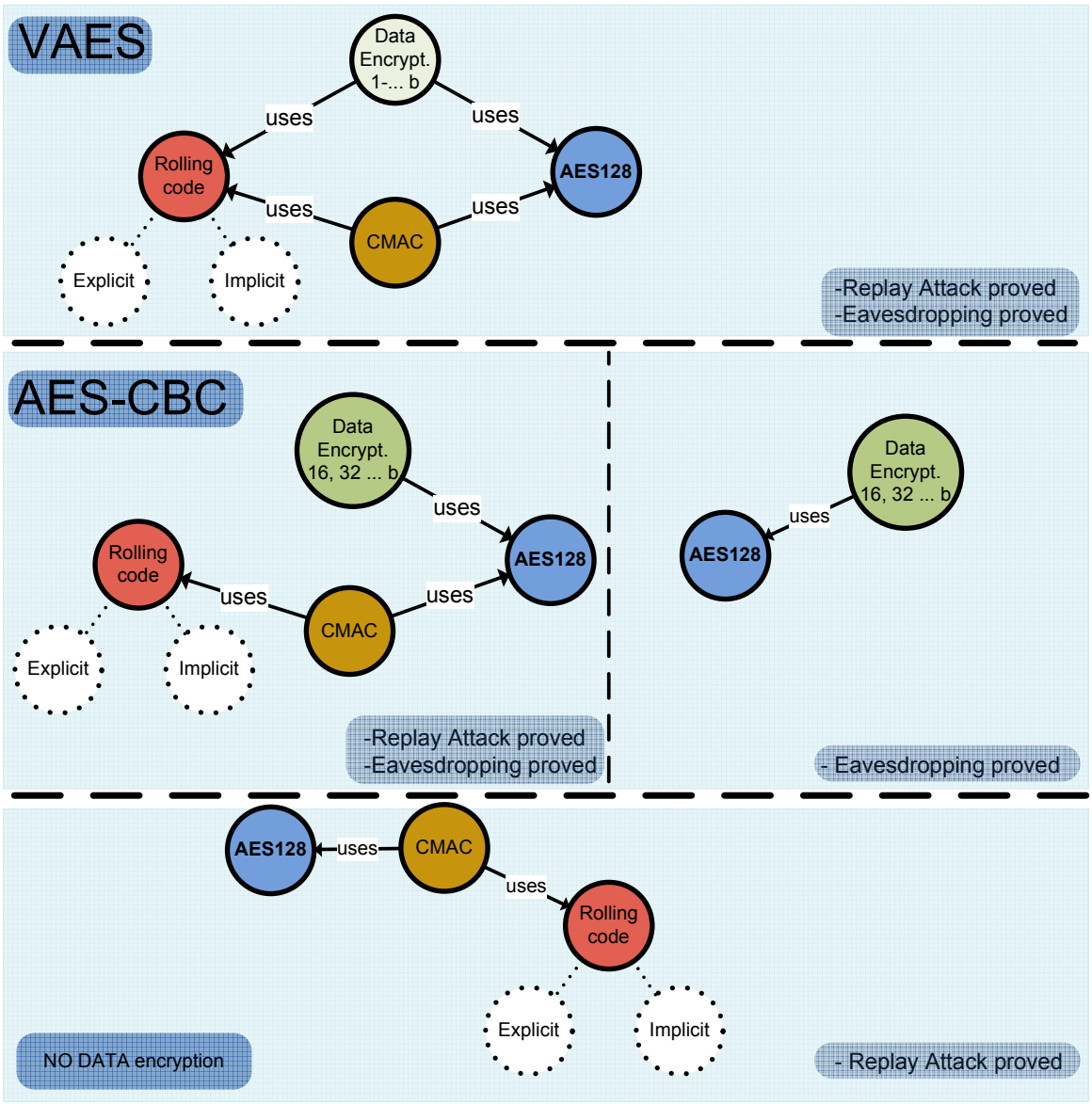

Figure 7 Security Algorithm overview

Resulting from the implementation following dependencies are defined:

- When using VAES then RLC (explicit / implicit) must be used
- When using CMAC then RLC (explicit / implicit) must be used

Resulting from Chapter 2.2.2 definition also this dependency is defined:

- When using RLC (implicit) then CMAC must be used

Content authentication – CMAC usage can be used also without data encryption. But since CMAC uses AES 128, there is only a very little step to exploit the AES 128 also for data encryption too. Therefore we recommend using data encryption, also when there is no immediate use case for it.

The defined security features as shown in Figure 6 are visualised and explained on examples of receiver and transmitter tasks in the following chapters.

4. SECURITY TASKS

4.1. Tasks in Transmitter

The tasks of a transmitter are as follows:

- Define how and what security features are used (e.g. what kind of encryption). This practically means to define the Security Level Format – SLF [1].
- Inform the receiver about the SLF used, security KEY and initial RLC. This practically means transmitting a security teach in message [1].
- Send the data communication with the defined security features.
- Store and update the RLC

It is important to note, that once a transmitter uses security features in communications it cannot fallback to unprotected communication. By doing this the transmitter would open the back door for an intruder, because the receiver cannot authenticate the origin of the unprotected communication.

4.2. Tasks in Receiver

The security tasks of a receiver are as follows:

- Teach in security device and store their information. Practically it means to receive the teach-in message and parse the included information. This information must be stored in non-volatile memory, because the teach-in is not being repeated.
- On reception of encrypted data telegrams decode them. Practically it means to recall the device security information (e.g. SLF, KEY and RLC) and then perform decoding operation.
- Update and maintain the RLC of every known security device. After a reception and successful decoding the RLC must be updated and then stored in non-volatile memory
- Look out for possible ongoing attacks. The receiver can gather indices that a possible intruder is trying to gain control.

Important to note here is that a receiver should not process unprotected messages from a transmitter once it transmitted its security teach-in. In this case it is to be assumed that an attacker has tried to gain control of the system.

4.3. Use of AES 128

The AES 128 algorithm is used in EnOcean network by these security features:

- Content authentication: CMAC
- Content protection – encryption: VAES
- Content protection – encryption: AES-CBC

The AES 128 algorithm [1] is a symmetrical algorithm. It uses the same key for encryption and decryption. By one encryption or decryption cycle it can process a 16 byte chunk (not more or less). If you concerned about the "strentgh" of the algorithm please see reference [2]. See simple graphical interpretation of AES function in Figure 8.

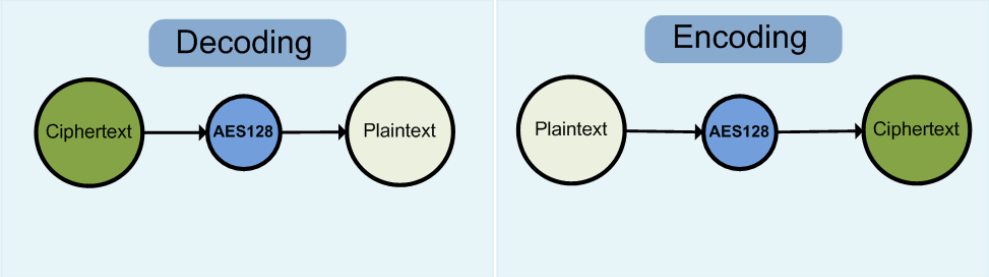

Figure 8 AES functions

For the understanding of the EnOcean Security we do not need to get deeper understanding of the algorithm it-self. It will enough to refer to it as algorithm with encryption and description functions.

4.3.1. Content authentication - CMAC with AES 128

In this chapter we will focus on the content authentication and use of RLC for this purpose.

As stated in chapter 2.2 content authentication creates a unique signature (CMAC) for every message, which allows the receiver to authenticate the incoming message at its transmitter. A possible attacker does not have the capabilities to create this signature (CMAC) and so the communication is protected against replay attacks.

Also, as explained before in chapter 2.2.2 and defined in chapter 3.1.1, with content authentication dynamic key modification must LAO be used. Or in a more practical meaning: when using CMAC then RLC (explicit / implicit) must be used. Applications which have a very large sensor value space (e.g. 24 bit sensor values) and have a low repeating ratio of measured values does not require this. But this is not a typical use case for energy harvesting sensors or the use case of residential building automatization. Therefore we will focus only on the use of CMAC and RLC together. The algorithm of CMAC counting is shown in Figure 9. For a more detailed explanation please refer to the specification[1].

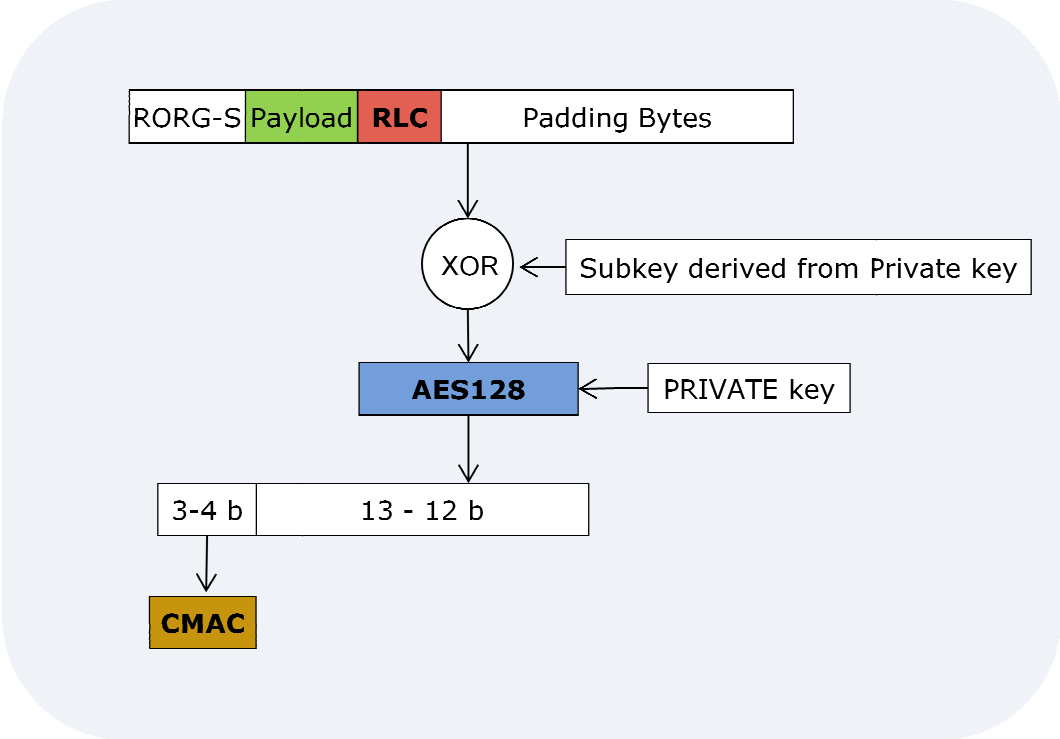

Figure 9 CMAC Algorithm

CMAC is a unique signature for a message in the telegram stream it includes:

- the included **payload** (in the figure green) data in the telegrams
- the **RLC** (in the figure red)

The CMAC is part of the transmitted telegram. It consumes additional payload in the telegram (3 or 4 bytes) but offers strong protection against replay attacks. The mechanism is simple:

1. The transmitter creates the CMAC based on the PAYLOAD and RLC.
2. Transmitter sends the telegram with CMAC to the receiver.
3. Receiver counts based on the PAYLOAD and actual RLC the CMAC too.
4. If the transmitted CMAC and counted CMAC are matching then the message is validated.

As stated before a dynamic key: RLC is an essential part of the CMAC process. It ensures that even if the payload is same the CMAC of any two telegrams in the telegram stream will be different. If no RLC would be present, then the CMAC would be same and thus the replay attack protection would be non-effective.

By using the RLC the CMAC and payload combination is always changing. The CMAC is 24 or 32 bit long. In the telegram stream the actual CMAC value can repeat itself, but the payload and CMAC combination is always unique and so ensures effective replay attack protection during the whole application live. This CMAC task with changing RLC on a transmitter is shown in Figure 10.

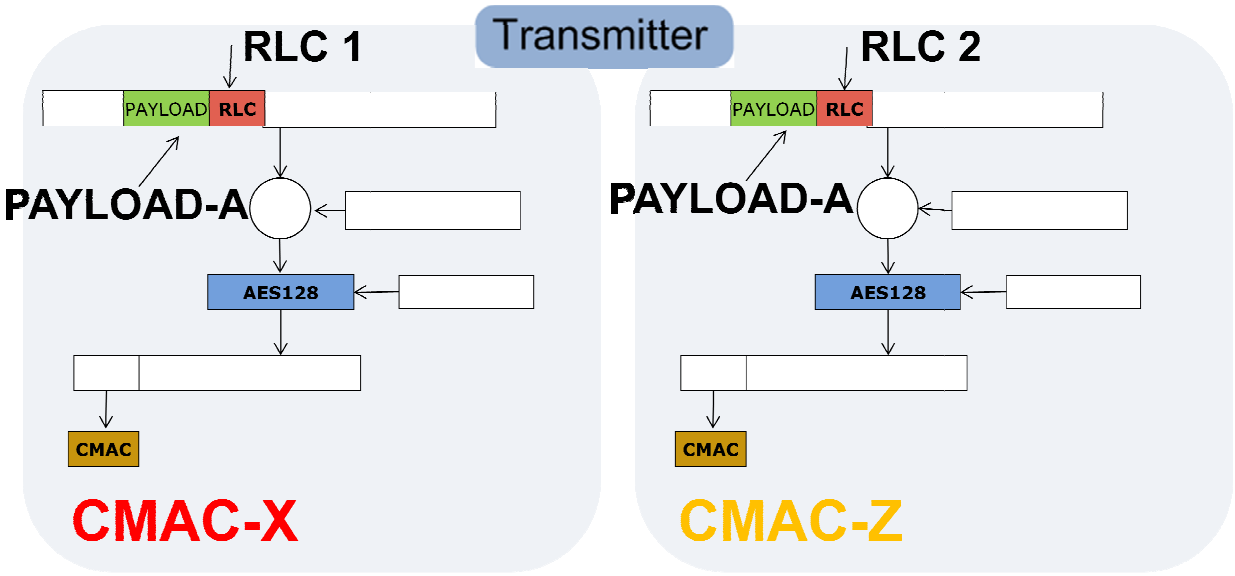

Figure 10 CMAC Task with RLC

In a wireless network some messages can get lost or the receiver is not able to receive it (e.g. is powered off). In this case the receiver has to try several RLCs until the CMAC is matched. The amount of this tries is also called the *rolling code window*. For more details please refer to the chapter 2.2.2 or the specification [1].

In the EnOcean Security specification the following RLC is defined as increment by 1. So RLCs are just a continuous row of number (e.g. ... ,50, 51, 52, 53, 54, ...). The receiver has to try all RLCs within the rolling code window (e.g. window size can be 100 RLCs) before he declares that this message cannot be authenticated. So if the receivers actual state was 50, then he tries to do the CMAC validation with 51, if not validated, then 52 and so on till he reaches 150 (with RLC window is 100). If the message was not validated till then than the transmitter and receiver lost its synchronisation or a potential attacker is intruding the system.

In Figure 11 you can see the above described algorithm. The transmitter uses the RLC X and gets the CMAC X as result then it transmits the telegram. The receiver did not receive messages L, M, ..., W and so he first computes the CMAC with his actual count. By using RLC L he gets CMAC L, but comparing CMAC X to CMAC L will result as not valid. So he tries RLC M, N until he gets a match with X. Then he updates his actual RLC state to X.

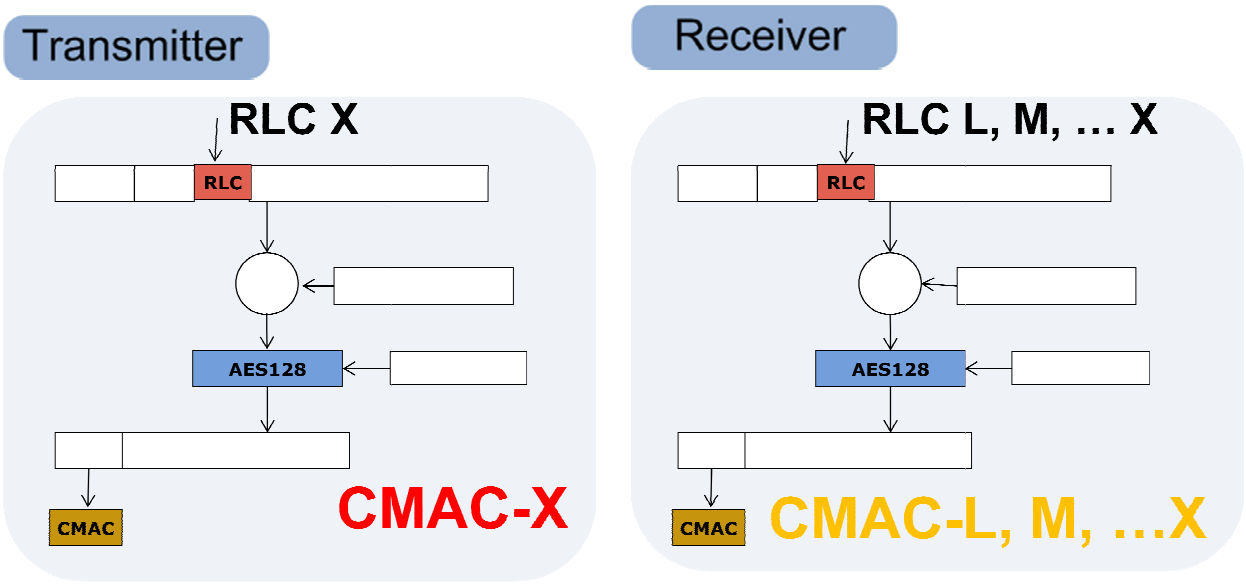

Figure 11 CMAC Task with use of rolling code window

4.3.2. Content protection with AES 128.

To enable encryption of variable length of data or to optimize the encryption for an ongoing data stream many use cases of the AES 128 algorithm were developed. For the needs of self powered radio devices the Security specification for EnOcean radio networks [1] uses these:

- AES – CBC: AES Cipher block chaining.
  Allows the chaining of byte streams. Minimum plaintext length is 16 bytes. This algorithm is used for application with huge data rates, which is not common in EnOcean networks.
- VEAS: Variable AES.
  Allows encryption of variable length of data (1 - ... b). This is mostly used by energy autarkic applications. In the next text we will focus more at this process.

The VAES tasks for encoding (transmitter) and decoding (receiver) are shown in figure. They look very similar. For detailed explanation please refer to specification [1]. There you can find also explanation on how to transmit larger amount (more than 16 bytes) of data with VAES (e.g. 18 bytes).

The essential difference between encoding and decoding is the last XOR operation. The XOR operation is used to add DATA to the ENC and then to retrieve the information back. This is valid:

DATA *XOR* **ENC** = **DATA_ENC** and **ENC** *XOR* **DATA_ENC** = **DATA**.

With this process it is then possible to encode variable length of data. The length of the ciphertext is equal the length of the plaintext.

The ENC is the result of an AES 128 operation. The Transmitter and Receiver have to use the same ENC to encode and decode the data. If we use same ENC in every following operation an intruder can easy guess the ENC if he is also aware of the context of the transmitted data. So we have to apply same principle of dynamic key modification. The change of RLC ensures that ENC is different for every following operation.

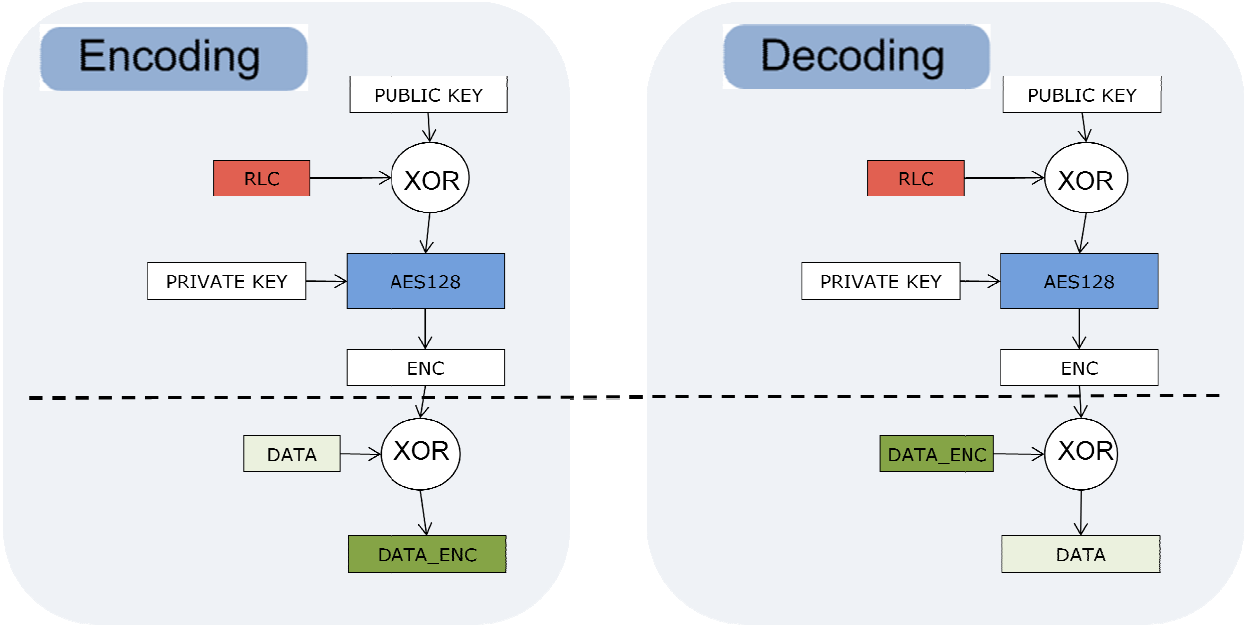

Figure 12 VAES tasks.

By using the RLC we must consider that the receiver and transmitter can have different RLC status for example due to telegram loss. This is a similar situation as with CMAC computing, see chapter 4.3.1. The receiver must use for ENC calculation the same RLC as the transmitter did, otherwise the decrypted data would not correspond to the original data.

The right RLC is obtained through CMAC validation, because CMAC and VEAS use the same RLC for one telegram encoding. First the CMAC is validated. Within this operation the right RLC is obtained. If the CMAC was not validated in the defined rolling code window, then the VEAS cannot decrypt data. This process is visualized in Figure 13.

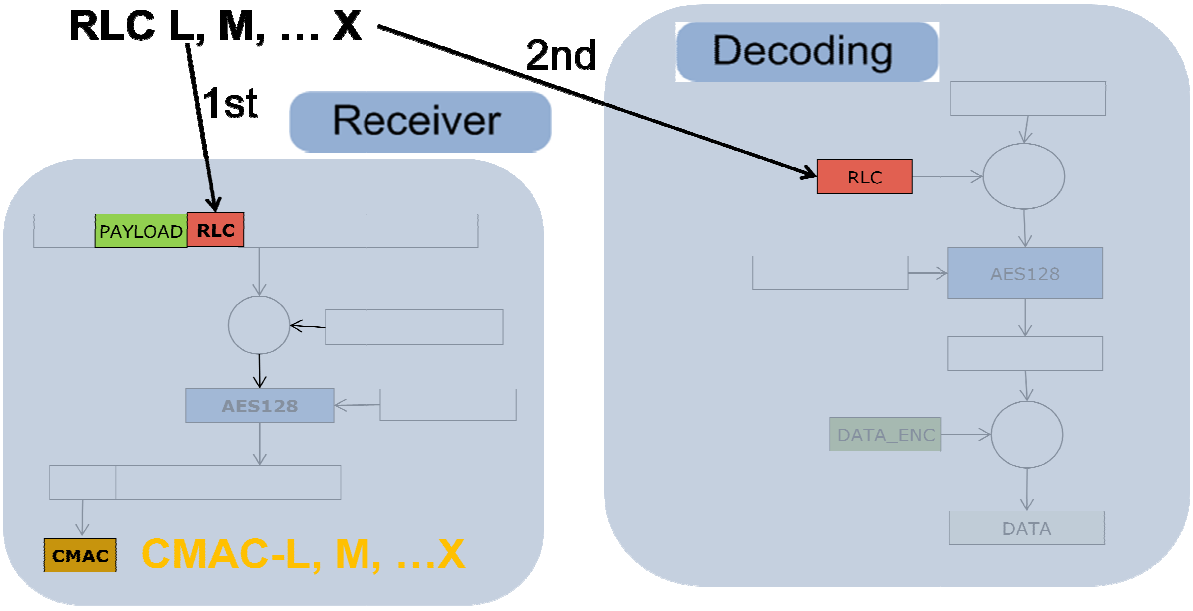

Figure 13 CMAC and VEAS tasks together

5. SECURITY AS A LAYER IN ENOCEAN PROTOCOL STACK

Security perfectly fits as a layer into the EnOcean Protocol stack. Please see the protocol stack visualisation in Figure 14. Security only affects the payload of a telegram. Therefore it does not affect any of the other layers. The security features defined do not take the context of the payload into consideration (e.g. it does not matter what the payload represents). Therefore security can be universally used with any other protocol.

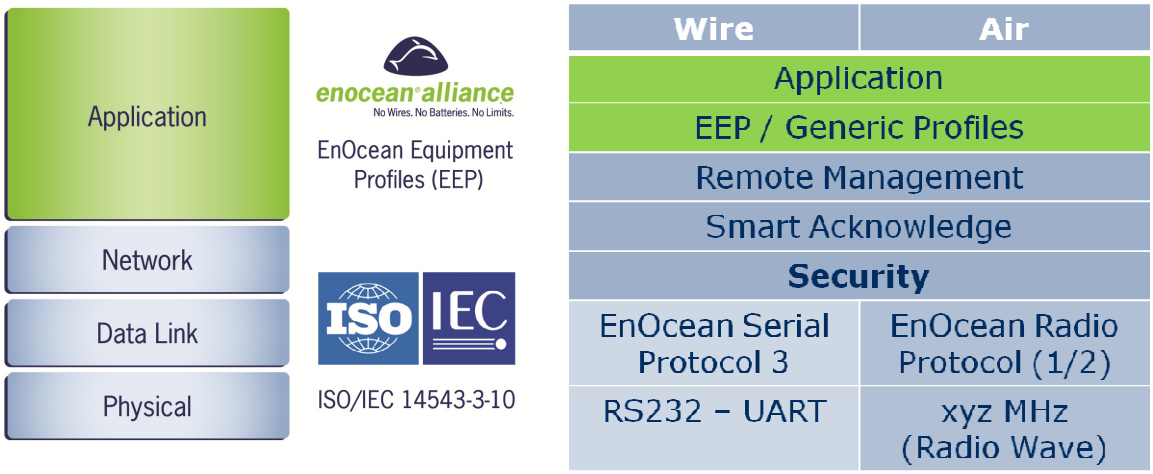

Figure 14 EnOcean protocol stack

If designing a new application or redesigning an existing application adding security features will not affect the application layer architecture or logic. The EnOcean protocols above Security are:

- EEP / GP
- Remote Management
- Smart Acknowledge

Their output is a radio message. The message consists of more parts, but for security only the header – RORG and payload are important. The security layer adds the security feature to the message and forwards it to the lower layers for transmission per radio or wire.

The security features can however extend the length of the bytes needed to transmit (e.g. adding CMAC, which is 3 or 4 bytes additional content to transmit). If the message length extends the capacities of one telegram, then the message is divided into several telegrams. The telegrams are then merged together to a message again at the receiver. For details on changing please refer to the security specification [1].

To underline the universal use of security with any above layer please refer to Figure 15. There you can see simple visualisation of the above description. The output message of the above layers is being process with security and transmitted to the air.

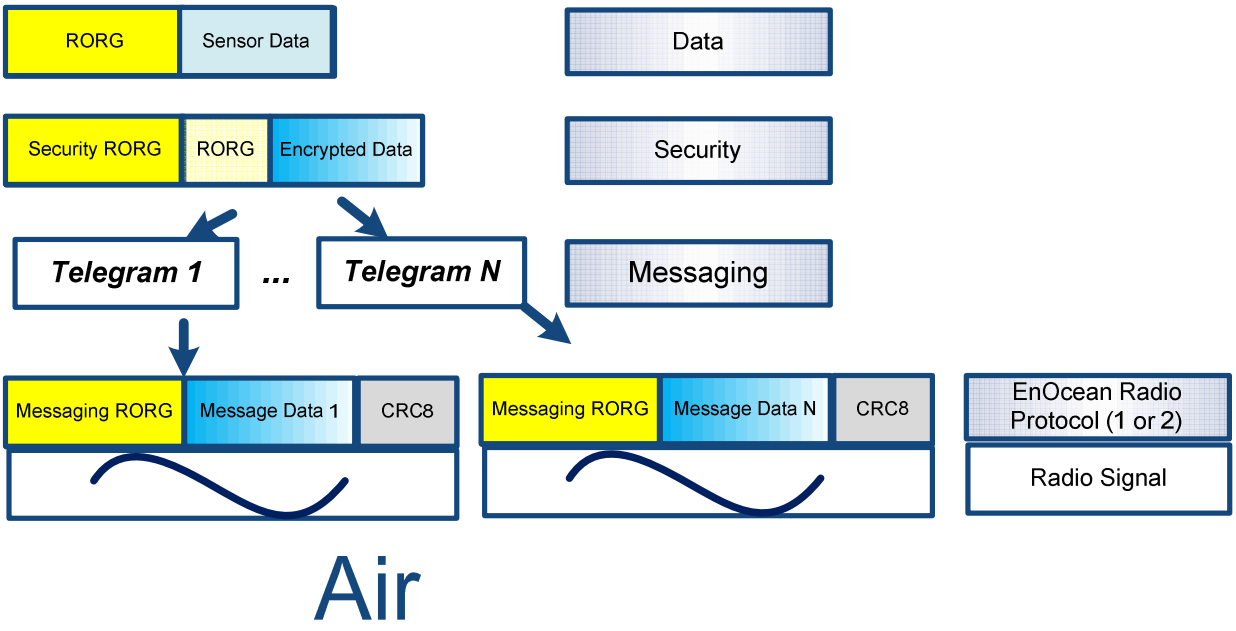

Figure 15 Security with underlying layers

This chapter describes how to use security with the above layers in detail.

5.1. Usage of EEP Profiles and GP

With the existing Generic Profiles – GP [4] and EnOcean Equipment Profiles – EEP [3] specification there is no change in the application because of using security features. The defined profiles can be used as before, because security features are not aware of the processed content. The only exceptions are profiles which use the telegram status field of EnOcean Radio Protocol 1 [5] (RPS - F6 Profiles). The status field is not part of the telegram payload and is not protected by the security features. Therefore these profiles are redefined in the VLD Family D2-03-XX [3].

5.1.1. Teach-in process with profiles

The teach-in telegram content of the teach-in telegram is not changed. But it is important to note that in applications using security, first the security teach-in must be performed and then the profile teach-in. This means:

- first the "security link" is established by security teach-in and
- then the profile teach-in information is already transmitted protected by the security features (e.g. the EEP profile number and manufacturer ID is encrypted)

The applications using RPS Profiles (EEP: F6-xx-xx) are again here an exception. These profiles did not have a profile teach-in due to the operative characteristics. Now with security they require to develop a security teach-in capability. Therefore also the security teach-in header was adjusted. Please see the specification for details [1].

5.1.2. Data Communication

As stated before data communication of devices is not affected by security features. The payload gets protected by the security features at the transmitter and then processed at the receiver. The payload from application layer aspect will stay unchanged. This enables a flexible design or redesign of application.

5.2. Usage of Smart Acknowledge and Remote Management

To use security features with Smart Acknowledge [6] and Remote Management [7] first a bidirectional "secure link" must be established. Essentially this means that a bidirectional security teach-in must be performed. Please see details on bidirectional security communication in the specification [1]. Then all transmitted data communication must be protected by the defined security features.

Reclaim messages and Signal messages in Smart Acknowledge have a special meaning and can be used also non-encrypted. There are also some other constrains in Smart Acknowledge due to security features which are out of the focus of this application note. If you are interested in a Smart Acknowledge application with security please contact us for details (support@enocean.com).

Remote Management communication is also protected by an remote management security key. Considering this fact the remote management communication can be also used without security features in a secure application. This decision is up to the application developer.

5.3. Bidirectional communication with security features

Prior to a bidirectional communication a bidirectional security teach-in must be performed. Details can be found in the security specification [1]. But in a bidirectional communication between devices A and B the way from A to B shall be protected by different KEY and RLC counter than the way from B to A.

This means:

- Device A sends in security teach-in its KEY A and RLC A to device B.
- Device B stores this information and sends to device A in a security teach in KEY B and RLC B.

The data communication will follow these rules:

- Device A uses for security features KEY A and RLC A.
- When device B receives a message from device A it updates the RLC A to its actual state.
- Device B uses for security features KEY B and RLC B.
- When device A receives a message from device B it updates the RLC B to its actual state.

6. WHAT NEXT?

If you like the security features of the EnOcean radio networks and are interested in developing an secure application there several steps how to continue:

- See Application Note for receivers
- See Application Note for transmitters
- See specification of Security in EnOcean radio networks
- See EnOcean Link User Manual for security implementation
- See Dolphin API User Manual for security implementation
- See Dolphin V4 API User Manual for security implementation
- See ready Security Products / Modules

Disclaimer

The information provided in this document describes typical features of the EnOcean radio 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.