ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 3, No. 2, 2018

SECURITY ARCHITECTURE TECHNICAL INVESTIGATION FOR IEEE 802.15.4 LOW-RATE WIRELESS PERSONAL AREA NETWORKS

Viktor Melnyk

Lviv Polytechnic National University, 12, S. Bandery Str., Lviv, 79013, Ukraine
Authors' e-mail*: viktor.a.melnyk@gmail.com*

Submitted on 30.11.2018

Abstract: **The paper aims at providing the technical investigation on implementation options for the security subsystem architecture in the IEEE 802.15.4 compatible devices. Since the security procedures typically consume most processing capacity of IEEE 802.15.4 device, efficient implementation of the security subsystem is essential. A brief functional overview of the security procedures has been provided. General investigations on security procedures implementation have been performed. Three general approaches for security subsystem implementation have been considered: a) software implementation; b) hardware implementation; and c) hardware-software implementation. The aim is to determine optimum implementation approach corresponding to low-cost and low-power consumption requirements. A hardware-software approach has been selected for the security subsystem implementation, and an expediency of the security procedures of hardware and software implementation has been estimated.**

***Index Terms*: Wireless Personal Area Networks, IEEE 802.15.4, AES-CCM*, wireless security.**

I. INTRODUCTION

The IEEE 802.15.4 standard [1], [2] defines the protocol and interconnection of devices via radio communication in a low-rate wireless personal area networks (LR-WPANs), which are used to transfer information over relatively short distances. The standard is intended to such wireless networks where devices are often battery-powered, and the power consumption must be as low as possible. It is the basis, for instance, for the ZigBee [3], ISA100.11a [4], MiWi, and Thread protocol specifications, each of which further extends the standard by developing the upper layers which are not defined in IEEE 802.15.4.

There is a wide variety of applications where IEEE 802.15.4 wireless technology is intended to be used. As a motivation for the standard creation, several leading companies and visionaries have proposed a number of appropriate applications. Some of them are listed below.

- Industrial and commercial control and monitoring, where high data throughput is not needed, but low power consumption of the battery-powered devices is essential.
- Home automation and networking, where cable replacement appears as significant advantage. Types of potential devices include consumer electronic devices, personal computer accessories, as well as home security, lighting control, air conditioning and heating systems.
- Automotive sensing, where wired applications cause a great impact on the installation cost, maintenance, diagnostic etc., and even have a limited installation feasibility.
- Other applications, which include agricultural realm, education, etc.

Alternatively, it can be used with 6LoWPAN and standard Internet protocols to build a wireless embedded Internet.

Among the main objectives of LR-WPAN there are reliable data transfer, extremely low cost, low power consumption and a reasonable battery life. This introduces restrictions and additional requirements into LR-WPAN infrastructure, as follows: a) communicating devices would have to use a small number of simple (as simple as possible) communication protocols, to consume extremely small power amount; b) communicating devices would be simple and cheap itself; and c) communicating devices would use reliable protocols of the data transfer, and would provide an appropriate level of the data security, thus corresponding to requirements of the critical application realms.

Two different device types can participate in an IEEE 802.15.4 network; a full-function device (FFD) and a reduced-function device (RFD). The FFD can operate in three modes serving as a personal area network (PAN) coordinator, a coordinator, or a device. An FFD can talk to RFDs or other FFDs, while an RFD can talk only to an FFD. An RFD is intended for applications that are extremely simple, such as a light switch or a passive infrared sensor; they do not have the need to send large amounts of data and may only be associated with a single FFD at a time. Consequently, the RFD can be implemented using minimal resources. An FFD has to be able to communicate with a number of RFDs or/and other FFDs; this feature makes it more "resources-hungry".

II. PROBLEM STATEMENT

The most resource-intensive part of the information processing in IEEE 802.15.4 std. WPAN devices (both RFDs and FFDs) is the cryptographic protection of the

data packets. Given the very severe constraints on power consumption, the realization of the information security subsystem, its architecture and implementation is a matter of special concern, and that is a goal of the study carried out in this paper. Basing on a detailed analysis of the security procedures complexity, the sequence of their execution and the links between them, this paper explores the implementation options for the security subsystem architecture in IEEE 802.15.4-Std. compatible devices and defines which procedures are appropriate to implement in software and which in hardware.

III. STRUCTURE OF THE ARTICLE

The paper consists of eight sections (including Introduction, Problem Statement, and the current one). The material of the following sections is structured as follows. Section IV provides an overview of the security services in IEEE 802.15.4 Std. Section V gives the detail description of the frame securing and unsecuring processes on the MAC sublayer, including the details of the outgoing and incoming frames security procedures and outgoing and incoming frames key retrieval procedures. In Section VI, the security procedures implementation is investigated, hardware-software partitioning criteria are formulated and general approaches for security subsystem architecture are outlined. In Section VII, the IEEE 802.15.4 security procedures implementation options are analyzed depending on where the initial frame is retained. The Section VIII – Conclusions, summarizes options for the security procedures implementation and the consequent security subsystem architecture in the IEEE 802.15.4 compatible devices.

IV. SECURITY SERVICES IN IEEE 802.15.4 STD.

The IEEE 802.15.4 standard specifies cryptographic procedures for protecting communications at the Medium Access Control (MAC) sublayer according to OSI Reference Model. The cryptographic mechanism in this standard is based on the symmetric-key cryptography and uses keys that are provided by higher layer processes. The establishment and maintenance of these keys are outside the scope of the standard. The mechanism assumes a secure implementation of cryptographic operations and secure and authentic storage of keying material.

The cryptographic mechanism provides particular combinations of the following security services: data confidentiality, data authenticity and replay protection. The actual frame protection provided can be adapted on a frame-by-frame basis and allows for varying levels of data authenticity and for optional data confidentiality.

Three editions of the IEEE 802.15.4 standard – in 2003, in 2006 and in 2011, and a number of amendments were issued. At the first edition in 2003 the eight security levels which were provided by eight security suits were offered, as it is shown in Table 1.

Table 1

Security suits by IEEE 802.15.4 – 2003rd edition standard

Security Attribute	Data confident.	Data authent.	Replay protection
NONE	no	no	no
AES-CTR	yes	no	optional
AES-CBC-MAC-32	no	yes	no
AES-CBC-MAC-64	no	yes	no
AES-CBC-MAC-128	no	yes	no
AES-CCM-32	yes	yes	optional
AES-CCM-64	yes	yes	optional
AES-CCM-128	yes	yes	optional

The symmetric algorithm that is employed for all security suits is Advanced Encryption Standard (AES) [5]. It is a block cipher adopted as an encryption standard by the U.S. government. AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) in November 26, 2001 after a 5-year standardization process. It became effective as a standard in May 26, 2002. As of 2016, AES is one of the most popular algorithms used in symmetric key cryptography.

The AES encrypts 128-bit data blocks with the usage of encryption keys of 128, 192, and 256 bits. For the 802.15.4 standard the 128-bit key size has been selected.

As we see in the Table 1, the security suits are broken down into four sections. The first of them (NONE) defines a suit that does not provide any cryptographic mechanisms application. Its inclusion is mandatory. The second section (AES-CTR) defines a suit that provides confidentiality performing encryption in accordance to the AES algorithm with the Counter (CTR) mode. The third section shows three security suits that perform authentication according to the CBC-MAC mode of the AES algorithm for message authentication code length of 32, 64, and 128 bits, respectively. The fourth section is built similarly to the previous one, but here the CCM (Counter + CBC-MAC) mode [6] of AES is used. It may provide confidentiality and authenticity as well.

Good explanation of the 802.15.4 – 2003 edition standard security suits is done in [7].

A receiver can optionally enable replay protection when using a security suite that provides confidentiality. This includes AES-CTR and all of the AES-CCM variants. The recipients use the frame and key counter as a 5 byte value, the replay counter, with the key counter occupying the most significant byte of this value. The recipient compares the replay counter from the incoming packet to the highest value seen, as stored in the access control list (ACL) entry1. If the incoming packet has a larger replay counter than the stored one, then the packet

[1] ACL mode has been defined in IEEE 802.15.4 standard edition of 2003 as a mechanism to provide authentication. It is rejected in edition of 2006 of the standard.

is accepted and the new replay counter is saved. If, however, the incoming packet has a smaller or an equal value, the packet is rejected and the application is notified of the rejection.

The IEEE 802.15.4 Std. edition of 2006 slightly changes the security suits specification. Instead of the usage of three different modes for encryption and authentication: CTR, CBC-MAC, and CCM, it just uses the CCM* mode, that is specified in Appendix B of [1]. The CCM* mode differs from the original CCM in the following. The length of an authentication field is changeable (this feature is lacking in CCM). The value of the length is encoded somewhere into nonce, that imposes additional restrictions to the last. Zero value for the length of an authentication field is acceptable, that means disabling authenticity since the authentication field is empty.

Thus, if the length of an authentication field is equal to zero, and confidentiality is disabled, then no security operations are performed. If the length of an authentication field is equal to zero, and confidentiality is enabled, then the CCM* is functioning as the CTR mode. If confidentiality is disabled and the length of an authentication field is more then zero, then the CCM* is functioning as the CBC-MAC mode. And, finally, if confidentiality is enabled and the length of an authentication field is more then zero, then the CCM* is functioning as the original CCM mode.

For variable-length authentication tags (i.e. variable-length results of authentication algorithm execution), the original CCM mode is known to be vulnerable to specific attacks [7]. These attacks may arise with the original CCM mode because the decryption transformation does not depend on the length of the authentication tag itself. The CCM* mode avoids these attacks altogether, by requiring that one shall be able to uniquely determine the length of the applicable authentication tag from the counters blocks.

The CCM* mode is employed for all security suits (excepting NONE) and is intended to perform authentication, encryption, or both authentication and encryption as it is shown in Table 2.

Table 2

Security suits by IEEE 802.15.4 – 2006th and 2011th editions standards

Security attribute	Data conf.	Data authent.	Replay protection	MIC length, in octets
NONE	no	no	no	0
MIC-32	no	yes	no	4
MIC-64	no	yes	no	8
MIC-128	no	yes	no	16
ENC	yes	no	optional	0
ENC-MIC-32	yes	yes	optional	4
ENC-MIC-64	yes	yes	optional	8
ENC-MIC-128	yes	yes	optional	16

The CCM* [1] uses Counter (CTR) mode for encryption in conjunction with message authentication method based on Cipher Block Chaining (CBC) mode. CBC is used to produce a message integrity code (MIC).

The CCM* mode has two parameters, M – the length (in bytes) of the MIC, valid values for which are 0, 4, 6, 8, 10, 12, 14, and 16 (the value M = 0 corresponds to disabling authenticity), and L – the length (in bytes) of the message length field, valid values for which are the integers 2, 3, …, 8 (the value L = 1 is reserved). It is based on the original CCM mode, which is described in [6].

The standards [1] and [2] specify CCM* application with M = 0, 4, 8, and 16 octets, and L = 2 octets. These parameters allow to provide data confidentiality and/or authenticity according to eight security suits that are specified in [1] and [2] and shown in Table 2.

V. IEEE 802.15.4 WIRELESS NETWORK SECURITY FUNCTIONAL OVERVIEW

Data security services in IEEE 802.15.4 wireless network are based on AES-CCM* transformation. This transformation provides data confidentiality, authenticity, and optional replay protection. This transformation applies AES encryption algorithm, which belongs to symmetric block ciphers and uses secret symmetric encryption keys.

According to [1], security is applied to the data on MAC sublayer. Corresponding data to be secured or unsecured are represented in MAC frame format. General format of the MAC frame is shown in Fig. 1.

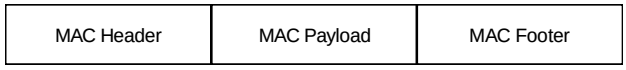

Fig. 1 . General MAC frame format

Generally, MAC frame constitutes of three parts: MAC header (MHR), MAC payload, and MAC footer (MFR). MAC payload represents data which come to the MAC sublayer from the upper layers. MHR part contains control information, sequence number, and addressing fields, and is added in MAC sublayer. Since frame securing in IEEE 802.15.4 is optional, in order to determine whether the frame needs to be secured or unsecured MHR contains control field which shows whether the frame security is enabled. If it does, the MHR contains information that specifies security level determining which security suit (see Table 2) shall be used for the frame unsecuring. This information is placed into the optional security header (Auxiliary Security Header) which is inserted during the frame securing into MHR if frame security is enabled.

Confidentiality service is applied to MAC payload only, when authenticity – to MHR and MAC payload. MFR represents a checksum of MHR and MAC payload and is added to the MAC frame before its transmission to the lower (physical) layer. Thus, MFR in security process is not used.

The standard [1] specifies four types of frames: beacon frame, data frame, MAC command frame, and acknowledgement frame. First three contain MAC payload, when acknowledgement frame does not. Therefore, security can not be applied to the last.

Each outgoing frame shall pass the securing process as well as each incoming frame shall pass the unsecuring process. Securing and unsecuring processes are specified in [1] by appropriate security procedures.

In this Section brief functional overview of the security procedures is provided. In order to perform investigations on MAC sublayer security subsystem implementation it is necessary to have better understanding of how the overviewed security procedures do operate.

For both securing and unsecuring main operation is execution of the AES-CCM* transformation. AES-CCM* forward transformation is executed during frame securing, while AES-CCM* inverse transformation is executed during unsecuring. All other operations consist in forming of the input data for AES-CCM* transformation, checking of security settings of the frame and the device, retrieving of the key and other information needed for frame securing or unsecuring.

The input data for securing/unsecuring process are outgoing or incoming frame and additional information which is needed in order to perform the process. Additional information represents device security settings, values determining key identification method and data for its identification, values of the outgoing and incoming frames counters that are used to provide and check sequential freshness and to construct the nonce, and other. Some of this information is stored in MAC PIB (PAN information base) of the device; some is taken from parameters of originating primitives, which come from the upper layers.

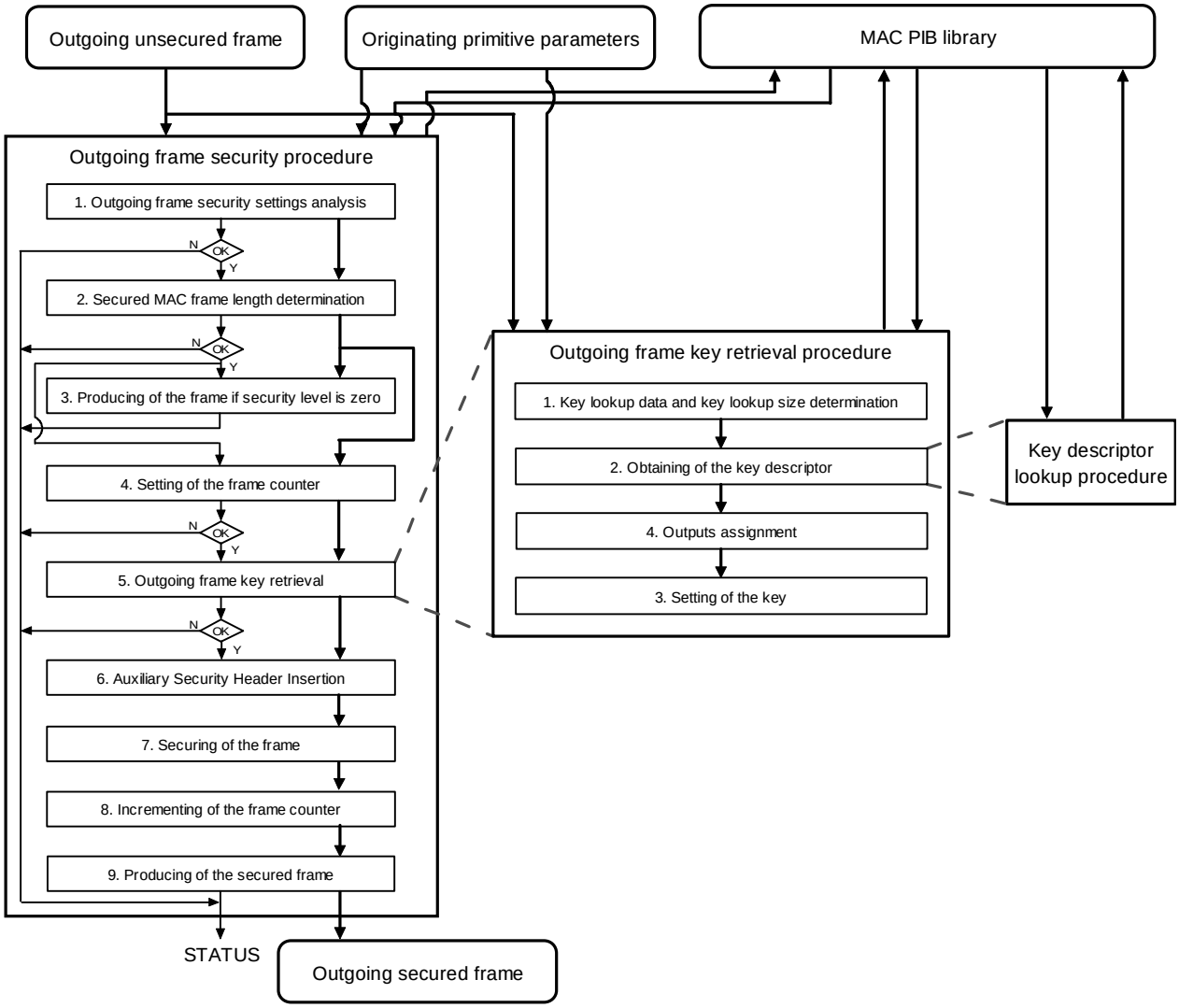

Fig. 2. The frame securing process diagram

The output data for securing/unsecuring process are corresponding secured or unsecured frame, as well as data needed to generate primitives for upper layers, and updated MAC PIB information. The updated MAC PIB information consists in new values for frames counters and values specifying future possibility of the keying material to be used.

Information whether securing/unsecuring process completed successfully is given by status output. If corresponding process competed successfully, then status output indicates SUCCESS. Otherwise, it indicated some type of error.

If error occurs during frame securing then process terminates and indicates upper layer about the reason of termination. If error occurs during the frame unsecuring, then process terminates with error identification and produces incoming frame and security-related data to generate originating primitives for upper layers.

Securing and unsecuring processes are functionally different, so we consider them both separately.

A. FRAME SECURING PROCESS

Functionally, frame securing process can be represented as it is shown in Fig. 2. It consists in execution of three procedures: outgoing frame security procedure, outgoing frame key retrieval procedure, and key descriptor lookup procedure. The outgoing frame key retrieval procedure is invoked by the outgoing frame security procedure. Also, it invokes the key descriptor lookup procedure.

Each procedure contains one or more functional operations, or steps. Let's take an overview of these procedures step by step.

1) The outgoing frame security procedure overview

The outgoing frame security procedure is main procedure that is executed during frame securing process. The aim of the procedure is to produce secured MAC frame. Procedure uses frame itself, information from MAC PIB of the device, and information from originating primitive parameters. Sequence of the procedure steps is listed and described below.

1. The outgoing frame security settings analysis. The aim of this step is to determine whether outgoing frame needs to be secured and whether the frame security settings are set correctly. Also, the step determines the device's security settings. If the checks fail procedure terminates and indicates that security is unsupported.

2. The secured MAC frame length determination. The aim of this step is to check whether the resulted secured frame length will not be greater than maximum length accepted for the MAC frame. This check is necessary since Auxiliary Security Header and the authentication tag shall be added to the initial frame. If check fails the procedure terminates and indicates that the frame is too long.

3. Producing of the secured frame if the security level is zero. If frame settings show that the frame does not need to be secured, and all previous checks have passed successfully, it is produced out. The procedure completes indicating successful frame processing.

4. Setting of the frame counter. The aim of this step is to obtain the frame counter value from the MAC PIB of device and to check that this value hasn't reached maximum acceptable value 0xFFFFFFFF (it determines maximum number of the key usage). If it has then associated keying material can no longer be used, thus, the requiring key has to be updated. In this case procedure terminates and indicates the counter error.

5. The outgoing frame key retrieval. This step invokes outgoing frame key retrieval procedure (see next subsection). If that procedure fails the procedure terminates and indicates that the key is unavailable.

6. Auxiliary Security Header insertion. At this step Auxiliary Security Header is inserted into MHR. Auxiliary Security Header contains security-related information which shall be used by corresponding recipient(s) during the frame unsecuring.

7. Securing of the frame. At this step the frame securing according to AES-CCM* forward transformation is performed. Before transformation execution corresponding input data sets and nonce must be formed. The secured frame, where secured payload concatenated with authentication tag substitutes unsecured payload, is formed after transformation execution.

8. Incrementing of the frame counter. At this step the frame counter is incremented by one. Corresponding frame counter value is updated in MAC PIB of device.

9. Producing of the secured frame. This step is formal and means that frame securing process is successfully completed. The procedure completes indicating successful frame processing.

2) The outgoing frame key retrieval procedure overview

The outgoing frame key retrieval procedure is invoked by the outgoing frame security procedure in order to obtain encryption key from the MAC PIB of device. The procedure uses frame itself, information from MAC PIB, and information from originating primitive parameters. The sequence of the procedure steps is listed and described below.

1. Key lookup data and key lookup size determination. The aim of this step is to form the identification data which shall be used to retrieve appropriate key from the MAC PIB of device. These identification data are named as key lookup data and key lookup size.

2. Obtaining of the key descriptor. This step invokes key descriptor lookup procedure, which is overviewed in the next subsection. If that procedure fails the procedure terminates and indicates failure.

3. Setting of the key. This step consists in assignment of key that was retrieved from the MAC PIB of device to key value which shall be used in AES-CCM* transformation.

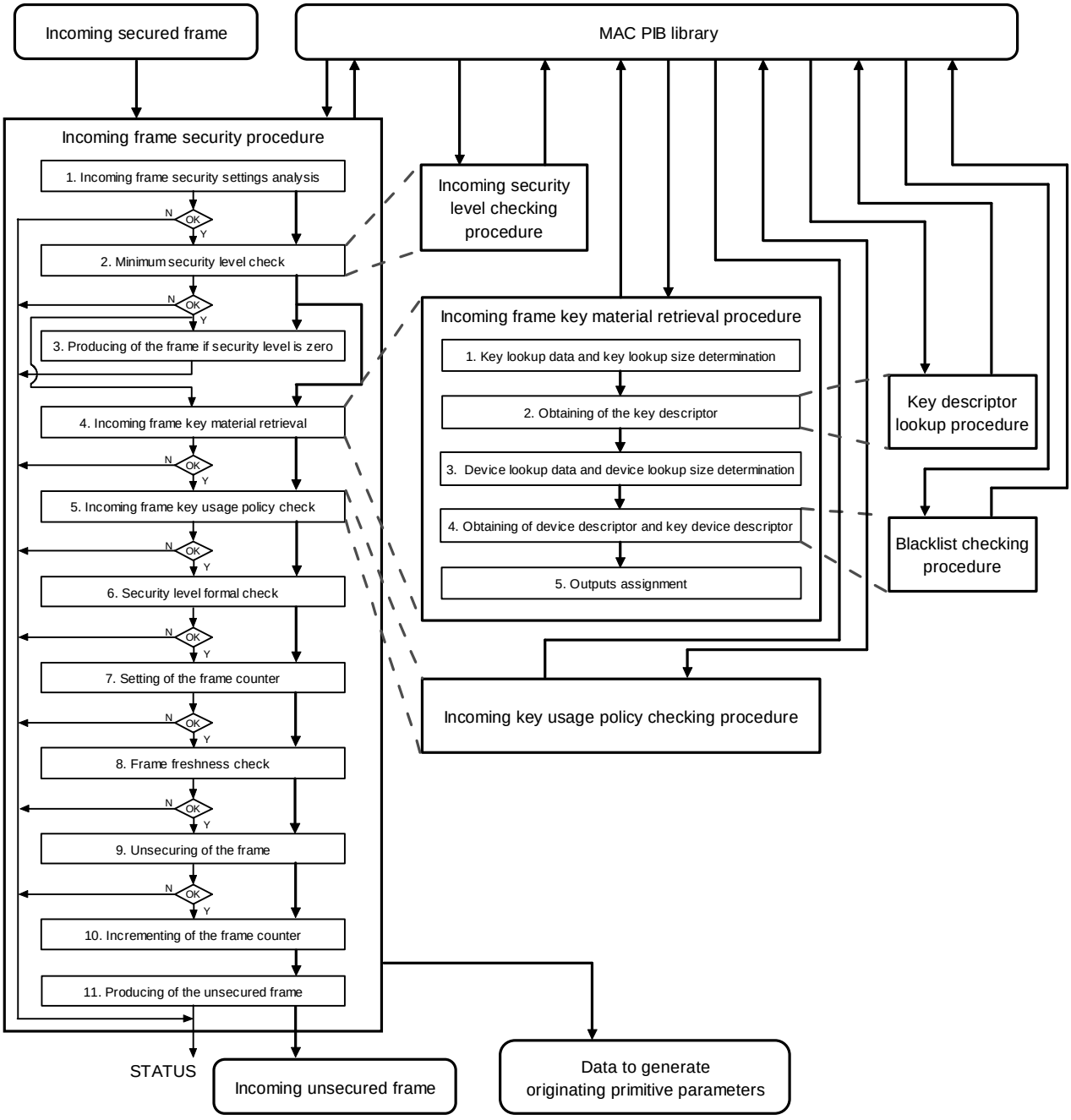

Fig. 3. The frame unsecuring process diagram

4. The outputs assignment. This step is formal and means that key retrieval process is successfully completed and key is produced.

3) The key descriptor lookup procedure overview

In order to communicate securely in the PAN the device may use one or more encryption keys. Keys and key-related information, which is used during securing/unsecuring process, are contained into the key descriptors. Each key descriptor contains exactly one key. Beside key and key-related information each key descriptor contains auxiliary information that identifies this key. All key descriptors are placed into MAC PIB of the device. The aim of key descriptor lookup procedure is to determine appropriate key among all others listed in the MAC PIB. Determination is performed by comparison of identification data formed at the first step of outgoing frame key retrieval procedure with that data contained info key descriptors in MAC PIB. If the matching entries weren't found the procedure would fail.

B. THE FRAME UNSECURING PROCESS

Functionally, the frame unsecuring process can be represented as it is shown in Fig. 3. It consists in execution of six procedures: incoming frame security procedure, incoming security level checking procedure, incoming frame key material retrieval procedure, incoming key usage policy checking procedure, key

descriptor lookup procedure (the same as overviewed earlier), and blacklist checking procedure. Let's take an overview of these procedures step by step.Incoming frame security procedure overview

Incoming frame security procedure is main procedure that is executed during the frame unsecuring process. The aim of the procedure is to produce the unsecured MAC frame. The procedure uses frame itself and information from MAC PIB of the device. The sequence of the procedure steps is listed and described below.

1. The incoming frame security settings analysis. The aim of this step is to determine whether incoming frame needs to be unsecured, whether the frame security settings are set correctly, and whether the frame version is suitable. Also, the step determines the device's security settings. If the checks fail the procedure terminates and indicates that security or legacy is unsupported.

2. The minimum security level check. This step invokes incoming security level checking procedure, which is overviewed in the next subsection. If that procedure fails the procedure terminates and indicates that security level of the frame is not proper.

3. Producing of the unsecured frame if the security level is zero. If frame settings show that the frame does not need to be unsecured, and all previous checks have passed successfully, it is produced out. The procedure completes indicating successful frame processing.

4. The incoming frame key material retrieval. This step invokes incoming frame key material retrieval procedure, which is overviewed in the subsection 3 of this section. If that procedure fails the procedure terminates and indicates that the key is unavailable.

5. The incoming key usage policy check. This step invokes incoming key usage policy checking procedure, which is overviewed in the subsection 5 of this section. If that procedure fails the procedure terminates and indicates that the key type is not proper.

6. The security level formal check. The aim of this step is to determine whether the remote device originating the frame can override security minimum level. If incoming security level checking procedure has returned conditionally passed status and remote device can not override security minimum, then the procedure terminates and indicates that security level of the frame is not proper.

7. Setting of the frame counter. The aim of this step is the same as in the frame securing process.

8. The frame freshness check. In order to check frame freshness the MAC PIB of device contains the frame counters values for each remote device that the current device communicates with. The check is performed by comparison of the frame counter value included into the incoming frame with the value of corresponding frame counter in the MAC PIB. If this value is less or equal to the stored one in MAC PIB, then it means that frame is not fresh. In this case the procedure terminates and indicates the counter error.

9. Unsecuring of the frame. At this step, the frame unsecuring according to AES-CCM* inverse transformation is performed. Before transformation execution corresponding input data sets and nonce must be formed. The unsecured frame, where unsecured payload substitutes secured payload concatenated with the authentication tag, is formed after transformation execution. During the unsecuring frame the authenticity is checked. It is performed by comparison of the computed authentication tag with the received one. If they do not match, the procedure terminates and indicates security error.

10. Incrementing of the frame counter. At this step frame counter is incremented by one. Corresponding frame counter value is updated in MAC PIB of device. If the updated value has reached maximum acceptable value 0xFFFFFFFF, then corresponding information is set in MAC PIB indicating that associated keying material can no longer be used, thus, requiring key to be updated.

11. Producing of the unsecured frame. This step is formal and means that frame unsecuring process is successfully completed and unsecured frame along with the security-related information is produced. The procedure completes indicating successful frame processing.

4) Incoming security level checking procedure overview

The aim of the procedure is to determine whether the security level specified in the incoming frame corresponds to the minimum requirements of the security level for actual frame type. Types and subtypes[2] of the frames with the associated minimum security levels are contained into the MAC PIB of device. Determination is performed by comparison of security level specified in the incoming frame with minimum security level for actual frame type and subtype. Beside the minimum security level value MAC PIB contains information indicating whether the remote device can override security minimum. The procedure completes successfully if security level specified in the incoming frame is greater than or equal to minimum security level. Otherwise, the procedure fails if remote device can not override security minimum, and passes conditionally if it can override.

5) Incoming frame key material retrieval procedure overview

The incoming frame key material retrieval procedure is invoked by the incoming frame security procedure in order to obtain encryption key and other key-related information from the MAC PIB of device. The procedure uses frame itself and information from MAC PIB. The sequence of the procedure steps is listed and described below (the first two steps are the same as in the outgoing frame key retrieval procedure).

1. Key lookup data and key lookup size determination.

2. Obtaining of the key descriptor.

[2] Frame subtype means type of the MAC command frame

3. Device lookup data and device lookup size determination. The aim of this step is to form the identification data which shall be used to retrieve security-related information regarding remote device originating the incoming frame. This information is stored in the MAC PIB of device. The identification data are named as the device lookup data and the device lookup size.

4. Obtaining of the device descriptor and the key device descriptor. This step invokes blacklist checking procedure, which is overviewed in the next subsection. If that procedure fails, the procedure terminates and indicates failure.

5. The outputs assignment. This step is formal and means that key material retrieval process is successfully completed and key and key-related information are produced.

6) Blacklist checking procedure overview

The aim of the procedure is to determine whether the remote device originating the incoming frame belongs to the devices that may use identified key. The list of devices associated with each key is contained into the MAC PIB of the device. Determination is performed by comparison of identification data formed at third step of incoming frame key material retrieval procedure with that data contained info the MAC PIB of the device. If the matching entries weren't found, the procedure fails. Otherwise, the procedure shall check whether the device is marked as "blacklisted", i.e. whether keying material associating with this device may be used. If so, the procedure completes successfully, in other case it fails.

7) Incoming key usage policy checking procedure overview

The aim of the procedure is to determine whether the identified key may be used to unsecure the frame of present type. Frames types and subtypes which can be unsecured with the appropriate key are listed in the MAB PIB of the device for each key. Determination is performed by comparison of incoming frame type and subtype with those ones associating with the identified key in MAB PIB. If the matching entries weren't found, the procedure fails.

VI. THE INVESTIGATIONS ON SECURITY PROCEDURES IMPLEMENTATION

The security operation procedures have to be performed by MAC sublayer security subsystem (further named security subsystem). Since the security procedures typically consume most processing capacity of IEEE 802.15.4 device, the efficient implementation of the security subsystem is essential. This subsection provides general investigations on the security procedures implementation. Three general approaches for security subsystem implementation are considered in this subsection: a) software implementation; b) hardware implementation; and c) hardware-software implementation. The aim is to determine optimum implementation approach corresponding to low-cost and low-power consumption requirements.

It is assumed that the security subsystem is a part of MAC sublayer implementation, which is normally implemented with MAC microcontroller – i.e. programmable microcontroller intended to perform primary MAC functions, excluding security. No additional constrains are used.

C. SECURITY SUBSYSTEM FUNCTIONAL PARTITIONING

Functionally structure of the security subsystem can be broken down into two main units: AES-CCM* Crypto Engine (AES-CCM*-CE) and Security Processing Support Unit (SPSU), as it is shown in Fig. 4.

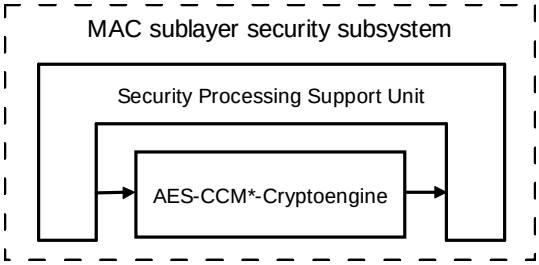

Fig. 4. The security subsystem general structure

AES-CCM*-CE unit performs AES-CCM* forward and inverse transformation and is a basic unit of the security subsystem. Further details of AES-CCM* processing as well as AES-CCM*-CE architecture are out of scope of this paper. We consider AES-CCM*-CE as a "black box", that receives appropriate input data and produces appropriate output data. All input data for AES-CCM*-CE are formed by SPSU. The output data of AES-CCM*-CE are used by SPSU to form the resulting secured or unsecured frame. We assume, that AES-CCM*-CE is hardware implemented, since its software implementation requires high performance microprocessor which energy consumption doesn't allow its application in IEEE 802.15.4 compatible devices.

SPSU unit interacts with AES-CCM*-CE and performs all overviewed security procedures:

Outgoing frame security procedure,
Outgoing frame key retrieval procedure,
Key descriptor lookup procedure,
Incoming frame security procedure,
Incoming security level checking procedure,
Incoming frame key material retrieval procedure,
Blacklist checking procedure,
Incoming key usage policy checking procedure.

All input data for the security subsystem are handled by SPSU. They are the frame to be secured/unsecured and security-related information that is used by security subsystem during the security processing. The security-related information involves the MAC PIB security attributes, the parameters of originating primitive, and the MAC sublayer constants. The output data are corresponding secured/unsecured frame, processing status, updated MAC PIB security attributes, and

security-related data to generate originating primitives for the upper layers.

D. SECURITY PROCEDURES SOFTWARE IMPLEMENTATION

Software implementation provides execution of all security functions by the programmable microprocessor. The security subsystem software implementation consists in the usage of the MAC microcontroller for the security subsystem software execution, as it is shown in Fig. 5.

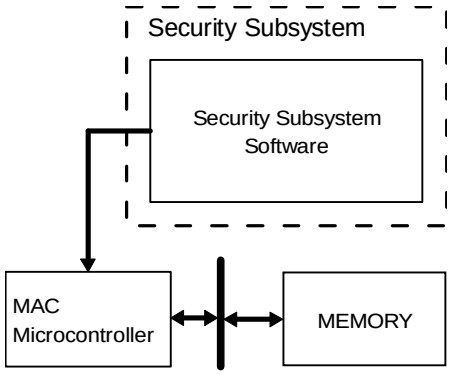

Fig. 5. The software-implemented security subsystem with the usage of the MAC microcontroller

Effective implementation of the security subsystem according to this approach is questionable since the system is intended to work in real time, which places strong restrictions on the frame securing/unsecuring (especially unsecuring) time. Moreover, in this case MAC microcontroller has to be able to execute all security procedures including AES-CCM* transformation along with the primary MAC sublayer functions. Due to the requirements of low cost and low power consumption this option seems to be not feasible.

Other disadvantages of the security software implementation are in following.

- Microprocessors use standard software and hardware interfaces which do not intended to perform operations on wide bit number data sets;
- Microprocessor's architecture is not efficient to perform cryptographic algorithms [8];
- Confidential data (keys and security-related information) are not protected since they are stored unsecured to be used by microprocessor and may be accessible for the unauthorized parties. Additional measures should be applied to protect confidential data. For more details see [9].

E. SECURITY PROCEDURES HARDWARE IMPLEMENTATION

Hardware implementation of the security subsystem provides execution of all security functions by specialized hardware means like ASIC or programmable hardware (e.g. FPGA, CPLD). Hardware-implemented security subsystem is shown in Fig. 6.

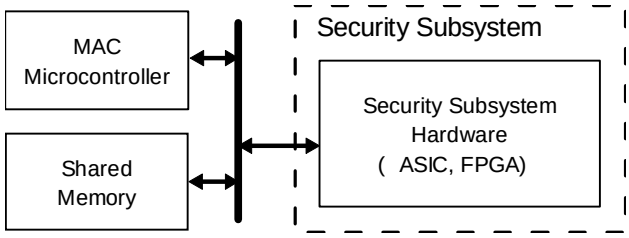

Fig. 6. The hardware-implemented security subsystem

The major advantage of the hardware implementation is the possibility to achieve needed performance level that shall satisfy the real-time requirements with minimum power consumption. The disadvantage is that the microcontroller may remain unemployed during the security subsystem operating time, what reflects non-effective usage of resources in the time of data processing, while resources of the dedicated security subsystem hardware will be significant.

F. SECURITY PROCEDURES HARDWARE-SOFTWARE IMPLEMENTATION

Hardware-software implementation of the security subsystem provides execution of security functions partly by the programmable microcontroller and partly by the specialized hardware means like ASIC or FPGA.

The general structure of hardware-software security subsystem is shown in Fig. 7.

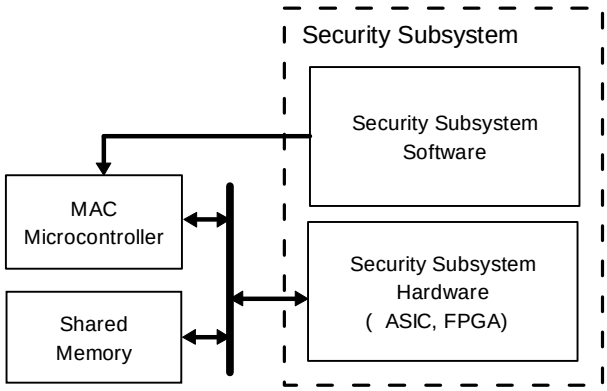

Fig. 7. The general structure of hardware-software-implemented security subsystem

The security subsystem here contains software and hardware parts. The execution of the software part is performed by the MAC microcontroller, thus, the interaction between software and hardware parts is realized through common bus. The security-related information is stored in shared memory, which is accessible by both MAC microcontroller and security subsystem hardware.

Hardware-software implementation represents a compromise solution which allows to avoid disadvantages of software implementation in sense of performance and expensive microcontroller usage and disadvantages of hardware implementation in sense of

high resources consumption. Furthermore it allows to employ the microcontroller along with specialized hardware for security procedures execution.

Due to these advantages, and having the MAC microcontroller available, the hardware-software implementation is being chosen as a basic for security subsystem implementation.

G. SECURITY SUBSYSTEM HARDWARE-SOFTWARE PARTITIONING CRITERIA

The first and main task that must be solved during security subsystem hardware-software implementation is effective system partitioning to the hardware and software parts. The following rules should be taken into account in this regard:

- Hardware part provides execution of the most time-critical tasks, especially for unsecuring process.
- The amount of information that is transferred between software (i.e. programmable microprocessor) and hardware parts has to be as small as possible, since the microprocessor uses its standard interface, and transition of large amount of data may take too long. If the large data portions must be transferred, the transfer may take a major time with a respect to the data processing time.
- The amount of separate interactions between HW and SW – i.e. complexity, has to be as small as possible.
- Scalability – i.e. possibility to configure the device to be FFD or RFD, to adjust the number of remote devices that actual FFD can communicate with.
- The amount of MAC hardware and software resources has to be reasonable.

Assuming that AES-CCM*-CE shall be implemented in hardware, hardware-software partitioning of the security subsystem comes to SPSU partitioning. The security subsystem with hardware-software general partitioning is shown in Fig. 8.

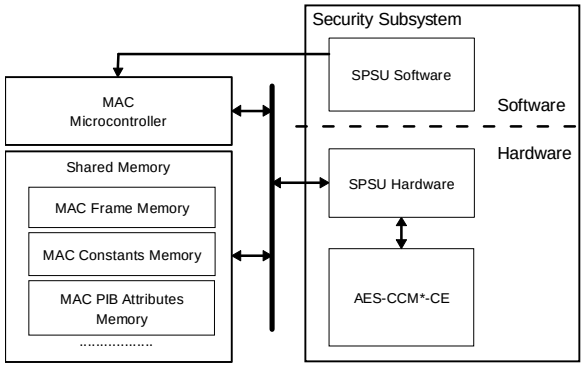

Fig. 8. The security subsystem with hardware-software general partitioning

Future task that has to be solved in this paper is security procedures elaboration in sense of their efficient hardware-software partitioning.

H. GENERAL APPROACHES FOR THE SECURITY SUBSYSTEM ARCHITECTURE

The most resource-intensive and power-consuming security subsystem module is the AES-CCM*-CE. It is used in both securing and unsecuring processes. There are two security subsystem architecture options in sense of AES-CCM*-CE usage.

8) The security subsystem architecture with the joined Rx/Tx path

Security subsystem architecture with the joined Rx/Tx path is shown in Fig. 9. It consumes one joined AES-CCM*-CE that is used for both securing (i.e. transmit – Tx) and unsecuring (i.e. receive – Rx) of the frames. Consequently, AES-CCM*-CE has to be able to perform both AES-CCM* forward and inverse transformations. If assume that AES-CCM*-CE may serve to process one frame in certain moment of time, then security subsystem may operate only to secure or unsecure the frame (i.e. send or receive operation).

Since security subsystem with the joined Rx/Tx path can potentially perform both securing and unsecuring of the frame, but can not perform them simultaneously, each moment of time one of SPSU hardware parts: Rx or Tx, and AES-CCM*-CE shall operate. Therefore, it is expedient to identify common parts in SPSU Rx and Tx hardware parts and to develop one Rx/Tx SPSU hardware unit.

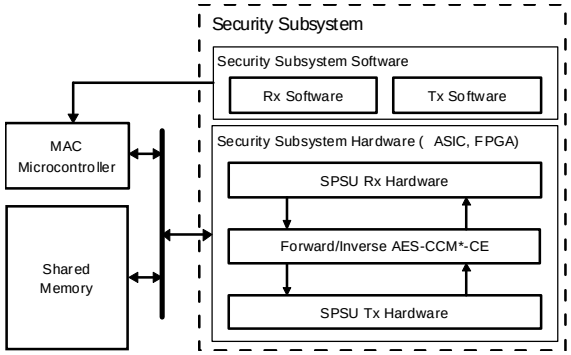

Fig. 9. The security subsystem architecture with the joined Rx/Tx path

9) Security subsystem architecture with the separate Rx and Tx paths

The security subsystem architecture with the separate Rx/Tx paths is shown in Fig. 10.

It consumes two AES-CCM*-CE modules and has fully detached Rx and Tx parts, that provides the security subsystem to operate simultaneously for both receiving and sending of the frames. The receiving part contains Rx SPSU and dedicated AES-CCM*CE. Since it operates for the unsecuring only it is enough that AES-CCM*CE performs AES-CCM* inverse transformation only. For the transmitting part the concept is similar. It operates for the securing only and contains Tx SPSU and dedicated AES-CCM*CE that performs AES-CCM* forward transformation. The optimisation of the AES-CCM*CE to perform one-direction processing lows down needed resources and power consumption.

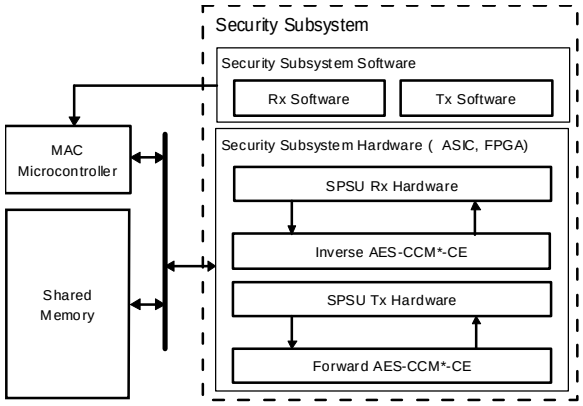

Fig. 10. The security subsystem architecture with the separate Rx and Tx paths

Given architecture imposes additional requirements to the shared memory organisation: probability of simultaneous execution of the MAC PIB lookup procedures in Rx and Tx paths requires simultaneous access to the memory.

VII. IEEE 802.15.4 SECURITY PROCEDURES IMPLEMENTATION OPTIONS

Since hardware-software approach has been selected for the security subsystem implementation, the aim of this Section is to estimate an expediency of the security procedures hardware and software implementation. During the estimation the following assumptions are being used:

1. The security subsystem shall be implemented both in hardware and software;
2. The AES-CCM* transformation is implemented in hardware;
3. The lookup procedures might be implemented in hardware or in software; keys and other security-related information from MAC PIB may initially be retained in hardware or in software.

I. OPERATION TYPES USED IN SECURITY PROCEDURES

The analysis of the security procedures shows that three main types of operations that are used there can be determined, as follows:

- Operations on originating primitive parameters, MAC PIB security attributes, and MAC constants;
- Operations on a frame fields;
- Lookup operations on MAC PIB security tables.

The operations on originating primitive parameters, MAC PIB security attributes, and MAC constants consist in their values analysis and usage for frame fields forming. Since these values have usually short length (in bits) and the operations are simple (comparing, assignment), they can easily be implemented by hardware or by software, depending on where the results have to be used (i.e. in software or in hardware part).

The operations on the frame fields involve analysis of the frame fields values and usage of the frame fields to form the resulting data sets and internal values. Although these operations are simple as well (comparing, assignment) however they need to operate with long length frame fields (e.g. MAC Header, MAC Payload) and software implementation may be ineffective here. Furthermore, operating with the variable length frame fields, as MAC Payload is, requires additional analysis of the frame, to determine the starting bit number of corresponding field in a frame and field length. Thus, the algorithm of the frame analysis and subfields extraction must be developed. These are the overhead expenses which can be eliminated, if operating only on originating primitive parameters but not on a frame fields.

The lookup operations on MAC PIB security tables are performed in MAC PIB. These operations require significant amount of processor time, since the tables are big enough (further estimations shall be done) and sometimes include several embedded lookups. If the tables are implemented in hardware memory, then the lookup operations can be implemented in software or in hardware as well. If they are in software, then the lookup operations shall be implemented in software only.

J. OUTGOING FRAME SECURITY PROCEDURE GENERAL IMPLEMENTATION OPTIONS

The outgoing frame security procedure steps are overviewed in Section V (A) “The frame securing process”. These steps are executed sequentially by default. In order to optimize execution time it would be expedient to analyze interdependencies among the steps. In order to do this it is necessary to develop the directed flow-graph [10] of the procedure algorithm. The nodes of the graph shall be the procedure steps. The layers of the graph are characterized so that the first layer nodes use only the input data of the algorithm, the nodes of each following layer use the outputs of the previous layer nodes and the input data possibly.

According to the analysis the steps can be executed in semi-parallel manner in 4 conditional periods (layers), as it is shown in Fig. 11, instead of 9 layers if sequential steps execution is performed.

The parallel (or semi-parallel) execution of procedure steps is reasonable for procedure hardware and hardware-software implementation, as it provides higher efficiency of hardware usage, reduces the execution time, and has no impact on the resources.

The procedure uses two types of operations: a) operations on originating primitive parameters, MAC PIB security attributes, and MAC constants, and b) operations on a frame fields. The operations on originating primitive parameters and the operations of the analysis of the frame fields’ values could be effectively implemented either on software and hardware. The implementation effectiveness of the operations of usage of the frame fields to form the resulting data sets and internal values depends on the size of appropriate fields.

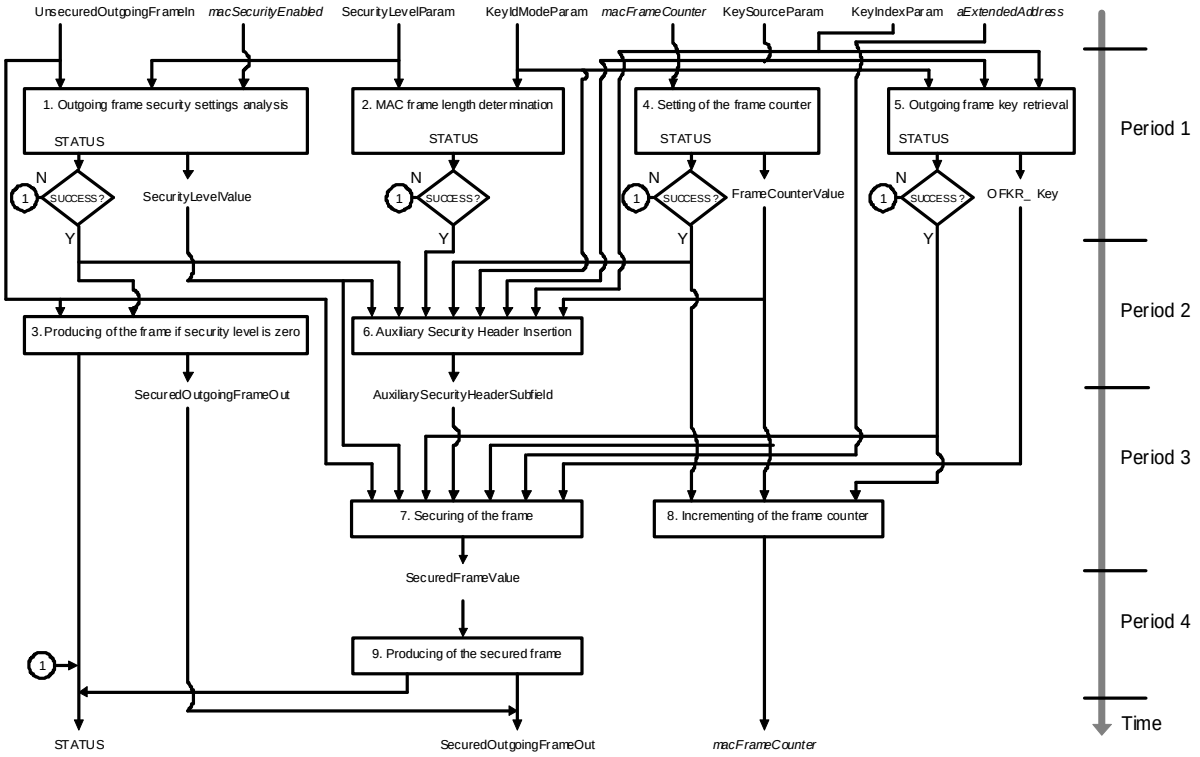

Fig. 11. The semi-parallel execution of the outgoing frame security procedure steps

10) The "Securing of the frame" implementation options

Since securing of the frame is a basic step in the outgoing frame security procedure it shall be analyzed first. The structure of the step is shown in Fig. 12. This step consists in execution of three actions:

- Forming of the input data sets for the AES-CCM* forward transformation;
- Execution of the AES-CCM* forward transformation;
- Forming of the secured MAC frame (i.e. the substitution of the unsecured payload fields in the original unsecured frame with the secured payload fields right-concatenated with the authentication tag).

The first and third actions use two types of operations: the operations on the frame fields and the operations on the MAC constant. For the input data sets forming the following input values are used: the frame to be secured (UnsecuredOutgoingFrameIn), *aExtendedAddress*, the frame counter (FrameCounterValue), the security level (SecurityLevelValue), and Auxiliary Security Header.

The values FrameCounterValue and Security LevelValue together with the *aExtendedAddress* are used to form the nonce. They can be taken from the dedicated inputs extracted from the Auxiliary Security Header, where they are present as well. If the Auxiliary Security Header is formed in hardware, the second approach is preferable.

The frame to be secured is used to form the unsecured payload fields and non-payload fields data sets according to the frame type. In order to form them it is necessary to:

- Determine the frame type;
- Perform assignments and concatenations of the appropriate frame fields.

Determination of the frame type is easy to implement as the FrameType subfield is 3 bits in length and has a fixed position in the first octet of the frame. The operations of the frame subfields assignments are more complicated to implement as it requires determination of the actual frame payload starting bit position and extraction of all subfields in the payload. Since the subfields lengths are variable, it is necessary to perform frame analysis.

The following data sets: unsecured payload fields and non-payload fields, and the Auxiliary Security Header are used to form the *a*_data and *m*_data. These data sets are formed according to the specified security level. In order to form them it is necessary to:

- Determine the security level;
- Perform assignments and appropriate concatenations of the unsecured payload fields and non-payload fields and the Auxiliary Security Header.

Determination of the security level is easy to implement as the SecurityLevel subfield is 3 bits in length and has a fixed position in the Auxiliary Security Header. The operations of the assignments and concatenations are easy to implement as well, since all values are already determined. As the resulting data sets are used as inputs for hardware-implemented AES-CCM* transformation, it is reasonable to perform assignments and concatenations in hardware.

Several scenarios of the data processing in the "securing of the frame" step are possible, depending

where the initial frame is stored and where the resulting frame should be placed. These scenarios determine hardware and software responsibilities during the step execution. We assume that the key is already available in hardware; other security-related information, like originating primitive parameters, is retained in software.

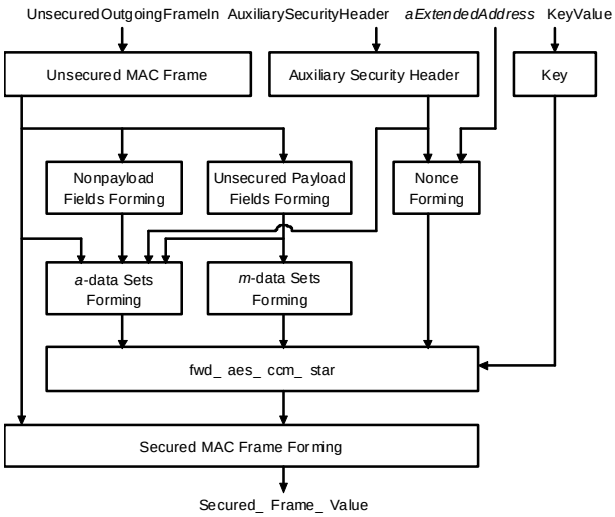

Fig. 12. The structure of the securing of the frame step

Scenario 1: initial unsecured frame is retained in software.

The following implementation options consistent with this scenario are determined:

1. Encryption software support:

Software responsibilities:

- Frame analysis and subfields extraction. Forming of unsecured payload fields and non-payload fields;
- Auxiliary Security Header forming;
- *a*_data and *m*_data data sets forming;
- Nonce forming;
- Transmission of *a*_data, *m*_data, and nonce into hardware;
- If the secured frame has to be transferred back from hardware into software: Secured MAC frame forming.

Hardware responsibilities:

- AES-CCM* forward transformation execution according to the specified security level;
- If the secured frame has to be transferred back from hardware into software: transmission of the *c*_data into software.Else: Secured MAC frame forming.

2. Encryption hardware-software support:

Software responsibilities:

- Frame analysis and subfields extraction. Forming of unsecured payload fields and non-payload fields;
- Transmission of unsecured payload fields and non-payload fields into hardware;
- Transmission of frame counter value and SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters into hardware.
- If the secured frame has to be transferred back from hardware into software: secured MAC frame forming.

Hardware responsibilities:

- Auxiliary Security Header forming;
- *a*_data and *m*_data data sets forming;
- Nonce forming;
- AES-CCM* forward transformation execution according to the specified security level;
- If secured frame has to be transferred back from hardware into software: transmission of the *c*_data into software.

Else: Secured MAC frame forming.

3. Encryption hardware support:

Software responsibilities:

- Transmission of the frame to be secured into hardware;
- Transmission of frame counter value and SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters into hardware.

Hardware responsibilities:

- Frame analysis and subfields extraction. Forming of unsecured payload fields and nonpayload fields;
- Auxiliary Security Header forming;
- *a*_data and *m*_data data sets forming;
- Nonce forming;
- AES-CCM* forward transformation execution according to the specified security level;
- Secured MAC frame forming;
- If the secured frame has to be transferred back from hardware into software: Transmission of the secured MAC frame into software.

Scenario 2: initial unsecured frame is retained in hardware.

The following implementation options consistent with this scenario are determined:

1. Encryption hardware support:

Software responsibilities:

- Transmission of frame counter value and SecurityLevel, KeyIdMode, KeySource, and KeyIndex parameters into hardware.

Hardware responsibilities:

- Frame analysis and subfields extraction. Forming of unsecured payload fields and non-payload fields;
- Auxiliary Security Header forming;
- *a*_data and *m*_data data sets forming;
- Nonce forming;
- AES-CCM* forward transformation execution according to the specified security level;
- Secured MAC frame forming.
- If the resulting secured frame has to be transferred into software: transmission of the secured MAC frame into software.

The options predefine implementation for the Auxiliary Security Header insertion step. This step has to be implemented in software in case of encryption software support and in hardware otherwise.

11) "Producing of the secured frame" implementation options

This step consists in trivial output assignment and its implementation is fully dependent on the "securing of the frame" implementation. If the secured MAC frame is formed in software, then this step shall be implemented in software. Else this step shall be implemented in hardware.

12) "Auxiliary Security Header insertion" implementation options

Input data for this step are SecurityLevelValue, FrameCounterValue, KeyIdModeParam, KeySourceParam, and KeyIndexParam. The output is formed by the Auxiliary Security Header.

Implementation of Auxiliary Security Header insertion shall be done according to the following: in software in case of encryption software support and in hardware otherwise. Either hardware or software approach is used the step is easy to implement as it consists in concatenations of the predefined input values. There is a possibility to integrate this step with the "securing of the frame" step in Period 3 according to 01.

13) "Producing of the frame if the security level is zero" implementation options

According to [1] and if the procedure is executed sequentially, this step is executed after the secured MAC frame length determination. The sense of such order is in the following. After security processing the frame may expand and may finally exceed the maximum accepted length, since the Auxiliary Security Header and the authentication tag are appended to the frame. However, if the security is disabled, then the length of the Auxiliary Security Header and the authentication tag are zero, and the frame length as well as the frame itself shall not be changed. Since the frame length initial check is performed outside of the security subsystem before the frame is going to be secured, in this case no check is required. It is, thus, reasonable to produce the secured frame if the security level is zero just after the security settings analysis.

Due to the simplicity (trivial output assignment) this step can practically be integrated with the "outgoing frame security settings analysis" step in Period 1 according to Fig. 11. The implementation options depend on the "outgoing frame security settings analysis" implementation options.

14) The "Outgoing frame security settings analysis" implementation options

The security settings are analyzed in this step. Its inputs are SecurityEnabled subfield, SecurityLevel Param, and *macSecurityEnabled*. The outputs are SecurityLevelValue, which is used for the Auxiliary Security Header forming and for the frame securing, and status. If the status is OK, then the procedure execution goes on; else the procedure is terminated with the status UNSUPPORTED_SECURITY.

Optimized pseudo code for this step is shown below.

```
IF SecurityEnabledSubfield = '1' THEN
    SecurityLevelValue <= SecurityLevelParam;
    IF (SecurityLevelValue = "000") OR
(macSecurityEnabled = FALSE) THEN
      STATUS <=
UNSUPPORTED_SECURITY;
    ELSE
      STATUS <= OK;
    END IF;
  ELSE
    SecurityLevelValue <= "000";
    STATUS <= OK;
  END IF;
```

Two types of operations are used in this step: operations on the frame subfields and operations on MAC PIB attribute.

The SecurityEnabled subfield is easy to determine as it is 1 bit in length and has a fixed position in the first octet of the frame. The SecurityLevelParam is 3 bit in length. Thus, this step is easy to implement in hardware or in software as well.

The hardware implementation is reasonable if the frame to be secured and *macSecurityEnabled* MAC PIB attribute are initially retained in hardware. Otherwise, software implementation is expedient.

15) The "Secured MAC frame length determination" implementation options

In order to estimate the implementation options for this step the following prerequisites shall be used:

- The frame length initial check is performed by MAC sublayer software outside of the security subsystem before the frame is going to be secured. Thus, value of the MAC frame length before it goes to be secured is in software possession;
- The input data for this step are KeyIdMode parameter and SecurityLevelValue. The output for this step is status.

The opportunity to use Security Level parameter instead of SecurityLevelValue brings an advantage of the parallel execution of the security settings analysis and the MAC frame length determination steps.

Since the step consists in the execution of arithmetic calculations and uses the originating primitive parameters and the value of the MAC frame length, which originate from software, it is expedient to implement it in software.

16) The "Setting of the frame counter" implementation options

The input for this step is *macFrameCounter* attribute. The step consists in checking of its value and its assigning to the FrameCounterValue, which is used to form the Auxiliary Security Header and nonce.

Since this step is concerned with the "incrementing of the frame counter" step it makes sense to consider them both and summarize implementation options then.

17) The "Incrementing of the frame counter" implementation options

The input for this step is FrameCounterValue. The incremented result shall be written into the *macFrameCounter* MAC PIB attribute.

This step shall be performed after security settings analysis, secured MAC frame length determination, and setting of the frame counter, but only if their statuses are OK.

Hardware implementation for both steps requires 32-bit comparator (or alternatively, 31 2-AND logic gates), 32-bit adder, and two 32-bit registers, so the resources are significant enough.

Incrementing operation is not time-critical and may be easy performed in software just after "setting of the frame counter" step. Thus, software implementation seems to be more expedient for both steps (setting of the frame counter and incrementing of the frame counter).

18) "Outgoing frame key retrieval" implementation options

This step represents a separate procedure. Its implementation options are out of scope of the present paper. It is necessary that the step execution has been completed before the "securing of the frame" is started.

K. INCOMING FRAME SECURITY PROCEDURE IMPLEMENTATION OPTIONS

The incoming frame security procedure steps are overviewed in Section V (B) "Frame unsecuring process". These steps are executed sequentially by default. In order to optimize execution time, as well as for the outgoing frame security procedure, it would be expedient to develop the directed flow-graph [10] of the procedure algorithm.

According to the analysis the steps can be executed in semi-parallel manner in 4 conditional periods (layers), as it is shown in Fig. 13, instead of 11 layers if sequential steps execution is performed. The procedure uses two types of operations: a) operations on the originating primitive parameters, MAC PIB security attributes, and MAC constants, and b) operations on the frame fields.

19) "Unsecuring of the frame" implementation options

- Since the unsecuring of the frame is a basic step in the incoming frame security procedure it shall be analyzed first. In fact, this step has a similar structure with the securing of the frame step in the outgoing frame security procedure. The structure of the step is shown in Forming of the input data sets for the AES-CCM* inverse transformation;
- Execution of the AES-CCM* inverse transformation;
- Forming of the unsecured MAC frame (i.e. the substitution of the secured payload fields right-concatenated with the authentication tag in the original secured frame with the unsecured payload fields) if ccm_mic_status output has PASSED value.

The first and third actions use two types of operations: the operations on the frame fields and the operations on the MAC PIB security attributes. For the input data sets forming following input values are used: frame to be unsecured (SecuredIncomingFrameIn), security level (SecurityLevelValue), frame counter (FrameCounterValue), ExtAddress element of the DeviceDescriptor, and Key element of the KeyDescriptor.

The SecurityLevelValue can be taken from the dedicated input or from the Auxiliary Security Header of the SecuredIncomingFrameIn, where it is present as well.

The following data sets: unsecured payload fields, non-payload fields, *a*_data, and *c*_data, are formed in the similar manner as during the "securing of the frame" step. In order to form them the frame analysis and subfields extraction shall be performed first.

Several scenarios of the data processing in the "unsecuring of the frame" step are possible, depending where the initial frame is stored and where the resulting frame should be placed. These scenarios determine hardware and software responsibilities during the step execution. We assume that the key is already available in hardware; other security-related information, like frame counter value, is retained in software.

Scenario 1: initial secured frame is retained in software.

The following implementation options consistent with this scenario are determined:

1. Decryption software support:

Software responsibilities:

- Frame analysis and subfields extraction. Forming of secured payload fields and non-payload fields;
- *a*_data and *c*_data data sets forming;
- Nonce forming;
- Transmission of *a*_data, *c*_data, and nonce into hardware;
- If unsecured frame has to be transferred back from hardware into software: Unsecured MAC frame forming.

Hardware responsibilities:

- AES-CCM* inverse transformation execution according to the specified security level;
- If unsecured frame has to be transferred back from hardware into software: transmission of the *m*_data into software; Else: Unsecured MAC frame forming.

2. Decryption hardware-software support:

Software responsibilities:

- Frame analysis and subfields extraction. Forming of secured payload fields and non-payload fields;
- Transmission of secured payload fields and non-payload fields into hardware;
- Transmission of frame counter value and ExtAddress element of the Key Descriptor into hardware;
- If unsecured frame has to be transferred back from hardware into software: unsecured MAC frame forming.
- Hardware responsibilities:
- *a*_data and *c*_data data sets forming;
- Nonce forming;
- AES-CCM* inverse transformation execution according to the specified security level;

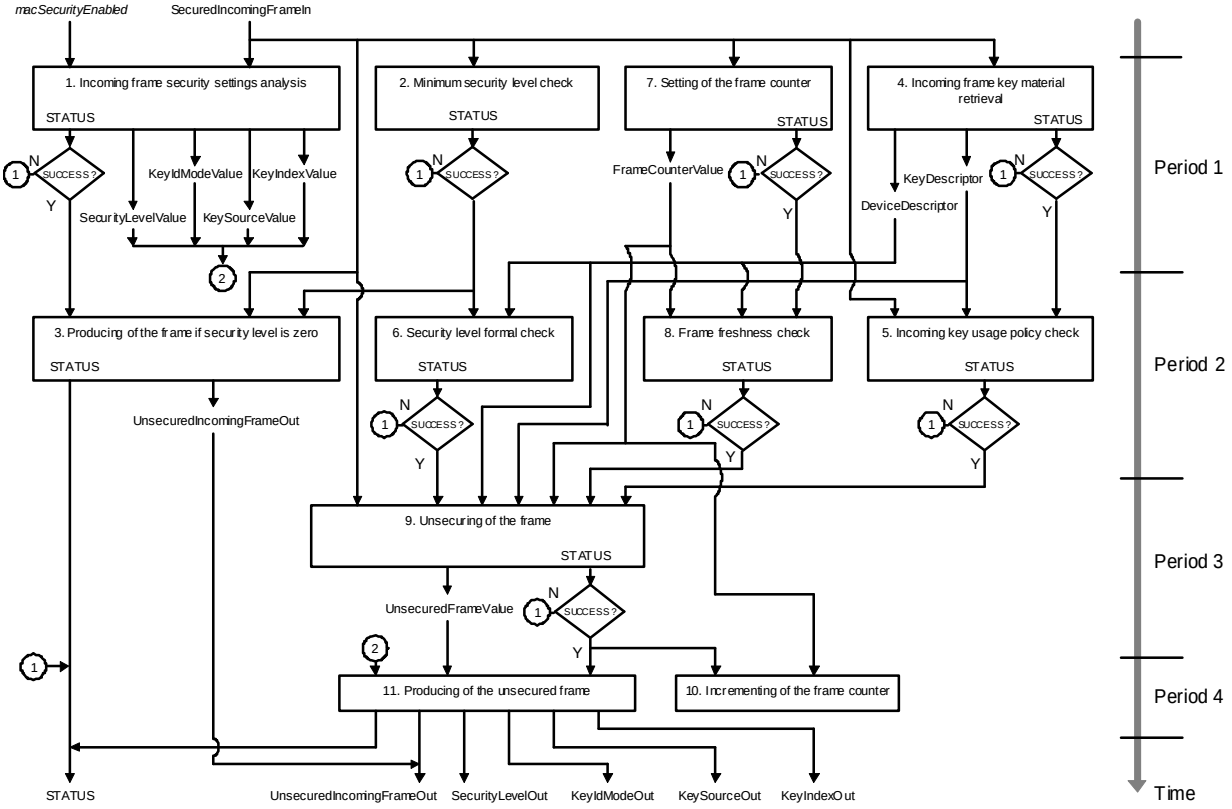

Fig. 13. The semi-parallel execution of incoming frame security procedure steps

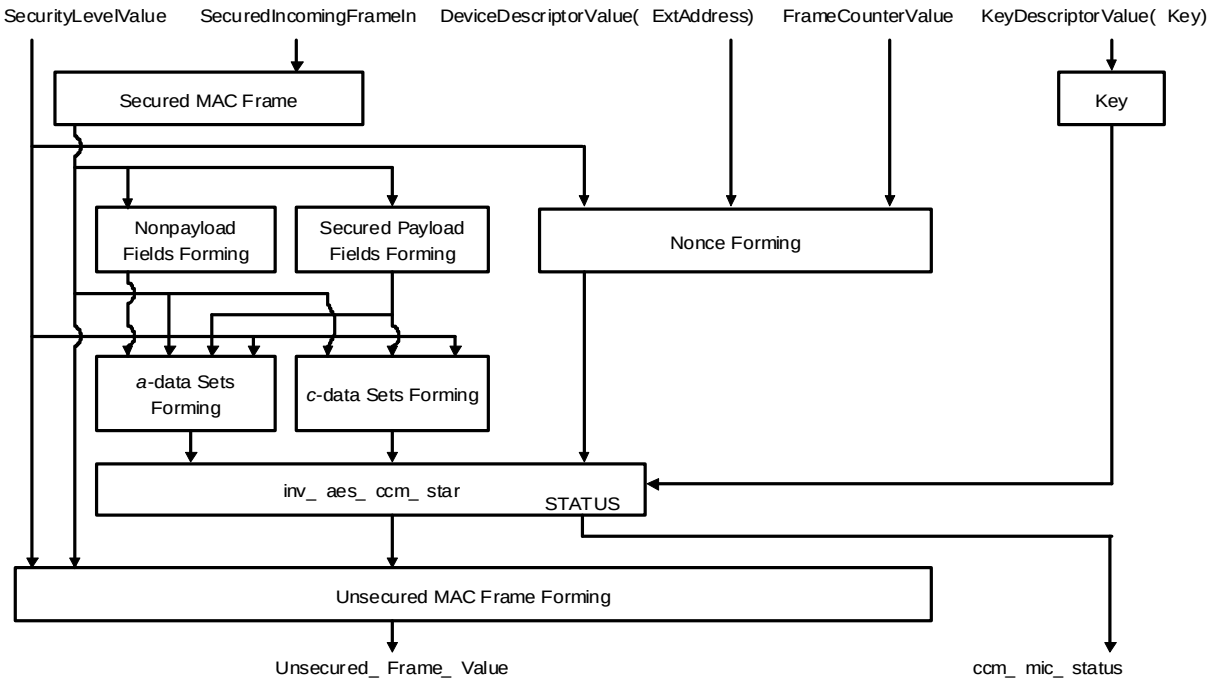

Fig. 14. The structure of the unsecuring of the frame step

- If unsecured frame has to be transferred back from hardware into software: transmission of the *m*_data into software; Else: Unsecured MAC frame forming.

3. Decryption hardware support:

Software responsibilities:

- Transmission of the frame to be unsecured into hardware;
- Transmission of frame counter value and ExtAddress element of the Key Descriptor into hardware;

Hardware responsibilities:

- Frame analysis and subfields extraction. Forming of secured payload fields and non-payload fields;
- *a*_data and *c*_data data sets forming;
- Nonce forming;
- AES-CCM* inverse transformation execution according to the specified security level;
- Unsecured MAC frame forming;
- If unsecured frame has to be transferred back from hardware into software: Transmission of the unsecured MAC frame into software.

Scenario 2: initial secured frame is retained in hardware.

The following implementation options consistent with this scenario are determined:

1. Decryption hardware support:

Software responsibilities:

- Transmission of frame counter value and ExtAddress element of the Key Descriptor into hardware.

Hardware responsibilities:

- Frame analysis and subfields extraction. Forming of secured payload fields and non-payload fields;
- *a*_data and *c*_data data sets forming;
- Nonce forming;
- AES-CCM* inverse transformation execution according to the specified security level;
- Unsecured MAC frame forming.
- If the resulting unsecured frame has to be transferred into software: Transmission of the unsecured MAC frame into software.

20) "Producing of the unsecured frame" implementation options

Implementation options for this step are fully dependent on the "unsecuring of the frame" implementation. If the unsecured MAC frame is formed in software, then this step shall be implemented in software. Else then this step shall be implemented in hardware.

21) "Incoming frame security settings analysis" implementation options

Two types of operations are used in this step: the operations on the frame fields and operations on the MAC PIB attribute. Input data are incoming frame to be unsecured and *macSecurityEnabled* MAC PIB attribute. The outputs are SecurityLevel, KeyIdMode, KeySource, and KeyIndex values, and status. The step implies execution of the simple operations and can be easily implemented either in hardware or software. The output values are formed by assignments of the values from the corresponding frame subfields. But the frame analysis and subfields extraction shall be done first in order to form them.

If the frame analysis and subfields extraction is implemented in hardware, it is reasonable to implement the entire step in hardware as well, since all used subfields are in hardware possession. Else it is reasonable to implement the step in software.

Since the frame must be processed in real time, software implementation of the frame analysis and subfields extraction may be ineffective. In order to perform timing estimations the algorithm of the frame analysis and subfields extraction has to be developed.

22) "Minimum security level check" implementation options

This step represents a separate procedure, which belongs to the lookup procedures. Its general implementation options were discussed earlier.

In order to execute this step the frame analysis and subfields extraction has to be performed first. It would be reasonable to perform the frame analysis and subfields extraction once in Rx path and use its results in all steps.

23) "Incoming frame key material retrieval" implementation options

This step represents separate procedure. Its implementation options are out of scope of the present paper.

24) "Incoming key usage policy check" implementation options

This step represents a separate procedure, which belongs to the lookup procedures. Its general implementation options were discussed earlier.

25) "Producing of the unsecured frame if the security level is zero" implementation options

This step consists in trivial output assignment. However its execution is possible only after security settings analysis and minimum security level check if their statuses are passed.

It is reasonable to implement this step in hardware if the frame analysis and subfields extraction before the "incoming frame security setting analysis" step is implemented in hardware as well. Otherwise, it is reasonable to implement this step in software.

26) "Security level formal check" implementation options

The inputs for this step are Exempt element of the DeviceDescriptor and status of the incoming security level checking procedure.

Implementation of this step depends on implementation of the data fetch and analysis facilities of the blacklist checking procedure and incoming security level checking procedure (i.e. whether appropriate DeviceDescriptor is retained in software or in hardware as well as the status of incoming security level checking procedure). It is reasonable to implement this step in hardware if they are retained in hardware, otherwise in software.

27) "Setting of the frame counter" implementation options

The input for this step is FrameCounter subfield of the frame to be unsecured.

Hardware implementation of this step requires 32-bit comparator (or alternatively 31 2-AND logic gates) and 32-bit register, so the amount of hardware resources is significant enough. Hence, we would suggest it to implement in software and transmit the result to hardware if the status is passed.

28) "Incrementing of the frame counter" implementation options

The input for this procedure is FrameCounterValue. The incremented result shall be written into the corresponding element of the DeviceDescriptor.

This step shall be executed after unsecuring of the frame, but only if ccm_mic_status is passed.

Hardware resources to implement this step consist in 32-bit adder and 32-bit register. Due to this factor we would suggest it to implement this step in software. Although it seems logically reasonable to implement the step in hardware if the KeyDeviceDescriptor and DeviceDescriptor are retained in hardware and in software in other case, since step's results are used to update appropriate entries of the KeyDeviceDescriptor and DeviceDescriptor.

29) "Frame freshness check" implementation options

The inputs for this procedure are Frame CounterValue and FrameCounter element of the DeviceDescriptor.

This step shall be executed after setting of the frame counter and incoming frame key material retrieval, so the appropriate FrameCounterValue and KeyDescriptor are available.

Hardware resources to implement this step consist in 32-bit comparator. Moreover, this step needs an access to KeyDescriptor. Due to these factors we would suggest it to implement in software.

VIII. CONCLUSIONS

The conclusions summarize options for the security procedures implementation and consequent security subsystem architecture in the IEEE 802.15.4 compatible devices.

1	**SW**	Outgoing frame security settings analysis, Producing of the frame if the security level is zero		
2	**SW**	MAC frame length determination		
3	**SW**	Setting of the frame counter		
4	**SW**	Incrementing of the frame counter	**HW-SW**	Auxiliary security header insertion, Securing of the frame, Producing of the secured frame

Fig. 15. The proposed implementation of the outgoing frame security procedure

30) Outgoing frame security procedure implementation options summary

The proposed implementation of the outgoing frame security procedure regardless the outgoing frame key retrieval, which represents separate procedure, is shown in Fig. 15. The procedure is executed in four conditional periods. Indexes HW (hardware) and SW (software) show the proposed implementation option for each step. The steps within the fourth period are executed in parallel. Note, that the key should be available before the fourth period.

1. The following steps: "Auxiliary Security Header insertion", "securing of the frame", and "producing of the secured frame", should be combined into one step and executed in one conditional period.

2. The following steps: "outgoing frame security settings analysis" and "producing of the frame if the security level is zero", should be combined into one step and executed in one conditional period.

3. It is suggested, that software implementation is expedient for the "secured MAC frame length determination" step. This statement gives challenges to the implementation of the "outgoing frame security settings analysis" step. Potentially, the last can be executed in parallel with the "secured MAC frame length determination" if it is hardware-implemented. It has to be executed before the last if it is software-implemented. Paralleling of these steps where their outputs are combined into another step adds quite a bit of complexity to the system. Also, it requires the output data to be sent back from hardware into software, what is ineffective. Hence, for both steps: the "outgoing frame security settings analysis" and the "secured MAC frame length determination" software implementation is expedient.

4. The "incrementing of the frame counter" step can be executed in parallel with the "securing of the frame" step, since their outputs aren't combined.

5. The "incrementing of the frame counter" step can be executed just after "setting of the frame counter step" if the last passed successfully, regardless whether the "securing of the frame" step execution is finished.

6. Implementation of the "securing of the frame" step has no impact on implementation of other steps except the "Auxiliary Security Header insertion" and "producing of the secured frame" steps.

31) Incoming frame security procedure implementation options summary

The proposed implementations of the incoming frame security procedure regardless incoming frame key material retrieval (including "incoming key usage policy check" step), which represents separate procedure, are shown in Fig. 16. Indexes HW (hardware) and SW (software) show the proposed implementation option for each step. Note, that the key should be available before execution of the "security level formal check" step.

1. The "incoming key usage policy check" step should be transmitted to the incoming frame key material retrieval procedure.

1	**SW**	Incoming frame security settings analysis
2	**SW**	Minimum security level check
3	**SW**	Producing of the unsecured frame if the security level is zero
5	**SW**	Security level formal check
6	**SW**	Setting of the frame counter
7	**SW**	Frame freshness check
8	**SW - HW**	Unsecuring of the frame, Producing of the unsecured frame
9	**SW**	Incrementing of the frame counter

a) Software lookup procedure implementation, software frame analysis and subfields extraction

1	**HW**	Incoming frame security settings analysis	**SW**	Minimum security level check
2	**HW**	Producing of the unsecured frame if the security level is zero	**SW**	Security level formal check
4	**SW**	Setting of the frame counter		
5	**SW**	Frame freshness check		
6	**SW - HW**	Unsecuring of the frame, Producing of the unsecured frame		
7	**SW**	Incrementing of the frame counter		

b) Software lookup procedure implementation, hardware frame analysis and subfields extraction

1	**SW**	Incoming frame security settings analysis	**HW**	Minimum security level check
2	**SW**	Producing of the unsecured frame if the security level is zero	**HW**	Security level formal check
3	**SW**	Setting of the frame counter		
4	**SW**	Frame freshness check		
5	**SW - HW**	Unsecuring of the frame, Producing of the unsecured frame		
6	**SW**	Incrementing of the frame counter		

c) Hardware lookup procedure implementation, software frame analysis and subfields extraction

1	**HW**	Incoming frame security settings analysis	**HW**	Minimum security level check
2	**HW**	Producing of the unsecured frame if the security level is zero	**HW**	Security level formal check
3	**SW**	Setting of the frame counter		
4	**SW**	Frame freshness check		
5	**SW - HW**	Unsecuring of the frame, Producing of the unsecured frame		
6	**SW**	Incrementing of the frame counter		

d) Hardware lookup procedure implementation, hardware frame analysis and subfields extraction

Fig. 16. The proposed implementations of the incoming frame security procedure

2. The following: “minimum security level check” step consists in the execution of the incoming security level checking procedure, which belongs to the lookup procedures. No certain option for the lookup procedures implementation has been proposed yet. Therefore, two implementation options: software and hardware, which are potentially possible, we will consider in this summary.

3. The lookup procedures implementation approach determines the implementation for the “security level formal check” step and may impact implementation for the “incrementing of the frame counter” and the “frame freshness check” steps.

4. Initial operation that has to be performed before the procedure execution is frame analysis and subfields extraction. The frame analysis algorithm has not been developed and its certain implementation approach has not been proposed yet. Therefore two implementation options: software and hardware, which potentially are possible, we will consider in this summary.

1. The frame analysis and subfields extraction implementation approach determines the implementation for the “incoming frame security settings analysis” and “producing of the frame if the security level is zero” steps.

2. The following steps: “unsecuring of the frame” and “producing of the unsecured frame”, should be combined into one step and executed in one conditional period.

3. The following steps: “setting of the frame counter”, “frame freshness check”, and “incrementing of the frame counter” should be implemented in software from the hardware resources point of view.

4. It is reasonable to execute the following steps: “producing of the frame if the security level is zero” and the “security level formal check” in parallel, if they both are implemented in hardware, or one of them is in hardware and another one in software, since their outputs are not combined.

5. It is reasonable to execute the following steps: the “incoming frame security settings analysis” and the “minimum security level check” in parallel, if they both are implemented in hardware, or one of them is in hardware and another one in software, because of different nature of these steps.

6. The implementation of the “unsecuring of the frame” step has no impact on the implementation of other steps except “producing of the unsecured frame”.

REFERENCES

[1] IEEE Std 802.15.4TM, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Second edition, September 2006.

[2] IEEE Std 802.15.4TM 2011, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs). Revision of IEEE Std 802.15.4-2006, Approved 14 August 2012 by American National Standards Institute.

[3] Gascón, David (February 5, 2009). "Security in 802.15.4 and ZigBee networks". [Online]. Available: http://www.libelium.com/security-802-15-4-zigbee/ [Accessed: Nov. 25, 2018].
[4] ISA100 Committee Home Page: https://www.isa.org/isa100/ [Accessed: Nov. 25, 2018]
[5] Federal Information Processing Standards (FIPS) Publication 197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). November 26, 2001. Available at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf [Accessed: Nov. 25, 2018]
[6] NIST Special Publication 800-38C. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. May 2004.
[7] Rogaway, P., and Wagner, D., "A Critique of CCM", IACR ePrint Archive 2003-070, April 13, 2003.
[8] Melnyk V. Korkishko T., Melnyk A., Algorithms and Processor for the Symmetric Block Ciphering. Lviv. BAK, 2003, 187 p.
[9] Security requirements for cryptographic modules. Federal information proceedings standard publication 140-2, 1999. – 50 p.
[10] Melnyk A. Digital Signal Processors. Preprint N 29-89. Applied Problems of Mechanics and Mathematics Institute of the Academy of Sciences of Ukraine, Lviv. – 1989, 63 p.

Viktor Melnyk is a professor of the Department of Information Technologies Security at Lviv Polytechnic National University, Ukraine. He was awarded with the academic degrees of Philosophy Doctor in 2004, and Doctor of Technical Sciences in 2013 at Lviv Polytechnic National University.

He has scientific, academic and hands-on experience in the field of computer systems research and design, proven contribution into IP Cores design methodology and high-performance reconfigurable computer systems design methodology. He is experienced in computer data protection, including cryptographic algorithms, cryptographic processors design and implementation, wireless sensor network security. Mr. Melnyk is an author of more than 90 scientific papers, patents and monographs.