International Standard

ISO/IEC/IEEE 8802-15-9

First edition
2024-11

Telecommunications and information exchange between systems — Local and metropolitan area networks specific requirements —

Part 15-9: Transport of Key Management Protocol (KMP) Datagrams

Télécommunications et échange d'information entre systèmes — Réseaux locaux et métropolitains — Exigences spécifiques —

Partie 15-9: Transport des datagrammes du protocole de gestion des clés (KMP)

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Reference number
ISO/IEC/IEEE 8802-15-9:2024(en)

© IEEE 2022

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

COPYRIGHT PROTECTED DOCUMENT

© IEEE 2022

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from IEEE at the address below.

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted.

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 8802-15-9 was prepared by IEEE (as IEEE 802.15.9:2021) and drafted in accordance with its editorial rules. It was adopted, under the "fast-track procedure" defined in the Partner Standards Development Organization cooperation agreement between ISO and IEEE, by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

A list of all parts in the ISO/IEC/IEEE 8802 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

IEEE Std 802.15.9™-2021
(Revision of IEEE Std 802.15-9-2016)

IEEE Standard for Transport of Key Management Protocol (KMP) Datagrams

Developed by the

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 16 June 2021

IEEE SA Standards Board

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Abstract: A message exchange framework based on information elements as a transport method for key management protocol (KMP) datagrams and guidelines for the use of some existing KMPs with IEEE Std 802.15.4™ is defined in this standard. A new KMP is not created in this standard. In support of KMP transmission and reception, a generic multiplexed data service layer that can be used to transmit large packets from the upper KMP to another peer and that provides for protocol discrimination is also provided in this standard. The multiplexed data service provides a fragmentation and multiplexing layer for those packets so they can be delivered over smaller MAC layer frames and multiplexed on the recipient end to the right processing service. The multiplexing provides for EtherType protocol discrimination.

Keywords: EtherType, fragmentation, IE, IEEE 802.15.9™, information element, key management protocol, KMP, multiplexed data service, security

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2022 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 18 January 2022. Printed in the United States of America.

IEEE and 802 are registered trademarks in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-7720-8 STD24805
Print: ISBN 978-1-5044-7721-5 STDPD24805

IEEE prohibits discrimination, harassment, and bullying.
For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE SA) Standards Board. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers are not necessarily members of IEEE or IEEE SA, and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning this standard, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE is the approved IEEE standard.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and Standards Coordinating Committees are not able to provide an instant response to comments, or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or in revisions to an IEEE standard is welcome to join the relevant IEEE working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system. An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under US and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE. For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website. Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE Standards are developed in compliance with the IEEE SA Patent Policy.

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. IEEE Standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Participants

At the time this standard was completed, the IEEE 802.15 Working Group had the following membership:

Robert F. Heile, *IEEE 802.15 Working Group Chair*
Rick Alfvin, *IEEE 802.15 Working Group Vice-Chair*
Patrick W. Kinney, *IEEE 802.15 Working Group Vice-Chair, IEEE 802.15 Working Group Secretary*
James P. K. Gilb, *IEEE 802.15 Working Group Technical Editor*
Benjamin A. Rolfe, *IEEE 802.15 Working Group Treasurer*

Tero Kivinen, *802.15.9 Chair, Technical Editor*
Peter Yee, *802.15.9 Vice Chair, Secretary*

Mounir Achir
Keiji Akiyama
Richard Alfvin
Hideki Aoyama
Arthur Astrin
Philip Beecher
Frederik Beer
Kiran Bynam
Edgar Callaway
Chris Calvert
Radhakrishna Canchi
Jaesang Cha
Kapseok Chang
Soo-Young Chang
Clint Chaplin
Stephen Chasko
Paul Chilton
Sangsung Choi
Hee-Sang Chung
Hendricus De Ruijter
Guido Dolmans
Igor Dotlic
Stefan Drude
Dietmar Eggert
Shahriar Emami
Andrew Estrada
David Evans
Kiyoshi Fukui
Matthew Gillmore
Tim Godfrey
Jussi Haapola
Shinsuke Hara
Timothy Harrington
James Hartman
Marco Hernandez
Ken Hiraga
Koji Horisaki
Iwao Hosako
Bing Hui
Yeong Min Jang
Seong-Soon Joo
Hyunduk Kang
Shuzo Kato
Toyoyuki Kato
Jeritt Kent
Jaehwan Kim
Junhyeong Kim
Youngsoo Kim
Shoichi Kitazawa
Ryuji Kohno
Fumihide Kojima
Thomas Kuerner
Byung-Jae Kwak
Hoosung Lee
Jae Seung Lee
Moon-Sik Lee
Myung Lee
Huan-Bang Li
Liang Li
Qing Li
Michael Lynch
Itaru Maekawa
Hiroyuki Matsumura
Michael Mc Laughlin
Michael McInnis
Kenichi Mori
Robert Moskowitz
Mohammad Nekoui
Chiu Ngo
Paul Nikolich
John Notor
Hiroyo Ogawa
Mitsuaki Oshima
Chandrashekhar P S Bhat
Park Taejoon
Glenn Parsons
Charles Perkins
Albert Petrick
Clinton Powell
Verotiana Rabarijaona
Demir Rakanovic
Ivan Reede
RobertsRichard
Behcet Sarikaya
Noriyuki Sato
Norihiko Sekine
Nikola Serafimovski
Kunal Shah
Stephen Shellhammer
Shusaku Shimada
Masashi Shimizu
Gyung Chul Sihn
Gary Stuebing
Don Sturek
Mineo Takai
Kou Togashi
Kiyoshi Toshimitsu
Murat Uysal
Billy Verso
Gabriel Villardi
Brian Weis
Makoto Yaita
Yu Zeng
Chunhui Zhu

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Robert Aiello	Raj Jain	Thomas Starai
Philip E. Beecher	Pranav Jha	Walter Struppler
Vern Brethour	Piotr Karocki	Gary Stuebing
William Byrd	Stuart Kerry	Gerald Stueve
Paul Cardinal	Yongbum Kim	Don Sturek
Pin Chang	Tero Kivinen	Mark Sturza
Suresh Channarasappa	Yasushi Kudoh	Mark-Rene Uchida
Michael Cowan	Hyeong Ho Lee	Dmitri Varsanofiev
Hendricus De Ruijter	Rajesh Murthy	Billy Verso
Donald Dunn	Bansi Patel	Xiaohui Wang
Liu Fangfang	Arumugam Paventhan	Lisa Ward
Avraham Freedman	Clinton Powell	Scott Willy
Hongmei He	Maximilian Riegel	Andreas Wolf
Marco Hernandez	Robert Robinson	Chun Yu Charles Wong
Werner Hoelzl	Benjamin Rolfe	Yu Yuan
Klaus Hueske	Naotaka Sato	Oren Yuen
	Kunal Shah	

When the IEEE SA Standards Board approved this standard on 16 June 2021, it had the following membership:

Gary Hoffman, *Chair*
Jon Walter Rosdahl, *Vice Chair*
John D. Kulick, *Past Chair*
Konstantinos Karachalios, *Secretary*

Edward A. Addy	Howard Li	Mehmet Ulema
Doug Edwards	Daozhuang Lin	Lei Wang
Ramy Ahmed Fathy	Kevin Lu	F. Keith Waters
J. Travis Griffith	Daleep C. Mohla	Karl Weber
Thomas Koshy	Chenhui Niu	Sha Wei
Joseph L. Koepfinger*	Damir Novosel	Howard Wolfman
David J. Law	Annette Reilly	Daidi Zhong
	Dorothy Stanley	

*Member Emeritus

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Introduction

This introduction is not part of IEEE Std 802.15.9™-2021, IEEE Standard for Transport of Key Management Protocol (KMP) Datagrams.

Key management has been recognized as critical component for network security, but IEEE Std 802.15.4™ does not provide any methods for key management and leaves it out of scope. So this standard was created to provide a methodology to enable key management by providing a transport for key management protocols (KMPs) outside the application layers.

The first revision of the 802.15.9-2016 was a Recommended Practice, and this revision of the 802.15.9 will change it to an IEEE Standard.

The scope of the 2016 version of IEEE Std 802.15.9 was as follows:

This Recommended Practice defines a message exchange framework based on Information Elements as a transport method for key management protocol (KMP) datagrams and guidelines for the use of some existing KMPs with IEEE Std 802.15.4. This Recommended Practice does not create a new KMP.

The current scope of IEEE Std 802.15.9-2021 is as follows:

This standard defines security key management extensions to address session key generation (both 128-bit and 256-bit key lengths), the creation and/or transport of broadcast/multicast keys, and security algorithm agility. This standard maintains backwards compatibility with IEEE Std 802.15.9-2016.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Contents

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

IEEE Standard for Transport Key Management Protocol (KMP) Datagrams

1. Overview

1.1 General

This document defines a standard for the transport of key management protocols (KMP) for WPANs.

1.2 Scope

This standard defines security key management extensions to address session key generation (both 128-bit and 256-bit key lengths), the creation and/or transport of broadcast/multicast keys, and security algorithm agility. This standard maintains backwards compatibility with IEEE Std 802.15.9-2016.

1.3 Purpose

This standard describes support for transporting KMP datagrams to support the security functionality present in IEEE Std 802.15.4™.[1] Significant in support of KMP transport is the definition of a general purpose multiplexed (MPX) data service supporting fragmentation, re-assembly, and protocol dispatch for payloads unable to fit in a single media access control (MAC) frame.

1.4 Deprecated features

This standard deprecates the use of PANA KMP defined in the Clause D.

1.5 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals is *required to*).[2,3]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals is *recommended that*).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

[1]Information on references can be found in Clause 2.
[2]The use of the word *must* is deprecated and cannot be used when stating mandatory requirements, *must* is used only to describe unavoidable situations.
[3]The use of *will* is deprecated and cannot be used when stating mandatory requirements, *will* is only used in statements of fact.

Copyright © 2022 IEEE. All rights reserved.

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals is *permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals is *able to*).

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE Std 802.15.4™, IEEE Standard for Low-Rate Wireless Networks.[4, 5]

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[6]

extended address: The address as specified in 7.1 of IEEE Std 802.15.4-2020.

key management protocol (KMP): A collection of data transactions that provide the mechanism to manage cryptographic keys. This includes dealing with the generation, exchange, storage, use, and replacement of keys.

3.2 Acronyms and abbreviations

4WH	4-Way Handshake
ACL	access control list
AAA	authentication, authorization and accounting
AP	access point
AS	authentication server
BEX	base exchange
CAK	connectivity association key
DEX	diet exchange
EAP	extensible authentication protocol
EAPOL	EAP over LAN
GKH	Gate Key Handshake
HI	host identity

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

[4]IEEE publications are available from the Institute of Electrical and Electronics Engineers (http://standards.ieee.org/.)

[5]The IEEE standards referred to in Clause 2 are trademarks belonging to the Institute of Electrical and Electronics Engineers, Inc.

[6]IEEE Standards Dictionary Online is available at: http://dictionary.ieee.org. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page..

Copyright © 2022 IEEE. All rights reserved.

HIP	host identity protocol
HIT	host identity tag
ICV	integrity check value
IE	information element
IETF	Internet Engineering Task Force
IKEv2	Internet Key Exchange version 2
IP	Internet protocol
IPsec	IP Security
KDF	key derivation function
KMP	key management protocol
MAC	media access control
MSDU	MAC Service Data Unit
MKA	MACsec key agreement
MPX	multiplexed
MPDU	MAC protocol data unit
MSK	master session key
N2N	node to node
PAN	personal area network
PANA	protocol for carrying authentication for network access
PHY	physical layer
PIB	personal area network information base
PRF	pseudo-random function
SA	security association
SPI	security parameter index
UDP	user datagram protocol
WPAN	wireless personal area network

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

4. Introduction

4.1 Overview

IEEE Std 802.15.4 supports datagram security, but does not specify how keys are managed. A variety of proprietary, non-interoperable approaches for key management have been implemented. This standard defines KMP approaches which meet the critical needs of an effective security framework.

This standard specifies the transportation of KMP datagrams within IEEE Std 802.15.4. It also provides guidelines for KMPs such as the Internet Engineering Task Force's (IETF) host identity protocol (HIP) (Moskowitz [B19]), Internet Key Exchange version 2 (IKEv2) (Kaufman, et al. [B16]), Protocol for carrying Authentication for Network Access (PANA) (Forsberg, et al. [B6]), Dragonfly, and IEEE Std 802.1X™ [B12].[7]

IEEE Std 802.15.4y™ [B14] provides a mechanism for using other encryption algorithms in the IEEE Std 802.15.4 than AES-CCM-128. Depending on the KMP to be used this standard can be used to negotiate the algorithm.

[7]The numbers in brackets correspond to those of the bibliography in Annex G.

This standard also provides a generic MPX data service layer that can be used to transmit large packets from the upper layer process to another peer. The MPX data service provides a fragmentation and multiplexing layer for those packets so they can be delivered over smaller MAC layer frames and multiplexed on the recipient end to the right processing service.

4.2 System view

The KMP protocol is intended to operate over a variety of wireless physical layers (PHYs). Within the family of PHYs available in the IEEE Std 802.15.4, some PHYs provide low data rates and limited PHY packet sizes. Even when the PHY layer supports large frames, there are benefits for fragmenting the packets to smaller frames. In some KMPs supported by this document the message size exceeds even the largest PHY packet size supported by any PHY defined in IEEE Std 802.15.4. For these reasons, fragmentation and reassembly of the KMP messages is essential. An interoperable fragmentation/reassembly mechanism is described in this document.

The key transport mechanism described in this document sits between the MAC services and higher layers and is shown in Figure 1.

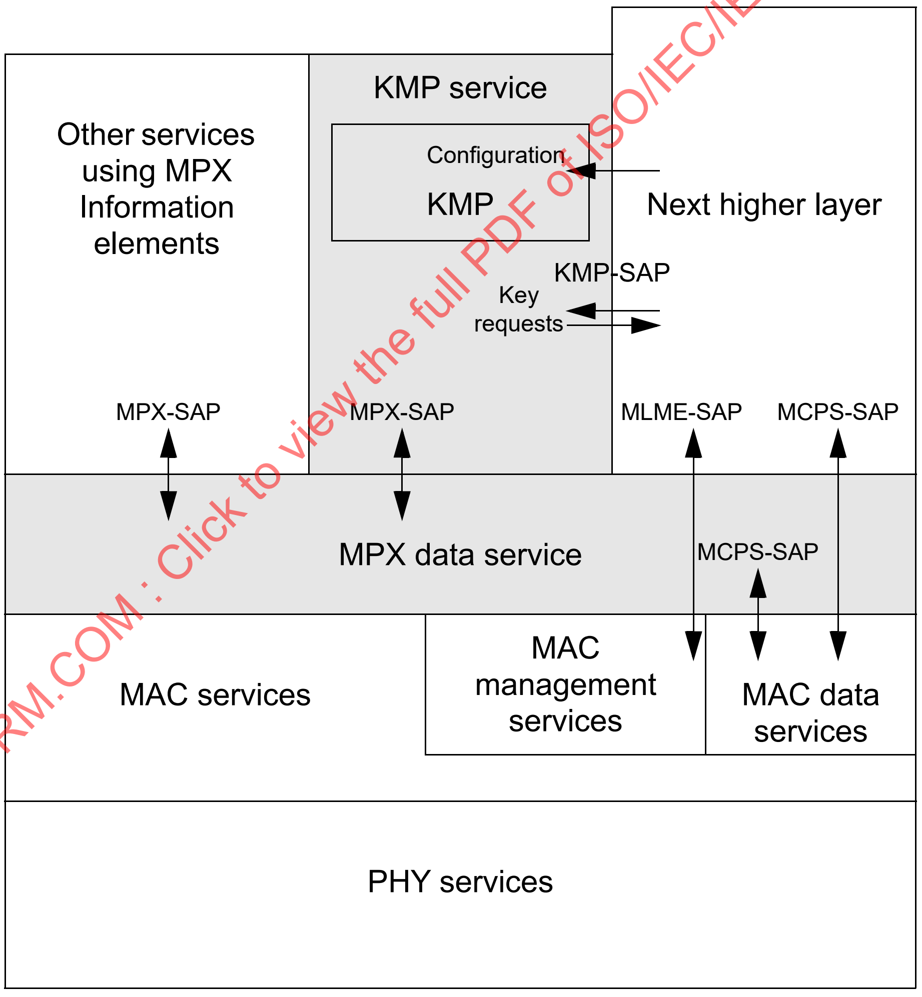

Figure 1—System overview

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

MPX data service provides the way to fragment and multiplex packets sent by the layers above it and send them out using the MAC data service. The interface between the MPX data service and the layers above it includes the MultiplexID and the payload to be sent or the payload received. The KMP Service will use MPX-DATA primitives to send and receive packets, and the next higher layer will talk to the KMP services through the KMP primitives.

The actual KMP is inside the KMP services, and the configuration of the KMP is provided by the next higher layer using a mechanism that is outside the scope of this document. The keys generated and the algorithms negotiated by the KMP are sent to the next higher layer, which will then install them to the MAC sublayer. The exact format of the keys and configuration depends on the KMP used, and is outside the scope of this document.

4.3 Network view

The KMP transport services are limited by the fragmentation/reassembly multiplexing service defined in this standard. The fragmentation defined in this document is intended for operation between two direct peer devices. This means that if applied to a multi-hop network topology, it is intended that fragmentation and reassembly be performed hop by hop.

The KMPs themselves are all capable of working over a multi-hop transport. If an implementation can ensure all KMP payloads fit in a single MAC protocol data unit (MPDU), the fragmentation/reassembly mechanism is not engaged. Further, the KMPs are normally used on lossy environments, and they have to work even when frames are lost, thus the lack of a multi-hop acknowledgment is not a limitation. Therefore if there is also a MAC forwarding mechanism, the KMP transport services will work in a multi-hop environment. It is up to the implementation to make sure that these constraints are met before using in a multi-hop environment.

The KMPs used by this standard may provide both link and group key management support.

4.4 Security associations

The security information between two devices is maintained in a security association (SA). The role of a KMP is to set up and maintain an SA between two or more devices. SAs may be unicast or group SAs.

Many types of SAs are possible as described in the security-related MAC personal area network (PAN) information base (PIB) attributes in 9.5 of IEEE Std 802.15.4-2020. An implementer of this standard will select from the options allowed to request the KMP higher layer to establish the desired SA and provide the necessary keys to the MAC PIB attribute. Short summary of the types of SAs in IEEE Std 802.15.4 can be found from Annex F.

4.5 Process flow

SAs can be established when a device joins the PAN, or any time prior to the first secured data transmission. There are different methods that can trigger the SA establishment. One is that the PIB *macSecurityEnabled* is set to TRUE in either device.

In a beaconing PAN, either device configured to require security may start the KMP. A coordinator may require SA establishment prior to association or the next step after association. A joining device may attempt SA establishment prior or after association if it is configured to only communicate over a secured link.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

In a non-beaconing PAN, if the sender of the initial data transmission is configured to require security it first establishes the SA by starting the KMP. If the recipient is configured to require security and receives a non-secured data transmission, it drops the data transmission and starts the KMP to establish the SA.

4.6 State machine

There are separate inbound and outbound processing state machines (see Clause 9).

4.7 Address formats

To be able to send or receive a packet securely, IEEE Std 802.15.4 requires that the originator and the recipient of the frame know the extended address (see IEEE Std 802.15.4) of the other end. This means that before any link keys between peers can be established, they need to know the extended address of the originator, as the SA will be associated with that. The KMPs can use that 64-bit extended address to identify the peer and use it to authenticate the other end.

4.8 KMP payload size

As the MPX data layer allows a KMP payload to be fragmented into a maximum of 256 fragments, this means that even if the PHY is using 127 octet PHY service data units (96 octets effective fragment size), this standard can use KMP payloads up to 24 576 octets. The Total Upper Layer Frame Size field of the MPX IE is two octets, thus regardless of the maximum size of the underlaying PHY the maximum KMP payload size is 65 535 octets.

4.9 Format conventions

This document follows the formatting conventions as defined in Clause 4 of IEEE Std 802.15.4-2020.

5. MPX data service

5.1 Description

The MPX data service provides a mechanism to carry multiple payloads of different higher layer services between two MAC entities, and multiplexing and fragmenting those payloads so they can be transmitted using MAC frames. The MPX data service delivers an MPX data payload from a higher protocol layer by fragmenting it into a number of MAC frames at the originator and sending them sequentially to the recipient. At the recipient, the MPX data payload is reassembled from the sequence of MAC frames and delivered to the higher protocol layer. The MPX data payload carried within MPX IEs can be sent within either Data or Multipurpose frames.

As MPX data service payloads may exceed the size of an MPDU, a fragmentation method using the MAC acknowledgment provides the needed fragmentation support. The use of the MAC acknowledgment allows the originator to be assured the receiving device has all the frames to reassemble the MPX data service payload. Retransmission of unacknowledged frames is handled by the MAC. The MPX data service shall limit each MPX IE Content field to at maximum of *macMpxMaxFragmentSize* octets.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

If the originator fails to send the data frame (for example because of the NO_ACK error status), the MPX data service shall fail the transaction and return an error to the next higher layer. As quite often there is no possibility to signal a state of failed transfer via the MAC, the recipient state machine uses a timeout defined by the *macMpxReassemblyTimeout* to determine when the originator has failed the transfer. In the MPX data service there is signaling mechanism allowing either end to request that the transfer be aborted. Recipient may request transfer to be aborted if it does not have enough resources to finish the transaction, and recipient may indicate the maximum size of transaction recipient can process in the abort message.

If a fragment is received out of order, the inbound fragmentation processing of the receiving device may be declared as failed and the transaction terminated. It is possible for the MAC layer to have acknowledged receipt of a fragment that is later dropped by a higher layer function (due to overload or lack of resources). Thus, the acknowledgment should only be taken to indicate that the fragment was received by the recipient, not that the recipient passed the fragment to the reassembly service and onto the service using the MPX IEs.

The following primitives, summarized in Table 1, are used by an entity at a higher layer to fragment and transfer an MPX payload to a remote device where it is reassembled and passed to the next higher layer.

Table 1—Summary of MPX primitives

Name	Request	Indication	Response	Confirm
MPX-DATA	5.2.2	5.2.4	—	5.2.3
MPX-PURGE	5.3.2	—	—	5.3.3

5.2 MPX data primitives

5.2.1 MPX data primitive use

Figure 2 shows a successful transfer of a payload, which must be split into three fragments for transmission by the MAC on the originator and is received and reassembled at the recipient. Figure 2 shows the next higher layer on the originator initiating the transfer by issuing the MPX-DATA.request primitive. At the recipient, on successful reassembly of the fragments into the full payload, the recipient next higher layer is signaled that a new payload has arrived using the MPX-DATA.indication. To complete the transfer, the originator next higher layer which started the transfer is signaled that the transfer completed successfully using the MPX-DATA.confirm with a status code of SUCCESS.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

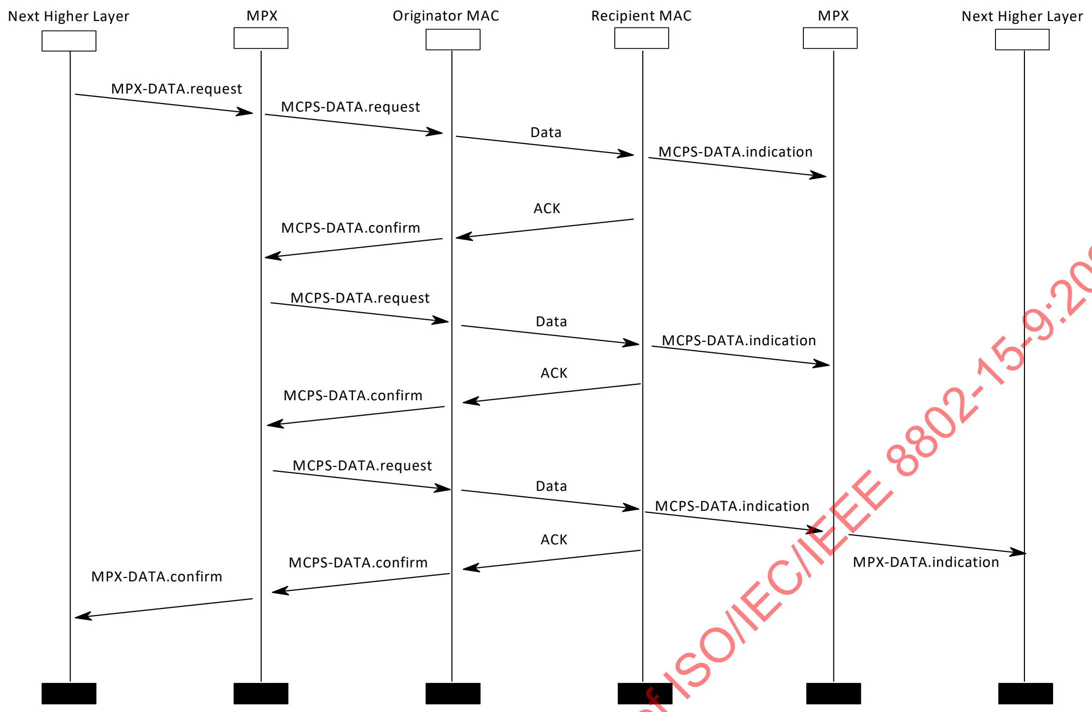

Figure 2—Message sequence for successful transfer of three fragments

Figure 3 shows an unsuccessful transfer of a fragmented payload. The figure shows the next higher layer on the originator initiating the transfer by issuing the MPX-DATA.request primitive. The transfer of the first fragment completes successfully as denoted by the ACK and MCPS-DATA.confirm. The second fragment is lost during transmission. The originator MAC (as shown in Figure 3) will try retransmitting the frame until it reaches the maximum number of retransmissions, at which point it will return failure; the MCPS-DATA.confirm returns a status of NO_ACK and the MPX-DATA.confirm carries the status to the originator next higher layer. After returning the NO_ACK status to the upper layer, the MPX data service can also optionally send an MPX IE with an abort code to inform the other end that it has aborted the transaction. As the other peer was not responding to the Acks, it is most likely because it is not in radio range, so it is quite likely that this abort message will not reach the other end either, so it is often sent without asking for Ack, and without retransmissions. If the other end receives the abort, it can immediately clear out its state. If the recipient does not receive the abort it shall drop the transaction after timeout specified by the *macMpxReassemblyTimeout* PIB attribute.

Figure 4 shows a case where the recipient cannot process the request and thus it aborts it by sending a separate Data frame having an MPX IE with an abort code. The MPX IE may also include the maximum transfer size which is then given to the originator's next higher layer along with the TRANSACTION_ABORTED status code. When receiving the MPX IE with Transfer type of aborted the Originator MAC can find matching transaction by matching the Source and Destination addresses and the Transaction ID.

In Figure 5 the recipient is configured not to accept any of the MPX IE transactions, thus it can immediately answer with the MPX IE inside the Enh-Ack it is sending back to the originator for the first fragment.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

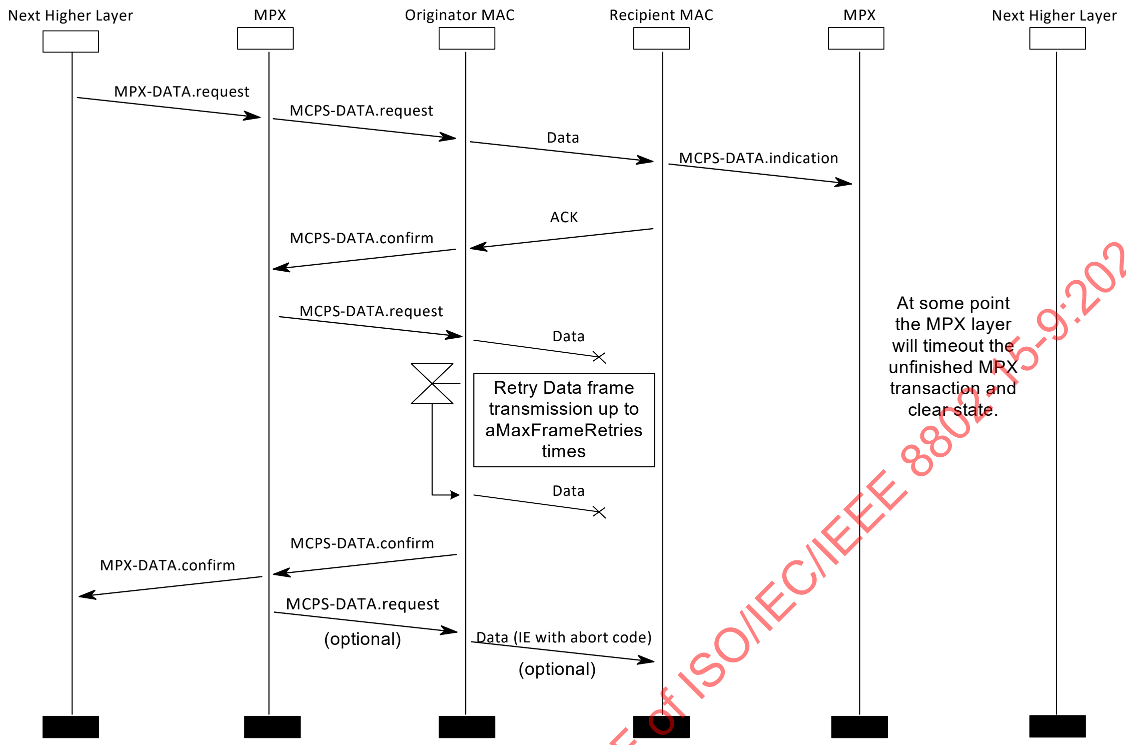

Figure 3—Failed fragment transfer

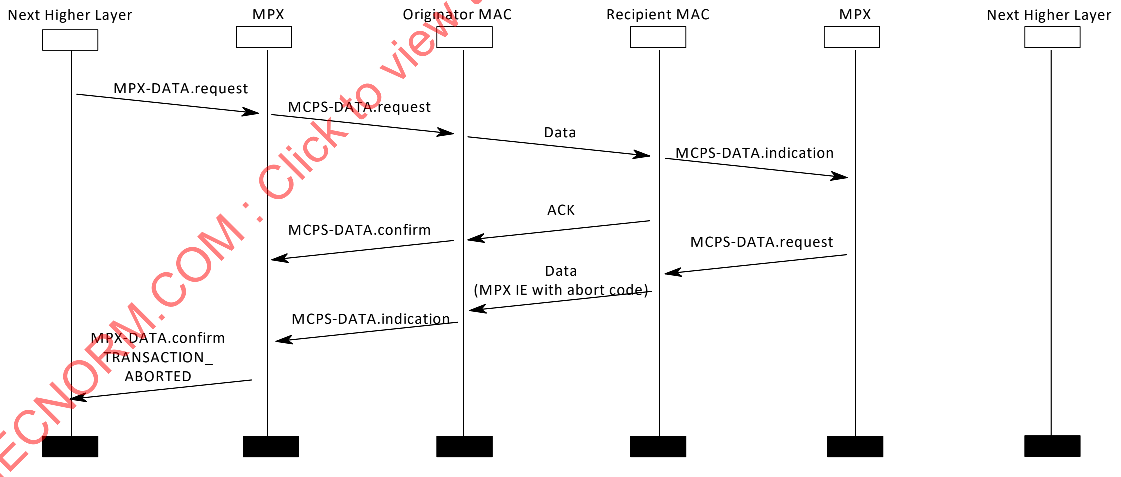

Figure 4—Aborted fragment transfer

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

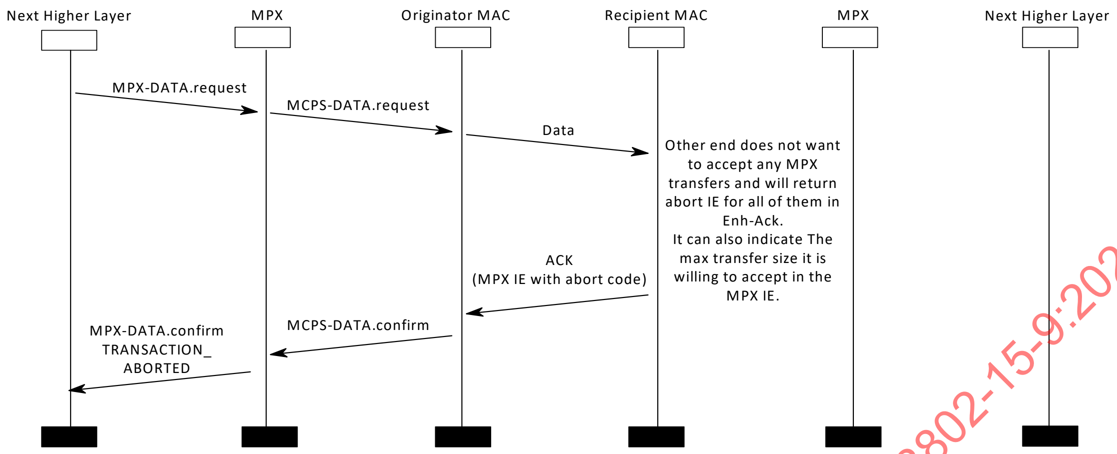
Next Higher Layer
MPX
Originator MAC
Recipient MAC
MPX
Next Higher Layer
MPX-DATA.request
MCPS-DATA.request
Data
Other end does not want to accept any MPX transfers and will return abort IE for all of them in Enh-Ack.
It can also indicate The max transfer size it is willing to accept in the MPX IE.
ACK
(MPX IE with abort code)
MCPS-DATA.confirm
MPX-DATA.confirm
TRANSACTION_
ABORTED

Figure 5—Aborted fragment transfer in Enh-Ack

5.2.2 MPX-DATA.request

The MPX-DATA.request primitive requests the transfer of an MPX payload to another device.

The semantics of this primitive are as follows:

```
MPX-DATA.request                (
                                SrcAddrMode,
                                DstAddrMode,
                                DstPanId,
                                DstAddr,
                                MultiplexId,
                                MpxData,
                                MpxHandle,
                                SecurityLevel,
                                KeyIdMode,
                                KeySource,
                                KeyIndex,
                                SendMultipurpose
                                )
```

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

The primitive parameters are described in Table 2.

Table 2—MPX-DATA.request parameters

Name	Type	Valid range	Description
SrcAddrMode	Enumeration	NONE, SHORT, EXTENDED	The source addressing mode for this MPX data.
DstAddrMode	Enumeration	NONE, SHORT, EXTENDED	The destination addressing mode for this MPX data.
DstPanId	Integer	0x0000–0xffff	The PAN identifier of the entity to which the MPX data is being transferred.
DstAddr	—	As specified by the DstAddrMode parameter	The address of the receiving (destination) device.
MultiplexId	Integer	0x0000–0xffff	The higher-layer protocol using the MPX data service. See 7.3.4.
MpxData	Set of octets	—	The set of octets forming the MPX data payload.
MpxHandle	Integer	0x00–0xff	An identifier which can be used to refer to the particular primitive transaction; used to match a confirm primitive with the corresponding request.
SecurityLevel	Integer	0–7	The combination of Message Integrity Check and Encryption to be applied to the payload of the MPX data service. For encoding see Table 9-6 in IEEE Std 802.15.4-2020.
KeyIdMode	Integer	As defined in Table 9-7 of IEEE Std 802.15.4-2020	The mode used to identify the key purportedly used by the originator of the received frame. This parameter is ignored if the SecurityLevel parameter is set to 0x00.
KeySource	Set of octets	As indicated by the KeyIdMode parameter	The originator of the key purportedly used by the originator of the received frame. The KeySource field, when present, indicates the originator of a group key. The KeySource parameter shall be formatted as described in the Section 9.4.4.2 of the IEEE Std 802.15.4-2020. This parameter is ignored if the KeyIdMode parameter is ignored or set to 0x00 or set to 0x01.
KeyIndex	Integer	0x01–0xff	The Key Index field allows unique identification of different keys with the same originator. It is the responsibility of each key originator to make sure that the actively used keys that it issues have distinct key indices and that the key indices are all different from 0x00. This parameter is ignored if KeyIdMode is ignored or set to 0x00.
SendMultipurpose	Boolean	TRUE, FALSE	If TRUE, use the Multipurpose frame type. If FALSE, use Data frame type. See 8.3.2 MCPS-DATA.request of IEEE Std 802.15.4-2020.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

5.2.3 MPX-DATA.confirm

The MPX-DATA.confirm primitive reports the results of a request to transfer data to another device.

The semantics of the MPX-DATA.confirm are as follows:

```
MPX-DATA.confirm              (
                              MpxHandle,
                              MaxTransferSize,
                              Status
                              )
```

The primitive parameters are described in Table 3. If there is no capacity to store the transaction, the Status shall be set to TRANSACTION_OVERFLOW. In case the other end aborts the transaction then the status shall be set to TRANSACTION_ABORTED and the MaxTransferSize is set to the value returned from the other end.

Table 3—MPX-DATA.confirm parameters

Name	Type	Valid range	Description
MpxHandle	Integer	0x00–0xff	An identifier that can be used to refer to a particular primitive transaction; used to match a confirm primitive with the corresponding request.
MaxTransfer-Size	Integer	0x0000–0xffff	In case of an aborted transaction this parameter can be returned from the other end to indicate the maximum size of transaction it can handle. In case an other end did not give a maximum size, this is set to zero.
Status	Enumeration	SUCCESS, TRANSACTION_ABORTED. Also see 8.2.2 of IEEE Std 802.15.4-2020 for list of generic errors	The status of the last MPX data transmission.

5.2.4 MPX-DATA.indication

The MPX-DATA.indication primitive delivers an MPX payload from another device.

The semantics of this primitive are as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

```
MPX-DATA.indication          (
                             SrcAddrMode,
                             SrcPanId,
                             SrcAddr,
                             DstAddrMode,
                             DstPanId,
                             DstAddr,
                             MultiplexId,
                             MpxData,
                             SecurityLevel,
                             KeyIdMode,
                             KeySource,
                             KeyIndex
                             )
```

The primitive parameters are described in Table 4.

Table 4—MPX-DATA.indication parameters

Name	Type	Valid range	Description
SrcAddrMode	Enumeration	NONE, SHORT, EXTENDED	The source addressing mode for this MPX data payload.
SrcPanId	Integer	0x0000–0xffff	The PAN identifier of the entity from which MPX data is being transferred.
SrcAddr	—	As specified by the SrcAddrMode parameter	The address of the transmitting (source) device.
DstAddrMode	Enumeration	NONE, SHORT, EXTENDED	The destination addressing mode for this MPX data payload.
DstPanId	Integer	0x0000–0xffff	The PAN identifier of the entity to which the MPX data is being transferred.
DstAddr	—	As specified by the DstAddrMode parameter.	The address of the receiving (destination) device.
MultiplexId	Integer	0x0000–0xffff	The higher-layer protocol using the MPX data service. See 7.3.4
MpxData	Set of octets	—	The set of octets forming the MPX data payload.
SecurityLevel	Integer	See Table 2	See Table 2.
KeyIdMode	Integer	See Table 2	See Table 2.
KeySource	Set of octets	See Table 2	See Table 2.
KeyIndex	Integer	See Table 2	See Table 2.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

5.3 MPX-PURGE primitive

5.3.1 MPX-PURGE primitive use

The MPX-PURGE primitives provide a means to remove or abort pending transfers from the MPX transaction queue of the originator. See Figure 6 for an example.

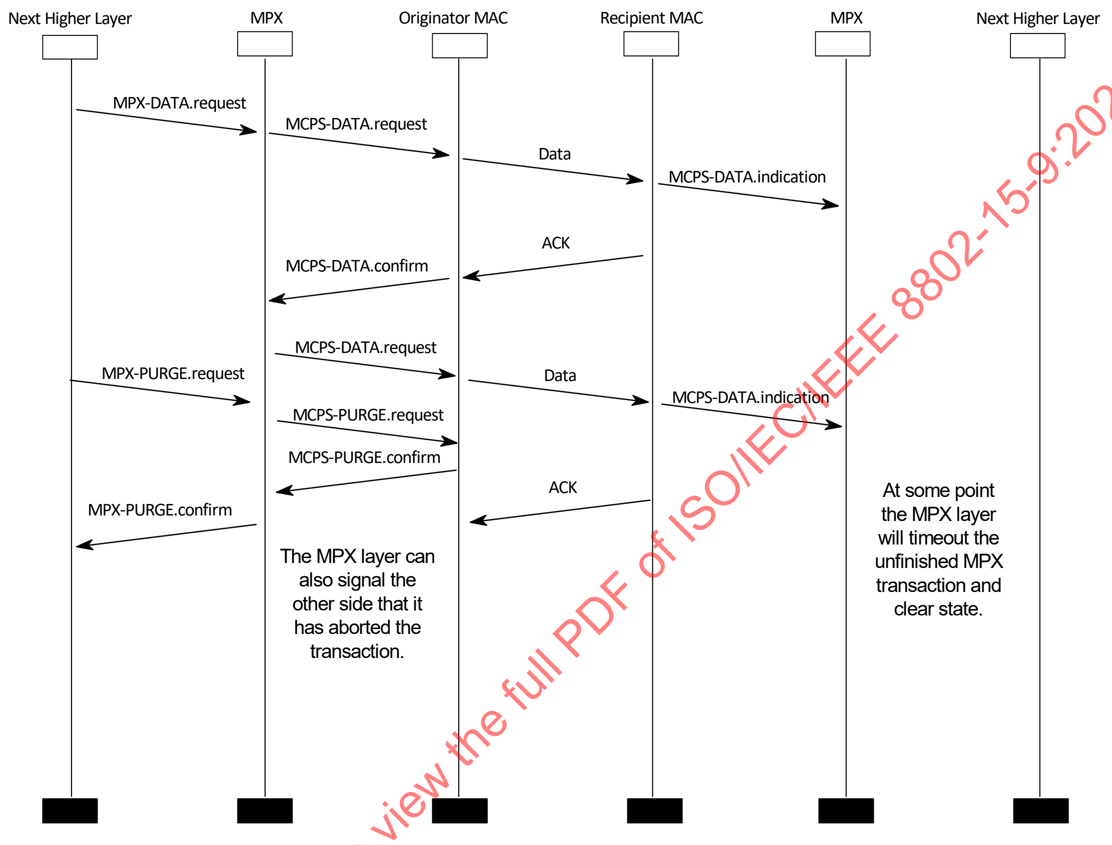

Figure 6—MPX-PURGE example

5.3.2 MPX-PURGE.request

The MPX-PURGE.request primitive allows the next higher layer to purge an MPX payload from the transaction queue.

The semantics of the MPX-PURGE.request are as follows:

```
MPX-PURGE.request                 (
                                  MpxHandle,
                                  SendAbort
                                  )
```

The primitive parameters are described in Table 5.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Table 5—MPX-PURGE.request parameters

Name	Type	Valid range	Description
MpxHandle	Integer	0x00–0xff	An identifier that can be used to refer to a particular primitive transaction; used to match an MPX-PURGE.request primitive with the corresponding MPX-DATA.request primitive.
SendAbort	Boolean	TRUE, FALSE	If this parameter is TRUE and the transaction is still active, the MPX data service sends an MPX IE with an abort code to the other end indicating that the transaction was aborted. If this parameter is FALSE, the transaction is just purged locally, and no information is sent to the other end.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

On receipt of the MPX-PURGE.request primitive, the MPX data service attempts to find in the transaction queue the payload indicated by the MpxHandle parameter. If an MPX payload associated with the provided handle has left the transaction queue, the MPX payload can not be purged and the MPX-PURGE.confirm primitive shall be generated with Status set to INVALID_HANDLE. If an MPX payload matching the given handle is found, the payload is discarded from the transaction queue. If SendAbort is set to TRUE, an abort message shall be sent to the intended recipient. When generated, the abort message is a data frame or a multi-purpose frame containing an MPX IE with the Transfer type field set to 0b110. If an abort message is sent to the other end that will allow the other end to clear out its state immediately without waiting for the timeout.

If the MPX-PURGE.request is received while the MPX service has an MCPS-DATA.request in process, a corresponding MCPS-PURGE.request is issued to the MAC data service

5.3.3 MPX-PURGE.confirm

The MPX-PURGE.confirm primitive allows the MPX data service to notify the next higher layer of the success of its request to purge an MPX payload from the transaction queue.

The semantics of this primitive are as follows:

```
MPX-PURGE.confirm          (
                           MpxHandle,
                           Status
                           )
```

The primitive parameters are described in Table 6.

Table 6—MPX-PURGE.confirm parameters

Name	Type	Valid range	Description
MpxHandle	Integer	0x00–0xff	An identifier which can be used to refer to a particular primitive transaction; used to match a confirm primitive with the corresponding request.
Status	Enumeration	SUCCESS, INVALID_HANDLE	The status of the request to purge MPX data from the transaction queue.

Copyright © 2022 IEEE. All rights reserved.

The MPX-PURGE.confirm shall be generated upon completion of the MPX-PURGE.request with the MpxHandle set to the value of MpxHandle provided in the corresponding MPX-PURGE.request, and with the Status value set according to the result of the request as described in 5.3.2

5.4 MPX PIB attributes

The MPX PIB includes the attributes required to manage the MPX data service of device. The attributes are contained in the Table 7. All attributes can be read or written by the next higher layer using the MLME-GET.request or MLME-SET.request primitives, described in the IEEE Std 802.15.4.

Table 7—MPX PIB attributes

Attribute	Type	Range	Description	Default
macMpxReassemblyTime-out	Integer	0–65 535	Timeout in seconds after which the partially reassembled frames are dropped.	30
macMpxMaxFragmentSize	Integer	7–2047	Max size of the MPX IE Content field used by the MPX data service when fragmenting the frame.	96

6. KMP transport service

6.1 Overview

KMP services provide the service primitives shown in Table 8. The KMP-CREATE primitive is used to initiate SA creation; the KMP-FINISHED is used to indicate when the key management process is ready and the keys are ready to be used. KMP-PURGE is used to delete ongoing KMP operations, and KMP-DELETE can be used to send delete request over the KMP to the other end. The support of KMP-DELETE depends on the KMP used, not all KMPs support deletion of the keys. In addition to those primitives the KMP may provide other methods.

Table 8—Summary of KMP Primitives

Name	Request	Indication	Response	Confirm
KMP-CREATE	6.2.2	6.2.4	6.2.5	6.2.3
KMP-FINISHED	—	6.3.2	—	—
KMP-DELETE	6.4.2	6.4.4	—	6.4.3
KMP-PURGE	6.5.2	—	—	6.5.3

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

The link keys negotiated between the peers using KeyIdMode 0x00 do not have a KeyIndex, i.e., there can be only one SA between the peers. This creates challenges for the rekeying of the SA, as a new SA cannot be created before the old one is removed. On the other hand, as there is only one SA, if a peer creates a new pairwise key between the peers, both ends will know that the previous SA has been removed. Because the installation of the key can happen at different times in peers, some traffic might be lost between the peers during rekey, as it is possible that one end has already installed a new key but that the other end is still using the old one. Because of this, the upper layer, desiring no data loss, should cease transmitting data using the key to be rekeyed while the rekey is in progress and only continue transmitting data after the KMP-FINISHED.indication primitive has been received.

The initial KMP frames need to be sent out without security (i.e., using security level 0 of IEEE 802.15.4), as there is not yet an SA set up between the peers. In IEEE Std 802.15.4, there is a way to do security filtering based on the IEs, i.e., the security PIB can be configured to allow an MPX IE without requiring security for suitable frame types. This allows the ability to process MPX IEs even when the frames would require security.

Note: the IEEE Std 802.15.4 only allows security filtering based on the IE type, i.e., it can allow all MPX IEs to be processed without security, but when configured so it will allow all MPX IEs, regardless whether they are using KMP as their Multiplex ID or not. The upper layer should do its own security filtering when using MPX IE for both upper layer data and KMP.

An alternative approach would be to use the IEEE 802.15.4 secDeviceDescriptor *secExempt* flag for the KMP frames (either for the MPX IEs or the frame types carrying KMP frames). The network join procedure would need to provide a means to identify the peer device to the local node ahead of the first KMP frame from the peer (such as a procedure to set the *secExempt* attribute in the *secDeviceList* entry corresponding to the peer device to TRUE after an initial frame exchange).

Because during the rekey the keys might be installed at different times, all key management packets should be sent without security, so they will not get dropped if the key used to encrypt/decrypt the frames has changed during the process.

6.2 KMP-CREATE primitives

6.2.1 KMP-CREATE primitive use

The KMP-CREATE primitives provide a means of initiating a KMP exchange that will ultimately result in an SA being established between the initiating and receiving devices.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Figure 7 shows an example of the successful KMP-CREATE.request operation with the KMP having a 3-packet exchange. The number of actual packets used by the key management protocols depends on the KMP used and options selected for them (for example the authentication method affects the number of frames needed). The location of the KMP-FINISHED.indication can differ depending on the KMP. For example in this case there is still one confirmation packet to be sent to the other end even after the KMP has already finished, and given the keys to the next higher layer. The KMP-FINISHED.indication could be delayed until the final MPX-DATA.confirm on the originator side is called.

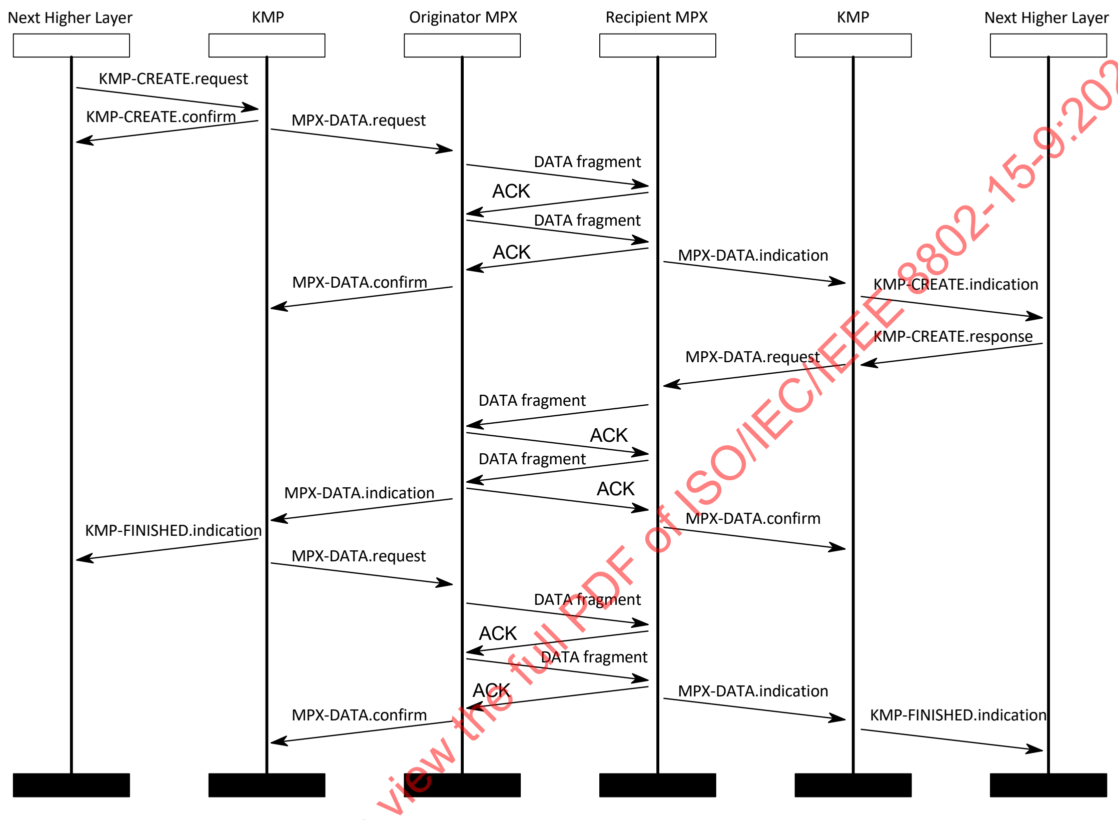

Figure 7—Successful key management operation

6.2.2 KMP-CREATE.request

A higher layer event initiates the KMP process directly by using the KMP-CREATE primitive. On receipt of the KMP-CREATE.request the KMP layer shall start the KMP which transmits KMP packets between the peers. The method of configuring the management protocol subsystem by the higher layer is outside the scope of this document.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

The semantics of this primitive are as follows:

```
KMP-CREATE.request            (
                              KmpPayloadHandle,
                              SrcAddrMode,
                              DstExtAddr,
                              KmpIdValue,
                              NewKeyIdMode,
                              NewKeySource,
                              NewKeyIndex,
                              NewAlgorithmIdList
                              )
```

The primitive parameters are described in Table 9.

Table 9—KMP-CREATE.request parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match a confirm primitive with the corresponding request. This identifier is also used to match the KMP-CREATE.request with its corresponding KMP-FINISHED.indication.
SrcAddrMode	Enumeration	NONE, SHORT, EXTENDED	The source addressing mode.
DstExtAddr	Extended address	Any valid extended address	The extended address of the receiving (destination) device.
KmpIdValue	Enumeration	See Table 22	The KMP to be used.
NewKeyIdMode	Integer	0x00–0x03	The KeyIdMode of the SA to be created.
NewKeySource	Set of octets	As indicated by the NewKeyIdMode parameter	The KeySource of the SA to be created.
NewKeyIndex	Integer	0x01–0xff	The KeyIndex of the SA to be created.
NewAlgorithmIdList	List of Integers	As defined in 802.15.4 ANA [B10]	List of algorithm identifiers allowed for this new key. The KMP shall then negotiate the exact algorithm to be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

The NewKeyIdMode specifies the KeyIdMode of the key to be created. If the NewKeyIdMode is 0x00, the link key between the peers is created and NewKeySource and NewKeyIndex are not used. If the NewKeyIdMode is 0x01, the NewKeySource is not used but NewKeyIndex specifies the KeyIndex to be used for the key to be created. If the NewKeyIdMode is 0x02 or 0x03, both NewKeyIndex and NewKeySource are used and they identify the key to be created as specified in 9.4.2.3 “Key Identifier Mode” in IEEE Std 802.15.4-2020.

Copyright © 2022 IEEE. All rights reserved.

If the KeyIdMode is 0x00, then the KMP will create a link key between the peers. For other KeyIdModes, the keys are group keys and the KMP will either create a new group key, if such is not yet installed to the security PIB, or transmit the currently installed group key to the other peer. To rekey group keys the higher layer simply creates a new key with different KeyIndex and KeySource, and deletes the old one. Note, that to do in-place rekey for group keys, the higher layer needs first to delete old group key from the security PIB and then create a new one.

Calling this primitive again with KeyIdMode 0x00 will replace the current link key between the peers in place, effectively rekeying it, in which case the old link key does not need to be deleted using KMP-DELETE.

6.2.3 KMP-CREATE.confirm

The KMP-CREATE.confirm primitive indicates the result of a request to start the KMP process with another device. This primitive shall be generated upon reception of the KMP-CREATE.request to indicate that the key management operation was successfully started or to indicate that the key management operation could not be started. If operation was successfully started KMP-CREATE.confirm shall be called with Status set to SUCCESS.

The semantics of this primitive are as follows:

```
KMP-CREATE.confirm            (
                              KmpPayloadHandle,
                              Status
                              )
```

The primitive parameters are described in Table 10.

Table 10—KMP-CREATE.confirm parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match a confirm primitive with the corresponding request.
Status	Enumeration	See Table 3	The status of the request to start the KMP process with a remote device.

6.2.4 KMP-CREATE.indication

The KMP-CREATE.indication primitive indicates the reception of a request to start the KMP process with a remote device. KMP-CREATE.indication shall be generated when the KMP starts its processing to inquire whether the KMP operation is allowed or not.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

The semantics of this primitive are as follows:

```
KMP-CREATE.indication          (
                               OriginatorExtAddr,
                               KmpIdValue,
                               KmpPayloadHandle,
                               )
```

The primitive parameters are described in Table 11.

Table 11—KMP-CREATE.indication parameters

Name	Type	Valid range	Description
OriginatorExtAddr	Extended address	Any valid extended address	The extended address of the initiating (originating) device.
KmpIdValue	Enumeration	See Table 22	The KMP in use.
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match an indication primitive with the corresponding response. This identifier is also used to match the KMP-CREATE.indication with its corresponding KMP-FINISHED.indication.

6.2.5 KMP-CREATE.response

The KMP-CREATE.response primitive is used to respond to the KMP-CREATE.indication call and specify whether the KMP shall continue with the peer indicated in the KMP-CREATE.indication call.

The semantics of this primitive are as follows:

```
KMP-CREATE.response            (
                               KmpPayloadHandle,
                               ContinueProcessing,
                               DenyTime
                               )
```

The primitive parameters are described in Table 12.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Table 12—KMP-CREATE.response parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match the response primitive with the initiating indication.
ContinueProcessing	Boolean	TRUE, FALSE	If this parameter is TRUE, processing with the current peer is continued. If this parameter is FALSE, then the KMP layer shall ignore this connection attempt from the other peer, effectively dropping all KMP frames received from the peer for time given in DenyTime parameter.
DenyTime	Integer	0–300	Number of seconds the KMP layer shall not forward requests from the same device to the higher-layer in case the ContinueProcessing was FALSE. If ContinueProcessing is TRUE this parameter is ignored.

6.3 KMP-FINISHED primitives

6.3.1 Overview

After the whole key management operation is finished, the status of the operation is indicated to the higher layer using the KMP-FINISHED.indication primitive. This call is called in both peers, i.e., regardless of who initiated the KMP-CREATE.request call. The KmpPayloadHandle given to the KMP-FINISHED.indication can either be one from KMP-CREATE.request or from KMP-CREATE.indication. This means the higher layer needs to use RemoteExtAddr to know to which call the KMP-FINISHED.indication relates.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

6.3.2 KMP-FINISHED. indication

This call indicates that the key management operation is finished. At this point the higher layer needs to extract the keying material from the key management subsystem and install it to the security PIB. If the group key is just transmitted to the other end from this end, the originator already has the group key installed in the PIB. The semantics of this primitive are as follows:

```
KMP-FINISHED.indication          (
                                 RemoteExtAddr,
                                 KmpPayloadHandle,
                                 KmpIdValue,
                                 CreatedKeyIdMode,
                                 CreatedKeySource,
                                 CreatedKeyIndex,
                                 CreatedKey,
                                 CreatedAlgorithmId,
                                 Status
                                 )
```

The primitive parameters are described in Table 13.

Table 13—KMP-FINISHED.indication

Name	Type	Valid range	Description
RemoteExtAddr	Extended address	Any valid extended address	The address of the remote entity.
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match the indication primitive with the initiating request or indication.
KmpIdValue	Enumeration	See Table 22	The KMP in use. See Table 22.
CreatedKeyIdMode	Integer	0x00–0x03	The mode used to identify the key created by this operation. See 9.4.2.3 in IEEE Std 802.15.4-2020.
CreatedKeySource	Set of octets	As specified by CreatedKeyIdMode	The originator of the key.
CreatedKeyIndex	Integer	0x01–0xff	The index of the key created by this operation.
CreatedKey	variable	—	The value of the key created.
CreatedAlgorithmId	Integer	As defined in 802.15.4 ANA [B10]	The algorithm identifier of the negotiated algorithm for this key.
Status	Enumeration	See Table 3	The status of the request.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

6.4 KMP-DELETE primitives

6.4.1 Overview

The KMP-DELETE primitive requests the other end to delete the SA.

6.4.2 KMP-DELETE.request

A higher layer initiates the KMP-DELETE.request when it wants to remove an SA, for example, because it has already created a new SA and started using it (for example because of a rekey). After this call is made, the higher layer should not use the given SA anymore when sending data. Note that there is no guarantee that this request will be successfully signaled to the other end so the other end might still continue sending data on the SA regardless whether this primitive was called or not.

The semantics of this primitive are as follows:

```
KMP-DELETE.request              (
                                KmpPayloadHandle,
                                SrcAddrMode,
                                DstExtAddr,
                                KmpIdValue,
                                SaToDeleteKeyIdMode,
                                SaToDeleteKeySource,
                                SaToDeleteKeyIndex
                                )
```

The primitive parameters are defined in Table 14.

Table 14—KMP-DELETE.request parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match a confirm primitive with the corresponding request.
SrcAddrMode	Enumeration	NONE, SHORT, EXTENDED	The source addressing mode for this request.
DstExtAddr	Extended address	Any valid extended address	The extended address of the receiving (destination) device.
KmpIdValue	Enumeration	See Table 22	The KMP in use.
SaToDeleteKeyIdMode	Integer	0x00–0x03	The KeyIdMode of the SA to be deleted. See 9.4.2.3 in IEEE Std 802.15.4-2020.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Table 14—KMP-DELETE.request parameters *(continued)*

Name	Type	Valid range	Description
SaToDeleteKeySource	Set of octets	As indicated by the Sa-ToDeleteKeyIdMode parameter	The KeySource of the SA to be deleted.
SaToDeleteKeyIndex	Integer	0x01–0xff	The KeyIndex of the SA to be deleted.

The key to be deleted is indicated by the combination of the SaToDeleteKeyIdMode, SaToDeleteKeySource, and SaToDeleteKeyIndex. If the SaToDeleteKeyIdMode is 0x00, the pairwise keys between the peers is deleted, and SaToDeleteKeySource and SaToDeleteKeyIndex are not used. If the SaToDeleteKeyIdMode is 0x01, SaToDeleteKeyIndex is used to identify the key to be deleted, and if SaToDeleteKeyIdMode is 0x02 or 0x03, both SaToDeleteKeyIndex and SaToDeleteKeySource are used to identify the key to be deleted.

6.4.3 KMP-DELETE.confirm

The KMP-DELETE.confirm primitive indicates the result of the KMP-DELETE operation. Depending on the KMP this might be called immediately, or it might be called only after the other end has responded and the SA is really deleted. After this call the higher layer can remove the SA key from the PIB.

The semantics of this primitive are as follows:

```
KMP-DELETE.confirm              (
                                KmpPayloadHandle,
                                Status
                                )
```

The primitive parameters are defined in Table 15.

Table 15—KMP-DELETE.confirm parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to the particular primitive transaction; used to match a confirm primitive with the corresponding request.
Status	Enumeration	See Table 3	The status of the request.

6.4.4 KMP-DELETE.indication

The KMP-DELETE.indication primitive is called when the other end requested an SA to be removed. The other end will be removing the SA from its PIB tables regardless of whether this end actually removed it or not, so there is no point in having a response to this.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

The semantics of this primitive are

```
KMP-DELETE.indication          (
                               SrcExtAddr,
                               DstExtAddr,
                               KmpIdValue,
                               SaToDeleteKeyIdMode,
                               SaToDeleteKeySource,
                               SaToDeleteKeyIndex
                               )
```

The primitive parameters are defined in Table 16.

Table 16—KMP-DELETE.indication parameters

Name	Type	Valid range	Description
SrcExtAddr	Extended address	Any valid extended address	The extended address of the initiating (source) device.
DstExtAddr	Extended address	Any valid extended address	The extended address of the receiving (destination) device
KmpIdValue	Enumeration	See Table 22	The KMP in use.
SaToDeleteKeyIdMode	Integer	0x00–0x03	The KeyIdMode of the SA the other end is requesting to be deleted.
SaToDeleteKeySource	Set of octets	As indicated by the SaToDeleteKeyIdMode parameter	The KeySource of the SA the other end is requesting to be deleted.
SaToDeleteKeyIndex	Integer	0x01–0xff	The KeyIndex of the SA the other end is requesting to be deleted

When receiving a delete request of the pairwise key (i.e., SaToDeleteKeyIdMode is 0x00), the KMP has usually (depending on the KMP) already verified that the originator of the frame is authentic, thus no additional authorization checks are needed when deleting the key. When receiving a delete request for a group key (i.e., SaToDeleteKeyIdMode is 0x01–0x03), before the next higher layer actually removes the keys from the security PIB, it needs to check whether this action should be allowed from the remote peer sending the delete request. Note that the KMP usually does authenticate the originator and then the RemoteExtAddr can be used when doing those checks.

6.5 KMP-PURGE primitives

6.5.1 Overview

The KMP-PURGE primitives provide means to remove KMP-CREATE or KMP-DELETE transactions from the KMP service.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

6.5.2 KMP-PURGE.request

The KMP-PURGE.request primitive allows the next higher layer to purge a KMP operation from the KMP service.

The semantics of the KMP-PURGE.request are as follows:

```
KMP-PURGE.request             (
                              KmpPayloadHandle
                              )
```

The primitive parameters are defined in Table 17.

Table 17—KMP-PURGE.request parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to a particular primitive transaction; used to match a KMP-PURGE.request primitive with the corresponding KMP operation.

On receipt of the KMP-PURGE.request primitive, the MPX data service attempts to find in the transaction queue the payload indicated by the KmpPayloadHandle parameter. If a KMP operation is already finished, the handle will not be found, and the KMP operation can no longer be purged. If a KMP operation matching the given handle is found, the operation is aborted and its status is discarded from the KMP service.

The KMP-PURGE request shall also issue a corresponding MPX-PURGE.request to the MPX data service, provided it has an MPX-DATA.request in process when the KMP-PURGE.request is called.

6.5.3 KMP-PURGE.confirm

The KMP-PURGE.confirm primitive allows the KMP service to notify the next higher layer of the success of its request to purge a KMP operation from the KMP service.

The semantics of this primitive are as follows:

```
KMP-PURGE.confirm             (
                              KmpPayloadHandle,
                              Status
                              )
```

The primitive parameters are defined in Table 18.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Table 18—KMP-PURGE.confirm parameters

Name	Type	Valid range	Description
KmpPayloadHandle	Integer	0x00–0xff	An identifier that can be used to refer to a particular primitive transaction; used to match a confirm primitive with the corresponding request.
Status	Enumeration	SUCCESS, INVALID_HANDLE	The status of the request to purge a KMP transaction from the transaction queue.

7. MPX IE format

7.1 IE overview

IEs consist of an ID, a length, and content. The content of the MPX IE is composed of multiple fields that are detailed in 7.3.

7.2 Payload IE group ID

The IE ID value is dependent on the standard in use. Figure 8 lists the values that are used.

Standard	Payload IE Group ID value
IEEE Std 802.15.4	3

Figure 8—IE ID

7.3 MPX IE content

7.3.1 Overall structure

The contents of an MPX IE are formatted as shown in Figure 9.

Octets: 1	0/1	0/2	0/2	Variable
Transaction Control	Fragment Number	Total Upper Layer Frame Size	Multiplex ID	Upper Layer Frame Fragment

Figure 9—MPX IE contents

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

The Transaction Control field is always there and the rest of the fields depend on the transfer type. The summary of different MPX IE formats can be found in Table 19.

Table 19—Summary of different MPX IE formats

Bits: 0–2	3–7	Octets: 0/1	0/2	0/2	Variable
Transfer type	**Transaction ID**	**Fragment number**	**Total Upper Layer Frame Size**	**Multiplex ID**	**Upper Layer Frame Fragment**
0b000	0x00–0x1f	Omitted	Omitted	0x0000–0xffff	Full upper layer frame.
0b001	0x00–0x1f	Omitted	Omitted	Omitted	Full upper layer frame for protocols having a small Multiplex ID. The Multiplex ID value of 0x00–0x1f shall be encoded inside the Transaction ID field, and the Multiplex ID field is omitted.
0b010	0x00–0x1f	0x00	0x0000–0xffff	0x0000–0xffff	First fragment(s), total size of the packet and Multiplex ID.
0b010	0x00–0x1f	0x01–0xfe	Omitted	Omitted	Middle fragment.
0b011	Reserved				
0b100	0x00–0x1f	0x01–0xff	Omitted	Omitted	Last fragment.
0b101	Reserved				
0b110	0x00–0x1f	Omitted	Omitted	Omitted	Omitted (abort without telling max size responder can handle).
0b110	0x00–0x1f	Omitted	0x0000–0xffff	Omitted	Omitted (abort with information what is max size responder can handle).
0b111	Reserved				

7.3.2 Transaction Control field

7.3.2.1 Overview of the Transaction Control field

The Transaction Control octet is as specified in Figure 10.

Bit: 0–2	3–7
Transfer type	Transaction ID

Figure 10—Transaction Control field

7.3.2.2 Transfer Type field

Where the transfer type can have the values as described in Table 20.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Table 20—Transfer Type Field

Transfer Type	Name	Description
0b000	Full frame	The Fragment Number and Total Upper Layer Frame Size fields are omitted, the Multiplex ID field is present, and the Upper Layer Frame Fragment field contains the whole upper layer frame packet.
0b001	Full frame with compressed Multiplex ID	The Fragment Number and Total Upper Layer Frame Size fields are omitted, the Multiplex ID field is not present; instead the Multiplex ID is transmitted inside the Transaction ID field, and the Upper Layer Frame Fragment field contains the whole upper layer frame packet.
0b010	Non-last fragment	The Fragment Number field is always present, and if it is 0x00 meaning this is the first fragment then the Total Upper Layer Frame Size and Multiplex ID fields are also present. The rest of the IE is upper layer frame fragment.
0b011	Reserved	
0b100	Last fragment	The Fragment Number field is always present, and the Total Upper Layer Frame Size and Multiplex ID fields are always omitted (as this cannot be the first fragment, as in that case the frame could have been fit in one fragment, i.e., the Full frame format transfer type would have been used). The rest of the IE is the upper layer frame fragment.
0b101	Reserved	
0b110	Abort	This is message from the responder to the originator indicating that it wishes to abort the transfer. This can be sent at any time, including the response to the first fragment. The Fragment Number field and Multiplex ID fields are always omitted; but the Total Upper Layer Frame Size field may be present and if it is present that tells the maximum size packet the responder is willing to accept. Whether the Total Upper Layer Frame Size field is present is known from the length of the IE, i.e., if the length is 1 octet, then it is omitted, if it is 3 octets, then it is there. This can also be sent in case when originator wants to abort the transfer in process. The Transaction ID field of this frame matches the Transaction ID this is aborting, i.e., if this is responder sending this message it matches the incoming Transaction ID, and if this is originator aborting its own transmission it matches the Transaction ID of its transaction.
0b111	Reserved	

7.3.2.3 Transaction ID field

The Transaction ID value shall be selected by the originator to uniquely identify the transaction while in process. It is selected by the originator to be unique between a pair of devices, and it cannot be reused until the transaction is finished or aborted. This allows 32 simultaneous transaction between two devices. Note that Transaction ID needs to be unique between the two devices for which the transaction is established regardless of who is originator and who is responder.

The Transfer Type 0b001 can be used if Multiplex ID is between 0x00–0x1f, and in that case the Transaction ID field contains the lowest five bits of the Multiplex ID and the Multiplex ID field is omitted.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

7.3.2.4 Fragment Number field

The Fragment Number field is only present if the Transfer type is 0b010 or 0b100. If the Fragment Number field has a value of 0x00, indicating the first fragment, the Total Upper Layer Frame Size and Multiplex ID fields are also present. This means that non-initial frames have only a two-octet overhead.

7.3.3 Total Upper Layer Frame Size field

The Total Upper Layer Frame Size field tells the size of the incoming upper layer frame. When the responder receives it, it can verify whether it can process the incoming upper layer frame or not. Total Upper Layer Frame Size field is only present if the Fragment Number is 0x00 and the Transfer type is 0b010, and it can optionally be present if the Transfer type is 0b110, in which case whether it is present or not can be seen from the length of the ID.

7.3.4 Multiplex ID field

The Multiplex ID field is used to multiplex different upper layer protocols. The Multiplex ID field takes one of two meanings, depending on its numeric value as follows:

a) If the value of this field is less than or equal to 1500, the Multiplex ID field takes values as specified in Table 21.

b) If the value of this field is greater than 1500, the Multiplex ID field indicates the EtherType of the MAC client protocol. This provides for EtherType protocol discrimination as described in 802-2014 [B11].

Table 21—Multiplex ID

Multiplex ID (decimal)	Multiplex ID (hex)	Description
1	0x0001	KMP
2–1279	0x0002–0x04ff	Reserved
1280–1500	0x0500–0x05dc	Vendor specific

The Multiplex ID field is present if the Frame number is 0x00 and the Transfer type is 0b010, or if the Transfer type is 0b000. If the Transfer type is 0b001, the Multiplex ID is stored inside the Transaction ID field and the Multiplex ID field is omitted.

7.3.5 Upper Layer Frame Fragment field

The Upper Layer Frame Fragment field is present in all other Transfer types than 0b110, but it can also be empty. For example, if the originator wants to first ask whether the other end is able to process this big of a transaction, i.e., it can send an IE having only Transfer type 0b010, with a Fragment Number of 0x00, a Total Upper Layer Frame Size, and the Multiplex ID, without any data in the Upper Layer Frame Fragment field. The responder can then either: send an Enhanced-Acknowledgment back in case the responder is willing to continue the transaction or, in the case that the responder cannot receive fragments now, it may send back an Enhanced-Acknowledgment with an MPX IE having Transfer type 0b110 and it can also include the maximum size it will accept.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

8. KMP Service

8.1 KMP ID

The KMP ID is added to the KMP payload as the first byte of the first MPX fragment as in Table 22.

Table 22—KMP ID values

KMP	KMP ID value
IEEE 802.1X/MKA	1
HIP	2
IKEv2	3
PANA	4
Dragonfly	5
IEEE 802.11/4WH	6
IEEE 802.11/GKH	7
ETSI TS 102 887-2	8
Reserved for future use	9 –254
Vendor-specific	255

8.2 Vendor-specific KMPs

The vendor-specific KMP is reserved for the use of other KMPs relevant only to certain implementations. The vendor-specific KMP shall start with the field containing the Vendor OUI as illustrated in Figure 11.

Octets: 3	Variable
Vendor OUI	Actual vendor KMP data

Figure 11—Vendor-specific KMP content

The Vendor OUI field shall contain an OUI or CID assigned by the IEEE Registration Authority (RA).[8] A value of the vendor OUI not understood by a receiving device causes the contents of the KMP data to be ignored.

The Actual Vendor KMP data field is defined by the vendor identified in the Vendor OUI field. Its use is outside the scope of this standard.

[8]Interested applicants should contact the IEEE Registration Authority, http://standards.ieee.org/develop/regauth/.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

9. State machines

9.1 Inbound state machine

The inbound flow diagram (Figure 12) is triggered by receipt of a frame containing the MPX IE. There are four outcomes on each execution of this machine:

- MPX payload delivered to MPX service
- MPX payload in partial assembly
- Frame dropped (duplicate, wrong fragment number, inconsistent security)
- Abort notification received

The recipient can clear out the partially reassembled frame either after it receives the abort message or when the frame has not been able to be reassembled after a timeout period specified by *macMpxReassemblyTimeout*. The specification of the timeout period depends on the environment and is outside the scope of this document.

The "Check fragment number" step in the inbound state machine means that the matching transaction is found (i.e., the source and destination addresses and transaction id match) and then the last fragment number received in that transaction is compared against the fragment number in the current frame. If the fragment number in the frame is smaller than or equal to the last fragment number received, the frame is a retransmission and shall be dropped. If the fragment number in the frame is the last fragment number received plus one, the frame is a new fragment, and is processed. If the fragment number in the frame is larger than the last fragment number received plus one, the state is out of sync, and all fragments are dropped.

The "Verify that security parameters are same" step in the inbound state machine means that when the first fragment is received, the SecurityLevel, KeyIdMode, KeyIndex (if present), and KeySource (if present) are stored, and, when further frames are later received, they are checked to verify that those security parameters stay the same for all fragments received for that transaction.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

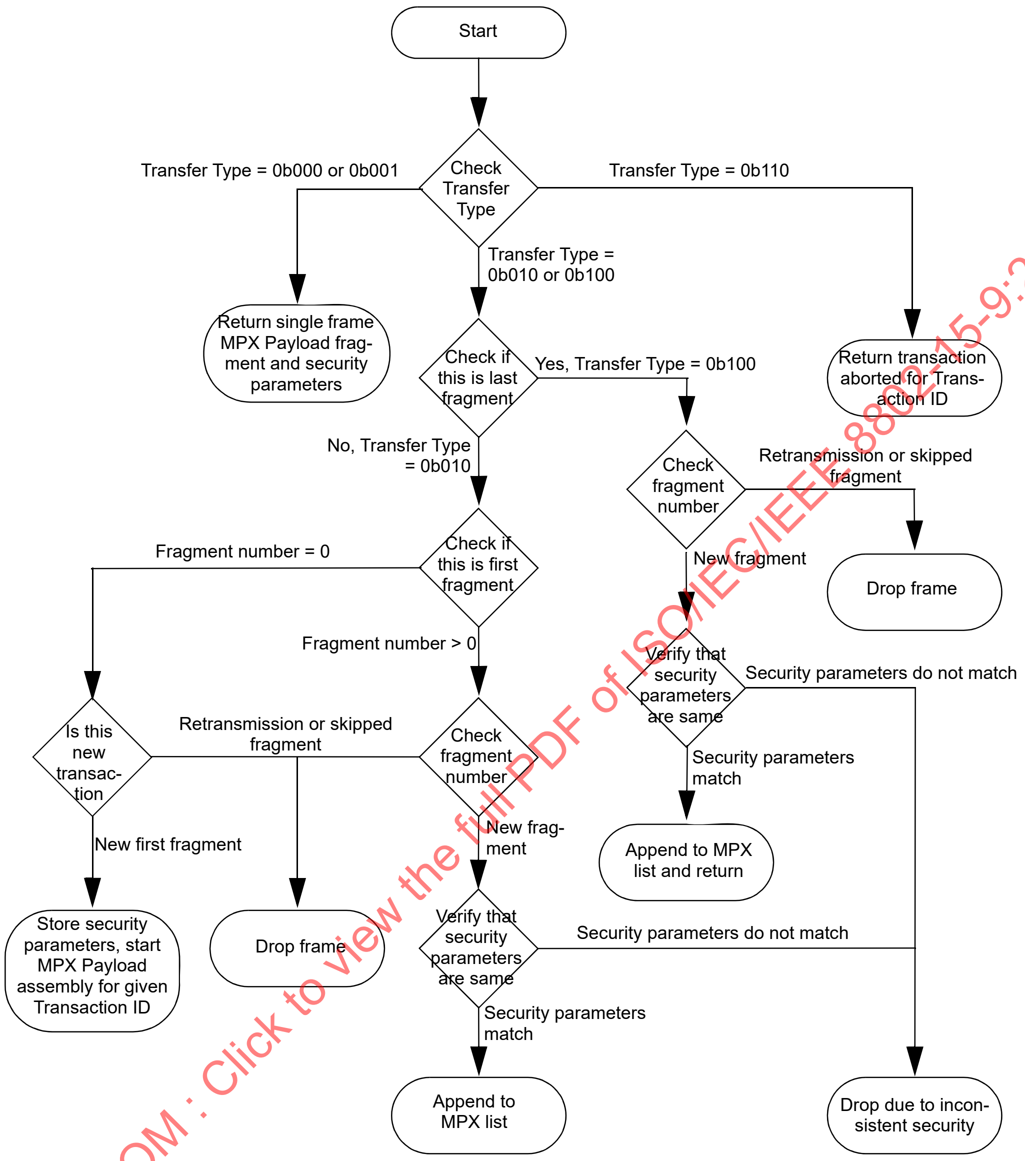
Start
Check Transfer Type
Transfer Type = 0b000 or 0b001
Transfer Type = 0b110
Transfer Type = 0b010 or 0b100
Return single frame MPX Payload fragment and security parameters
Check if this is last fragment
Yes, Transfer Type = 0b100
Return transaction aborted for Transaction ID
No, Transfer Type = 0b010
Check fragment number
Retransmission or skipped fragment
Drop frame
New fragment
Check if this is first fragment
Fragment number = 0
Fragment number > 0
Verify that security parameters are same
Security parameters do not match
Security parameters match
Append to MPX list and return
Is this new transaction
Retransmission or skipped fragment
Check fragment number
New first fragment
New fragment
Store security parameters, start MPX Payload assembly for given Transaction ID
Drop frame
Verify that security parameters are same
Security parameters do not match
Security parameters match
Append to MPX list
Drop due to inconsistent security

Figure 12—Inbound flow diagram

A buffer is needed for fragment reassembly into the complete MPX payload, and this buffer is initialized on receipt of the first fragment for a new transaction. Each transaction is identified by a Transaction ID and source/destination addresses, and there may be multiple simultaneous transactions between devices. There may be duplicate frames received if the acknowledgment was lost and the packet was resent. The MPX service only receives complete MPX payloads when the final fragment is received. There may be a partially received MPX payload that timed out for any number of reasons.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

9.2 Outbound state machine

The outbound flow diagram (Figure 13) creates a transaction with the destination device. In the Figure 13, the send is considered successful only if the recipient acknowledged the fragment to be sent. In the event of a failed transmission, it flushes all state on the payload. Upon reception of MPX-PURGE.request, all the state associated with the specific MPX payload corresponding to the MpxHandle parameter of the MPX-PURGE.request is cleared.

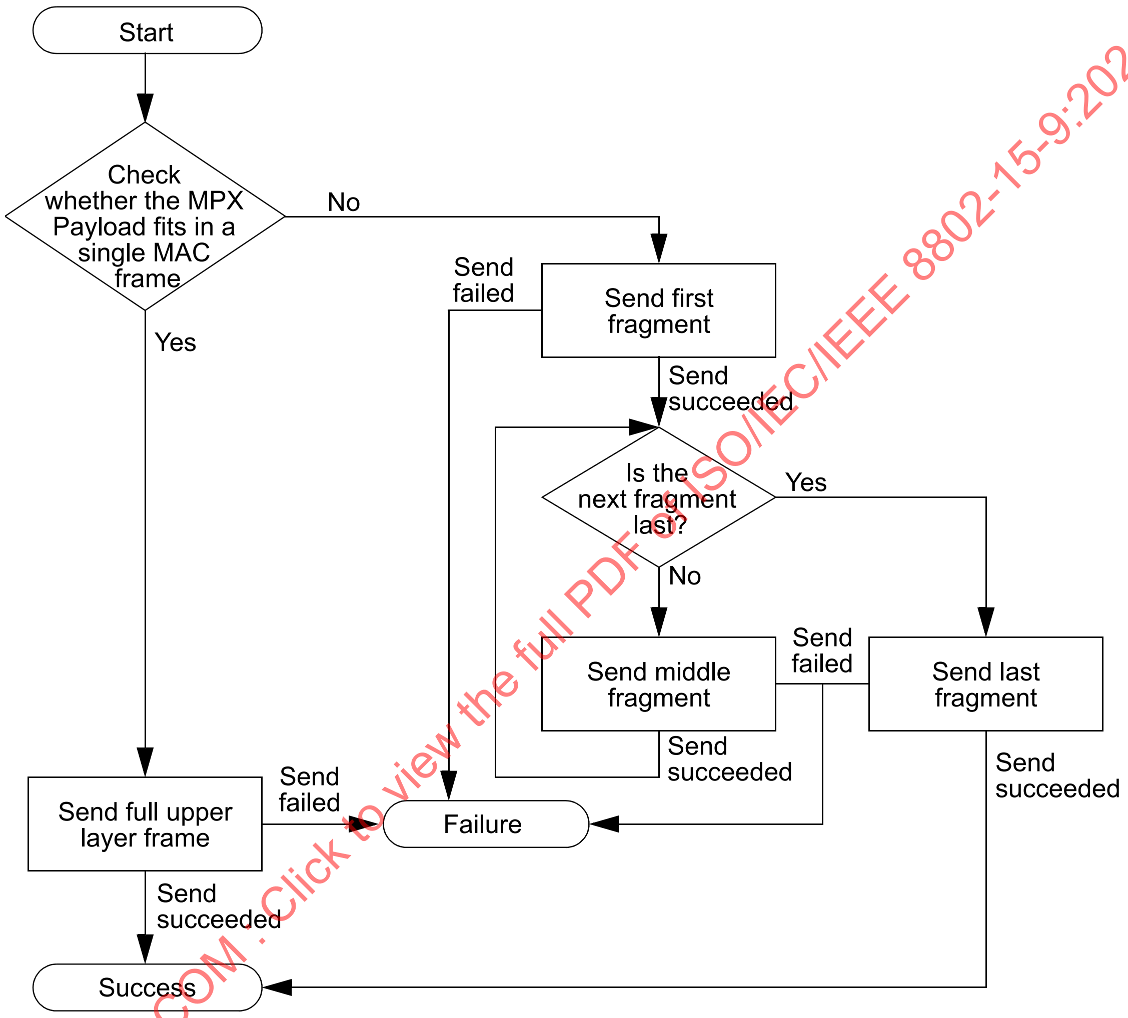

Figure 13—Outbound flow diagram

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

The principal function of this machine is the management of fitting the MPX payload into the MPDU through the fragmentation mechanism. The MPX data service shall fragment the frame so that each MPX IE content field is less than *macMpxMaxFragmentSize* bytes, and shall pass the resulting fragments for transmission through the MCPS-DATA.request SAP. The AckTx parameter of each MCPS-DATA.request shall be set to TRUE in order to request an acknowledgment of the fragmented payloads, and may be set to TRUE for full MPX payload.

Annex A

(informative)

KMP specifics—IEEE 802.1X/MKA

A.1 Description

IEEE Std 802.1X [B12] defines a port-based network access control for IEEE 802® networks. It allows network administrators to restrict the use of IEEE 802 LAN service access points (ports) to secure communication with authenticated and authorized systems. Before a device is granted access to a network it is authenticated. Extensible Authentication Protocol (EAP) (Aboba, et al. [B1]) is transported between a supplicant port-access entity (PAE) (6.3 of IEEE Std 802.1X-2020 [B12]) and an Authenticator PAE.

Device authentication does not itself provide for any protection of frames between wireless personal area network (WPAN) devices themselves. Successful authentication can be accompanied by the secure delivery, to both PAEs, of a secret key that can be used to prove mutual authentication and to distribute or agree further secret keys. These secret keys are then used to provide security services (e.g., confidentiality, integrity, and replay protection) for WPAN frames. Three such key agreement protocols to agree upon the secret keys are as follows:

— IEEE Std 802.1X-2020 [B12] (Clause 9) specifies the MACsec Key Agreement (MKA) protocol, where the secret key derived from EAP is the connectivity association key (CAK). The CAK is used to discover other PAEs attached to the same LAN, to confirm mutual possession of a CAK and hence prove a past mutual authentication, and to agree the secret keys used by a datagram security services. The CAK can either be derived from an EAP exchange, or pre-shared between a set of stations that are authorized to communicate between themselves.
— IEEE Std 802.11™ [B13] describes the use of IEEE Std 802.1X, where the secret key derived from EAP is the master session key (MSK). The MSK is used as the basis to protect 4-Way Handshake (4WH) and Group Key Handshake (GKH) protocols, which are encapsulated in EAP over LAN-KEY (EAPOL-KEY) message types defined in IEEE Std 802.1X.
— ETSI TS 102-887-2 [B4] defines a node-to-node pairwise link key establishment protocol.

In addition to support for group keys based on IEEE Std 802.11, ETSI TS 102-887-2 node-to-node pairwise link key establishment is defined.

A.1.1 Device authentication

When a WPAN network includes a network access point (AP) controlling access to devices on a secured network behind the AP, the AP can act as an authenticator. The authenticator facilitates authentication and authorization of each supplicant (i.e., a WPAN device) desiring to communicate to the secured network behind it and discards Data frames prior to the completion of the authentication process. An Authenticator PAE relies upon an authentication server (AS) to perform supplicant PAE authentication and authorization. The AS may be either co-located with the authenticator or the authenticator may access the AS remotely via a network to which the authenticator has access.

The supplicant PAE and AS use EAP (Aboba, et al. [B1]) transported in EAPOL messages. The supplicant PAE and AS share credentials appropriate for the EAP method used for authentication. Credentials can be passwords or device identifiers (e.g., digital certificates).

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

All of the KMPs defined in Table A.1 can also do an optional authentication step using IEEE 802.1X/EAP protocol as defined in IEEE Std 802.1X-2020 Clause 8, in which case the PDU format shall be EAPOL-EAP as defined in Section 11.8 of IEEE Std 802.1X-2020.

A.1.2 Device authentication and cryptographic key agreement

A.1.2.1 Overview

Device authentication does not itself provide for any protection of frames between WPAN devices themselves. However, when AS policy includes the use of an EAP method that produces a shared secret [e.g., EAP-TLS (Simon, et al. [B23])], and the authenticator and supplicant policy indicates a key agreement protocol, then it is possible for the devices to agree upon keys and policy to provide security services (e.g., confidentiality, integrity, and replay protection) for WPAN frames.

Several key agreement protocols are available to use the EAP shared secret. They are described in the following subclauses, summarized in Table A.1.

Table A.1—KMP Protocols following IEEE Std 802.1X

KMP	PDU Format	Reference
IEEE 802.1X/MKA	EAPOL-MKA	IEEE Std 802.1X-2020 Clause 9
IEEE 802.11/4WH	EAPOL-KEY	IEEE Std 802.11-2016 Subclause 12.7.6
IEEE 802.11/GKH	EAPOL-KEY	IEEE Std 802.11-2016 Subclause 12.7.7
ETSI N2N	ETSI TS 102 887-2	ETSI TS102-887-2, Sections 7.9.4.1 through 7.9.4.4

A.1.2.2 IEEE 802.1X/MKA

A.1.2.2.1 Overview

As defined in IEEE Std 802.1X, MKA provides agreement of MACsec policy and keying material. MKA can also be used as a WPAN KMP key agreement method. An additional definition defining the CCM* cipher has been defined for WPAN use.

The scope of MKA can be pairwise (i.e., between the authenticator and a single supplicant), effectively extending the pair-wise security obtained during device authentication to WPAN frames. Alternatively, an MKA session can be shared between a set of devices, effectively providing for shared secure communications including broadcast and multicast frames.

While other uses of MKA allow MKA to autonomously generate replacement keys, when MKA is used as a WPAN KMP service it only creates keys as the result of a new key exchange created by a KMP-CREATE.request. This restriction is required because the higher layer protocols manage keys and their properties (see Annex F). MKA can continue to exchange messages following the successful installation of a new key, but it only installs subsequent keys as indicated by the higher layer and will ignore indications sent within MKA (e.g., the delivery of PendingPNExhaustion or discovery of a new live peer). MKA should deliver a notification to the higher layer when it receives these events.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Figure A.1 shows the protocol exchanges between the higher layers of two PAN devices using IEEE 802.1X/MKA. As shown, the first KMP-CREATE.request results in the IEEE 802.1X/EAP protocol followed by the IEEE 802.1X/MKA protocol, because authentication is required before MKA can agree upon a IEEE 802.15.9 SA. Higher layer protocols can make additional KMP-CREATE.requests using the same IEEE 802.1X/MKA instance (shown as the second KMP-CREATE.request in the figure).

When a KMP-FINISHED.indication is returned to the higher layer, one WPAN SA can be installed by the higher layer.

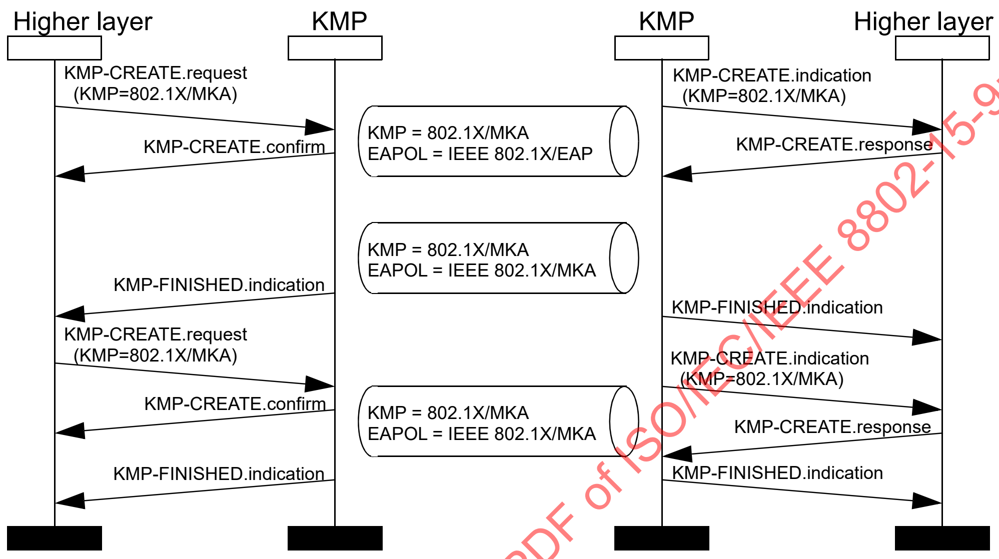
Higher layer
KMP
KMP
Higher layer
KMP-CREATE.request (KMP=802.1X/MKA)
KMP-CREATE.confirm
KMP = 802.1X/MKA EAPOL = IEEE 802.1X/EAP
KMP-CREATE.indication (KMP=802.1X/MKA)
KMP-CREATE.response
KMP = 802.1X/MKA EAPOL = IEEE 802.1X/MKA
KMP-FINISHED.indication
KMP-FINISHED.indication
KMP-CREATE.request (KMP=802.1X/MKA)
KMP-CREATE.confirm
KMP-CREATE.indication (KMP=802.1X/MKA)
KMP = 802.1X/MKA EAPOL = IEEE 802.1X/MKA
KMP-CREATE.response
KMP-FINISHED.indication
KMP-FINISHED.indication

Figure A.1—IEEE 802.1X/EAP and IEEE 802.1X/MKA as a PAN KMP

A.1.2.2.2 Cryptographic key agreement with pre-shared CAK

Some use cases will require key agreement without WPAN devices having previously executed supplicant PAE or authenticator PAE state machines. This is possible by pre-installing a CAK on each of the WPAN devices, and setting a policy on each device requiring that no WPAN Data frames be transmitted and received WPAN Data frames are discarded until MKA has obtained and installed WPAN cipher keys. When a pre-shared CAK is used, the IEEE 802.1X/EAP protocol shown in Figure A.1 is omitted.

The use of a pre-shared CAK is most expedient for resource-constrained WPAN devices, however a secure method of installing and refreshing CAKs to the WPAN devices would be critical to maintaining strong security.

A.1.2.3 IEEE 802.11/4WH and IEEE 802.11/GKH

The IEEE 802.11 4WH and GKH protocols can also be used as WPAN KMP key agreement methods. The protocols are inherently pair-wise protocols between two peers but the scope of the agreed upon keys differs: The result of an IEEE 802.11/4WH is a pairwise key, and the result of an IEEE 802.11/GKH is a group key that may be shared with multiple WPAN devices.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Each invocation of IEEE 802.11/4WH and IEEE 802.11/GKH is independent and the exchanges can be delivered in any order, according to the policy of the Higher Layer. Figure A.2 shows an example protocol flow where the IEEE 802.11/4WH exchange is followed by an IEEE 802.11/GKH. Because IEEE 802.1X/4WH was the first protocol between the two KMP elements, IEEE 802.1X/EAP precedes IEEE 802.1X/4WH.

Only one WPA Security Association can be returned to the higher layer, so the KMP-FINISHED.indication is restricted to returning just one SA. The optional distribution of a GTK in the IEEE 802.1X/4WH cannot be used. When an GTK is needed, it should be distributed using an IEEE 802.11/GKH KMP.

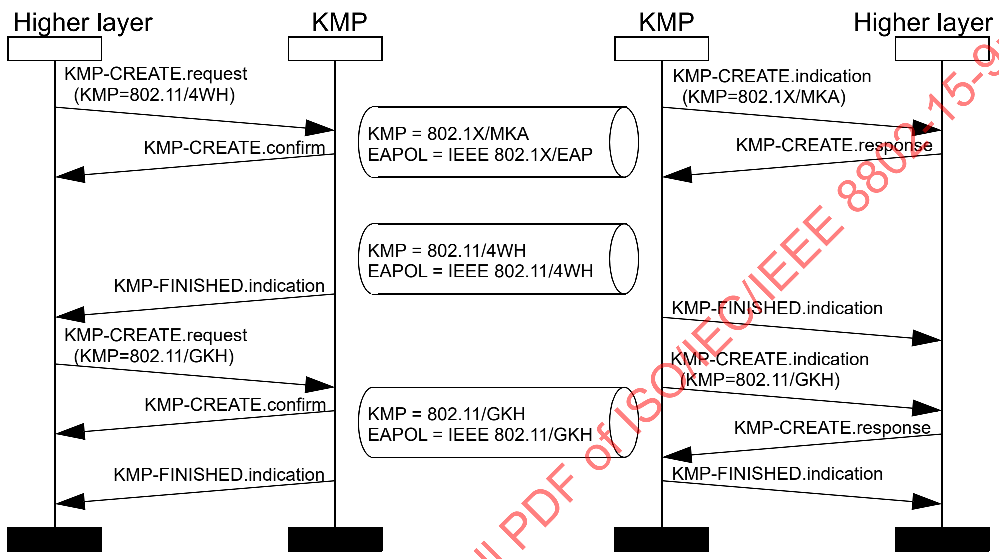
Higher layer
KMP
KMP
Higher layer
KMP-CREATE.request (KMP=802.11/4WH)
KMP-CREATE.confirm
KMP = 802.1X/MKA EAPOL = IEEE 802.1X/EAP
KMP-CREATE.indication (KMP=802.1X/MKA)
KMP-CREATE.response
KMP = 802.11/4WH EAPOL = IEEE 802.11/4WH
KMP-FINISHED.indication
KMP-FINISHED.indication
KMP-CREATE.request (KMP=802.11/GKH)
KMP-CREATE.indication (KMP=802.11/GKH)
KMP-CREATE.confirm
KMP = 802.11/GKH EAPOL = IEEE 802.11/GKH
KMP-CREATE.response
KMP-FINISHED.indication
KMP-FINISHED.indication

Figure A.2—IEEE 802.1X/EAP, IEEE 802.1X/4WH and IEEE 802.1X/GKH as a PAN KMP

A.1.2.4 ETSI TS102 887-2 node-to-node (N2N) link key establishment

The ETSI N2N key establishment is used to establish session keys between pairs of communicating one-hop neighbor nodes in the mesh. A unique session key is established between each pair of one-hop neighbor nodes. Section 7.9.3 of ETSI TS 102-887-2 serves as the reference for this exchange. Section 7.9.4 of ETSI TS 102-887-2 details the message exchange between source and destination one-hop neighbors. The source device is simply the first to send the New Session Create message to the one-hop destination. In practice, either of the devices in the pairwise key establishment exchange can serve as source or destination for new session establishment (although, once the new session exchange begins, the devices remain in the role of source or destination until the new session is created). Communication between the devices using the session SA can originate on either device once secured communication begins.

The N2N SA supports four message constructs as shown in Table A.2.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Table A.2—ETSI TS102 887-2 KMP protocol

KMP	PDU Format	Reference
ETSI TS 102 887-2	New Session Create	ETSI TS 102-887-2, Section 7.9.4.1
ETSI TS 102 887-2	New Session Created	ETSI TS 102-887-2, Section 7.9.4.2
ETSI TS 102 887-2	New Session Acknowledgment	ETSI TS 102-887-2, Section 7.9.4.3
ETSI TS 102 887-2	New Session Destruction	ETSI TS 102-887-2, Section 7.9.4.4

Figure A.3 shows the PDU exchange for ETSI N2N link key establishment. Note that the link key establishment uses the IEEE 802.1X/EAP KMP for authentication.

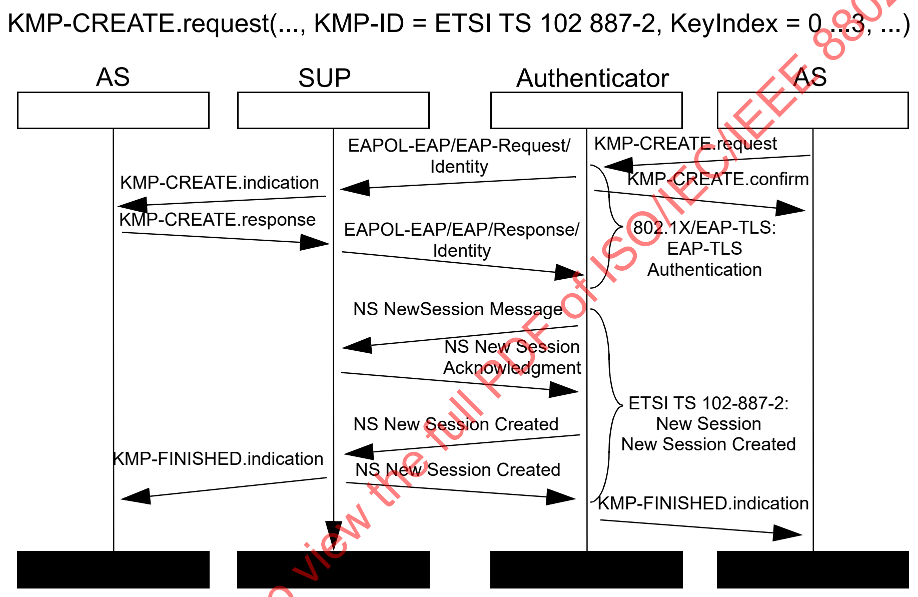
KMP-CREATE.request(..., KMP-ID = ETSI TS 102 887-2, KeyIndex = 0...3, ...)
AS
SUP
Authenticator
AS
EAPOL-EAP/EAP-Request/ Identity
KMP-CREATE.request
KMP-CREATE.indication
KMP-CREATE.confirm
KMP-CREATE.response
EAPOL-EAP/EAP/Response/ Identity
802.1X/EAP-TLS: EAP-TLS Authentication
NS NewSession Message
NS New Session Acknowledgment
ETSI TS 102-887-2: New Session New Session Created
NS New Session Created
KMP-FINISHED.indication
NS New Session Created
KMP-FINISHED.indication

Figure A.3—PDU exchange for N2N link key establishment

A.2 Use cases

A.2.1 Overview

The flexibility of device authentication makes IEEE 802.1X/EAP suitable for a variety of use cases. This clause describes optimal uses for each key agreement protocol used with IEEE 802.1X/EAP.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

A.2.2 Isolated enclave

IEEE 802.1X/MKA was designed to provide cryptographic key agreement services to a group (i.e., two or more) of devices that need to communicate together. An isolated enclave of WPAN devices that need to communicate amongst themselves with a mix of unicast, broadcast, and/or multicast frames could benefit from MKA's shared security model. If needed, strong authentication of group members can be first obtained using device authentication, when one device acts as both an authenticator and an AS. Alternatively, if strong device authentication is not required, the WPAN devices can use a pre-shared CAK to establish group authorization.

IEEE 802.11/GKH could also be used to distribute a shared key to a group of devices in an isolated enclave of WPAN devices.

ETSI TS102 887-2 can further be used to establish pairwise link keys to one-hop neighbors in an isolated enclave of WPAN devices.

A.2.3 Star topology

A network topology where a set of WPAN devices communicate privately with a PAN coordinator would benefit from pairwise key agreement services. A PAN coordinator could use either IEEE 802.1X/MKA or IEEE 802.11/4WH to create pairwise keys between itself and each PAN participant.

The controller may use device authentication to strongly authenticate the WPAN devices or, in the case of IEEE 802.1X/MKA, may depend on a pre-shared pair-wise CAK in order to generate keying material shared only with the expected WPAN device.

A.2.4 Mesh

If a mesh of WPAN devices comprises more than one broadcast domain, each WPAN needs be considered an individual isolated enclave or star.

A.3 IEEE 802.15 specifics

A.3.1 EAPOL message framing

IEEE 802.1X messages are specified as being carried in EAPOL PDUs. When carried in an MPX IE, the EAPOL PDU (beginning with the Protocol Version field) follows the KMP ID value in the first KMP Fragment.

A.3.2 EAPOL-MKA

A.3.2.1 General

Several modifications are necessary to the semantics and parameter sets within EAPOL-MKA messages.

A.3.2.2 ICV calculation

An integrity check value (ICV) is calculated using AES-CMAC. The bytes protected by the ICV are known as M and are defined in IEEE Std 802.1X as follows:

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

$$M = DA + SA + (MSDU – ICV) \tag{A.1}$$

where DA and SA refer to the destination and source addresses respectively, the IEEE Std 802.1X MAC Service Data Unit (MSDU) comprises all of the data bytes beginning with the EtherType but not including the ICV bytes themselves.

The IEEE Std 802.15.4 MSDU cannot be included in M, because the EAPOL message is likely to be fragmented and sent in a set of MPX IEs, each of which comprises an MSDU. However, the EAPOL-MKA is reassembled before being validated. Therefore, when an EAPOL-MKA message is passed in an MPX IE, M is defined to cover the EAPOL PDU (See 11.3 of IEEE Std 802.1X-2020) as shown below:

$$M = DA + SA + (EAPOL – ICV) \tag{A.2}$$

where

DA is the destination address used to deliver the EAPOL-MKA message
SA is the source address used to deliver the EAPOL-MKA message
EAPOL is the EAPOL PDU (beginning with Protocol Version)
ICV is the bytes of the EAPOL message comprising the ICV

A.3.2.3 MKA Basic parameter set

The MACsec Capability field indicates whether or not an MKA participant can perform MACsec. For the purposes of IEEE 802.15, this field indicates whether a WPAN device is capable of WPAN Security features.

The MACsec Desired field aids in incremental deployment. For the purposes of IEEE 802.15, the MACsec Desired fields indicates whether or not WPAN security is desired.

The SCI field is taken to be the 64-bit ExtendedAddress of the originator.

A.3.2.4 MACsec SAK Use parameter set

The following fields have unique semantics when delivered in an MPX IE.

— Latest Key AN is the KeyIndex. Note that the size of KeyIndex is effectively reduced to 0b01–0b11.
— Old Key AN, Old Key tx, Old Key rx are unused.
— Delay protect is unused.
— Latest Key Lowest Acceptable PN and Old Key Lowest Acceptable PN are unused.

A.3.2.5 Distributed SAK parameter set

The following fields have unique semantics when delivered in an MPX IE.

— Distributed AN is the KeyIndex in the range of 0b01–0b11.
— Offset Confidentiality is unused.
— MACsec Cipher Suite is taken to mean “WPAN CCM* Cipher Suite”. The Cipher Suite Identifier 00-80-C2-00-01-00-00-05 is defined by the IEEE 802.1 group for this cipher suite.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

A.3.2.6 MKA state machine—suspension

MKA requires each station to emit an MKA frame at least every 2 s, which may not be possible in all use cases. In particular, when devices are battery powered this will provide an unwelcome drain on the battery. Also when the WPAN includes a coordinator, messages may only be sent during the active portion of a superframe.

The MKA suspension semantics defined in IEEE Std 802.1X [B12] intended for managing in-service upgrades may be useful here as well. A device sets the MKA suspension time in the XPN parameter set, which causes the MKA key server to not expect MKA messages from the device for MKA suspension time. This allows the device to deliver infrequent data messages (e.g., between sleep periods) without having to deliver MKA messages. If the device desires to renew the XPN suspension time at any time, it may send MKA frames again and deliver a new, larger suspension time.

A.3.2.7 MKA state machine—Key Server selection

When a WLAN includes a PAN Coordinator (e.g., in a Star Topology), the PAN Coordinator should be the key server for the group. A PAN Coordinator is required to be a full functioned device, and it must also have the capability to generate high quality keying material.

A.3.3 EAPOL-KEY

EAPOL-KEY messages are framed according 11.9 of IEEE Std 802.1X-2020 and used as specified in 12.7 of IEEE Std 802.11-2016. Minor modifications are necessary.

IEEE Std 802.11-2016, in 6.3.19, describes a SetKeyDescriptor, which describes the interface whereby the key management algorithm describes the material generated for the use of the 4WH and GKH protocol. The following fields of the SetKeyDescriptor will require re-interpretation when used with this standard:

— Key ID: maps to the IEEE 802.15.4 Key Index field
— Cipher Suite Selector: The Cipher Suite Selector shall be 00-0F-AC-12 (00-0F-AC:18) for AES-CCM*-128.

A.3.4 ETSI TS 102 887-2

The ETSI TS 102 887-2 SA messages are framed as described in ETSI TS 102 887-2 Section 7.9.4. The specific frames used in this specification are included in Table A.1.

A.3.5 Group Traffic Key Generation

Once the IEEE 802.1X/GKH completes successfully and group key material is placed onto the supplicant, AES key expansion must be performed as described in section 5.2 of FIPS 197 [B5] with the resulting group traffic key then generated as follows:

Group Traffic Key = Truncate-<key size>(SHA-256(Network Identifier||group key material))

where:

— Group Traffic Key is the resulting secKey in the IEEE 802.15.4.
— Truncate-<key size> performs truncation of the results from the inner SHA-256 operation for key sizes of 128, 192 or 256 bits (e.g. Truncate-128, Truncate-192, Truncate-256)

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

- Network Identifier is a text string identifying the deployment network
- group key material was the key placed by the IEEE 802.1X/GKH or IEEE 802.1X/4WH
- || is the concatenation operation

The KeyIdMode, KeySource, and KeyIndex are not negotiated by the IEEE 802.1X/GKH or IEEE 802.1X/4WH, and it is outside the scope of this standard to define what values are used.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Annex B

(informative)

KMP specifics—IKEv2

B.1 Description

IKEv2 (Kaufman, et al. [B16]) is an IETF standard used for key management in IP security (IPsec). IKEv2 provides a means of performing mutual authentication between parties and establishing and maintaining SAs. IKEv2 supports a wide range of authentication methods such as shared keys, certificates, raw public keys, and EAP; depending on the environment in which IKEv2 will be used, only some of those are needed. In order to indicate which optional features of IKEv2 are used in a particular environment, it is necessary to list them in a profile for that environment.

The basic IKEv2 key exchange protocol is a two-exchange protocol consisting of four messages. In cases where EAP is used, there may be extra exchanges required, depending on the EAP method used.

B.2 Use cases

B.2.1 General

As IKEv2 provides a very wide range of options, it can be used in very different environments. It can be used in situations where there is no back-end AS, in which case shared keys or raw public key authentication may be used, or it can be deployed in an environment where there is extensive authentication, authorization, and accounting (AAA) infrastructure already in place, which allows elements to be reused to minimize program storage requirements.

B.2.2 Minimal IKEv2 use cases

A use case for a minimal IKEv2 implementation is where a small device engages in machine-to-machine communication. In such environments the node initiating connections usually has a small memory footprint and at the other end of the communication channel is some kind of larger device.

An example of the small initiating node could be a remote garage door opener; such a device has buttons, which open and close the garage door, and connects to the home area network server over a wireless link.

Another example of such a device is a sensor device, for example, a room temperature sensor that sends periodic temperature data to a centralized server node.

Such devices usually sleep for long periods, and only wake up because of user interaction or on a predetermined schedule. The data transfer is initiated from the sleeping node, and after the sleeping device sends its data packets to the server there may be a corresponding acknowledgment packet sent from the server to the sleeping device. In some cases, there may also be command or data packets to be sent from the server to the sleeping device that must be received before the sleeping device can go back to sleep. Data transfers from the server to the sleeping device can be implemented by the sleeping device regularly waking and polling the server.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

B.2.3 Enterprise or large-scale IKEv2 use cases

As IKEv2 also provides a way to use a full AAA infrastructure, including performing EAP authentication, it can be used in environments where reuse of existing AAA infrastructure is needed. It can also use full public key infrastructure X.509 certificates for authentication. In most cases, particularly for very constrained devices, these processes are too heavyweight; however, if the server that talks to the small devices is required to authenticate itself to the AAA infrastructure using IPsec, the ability to reuse IKEv2 in the small devices would be useful.

B.3 IKEv2 and IEEE 802.15 specifics

B.3.1 Overview

IKEv2 is normally run over Internet protocol (IP) and user datagram protocol (UDP), but there are already documents reusing the IKEv2 protocol over Fiber Channel to perform SA management (See Maino and Black [B18]).

B.3.2 Supported IKEv2 features

The following features of the IKEv2 should be supported:

— Basic IKEv2 exchange, IKE_SA_INIT, and IKE_AUTH (Kaufman, et al. [B16])
— Childless Initiation of IKEv2 (Nir, et al. [B21])
— Using Authenticated Encryption Algorithm with the Encrypted payload with IKEv2 (Black and McGrew [B2])

B.3.3 Unused IKEv2 features

The following IKEv2 features are not used:

— Negotiation of multiple protocols within same proposal
— Capability to handle multiple outstanding requests
— Cookies
— Configuration Payload
— NAT-Traversal

B.3.4 Message framing

The normal IP and UDP headers are not used; instead, messages are started directly with the IKEv2 header [See Section 3.1 of RFC 7296 (Kaufman et al. [B16])]. As the NAT-Traversal is not needed or supported, the first field of the header is the IKE SA initiator's security parameter index (SPI).

As the cipher algorithms supported by IEEE 802.15.4 are normally combined-mode ciphers (AES-CCM) the Encrypted Payload format described in RFC5282 (Black and McGrew [B2]) is needed.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

B.3.5 Algorithm negotiation

IKEv2 provides algorithm negotiation that negotiates which cryptographic algorithms are used to protect IKEv2.

The negotiated pseudo-random function (PRF) should use the same AES primitives, i.e., PRF_AES128_CMAC.

B.3.6 Key derivation

After the IKEv2 IKE_SA_INIT and IKE_AUTH exchanges are completed, IKEv2 generates SKEYSEED for keying material generation, and from there it generates the master secret SK_d, which is to be used for key generation for the child SAs. This SK_d is used to generate the necessary keying material for use in the IEEE 802.15.4 security layer, as specified in section 2.17 of RFC 7296 (Kaufman et al. [B16]):

$$\text{KEYMAT} = \text{prf}+(\text{SK\_d}, \text{Ni} \mid \text{Nr}) \quad \text{(B.1)}$$

The first octets of the KEYMAT are used as the key for the link key SA with KeyIdMode of 0x00.

B.3.7 Broadcast and multicast key distribution

If broadcast or multicast key distribution is needed, GSA, KD, and D payloads of the Group Key Management using IKEv2 [B22] can be used to distribute keys using GSA_INBAND_REKEY. The SPI Size and SPI fields inside GSA and D payloads is used to specify KeyIdMode, KeySource and KeyIndex as follows: If SPI Size is one then KeyIdMode of 0x01 is used, if SPI size is five, then KeyIdMode of 0x02 is used, and if SPI Size is nine, then KeyIdMode of 0x03 is used. The SPI field contains the KeySource and KeyIndex as specified in 9.4.4.1 of IEEE Std 802.15.4-2020. Traffic selectors inside GSA are ignored, and ENCR GSA Transform is used to specify the algorithm to be used.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

Annex C

(informative)

KMP specifics—HIP

C.1 Description

The HIP (Moskowitz [B20]) is an IETF standard that provides a KMP. Although primarily used to establish SAs between hosts at the IP layer, it is truly layer independent and can as readily establish SAs at the MAC layer. HIP has a Base Exchange (BEX) protocol and a simpler Diet Exchange (DEX) protocol (Moskowitz [B19]). Highly constrained systems will benefit from the minimal architecture of DEX. A less constrained system like a coordinator can easily support BEX and DEX peers, as DEX is largely a subset of BEX.

A central concept in HIP is a cryptographic host identity (HI) namespace. The self-asserting nature of HIs make for simple to implement and deploy models for small enclaves of constrained devices. As the HI format is determined by the underlying cryptographic algorithm, a host identity tag (HIT) is used as a normalized identity format. HITs are valid IPv6 addresses (2001:20::/28 prefix) and can be used in applications to provide mobility or where there is no real IP layer supported. HITs are also used in access control lists (ACLs) for simple authentication control. Third-party assertion authorities are optional with HIP. HIP does support both PKI and RADIUS based assertion authorities.

HIP is a four-packet peer-exchange protocol. These packets can be very small (particularly DEX), thus requiring very little fragmentation support. Either party may initiate an exchange, and a responder can force itself to be the initiator.

C.2 Use cases

C.2.1 General

HIP's lightweight nature works well with highly constrained devices and scenarios with no back-end authentication services. It can use an asymmetric defense against man-in-the-middle attacks for initial deployments; that is, if a sensor's HI (actually HIT) is registered to a coordinator, the coordinator can control permitted connections. Sensors are still exposed to connecting to rogue controllers, but this is detectable as the sensor would not be working in the desired network.

C.2.2 Isolated enclave

HIP works well in an isolated enclave where there is no apparent AS. It can work strictly ad hoc. There can be no authentication—that is the mode where "I don't know who you are, but we will communicate securely." There can be at authentication validation of the HITs on devices with screens that can display HITs. Finally HITs could be preloaded into devices that can read QR codes where HITs can be encoded for devices.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.

C.2.3 Home net

Home nets may use the isolated enclave approach, e.g., for a lighting system, or could use a RADIUS approach where the RADIUS server resides in a home gateway/server. HITs can be readily added to the RADIUS server either by a UI for entry, or more simply (for the user) by scanning QR codes on sensors that would contain encoded HITs.

C.2.4 City net

City nets will be have a mix of rich-function devices that have higher layer requirements for digital certificate based identities and highly constrained sensors that would benefit from the small footprint HIP DEX and HIT-based authentication. HIP BEX with certificate-based (Heer and Varjonen [B9]) and HIP DEX with HIT-based authentication can readily coexist; the initial packet, I1, identifies which exchange is called for. As DEX is basically a subset of BEX (with the switch to ECDH identities), coordinators can readily support both exchanges.

C.2.5 RFID networks

Although there are no known transmit-only RFID devices that can support any KMP, many RFID devices used supporting IEEE 802.15.4 have a receive mode for initial PAN joining. This can be used to provide the KMP support. The nature of RFID PANs of three receivers visible to any device, tends to produce deployments where most devices use the same key for data protection. In this scenario, the KMP is used to register devices to the PAN and distribute the group keys.

C.2.6 Infrastructure sensor nets

Highly constrained, low energy, critical infrastructure sensors are well matched with HIP DEX for a small memory/CPU/power footprint. The typical large infrastructure deployments in LECIM would work well with a RADIUS-styled device registration and authentication solution.

C.3 IEEE 802.15™ specifics

C.3.1 Message framing

HIP payloads are directly supported within KMP payloads. There is no IP or TCP/UDP content within BEX or DEX.

C.3.2 Key derivation and security PIB interaction

IEEE 802.15.4 supports complex security models through the Security PIB. As such, the higher-layer may use the ESP_TRANSFORM (Jokela, et al. [B15]), the ENCRYPTED_DATA, or the ENCRYPTED_KEY parameters as is appropriate. These can support unicast or group key distributions.

C.3.3 Deployment recommendations

HIs and HITs can be preinstalled in devices to avoid the processing cost of generation. The HIT can be provided externally as in a QR code for loading into an authentication service.

IECNORM.COM : Click to view the full PDF of ISO/IEC/IEEE 8802-15-9:2024

Copyright © 2022 IEEE. All rights reserved.