802.11w™

IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Amendment 4: Protected Management Frames

IEEE Computer Society

Sponsored by the
LAN/MAN Standards Committee

IEEE
3 Park Avenue
New York, NY 10016-5997, USA

30 September 2009

IEEE Std 802.11w™-2009
(Amendment to IEEE Std 802.11™-2007
as amended by IEEE Std 802.11k™-2008,
IEEE Std 802.11r™-2008, and
IEEE Std 802.11y™-2008)

IEEE Std 802.11w™-2009
(Amendment to IEEE Std 802.11™-2007
as amended by IEEE Std 802.11k™-2008,
IEEE Std 802.11r™-2008, and
IEEE Std 802.11y™-2008)

IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Amendment 4: Protected Management Frames

Sponsor

LAN/MAN Standards Committee
of the
IEEE Computer Society

Approved 11 September 2009

IEEE SA-Standards Board

Abstract: This amendment specifies the extensions to IEEE Std 802.11 for wireless local area networks (WLANs) providing mechanisms for protecting management frames.
Keywords: local area network (LAN)

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

PDF: ISBN 978-0-7381-6048-1 STD95962
Print: ISBN 978-0-7381-6049-8 STDPD95962

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 or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. The IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained herein is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

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. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation, or every ten years for stabilization. When a document is more than five years old and has not been reaffirmed, or more than ten years old and has not been stabilized, 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 publishing and making this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon his or her 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.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration. A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal interpretation of the IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Recommendations to change the status of a stabilized standard should include a rationale as to why a revision or withdrawal is required. Comments and recommendations on standards, and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854
USA

Introduction

This introduction is not part of IEEE Std 802.11w-2009, IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications—Amendment 4: Protected Management Frames.

IEEE Std 802.11-2007 as amended has been used in a wide range of mainstream business and personal applications. The success of products has resulted in an increased dependency on IEEE 802.11 as a primary method for the interconnection of networking equipment. This increased dependence has resulted in a need for increased assurance that the system will not be disrupted by the actions of unauthorized equipment. Such disruption can be caused by malicious systems generating false information and impersonating valid equipment. IEEE Std 802.11-2007 as amended addresses security of data frames but systems are still vulnerable to malicious attack because management frames are unprotected. At the same time, there is an increased dependence on management frames as a result of IEEE 802.11 amendments. The purpose of this amendment is to reduce the susceptibility of systems to such attack.

Notice to users

Laws and regulations

Users of these documents should consult all applicable laws and regulations. Compliance with the provisions of this standard does not imply 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.

Copyrights

Updating of IEEE documents

Users of IEEE standards 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. 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 the IEEE Standards Association website at http://ieeexplore.ieee.org/xpl/standards.jsp, or contact the IEEE at the address listed previously.

For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA website at http://standards.ieee.org.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/reading/ieee/updates/errata/index.html. Users are encouraged to check this URL for errata periodically.

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

Patents

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 with respect to the existence or validity of any patent rights in connection therewith. A patent holder or patent applicant has filed a statement of assurance that it will grant licenses under these 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. Other Essential Patent Claims may exist for which a statement 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.

Participants

At the time this amendment was sent to sponsor ballot, the IEEE 802.11 Working Group had the following officers:

Bruce P. Kraemer, *Chair*
Jon Walter Rosdahl, *Vice Chair, Treasurer, and Chair, Task Group mb*
Adrian P. Stephens, *Vice Chair*
Stephen McCann, *Secretary and Chair, Publicity Standing Committee*
Terry L. Cole, *Technical Editor and Assigned Number Authority*
Teik-Kheong Tan, *Chair, Wireless Next Generation Standing Committee*

David Bagby, *Chair, Architecture Standing Committee*
Bruce P. Kraemer, *Chair, Task Group n and Co-Chair, IMT-Advanced Ad hoc Committee*
Sheung Li, *Vice Chair, Task Group n*
Lee Armstrong, *Chair, Task Group p*
Donald E. Eastlake III, *Chair, Task Group s*
Neeraj Sharma, *Chair, Task Group t*
Stephen McCann, *Chair, Task Group u*
Dorothy V. Stanley, *Chair, Task Group v and IETF Ad hoc Committee*
Jesse Walker, *Chair, JCT1 Ad hoc Committee*
Peter Ecclesine, *Chair, Task Group y*
Menzo Wentink, *Chair, Task Group z*
Ganesh Venkatesan, *Chair, Task Group aa*
Eldad Perahia, *Chair, Very High Throughput Study Group*
Darwin Engwer, *Co-Chair, IMT-Advanced Ad hoc Committee*

When the IEEE 802.11 Working Group approved this amendment, Task Group w had the following officers:

Jesse Walker, *Chair*
Nancy Cam-Winget, Jon Edney, *Technical Editors*
Sandy Turner, *Secretary*

When this amendment was sent to sponsor ballot, the IEEE 802.11 Working Group had the following membership:

Osama S. Aboulmagd
Tomoko Adachi
Alok Aggarwal
Carlos H. Aldana
Thomas Alexander
Keith Amann
Lee Armstrong
Alex Ashley
Malik Audeh
David Bagby
Michael Bahr
Fan Bai
Gabor Bajko
Dennis J. Baker
Amit Bansal
John R. Barr
Gal Basson
Moussa Bavafa
Tuncer Baykas
Mathilde Benveniste
Bjorn A. Bjerke
Daniel Borges
Anthony Braskich
David Britz
G. Bumiller
Nancy Cam-Winget
Necati Canpolat
Javier Cardona
Douglas S. Chan
Lidong Chen
Nakjung Choi
Liwen Chu
Terry L. Cole
Ryon K. Coleman
Charles I. Cook
Todor Cooklev
Xavier P. Costa
David E. Cypher
Marc De Courville
Theodorus Denteneer
Jeremy deVries
Susan Dickey
Zhiming Ding
Yoshiharu Doi
John Dorsey
Roger P. Durand
Srinivasa Duvvuri
Donald E. Eastlake III
Peter Ecclesine
Mike Ellis
Stephen P. Emeott
Marc Emmelmann
Darwin Engwer
Joseph Epstein
Leonid Epstein
Vinko Erceg
Lars P. Falk
Robert Fanfelle
Stefan Fechtel
Paul H. Feinberg
Matthew J. Fischer
Wayne K. Fisher
Roberta Fracchia
James P. Gilb
Reinhard Gloger
David Goodall
Tugrul Guener
Jianlin Guo
Pratibha Gupta
Mark Hamilton
Christopher J. Hansen
Dan N. Harkins
Brian D. Hart
Amer A. Hassan
Vegard Hassel
Shigenori Hayase
Kevin V. Hayes
Robert F. Heile
Guido R. Hiertz
Junling Hu
Robert Y. Huang
David Hunter
Yasuhiko Inoue
Akio Iso
Junghoon Jee
Hongseok Jeon
Yeonkwon Jeong
Jorjeta G. Jetcheva
Lusheng Ji
Daniel Jiang
Padam Kafle
Carl W. Kain
Naveen K. Kakani
Masato Kato
Douglas Kavner
John Kenney
Stuart J. Kerry
Joonsuk Kim
Kyeongpyo Kim
Seong S. Kim
Yongsun Kim
Youjin Kim
Gunter Kleindl
Jarkko Kneckt
Mark M. Kobayashi
Fumihide Kojima
Tom Kolze
Bruce P. Kraemer
Johannes P. Kruys
Thomas Kuehnel
Rajendra Kumar
Thomas M. Kurihara
Joseph Kwak
Edwin Kwon
Jeremy A. Landt
Joseph P. Lauer
Jin Lee
Tae H. Lee
Joseph Levy
Sheung Li
Paul Lin
Hang Liu
Michael Livshitz
Peter Loc
Daniel Lubar
Anthony F. Maida
Jakub Majkowski
Alastair Malarky
Jouni K. Malinen

Bill Marshall
Sudheer Matta
Stephen McCann
Justin P. McNew
Robert R. Miller
Michael Montemurro
Rajendra T. Moorti
Hitoshi Morioka
Peter Murray
Andrew Myles
Rohit Nabar
Kengo Nagata
Chiu Ngo
Eero Nikula
Erwin Noble
Richard H. Noens
Hideaki Odagiri
Jisung Oh
Chandra S. Olson
Satoshi Oyama
Richard H. Paine
Arul Palanivelu
Jungsoo Park
Minyoung Park
Vijaykumar Patel
Bemini H. Peiris
Eldad Perahia
James E. Petranovich
Al Petrick
Fahd Pirzada
Subburajan Ponnuswamy
James D. Portaro
Henry S. Ptasinski
Chang W. Pyo
Emily H. Qi
Luke Qian
Jim E. Raab
Vinuth Rai
Ali Raissinia
Stephen G. Rayment
Leonid Razoumov
Ivan Reede
Joseph A. Repice
Edward Reuss
Randal Roebuck
Jon Walter Rosdahl
Richard Roy
Alexander Safonov
Kazuyuki Sakoda
Nicholas Sargologos
Katsuyoshi Sato
Hirokazu Sawada
Vincenzo Scarpa
Don Schultz
Yongho Seok
Huairong Shao
Neeraj Sharma
Stephen J. Shellhammer
Ian Sherlock
Kai Shi
Shusaku Shimada
Francois Simon
Harkirat Singh
Graham K. Smith
Matt Smith
Kapil Sood
Vinay Sridhara
Dorothy V. Stanley
Adrian P. Stephens
David S. Stephenson
Carl R. Stevenson
Guenael T. Strutt
Eiji Takagi
Mineo Takai
Teik-Kheong Tan
Allan Thomson
Jerry Thrasher
Alexander Tolpin
Jason Trachewsky
Solomon B. Trainin
Ashley Uyehara
Allert Van Zelst
Mathieu Varlet-Andre
Prabodh Varshney
Ganesh Venkatesan
Dalton T. Victor
George A. Vlantis
Jesse Walker
Qi Wang
Craig D. Warren
Fujio Watanabe
Patrick Waye
Menzo Wentink
Kyle Williams
James Worsham
Harry R. Worstell
Songping Wu
Pengfei Xia
Akiyoshi Yagi
Akira Yamada
Takeshi Yamamoto
Tomoya Yamaura
Zhiyu Yang
Peter Yee
Su K. Yong
Christopher Young
Artur Zaks
Hongyuan Zhang
Huimin Zhang
Meiyuan Zhao
Jing Zhu
Juan Zuniga
Jonathan Zweig

Major contributions were received from the following individuals:

Clint Chaplin
Lily Chen
Abhijit Choudhury
Frank Ciotti
David E. Cypher
Joseph Epstein
Matthew Gast
Dan N. Harkins
Kevin V. Hayes
Moo Ryong Jeong
Philip MacKenzie
Jouni K. Malinen
Robert Moskowitz
Emily H. Qi
Henry S. Ptasinski
Richard Roy
Kapil Sood
Fabrice Stevens
Fujio Watanabe
Marcus Wong
Peter Yee

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

Tomoko Adachi
Toru Aihara
Thomas Alexander
Butch Anton
Danilo Antonelli
Matthew Ball
Raja Banerjea
Gennaro Boggia
Nancy S. Bravin
William Byrd
Peter J. Calderon
Edward Carley
James Carlo
Juan Carreon
Clint Chaplin
Yung-Mu Chen
Hong Cheng
Keith Chow
Ryon K. Coleman
Charles I. Cook
Todor Cooklev
Susan Dickey
Russell Dietz
Petar Djukic
Sourav Dutta
Donald E. Eastlake III
Paul Eastman
Richard Eckard
Stephen P. Emeott
Marc Emmelmann
Joseph Epstein
Bernard Eydt
Matthew J. Fischer
Andre Fournier
Prince Francis
Matthew Gast
Devon Gayle
Michael Geipel
Mariana Goldhamer
Sergiu Goma
Randall Groves
C. Guy
Christopher J. Hansen
John Hawkins
Russell Housley
Yasuhiko Inoue
Atsushi Ito
Raj Jain
David Johnston
Bobby Jose
Joe Natharoj Juisai
Shinkyo Kaku
Piotr Karocki
Stuart J. Kerry
Brian Kiernan
Yongbum Kim
Joseph Kubler
Thomas M. Kurihara
Jeremy A. Landt
Daniel Levesque
Jan-Ray Liao
Arthur Light
Daniel Lubar
William Lumpkins
G. Luri
Jouni K. Malinen
Peter Martini
W. Kyle Maus
Stephen McCann
Gary Michel
Michael Montemurro
Jose Morales
Joseph Moran
Andrew Myles
Juichi Nakada
Hiroyuki Nakase
Michael S. Newman
Charles Ngethe
Satoshi Obara
Robert O'Hara
Stephen Palm
Subburajan Ponnuswamy
Cam Posani
Michael Probasco
Henry S. Ptasinski
Maximilian Riegel
Robert Robinson
Jon W. Rosdahl
Richard Roy
Herbert Ruck
Randall Safier
Anil Sanwalka
John Sargent
Oyama Satoshi
Kapil Sood
Amjad Soomro
Dorothy V. Stanley
Thomas Starai
Walter Struppler
Alourdes Sully
Masahiro Takagi
Joseph Tardo
Solomon Trainin
Mark-Rene Uchida
Prabodh Varshney
Jesse Walker
Stanley Wang
Fujio Watanabe
Stephen Webb
David Willow
Harry R. Worstell
Xuyong Wu
Oren Yuen
Paolo Zangheri
Meiyuan Zhao
Wenhao Zhu

When the IEEE-SA Standards Board approved this amendment on 11 September 2009, it had the following membership:

Robert M. Grow, *Chair*
Thomas Prevost, *Vice Chair*
Steve M. Mills, *Past Chair*
Judith Gorman, *Secretary*

John Barr
Karen Bartleson
Victor Berman
Ted Burse
Richard DeBlasio
Andy Drozd
Mark Epstein
Alexander Gelman
Jim Hughes
Richard H. Hulett
Young Kyun Kim
Joseph L. Koepfinger*
John Kulick
David J. Law
Ted Olsen
Glenn Parsons
Ronald C. Petersen
Narayanan Ramachandran
Jon Walter Rosdahl
Sam Sciacca
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Satish K. Aggarwal, *NRC Representative*
Michael Janezic, *NIST Representative*

Michelle Turner
IEEE Standards Program Manager, Document Development

Michael D. Kipness
IEEE Standards Program Manager, Technical Program Development

Contents

List of figures

List of tables

IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements

Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications

Amendment 4: Protected Management Frames

(This amendment is based on IEEE Std 802.11™-2007, as amended by IEEE Std 802.11k™-2008, IEEE Std 802.11r™-2008, and IEEE Std 802.11y™-2008.)

The editing instructions contained in this amendment define how to merge the material contained herein into the existing base standard and its amendments to form the comprehensive standard.[1]

The editing instructions are shown in ***bold italic***. Four editing instructions are used: ***change***, ***delete***, ***insert,*** and ***replace***. ***Change*** is used to make corrections in existing text or tables. The editing instruction specifies the location of the change and describes what is being changed by using ~~strikethrough~~ (to remove old material) and underscore (to add new material). ***Delete*** removes existing material. ***Insert*** adds new material without disturbing the existing material. Insertions may require renumbering. If so, renumbering instructions are given in the editing instructions. ***Replace*** is used to make changes in figures or equations by removing the existing figure or equation and replacing it with a new one. Editing instructions, change markings, and this NOTE will not be carried over into future editions because the changes will be incorporated into the base standard.

[1]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement the standard.

3. Definitions

Insert the following definitions alphabetically into Clause 3, renumbering as necessary:

3.223 Integrity GTK (IGTK): A random value, assigned by the broadcast/multicast source STA, which is used to protect group addressed medium access control (MAC) management protocol data units (MMPDUs) from that source STA.

3.224 Protected Dual of Public Action frame: Protected Dual of Public Action frames are defined as Action frames with the category value specified in 7.3.1.11 Table 7-24. For each Protected Dual of Public Action frame, there is a dual Action frame in a category that is specified with "No" in the "Robust" column of Table 7-24.

3.225 Robust Action frame: Robust Action frames are defined as Action frames with category values specified in 7.3.1.11 Table 7-24 with "Yes" in the "Robust" column.

3.226 Robust Management frame: A management frame that is eligible for protection.

4. Abbreviations and acronyms

Insert the following new abbreviations and acronyms in alphabetical order:

BIP	Broadcast/Multicast Integrity Protocol
IGTK	Integrity GTK
IPN	IGTK packet number
MFPC	Management Frame Protection Capable
MFPR	Management Frame Protection Required
MMIE	Management MIC Information Element
SA Query	Security Association Query

5. General description

5.2 Components of the IEEE 802.11 architecture

5.2.3 Distribution system (DS) concepts

5.2.3.2 RSNA

Insert the following item at the end of the dashed list in 5.2.3.2:

— Enhanced cryptographic encapsulation mechanisms for Robust Management frames

5.4 Overview of the services

5.4.2 Services that support the distribution service

5.4.2.4 Disassociation

Change the third paragraph of 5.4.2.4 as follows:

The disassociation service may be invoked by either party to an association (non-AP STA or AP). Disassociation is a notification, not a request. Disassociation cannot be refused by ~~either party to the association.~~the receiving STA except when Management Frame Protection is negotiated and the message integrity check fails.

5.4.3 Access control and data confidentiality services

5.4.3.2 Deauthentication

Change the second paragraph of 5.4.3.2 as follows:

In an ESS, because authentication is a prerequisite for association, the act of deauthentication shall cause the station to be disassociated. The deauthentication service may be invoked by either authenticated party (non-AP STA or AP). Deauthentication is not a request; it is a notification. ~~Deauthentication shall not be refused by either party. When an AP sends a deauthentication notice to an associated STA, the association shall also be terminated~~. The association at the transmitting STA is terminated when the STA sends a deauthentication notice to an associated STA. Deauthentication, and if associated, disassociation can not be refused by the receiving STA except when Management Frame Protection is negotiated and the message integrity check fails.

Change the fourth paragraph of 5.4.3.2 as follows:

In an RSNA, deauthentication also destroys any related pairwise transient key security association (PTKSA), group temporal key security association (GTKSA), station-to-station link (STSL) master key security association (SMKSA), ~~and~~ STSL transient key security association (STKSA), and integrity group temporal key security association (IGTKSA) that exist in the STA and closes the associated IEEE 802.1X Controlled Port. If pairwise master key (PMK) caching is not enabled, deauthentication also destroys the pairwise master key security association (PMKSA) from which the deleted PTKSA was derived.

5.4.3.3 Data confidentiality

Change 5.4.3.3 by inserting a new fourth paragraph and changing the last paragraph as follows:

IEEE Std 802.11 provides one security protocol, CCMP, for protection of unicast Robust Management frames. This standard does not provide data confidentiality for group addressed Robust Management frames.

The default data confidentiality state for all IEEE 802.11 STAs is "in the clear." If the data confidentiality service is not invoked, all ~~messages~~ frames shall be sent unprotected. If this policy is unacceptable to the sender, it shall not send data frames; and if the policy is unacceptable to the receiver, it shall discard any received data frames. Unprotected data frames and unprotected Robust Management frames received at a STA configured for mandatory data confidentiality, as well as protected data frames and protected Robust Management frames using a key not available at the receiving STA, are discarded without an indication to LLC (or without indication to distribution services in the case of "To DS" frames received at an AP). These frames are acknowledged on the WM [if received without frame check sequence (FCS) error] to avoid wasting WM bandwidth on retries of frames that are being discarded.

5.4.3.4 Key management

Change the text of 5.4.3.4 as follows:

The enhanced data confidentiality, data authentication, and replay protection mechanisms require fresh cryptographic keys and corresponding security associations. The procedures defined in this standard provide fresh keys by means of protocols called the 4-Way Handshake, FT 4-Way Handshake, FT Protocol, FT Resource Request Protocol, and Group Key Handshake.

5.4.3.5 Data origin authenticity

Change the text of 5.4.3.5 as follows:

The data origin authenticity mechanism defines a means by which a STA that receives a data or protected Robust Management frame can determine which STA transmitted the MAC protocol data unit (MPDU). This feature is required in an RSNA to prevent one STA from masquerading as a different STA. ~~This mechanism is provided for STAs that use CCMP or TKIP.~~

Data origin authenticity is only applicable to unicast data frames, and unicast Robust Management frames. The protocols do not guarantee data origin authenticity for broadcast/multicast ~~data~~ frames, as this cannot be accomplished using symmetric keys and public key methods are too computationally expensive.

5.4.3.6 Replay detection

Change the text of 5.4.3.6 as follows:

The replay detection mechanism defines a means by which a STA that receives a data or protected Robust Management frame from another STA can detect whether the received ~~data~~ frame is an unauthorized retransmission. This replay protection mechanism is provided for data frames for STAs that use CCMP or TKIP. The replay protection mechanism is also provided for Robust Management frames for STAs that use CCMP and Broadcast/Multicast Integrity Protocol (BIP).

5.4.3.7 Fast BSS transition

Insert the following new subclause (5.4.3.8) after 5.4.3.7 as follows:

5.4.3.8 Robust Management frame protection

Robust Management frames are a set of management frames that can be protected by the Management Frame Protection service. The Robust Management frames are Disassociation, Deauthentication, and Robust Action frames. Action frames specified with “No” in the “Robust” column of Table 7-24 are not Robust Management frames and shall not be protected.

Management Frame Protection protocols apply to Robust Management frames after RSNA PTK establishment for protection of unicast frames is completed and after delivery of the IGTK to protect group addressed frames. Robust Management frame protection is implemented by the CCMP and BIP protocols and the SA Query procedure.

5.8 IEEE Std 802.11 and IEEE Std 802.1X-2004

5.8.2 Infrastructure functional model overview

5.8.2.1 Authentication and key management (AKM) operations with Authentication Server (AS)

Change the second paragraph of 5.8.2.1 as follows:

A 4-Way Handshake or FT 4-Way Handshake utilizing EAPOL-Key frames is initiated by the Authenticator to do the following:

- Confirm that a live peer holds the PMK.
- Confirm that the PMK is current.
- In the case of fast BSS transition, derive PMK-R0s and PMK-R1s.
- Derive a fresh pairwise transient key (PTK) from the PMK or, in the case of fast BSS transition, from the PMK-R1.
- Install the pairwise encryption and integrity keys into IEEE Std 802.11.
- Transport the group temporal key (GTK) and GTK sequence number from Authenticator to Supplicant and install the GTK and GTK sequence number in the STA and, if not already installed, in the AP.
- If Management Frame Protection is negotiated, transport the IGTK and the IGTK packet number (IPN) from the Authenticator to the Supplicant and install these values in the STA and, if not already installed, in the AP.
- Verify that the RSN capabilities negotiated are valid as defined in 7.3.2.25.3.
- Confirm the cipher suite selection.

Insert the following paragraph at the end of 5.8.2.1:

When Management Frame Protection is negotiated, the Authenticator also uses the Group Key Handshake with all associated STAs to change the IGTK. The Authenticator encrypts the GTK and IGTK values in the EAPOL-Key frame as described in 8.5.

Replace Figure 5-13 with the following figure:

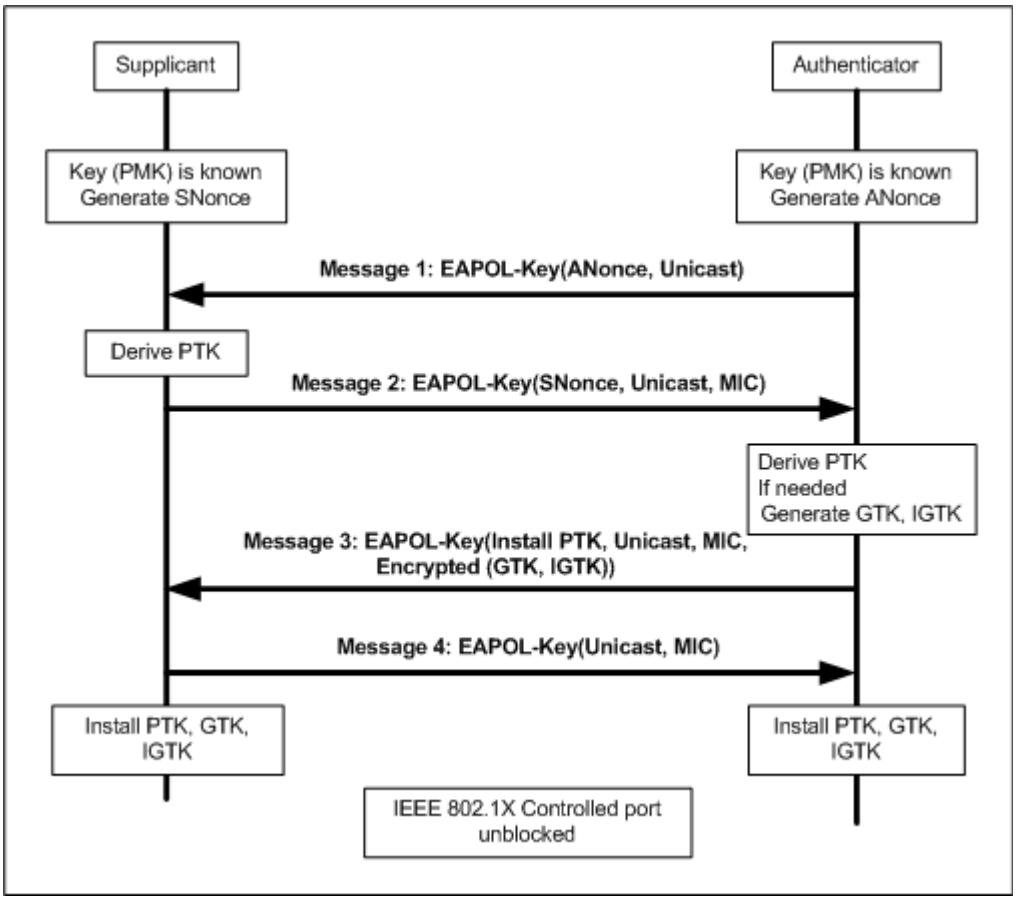

Figure 5-13—Establishing pairwise and group keys

Replace Figure 5-14 with the following figure:

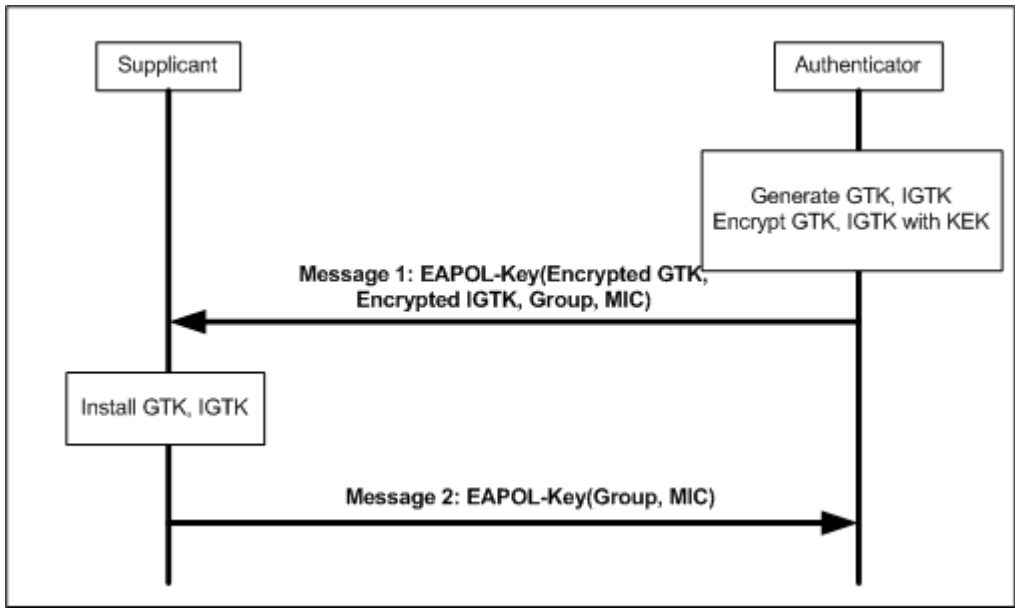

Figure 5-14—Delivery of subsequent group keys

5.8.2.2 Operations with PSK

Insert a new item after the third item in 5.8.2.2 as follows:

— If Management Frame Protection is negotiated, the IGTK and IGTK packet number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 5-13 and Figure 5-14.

5.8.5 PMKSA caching

Insert a new subclause (5.8.6) after 5.8.5 as follows:

5.8.6 Protection of broadcast and multicast Robust Management frames

When Management Frame Protection is negotiated, all group addressed Robust Management frames shall be encapsulated using the procedures defined in 11.12. This service provides integrity protection of group addressed Robust Management frames using BIP.

6. MAC service definition

6.1 Overview of MAC services

6.1.2 Security services

Change the text of 6.1.2 as follows:

Security services in IEEE Std 802.11 are provided by the authentication service and the TKIP, ~~and~~ CCMP, and BIP mechanisms. The scope of the security services provided is limited to station-to-station data and Robust Management frame transmissions ~~exchange~~. The data confidentiality service offered by an IEEE 802.11 TKIP ~~and CCMP~~ implementation is the protection of the MSDU. When CCMP is used, the data confidentiality service is provided for data frames and unicast Robust Management frames. For the purposes of this standard, TKIP and CCMP are viewed as logical services located within the MAC sublayer as shown in the reference model, Figure 5-10 (in 5.7). Actual implementations of the TKIP and CCMP services are transparent to the LLC and other layers above the MAC sublayer.

The security services provided by TKIP and CCMP in IEEE Std 802.11 are as follows:

a) Data Confidentiality;

b) Authentication; and

c) Access control in conjunction with layer management~~;~~.

BIP provides message integrity and access control for group addressed Robust Management frames.

During the authentication exchange, both parties exchange authentication information as described in Clause 8 and Clause 11A.

The MAC sublayer security services provided by TKIP, ~~and~~ CCMP, and BIP rely on information from nonlayer-2 management or system entities. Management entities communicate information to TKIP, ~~and~~ CCMP, and BIP through a set of MAC sublayer management entity (MLME) interfaces and MIB attributes; in particular, the decision tree for TKIP, ~~and~~ CCMP, and BIP defined in 8.7 is driven by MIB attributes.

The use of WEP for confidentiality, authentication, or access control is deprecated. The WEP algorithm is unsuitable for the purposes of this standard.

A STA that has associated with Management Frame Protection enabled shall not use pairwise cipher suite selectors WEP-40, WEP-104, TKIP, or "Use Group cipher suite."

7. Frame formats

7.1 MAC Frame formats

7.1.3 Frame fields

7.1.3.1 Frame control field

7.1.3.1.8 Protected Frame field

Change the text of 7.1.3.1.8 as follows:

The Protected Frame field is 1 bit in length. The Protected Frame field is set to 1 if the Frame Body field contains information that has been processed by a cryptographic encapsulation algorithm. The Protected Frame field is set to 1 only within data frames and within management frames of subtype Authentication, and unicast Robust Management frames. The Protected Frame field is set to 0 in all other frames. When the Protected Frame field is set to 1, the Frame Body field is protected utilizing the cryptographic encapsulation algorithm and expanded as defined in Clause 8. The Protected Frame field is set to 0 in Data frames of subtype Null Function, CF-ACK (no data), CF-Poll (no data), and CF-ACK+CF-Poll (no data) (see 8.3.2.2 and 8.3.3.1 that ~~, which~~ show that the frame body must be one octet or longer to apply the encapsulation).

7.2 Format of individual frame types

7.2.3 Management frames

7.2.3.3 Disassociation frame format

Change Table 7-9 and insert a NOTE at the end of the table as follows:

The frame body of a management frame of subtype Disassociation contains the information shown in Table 7-9.

Table 7-9—Disassociation frame body

Order	Information
1	Reason Code
2 — (Last – 1)	One or more vendor-specific information elements may appear in this frame.
Last	The Management MIC IE (MMIE) is present when Management Frame Protection is enabled at the AP and the frame is a group addressed frame.
NOTE—The MMIE appears after all fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 8.3.4.	

7.2.3.5 Association Response frame format

Insert order 13 into Table 7-11 as follows:

The frame body of a management frame of subtype Association Response contains the information shown in Table 7-11.

Table 7-11—Association Response frame body

Order	Information	Notes
13	Timeout Interval (Association Comeback time)	A Timeout Interval information element containing the Association Comeback time is present when dot11RSNAEnabled is true, dot11RSNAProtectedManagementFramesEnabled is true and the association request is rejected with a status code 30.
Last	Vendor Specific	One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements.

7.2.3.7 Reassociation Response frame format

Insert order 15 into Table 7-13 as follows:

The frame body of a management frame of subtype Association Response contains the information shown in Table 7-13.

Table 7-13—Reassociation Response frame body

Order	Information	Notes
15	Timeout Interval (Association Comeback time)	A Timeout Interval information element containing the Association Comeback time is present when dot11RSNAEnabled is true, dot11RSNAProtectedManagementFramesEnabled is true and the reassociation is rejected with status code 30.
Last	Vendor Specific	One or more vendor-specific information elements may appear in this frame. This information element follows all other information elements.

7.2.3.11 Deauthentication

Change Table 7-18 and insert a NOTE at the end of the table as follows:

The frame body of a management frame of subtype Deauthentication contains the information shown in Table 7-18.

Table 7-18—Deauthentication frame body

Order	Information
1	Reason code
2 — (Last – 1)	One or more vendor-specific information elements may appear in this frame. ~~This information element follows all other information elements.~~
Last	The Management MIC IE (MMIE) is present when Management Frame Protection is enabled at the AP and the frame is a group addressed frame.
NOTE—The MMIE appears after all fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 8.3.4.	

7.2.3.12 Action frame format

Change Table 7-19 and insert a NOTE at the end of the table as follows:

The frame body of a management frame of subtype Action contains the information shown in Table 7-19.

Table 7-19—Action frame body

Order	Information
1	Action
2 — (Last – 1)	One or more vendor-specific information elements may appear in this frame. ~~This information element follows all other information elements.~~
Last	The Management MIC IE (MMIE) is present when Management Frame Protection is enabled at the AP and the frame is a group addressed Robust Action frame.
NOTE—The MMIE appears after any fields that it protects. Therefore, it appears last in the frame body to protect the frames as specified in 8.3.4.	

7.3 Management frame body components

7.3.1 Fields that are not information elements

7.3.1.9 Status Code field

Change Table 7-23 as follows:

Table 7-23—Status codes

Status code	Meaning
29~~–31~~	Reserved
30	Association request rejected temporarily; try again later
31	Robust Management frame policy violation

7.3.1.11 Action field

Change Table 7-24 as follows:

Table 7-24—Category values

Code	Meaning	See subclause	Robust
0	Spectrum management	7.4.1	Yes
1	QoS	7.4.2	Yes
2	DLS	7.4.3	Yes
3	Block Ack	7.4.4	Yes
4	Public	7.4.7	No
5	Radio measurement	7.4.6	Yes
6	Fast BSS Transition	7.4.8	Yes
7	Reserved	—	—
8	SA Query	7.4.9	Yes
9	Protected Dual of Public Action	7.4	Yes
10–125 ~~6–126~~	Reserved	—	—
126	Vendor-specific Protected	7.4.5	Yes
127	Vendor-specific	7.4.5	No
128–255	Error	—	—

7.3.2 Information elements

Change the second to last paragraph in 7.3.2 as follows:

A STA that encounters an unknown or reserved element ID value in a management frame received without error shall ignore that element and shall parse any remaining management frame body for additional information elements with recognizable element ID values. The frame body components specified for many management subtypes result in elements ordered by ascending element ID, with the exception of the MIC Management IE (7.3.2.55). If present, the MIC Management IE appears at the end of the Robust Management frame body.

Change Table 7-26 as follows (note that the entire table is not shown):

Table 7-26—Element IDs

Information Element	Element ID	Length (in octets)	Extensible
Fast BSS Transition (FTIE) (see 7.3.2.48)	55	84 to 257	Yes
Timeout interval (see 7.3.2.49)	56	7	
RIC Data (RDIE) (see 7.3.2.50)	57	6	
DSE Registered Location (see 7.3.2.52)	58	22	
Supported Regulatory Classes (see 7.3.2.54)	59	4 to 255	
Extended Channel Switch Announcement (see 7.3.2.53)	60	6	
Reserved	61–62		
BSS Average Access Delay (see 7.3.2.39)	63	3	Yes
Antenna Information (see 7.3.2.40)	64	3	Yes
RSNI (see 7.3.2.41)	65	3	Yes
Measurement Pilot Transmission Information (see 7.3.2.42)	66	3 to 257	Subelements
BSS Available Admission Capacity (see 7.3.2.43)	67	4 to 28	Yes
BSS AC Access Delay (see 7.3.2.44)	68	6	Yes
Reserved	69		
RRM Enabled Capabilities (see 7.3.2.45)	70	7	Yes
Multiple BSSID (see 7.3.2.46)	71	3 to 257	Subelements
Reserved	72–74		
RIC Descriptor (see 7.3.2.51)	75	3 to 257	
Management MIC [see 7.3.2.55 (MMIE)]	76	18	
Reserved	77~~76~~–126		

7.3.2.25 RSN information element

Change the first paragraph of 7.3.2.25 as follows:

The RSN information element contains authentication and pairwise cipher suite selectors, a single group data cipher suite selector, an RSN Capabilities field, the PMK identifier (PMKID) count, a~~and~~ PMKID list~~.~~, and a single group management cipher suite selector. See Figure 7-72. All STAs implementing RSNA ~~shall~~ support this element. The size of the RSN information element is limited by the size of an information element, which is 255 octets. Therefore, the number of pairwise cipher suites, AKM suites, and PMKIDs is limited.

Replace Figure 7-72 with the following figure:

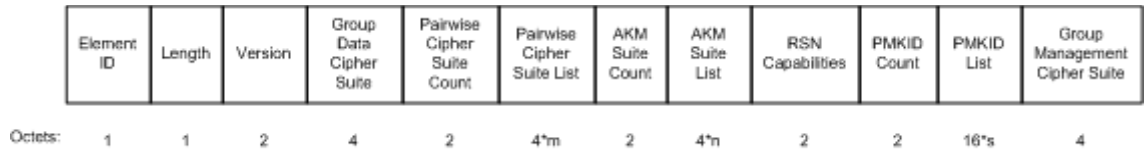

Figure 7-72—RSN Information Element format

Change the third paragraph in 7.3.2.25 as follows:

All fields use the bit convention from 7.1.1. The RSN information element shall contain up to and including the Version field. All fields after the Version field are optional. If any ~~optional~~ non-zero length field is absent, then none of the subsequent fields are ~~shall be~~ included.

Change the NOTE at the end of 7.3.2.25 as follows:

NOTE—The following represent sample information elements:

802.1X authentication, CCMP pairwise and group data cipher suites (WEP-40, WEP-104, and TKIP not allowed):

30, // information element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data key cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication
00 00 // No capabilities

802.1X authentication, CCMP pairwise and group data cipher suites (WEP-40, WEP-104, and TKIP not allowed), preauthentication supported:

30, // information element id, 48 expressed as Hex value
14, // length in octets, 20 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data key cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite

01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication
01 00 // Preauthentication capabilities

802.1X authentication, Use GTK for pairwise cipher suite, WEP-40 group data cipher suites, optional RSN Capabilities omitted:

30, // information element id, 48 expressed as Hex value
12, // length in octets, 18 expressed as Hex value
01 00, // Version 1
00 0F AC 01, // WEP-40 as group data key cipher suite
01 00, // pairwise cipher suite count
00 0F AC 00, // Use group key as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication

802.1X authentication, Use CCMP for pairwise cipher suite, CCMP group data cipher suites, preauthentication and a PMKID:

30, // information element id, 48 expressed as Hex value
26, // length in octets, 38 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication
01 00 // Preauthentication capabilities
01 00 // PMKID Count
01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 // PMKID

802.1X authentication, CCMP pairwise and group data key cipher suites (WEP-40, WEP-104, and TKIP are not allowed), and Management Frame Protection with AES-128-CMAC as the group management suite selector.

30, // information element id, 48 expressed as Hex value
1A, // length in octets, 26 expressed as Hex value
01 00, // Version 1
00 0F AC 04, // CCMP as group data key cipher suite
01 00, // pairwise cipher suite count
00 0F AC 04, // CCMP as pairwise cipher suite
01 00, // authentication count
00 0F AC 01 // IEEE 802.1X authentication
80 00 // Management Frame Protection is enabled but not required
00 00 // No PMKIDs
00 0F AC 06, // BIP as the broadcast/multicast management cipher suite

7.3.2.25.1 Cipher suites

Change the first paragraph of 7.3.2.25.1 as follows:

The Group Data Cipher Suite field contains the cipher suite selector used by the BSS to protect ~~broadcast/ multicast~~ group addressed data frames~~traffic~~.

Insert the following two new paragraphs after the third paragraph of 7.3.2.25.1 as follows:

The Group Management Cipher Suite field contains the cipher suite selector used by the BSS to protect group addressed Robust Management frames.

When Management Frame Protection is negotiated, the negotiated pairwise cipher suite is used to protect unicast Robust Management frames, and the group management cipher suite is used to protect group addressed Robust Management frames. Use of AES-128-CMAC is not valid as a data cipher suite.

Change Table 7-32 as follows:

Table 7-32—Cipher suite selectors

OUI	Suite type	Meaning
00-0F-AC	0	Use group cipher suite
00-0F-AC	1	WEP-40
00-0F-AC	2	TKIP
00-0F-AC	3	Reserved
00-0F-AC	4	CCMP—default pairwise cipher suite and default group cipher suite for data frames in an RSNA
00-0F-AC	5	WEP-104
00-0F-AC	6	BIP—default group management cipher suite in an RSNA with Management Frame Protection enabled
00-0F-AC	~~6~~7–255	Reserved
Vendor OUI	Other	Vendor-specific
Other	Any	Reserved

Change Table 7-33 as follows:

Table 7-33—Cipher suite usage

Cipher suite selector	GTK	PTK	IGTK
Use group key	No	Yes	No
WEP-40	Yes	No	No
WEP-104	Yes	No	No
TKIP	Yes	Yes	No
CCMP	Yes	Yes	No
BIP	No	No	Yes

7.3.2.25.2 AKM suites

Change Table 7-34 as follows:

Table 7-34—AKM suite selectors

OUI	Suite type	Meaning		
		Authentication type	Key management type	Key derivation type
00-0F-AC	0	Reserved	Reserved	Reserved
00-0F-AC	1	Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 8.4.6.2—RSNA default	RSNA key management as defined in 8.5 or using PMKSA caching as defined in 8.4.6.2—RSNA default	Defined in 8.5.1.1
00-0F-AC	2	PSK	RSNA key management as defined in 8.5, using PSK	Defined in 8.5.1.1
00-0F-AC	3	FT authentication negotiated over IEEE 802.1X	FT key management as defined in 8.5.1.5	Defined in 8.5.1.5.2
00-0F-AC	4	FT authentication using PSK	FT key management as defined in 8.5.1.5	Defined in 8.5.1.5.2
00-0F-AC	5	Authentication negotiated over IEEE 802.1X or using PMKSA caching as defined in 8.4.6.2 with SHA256 Key Derivation	RSNA Key Management as defined in 8.5 or using PMKSA caching as defined in 8.4.6.2, with SHA256 Key Derivation	Defined in 8.5.1.5.2
00-0F-AC	6	PSK with SHA256 Key Derivation	RSNA Key Management as defined in 8.5 using PSK with SHA256 Key Derivation	Defined in 8.5.1.5.2
00-0F-AC	~~5~~7–255	Reserved	Reserved	Reserved
Vendor OUI	Any	Vendor Specific	Vendor Specific	Vendor Specific
Other	Any	Reserved	Reserved	Reserved

7.3.2.25.3 RSN capabilities

Replace Figure 7-74 with the following figure:

B0	B1	B2-B3	B4-B5	B6	B7	B8	B9	B10-15
Pre-Auth	No Pairwise	PTKSA Replay Counter	GTKSA Replay Counter	Management Frame Protection Required (MFPR)	Management Frame Protection Capable (MFPC)	Reserved	Peer Key Enabled	Reserved

Figure 7-74—RSN Capabilities field format

Insert the following two items after dashed list item "Bits 4–5":

— Bit 6: Management Frame Protection Required (MFPR). A STA sets this bit to 1 to advertise that protection of Robust Management Frames is mandatory. A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesEnabled is TRUE and dot11RSNAUnprotectedManagementFramesAllowed is FALSE, otherwise it sets this bit to 0. If a STA sets this bit to 1, then that STA only allows RSNAs with STAs that provide Management Frame Protection.

— Bit 7: Management Frame Protection Capable (MFPC). A STA sets this bit to 1 when dot11RSNAProtectedManagementFramesEnabled is set to TRUE, to advertise that protection of Robust Management Frames is enabled.

Change dashed list item "Bits 6–8 and 10–15" as follows:

— Bit~~s 6–~~8 and 10–15: Reserved. The remaining subfields of the RSN Capabilities field are reserved ~~and shall be set to 0 on transmission and ignored on reception~~.

7.3.2.48 Fast BSS transition information element (FTIE)

Change Table 7-43g as follows:

Table 7-43g—Sub-element IDs

Value	Contents of data field	Length (in octets)
4	IGTK	Variable
~~4~~5–255	Reserved	

Insert the following text and Figure 7-95o6a after Figure 7-95o6 as follows:

The IGTK field contains the Integrity GTK, used for protecting Robust Management frames. The IGTK sub-element format is shown in Figure 7-95o6a.

	Sub-element ID	Length	KeyID	IPN	Key Length	Key
Octets	1	1	2	6	1	24

Figure 7-95o6a—IGTK sub-element format

The KeyID field indicates the value of the BIP key ID.

The IPN field indicates the receive sequence counter for the IGTK being installed. The PN field gives the current message number for the IGTK, to allow a STA to identify replayed MPDUs.

The Key Length field is the length of IGTK in octets. This length value does not include the possible padding (see 11A.8.5).

The Key field is the IGTK being distributed. The length of the resulting AES-Keywrapped IGTK in the Key field is Key Length + 8 octets.

7.3.2.49 Timeout Interval information element (TIE)

Change Table 7-43h as follows:

Table 7-43h—Timeout Interval Type field value

Timeout interval type	Meaning	Units
0	Reserved	
1	Reassociation deadline interval	Time units (TUs)
2	Key lifetime interval	Seconds
3	Association Comeback time	Time units (TUs)
~~43~~–255	Reserved	

7.3.2.54 Supported Regulatory Classes element

Insert a new subclause (7.3.2.55) at the end of 7.3.2.54 as follows:

7.3.2.55 Management MIC information element

The Management MIC information element (MMIE) provides message integrity and protects group addressed Robust Management frames from forgery and replay. Figure 7-95o15 shows the MMIE format.

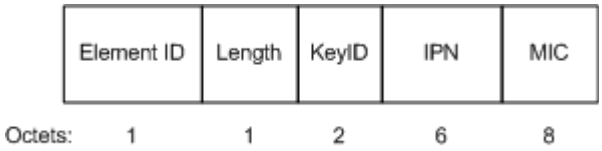

Figure 7-95o15—Management MIC information element format

The value of the Element ID field is 76 decimal (4c hex).

The Length field is set to 16.

The Key ID field identifies the IGTK used to compute the MIC. Bits 0–11 define a value in the range 0–4095. Bits 12–15 are reserved and set to 0 on transmission and ignored on reception. The IGTK Key ID is either 4 or 5. The remaining Key IDs are reserved.

The IPN field contains a 6 octet value, interpreted as a 48-bit unsigned integer and used to detect replay of protected group addressed Robust Management frames.

The MIC field contains a message integrity code calculated over the Robust Management frame as specified in 8.3.4.5 and 8.3.4.6.

7.4 Action frame format details

7.4.5 Vendor-specific action details

Insert a NOTE at the end of the first paragraph of 7.4.5 as follows:

NOTE—If Management Frame Protection is negotiated, then Vendor Specific Protected Action frames (see Table 7-24) are protected; otherwise they are unprotected.

7.4.7 Public Action details

7.4.7.1 Public Action frames

Change Table 7-57e as follows:

Table 7-57e—Public Action field values

Action field value	Description
0	Reserved
1	DSE enablement
2	DSE deenablement
3	DSE Registered Location Announcement
4	Extended Channel Switch Announcement
5	DSE measurement request
6	DSE measurement report
7	Measurement Pilot
8	DSE power constraint
9	Vendor Specific
10~~9~~–255	Reserved

7.4.7.9 DSE Power Constraint frame format

Insert a new subclause (7.4.7.10) at the end of 7.4.7.9 as follows:

7.4.7.10 Vendor Specific Public Action frame format

The Vendor Specific Public Action frame is defined for vendor-specific signaling between unassociated STAs. The format of the Vendor Specific Public Action frame is shown in Figure 7-95o16.

	Category	Action	OUI	Vendor Specific Content
Octets:	1	1	3	variable

Figure 7-95o16—Vendor Specific Public Action frame format

The Category field is set to the value indicating the Public category, as specified in Table 7-24 in 7.3.1.11.

The Action field is set to the value indicating Vendor Specific Public, as specified in Table 7-57e in 7.4.7.1.

The OUI field is a public OUI assigned by the IEEE. It is 3 octets in length. It contains the OUI of the entity that has defined the content of the particular Vendor Specific Public Action.

The Vendor Specific Content contains the vendor-specific fields and may include Information Elements defined in the standard. The length of the Vendor Specific Content in a Vendor Specific Public Action frame is limited by the maximum allowed MMPDU size.

7.4.8 Action frame details

Insert two new subclauses (7.4.9 and 7.4.9a) at the end of 7.4.8 as follows:

7.4.9 SA Query Action frame details

Two Action frame formats are defined for the SA Query procedure. An Action field, in the octet field immediately after the Category field, differentiates the formats. The Action field values associated with each frame format are defined in Table 7-57l.

NOTE—The SA query functionality defined in this standard is used to prevent the Association Lockout problem (defined in 11.3).

Table 7-57l—SA Query Action fields

Action field value	Description
0	SA Query Request
1	SA Query Response

7.4.9.1 SA Query Request frame

The SA Query Request frame is used to request a SA Query Response from the receiving STA. The format of the frame is shown in Figure 7-101m.

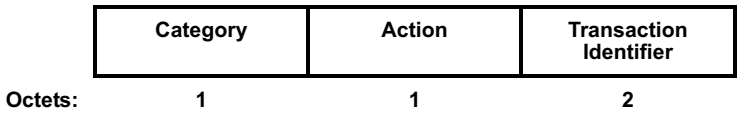

Figure 7-101m—SA Query Request frame details

The Category field is set to the value indicating the SA Query category, as specified in Table 7-24 in 7.3.1.11.

The Action field is set to the value indicating SA Query Request frame, as specified in Table 7-57l in 7.4.9.

The Transaction Identifier field is a 16-bit non-negative counter value set by the STA sending the SA Query Request frame to identify any outstanding request/response transaction.

7.4.9.2 SA Query Response frame

The SA Query Response frame is used to respond to an SA Query Request frame from another STA. The format of the frame is shown in Figure 7-101n.

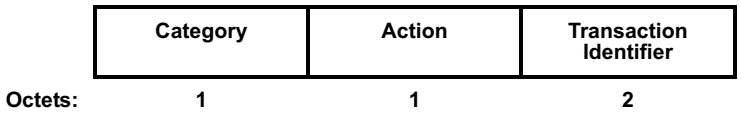

Figure 7-101n—SA Query Response frame details

The Category field is set to the value indicating the SA Query category, as specified in Table 7-24 in 7.3.1.11.

The Action field is set to the value indicating SA Query Response frame, as specified in Table 7-57l in 7.4.9.

The Transaction Identifier field is set to the same value as the Transaction Identifier field in the corresponding SA Query Request frame.

7.4.9a Protected Dual of Public Action frames

7.4.9a.1 Protected Dual of Public Action details

The Protected Dual of Public Action frame is defined to allow robust STA-STA communications of the same information that is conveyed in action frames that are not robust (see 7.3.1.11). The defined Protected Dual of Public Action frames are listed in Table 7-57m.

Table 7-57m—Protected Dual of Public Action field values

Action field value	Description
0	Reserved
1	Protected DSE Enablement
2	Protected DSE Deenablement
3	Reserved
4	Protected Extended Channel Switch Announcement
4	Protected Measurement Request
6	Protected Measurement Report
7	Reserved
8	Protected DSE Power Constraint
9–255	Reserved

7.4.9a.2 Protected DSE Enablement frame format

The Protected DSE Enablement frame format is the same as the DSE Enablement frame format (see 7.4.7.3). It is used instead of the DSE Enablement Action frame when Management Frame Protection is negotiated.

7.4.9a.3 Protected DSE Deenablement frame format

The Protected DSE Deenablement frame format is the same as the DSE Deenablement frame format (see 7.4.7.4). It is used instead of the DSE Deenablement Action frame when Management Frame Protection is negotiated.

7.4.9a.4 Protected Extended Channel Switch Announcement frame format

The Protected Extended Channel Switch Announcement frame format is the same as the Extended Channel Switch Announcement frame format (see 7.4.7.6). It is used instead of the Extended Channel Switch Announcement Action frame when Management Frame Protection is negotiated.

7.4.9a.5 Protected DSE Measurement Request frame format

The Protected DSE Measurement Request frame format is the same as the DSE Measurement Request frame format (see 7.4.7.7). It is used instead of the DSE Measurement Request Action frame when Management Frame Protection is negotiated.

7.4.9a.6 Protected DSE Measurement Report frame format

The Protected DSE Measurement Report frame format is the same as the DSE Measurement Report frame format (see 7.4.7.8). It is used instead of the DSE Measurement Report Action frame when Management Frame Protection is negotiated.

7.4.9a.7 Protected DSE Power Constraint frame format

The Protected DSE Power Constraint frame format is the same as the DSE Power Constraint frame format (see 7.4.7.9). It is used instead of the DSE Power Constraint Action frame when Management Frame Protection is negotiated.

8. Security

8.1 Framework

8.1.1 Security methods

Insert the following item in the second dashed list, between CCMP and RSNA:

— BIP, described in 8.3.4

8.1.3 RSNA establishment

Insert item 7) at the end of the list beginning with a) as follows:

7) If the STAs negotiate Management Frame Protection, the STA programs the TK and pairwise cipher suite into the MAC for protection of unicast Robust Management frames. It also installs the IGTK and IPN for protection of group addressed Robust Management frames.

Insert item 6) at the end of the list beginning with b) as follows:

6) If the STAs negotiate Management Frame Protection, the STA programs the negotiated pairwise cipher suite and established TK, IGTK, and IPN.

Change the title of 8.3 as follows:

8.3 RSNA ~~data~~ confidentiality and integrity protocols

8.3.1 Overview

Change the text of 8.3.1 as follows:

This standard defines two RSNA data confidentiality and integrity protocols: TKIP and CCMP. ~~Implementation of CCMP shall be mandatory in all IEEE 802.11 devices claiming RSNA compliance.~~ This standard defines one integrity protocol: BIP.

Implementation of TKIP is optional for an RSNA and used only for the protection of data frames. A design aim for TKIP was that the algorithm should be implementable within the capabilities of most devices supporting only WEP, so that many such devices would be field-upgradeable by the supplier to support TKIP.

BIP is a mechanism, used only when Management Frame Protection is negotiated. BIP provides integrity protection for group addressed Robust Management frames. BIP is only used to protect management frames within the BSS.

8.3.3 CTR with CBC-MAC Protocol (CCMP)

8.3.3.1 CCMP Overview

Insert the following paragraph at the end of 8.3.3.1:

When CCMP is selected as the RSN pairwise cipher and Management Frame Protection is negotiated, unicast Robust Management frames shall be protected with CCMP.

8.3.3.3 CCMP cryptographic encapsulation

8.3.3.3.2 Construct AAD

Replace Figure 8-17 with the following figure:

Figure 8-17—AAD construction

Change the third paragraph of 8.3.3.3.2 as follows:

The AAD is constructed from the MPDU header. The AAD does not include the header Duration field, because the Duration field value can change due to normal IEEE 802.11 operation (e.g., a rate change during retransmission). For similar reasons, several subfields in the Frame Control field are masked to 0. AAD construction is performed as follows:

a) FC—MPDU Frame Control field, with:
   1) Subtype bits (bits 4 5 6) in a Data MPDU masked to 0
   2) Retry bit (bit 11) masked to 0
   3) PwrMgt bit (bit 12) masked to 0
   4) MoreData bit (bit 13) masked to 0
   5) Protected Frame bit (bit 14) always set to 1
b) A1—MPDU Address 1 field.
c) A2—MPDU Address 2 field.
d) A3—MPDU Address 3 field.
e) SC—MPDU Sequence Control field, with the Sequence Number subfield (bits 4–15 of the Sequence Control field) masked to 0. The Fragment Number subfield is not modified.
f) A4—MPDU Address field, if present ~~in the MPDU~~.
g) QC—QoS Control field, if present, a 2-octet field that includes the MSDU priority. The QC TID field is used in the construction of the AAD, and the remaining QC fields are set to 0 for the AAD calculation (bits 4 to 15 are set to 0).

8.3.3.3.3 Construct CCM nonce

Replace Figure 8-18 with the following figure:

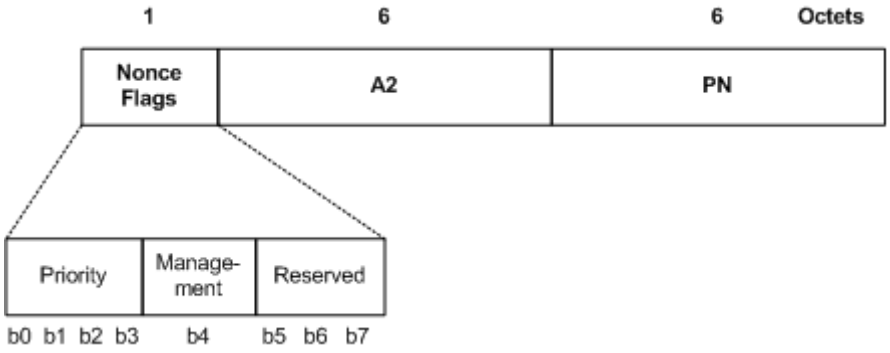

Figure 8-18—Nonce construction

Change the second paragraph 8.3.3.3.3 as follows:

The Nonce field has an internal structure of ~~Priority Octet~~ Nonce Flags || A2 || PN ("||" is concatenation), where

— ~~The Priority Octet field shall be set to the fixed value 0 (0x00) when there is no QC field present in the MPDU header. When the QC field is present, bits 0 to 3 of the priority Octet field shall be set to the value of the QC TID (bits 0 to 3 of the QC field). Bits 4 to 7 of the Priority Octet field are reserved and shall be set to 0.~~

— The Priority subfield of the Nonce Flags field shall be set to the fixed value 0 when there is no QC field present in the MPDU header. When the QC field is present, bits 0 to 3 of the Priority field shall be set to the value of the QC TID (bits 0 to 3 of the QC field).

— When Management Frame Protection is negotiated, the Management field of the Nonce Flags field shall be set to 1 if the Type field of the Frame Control field is 00 (Management frame); otherwise it is set to 0.

— Bits 5 to 7 of the Nonce Flags field are reserved and shall be set to 0 on transmission.

— MPDU Address A2 field occupies octets 1–6. This shall be encoded with the octets ordered with A2 octet 0 at octet index 1 and A2 octet 5 at octet index 6.

— The PN field occupies octets 7–12. The octets of PN shall be ordered so that PN0 is at octet index 12 and PN5 is at octet index 7.

8.3.3.3.5 CCM originator processing

Insert the following new paragraph at the end of 8.3.3.3.5:

A CCMP protected unicast Robust Management frame shall be protected with the TK.

8.3.3.4 CCMP decapsulation

Change list item c) as follows:

c) The Nonce value is constructed from the A2, PN, and ~~Priority Octet~~ Nonce Flags fields.

Insert the following paragraph at the end of 8.3.3.4 (before 8.3.3.4.1):

When the received frame is a CCMP protected unicast Robust Management frame, contents of the MMPDU body after protection is removed shall be delivered to the SME via the MLME primitive designated for that management frame rather than through the MA-UNITDATA.indication primitive.

8.3.3.4.1 CCM recipient processing

Insert the following sentence at the end of the first paragraph in 8.3.3.4.1:

A CCMP protected unicast Robust Management frame shall use the same TK as a Data MPDU.

8.3.3.4.3 PN and replay detection

Change list item e) as follows:

e) For each PTKSA, GTKSA, and STKSA, the recipient shall maintain a separate replay counter for each IEEE 802.11 MSDU priority and shall use the PN recovered from a received frame to detect replayed frames, subject to the limitation of the number of supported replay counters indicated in the RSN Capabilities field (see 7.3.2.25). A replayed frame occurs when the PN extracted from a received frame is less ~~that~~ than or equal to the current replay counter value for the frame's MSDU priority and frame type. A transmitter shall not use IEEE 802.11 MSDU priorities without ensuring that the receiver supports the required number of replay counters. The transmitter shall not reorder frames within a replay counter, but may reorder frames across replay counters. One possible reason for reordering frames is the IEEE 802.11 MSDU priority.

Insert the following list item after e):

e1) If dot11RSNAProtectedManagementFramesEnabled is TRUE, the recipient shall maintain a single replay counter for received unicast Robust Management frames and shall use the PN from the received frame to detect replays. A replayed frame occurs when the PN from the frame is less than or equal to the current management frame replay counter value. The transmitter shall preserve the order of protected Robust Management frames sent to the same DA.

Change list item f) as follows:

f) The receiver shall discard MSDUs and MMPDUs whose constituent MPDU PN values are not sequential. A receiver shall discard any MPDU that is received with its PN less than or equal to the replay counter. When discarding a frame, the receiver~~and~~ shall increment by 1 the value of dot11RSNAStatsCCMPReplays for data frames or dot11RSNAStatsRobustMgmtCCMPReplays for Robust Management frames~~this key~~.

Insert the following new subclause (8.3.4) after 8.3.3:

8.3.4 The Broadcast/Multicast integrity protocol

The Broadcast/Multicast Integrity Protocol (BIP) provides data integrity and replay protection for group addressed Robust Management frames after successful establishment of an IGTKSA (see 8.4.1.1.3a).

8.3.4.1 BIP overview

BIP provides data integrity and replay protection, using AES-128 in CMAC Mode. NIST SP 800-38B defines the CMAC algorithm. All BIP processing uses AES with a 128-bit integrity key and a 128-bit block size, and a CMAC TLen value of 128 (16 octets). The CMAC output is truncated to 64 bits:

$$MIC = L(CMAC\ Output, 0, 64) \quad (1)$$

Where L is defined in 8.5.1.

BIP uses the IGTK to compute the MMPDU MIC. The authenticator shall distribute one new IGTK and IGTK PN (IPN) whenever it distributes a new GTK. The IGTK is identified by the MAC address of the transmitting STA plus an IGTK identifier that is encoded in the MMIE Key ID field.

8.3.4.2 BIP MMPDU format

The Management MIC IE shall follow all of the other IEs in the management frame body but precede the FCS. See 7.3.2.55 for the format of the Management MIC IE. Figure 8-19a shows the BIP MMPDU.

IEEE 802.11 Header	Management Frame Body including MMIE	FCS

Figure 8-19a—BIP Encapsulation

8.3.4.3 BIP AAD construction

The BIP Additional Authenticated Data (AAD) shall be constructed from the MPDU header. The Duration field in the AAD shall be masked to 0. The AAD construction shall use a copy of the IEEE 802.11 header without the SC field for the MPDU, with the following exceptions:

a) FC—MPDU Frame Control field, with:
   1) Retry bit (bit 11) masked to zero
   2) PwrMgt bit (bit 12) masked to zero
   3) MoreData bit (bit 13) masked to zero
b) A1—MPDU Address 1 field.
c) A2—MPDU Address 2 field.
d) A3—MPDU Address 3 field.

Figure 8-19b depicts the format of the AAD. The length of the AAD is 20 octets.

	FC	A1	A2	A3
Octets:	2	6	6	6

Figure 8-19b—BIP AAD Construction

8.3.4.4 BIP replay protection

The MMIE Sequence Number field represents a sequence number whose length is 6 octets.

When Management Frame Protection is negotiated, the receiver shall maintain a 48-bit replay counter for each IGTK. The receiver shall set the receive replay counter to the value of the IPN in the IGTK KDE

provided by the Authenticator in either the 4-Way Handshake, FT 4-Way Handshake, FT Handshake, or Group Key Handshake. The transmitter may reinitialize the sequence counter when the IGTK is refreshed. See 8.3.4.5 and 8.3.4.6 for per packet BIP processing.

NOTE—When the IPN space is exhausted, the choices available to an implementation are to replace the IGTK or to end communications.

8.3.4.5 BIP transmission

When a STA transmits a protected group addressed Robust Management frame, it shall:

a) Select the IGTK currently active for transmission of frames to the intended group of recipients and construct the MMIE (see 7.3.2.55) with the MIC field masked to zero and the KeyID field set to the corresponding IGTK KeyID value. The transmitter shall insert a monotonically increasing non-negative integer into the MMIE IPN field.

b) Compute AAD as specified in 8.3.4.3.

c) Compute AES-128-CMAC over the concatenation of (AAD || Management Frame Body including MMIE), and insert the 64-bit output into the MMIE MIC field.

d) Compose the frame as the IEEE 802.11 header, management frame body, including MMIE, and FCS. The MMIE shall appear last in the frame body.

e) Transmit the frame.

8.3.4.6 BIP reception

When a STA with Management Frame Protection negotiated receives a group addressed Robust Management frame protected by BIP, it shall:

a) Identify the appropriate IGTK key and associated state based on the MMIE KeyID field. If no such IGTK exists, silently drop the frame.

b) The STA shall perform replay protection on the received frame. The receiver shall interpret the MMIE IPN field as a 48-bit unsigned integer. It shall compare this MMIE IPN integer value against the receive replay counter for the IGTK identified by the MMIE Key ID field. If the integer value from the received MMIE IPN field is less than or equal to the replay counter value for this IGTK, the receiver shall discard the frame and increment the dot11RSNAStatsCMACReplays counter by 1. The receiver shall extract and save the received MIC value, and compute the AES-128-CMAC over the concatenation of (AAD || Management Frame Body including MMIE) with the MIC field masked to zero in the MMIE. If the result does not match the received MIC value, then the receiver shall discard the frame and increment the dot11RSNAStatsCMACICVErrors counter by 1.

c) If the replay protection succeeds, the receiver shall compute AAD for this management frame, as specified in 8.3.4.3.

d) The receiver shall extract and save the received MIC value, and compute the AES-128-CMAC over the concatenation of (AAD || Management Frame Body || MMIE) with the MIC field masked to zero in the MMIE. If the result does not match the received MIC value, then the receiver shall discard the frame and increment the dot11RSNAStatsCMACICVErrors counter by 1.

e) The receiver shall update the replay counter for the IGTK identified by the MMIE Key ID field with the integer value of the MMIE IPN field.

If Management Frame Protection is negotiated, group addressed Robust Management frames that are received without BIP protection shall be discarded.

8.4 RSNA security association management

8.4.1 Security associations

8.4.1.1 Security association definitions

Change the second paragraph as follows:

A security association is a set of policy(ies) and key(s) used to protect information. The information in the security association is stored by each party of the security association, must be consistent among all parties, and must have an identity. The identity is a compact name of the key and other bits of security association information to fit into a table index or an MPDU. ~~There are four types of security associations supported by an RSN STA:~~The following types of security associations are supported by an RSN STA:

Insert the following item after the third dashed item (after GTKSA):

— IGTKSA: A result of a successful Group Key Handshake, successful 4-Way Handshake, FT 4-Way Handshake, or the Reassociation Response message of the Fast BSS Transition protocol.

8.4.1.1.3 GTKSA

Insert the following new subclause (8.4.1.1.3a) after 8.4.1.1.3:

8.4.1.1.3a IGTKSA

When Management Frame Protection is enabled, a non-AP STA's SME creates an IGTKSA when it receives a valid Message 3 of the 4-Way Handshake or FT 4-Way Handshake, the Reassociation Response message of the Fast BSS Transition protocol with a status code indicating success, or a valid Message 1 of the Group Key Handshake. The Authenticator's SME creates an IGTKSA when it establishes or changes the IGTK with all STAs to which it has a valid PTKSA.

An IGTKSA consists of the following elements:

— Direction vector (whether the IGTK is used for transmit or receive)
— KeyID
— IGTK
— Authenticator MAC address

8.4.1.2 Security association life cycle

8.4.1.2.1 Security association in an ESS

Change lettered list item d) in 8.4.1.2.1 as follows:

d) The last step is key management. The authentication process creates cryptographic keys shared between the IEEE 802.1X AS and the STA. The AS transfers these keys to the AP, and the AP and STA use one key confirmation handshake, called the 4-Way Handshake, to complete security association establishment. The key confirmation handshake indicates when the link has been secured by the keys and is ready to allow normal data traffic and protected Robust Management frames.

Change the second dashed item in 8.4.1.2.1 as follows:

— A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSN information element of its (Re)Association Request frame. An AP whose Authenticator has retained the PMK for one or more of the PMKIDs can skip the IEEE 802.1X authentication and proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, or if the AKM of the cached PMKSA differs from that offered in the (Re)Association Request, or the PMK in the cached PMKSA is no longer valid, then the Authenticator shall perform another IEEE 802.1X authentication. Similarly, if the STA fails to send a PMKID, the STA and AP must perform a full IEEE 802.1X authentication.

Change the last sentence of the last paragraph as follows:

A STA's SME uses this primitive when it deletes a PTKSA, ~~or~~ GTKSA, or IGTKSA.

8.4.3 RSNA policy selection in an ESS

Insert the following at the end of 8.4.3 (before 8.4.3.1):

An RSNA-enabled AP shall use Table 8-1a and the Management Frame Protection Capable (MFPC) and Management Frame Protection Required (MFPR) advertised in the RSN IEs to determine if it may associate with a non-AP STA. An RSNA enabled non-AP STA shall use Table 8-1a and the values of the Management Frame Protection Capable and Management Frame Protection Required bits advertised in the RSN IEs to determine if it may associate with an AP. Management Frame Protection is enabled when dot11RSNAProtectedManagementFramesEnabled is set to 1. Management Frame Protection is negotiated when an AP and non-AP STA set the Management Frame Protection Capable field to 1 in their respective RSN IEs in the (re)association procedure, and both parties confirm the Management Frame Protection Capable bit set to 1 in the 4-Way Handshake, FT 4-Way Handshake, or the FT Fast BSS Transition protocol.

Table 8-1a—Robust Management frame selection in an ESS

AP MFPC	AP MFPR	STA MFPC	STA MFPR	AP Action	STA Action
0	0	0	0	The AP may associate with the STA	The STA may associate with the AP
1	0	0	0	The AP may associate with the STA	The STA may associate with the AP
1	0 or 1	1	0 or 1	The AP may associate with the STA	The STA may associate with the AP
1	1	0	0	The AP shall reject associations from the STA with the Status Code "Robust Management frame policy violation"	The STA shall not associate with the AP
0	0	1	1	No action	The STA shall not try to associate with the AP
0	0	1	0	The AP may associate with the STA	The STA may associate with the AP

Table 8-1a—Robust Management frame selection in an ESS *(continued)*

AP MFPC	AP MFPR	STA MFPC	STA MFPR	AP Action	STA Action
1	0 or 1	0	1	The STA advertises an invalid setting. The AP shall reject associations from the STA with the Status Code "Robust Management frame policy violation"	The STA shall not try to associate with the AP
0	1	1	0 or 1	No action	The AP advertises an invalid setting. The STA shall not try to associate with the AP

8.4.4 RSNA policy selection in an IBSS

Insert the following at the end of 8.4.4 (before 8.4.4.1) as follows:

To establish a connection with a peer STA, an RSNA enabled STA that implements Management Frame Protection shall use Table 8-1b and the MFPC and MFPR values advertised in the RSN IEs exchanged in the 4-Way Handshake initiated by the Authenticator of the STA with the larger MAC address to determine if the communication is allowed. Management Frame Protection is enabled when dot11RSNAProtectedManagementFramesEnabled is set to 1. The STAs negotiate protection of management frames when the both STAs set the Management Frame Protection Capable subfield to 1 during the 4-Way Handshake.

Table 8-1b — Robust Management frame selection in an IBSS

MFPC	MFPR	Peer STA MFPC	Peer STA MFPR	STA Action
0	0	0	0	The STA may exchange data with the peer STA.
1	0	0	0	The STA may exchange data with the peer STA.
1	0 or 1	1	0 or 1	The STA may exchange data with the peer STA.
1	1	0	0	The STA shall not exchange data with the peer STA and shall reject security association attempts from the peer STA with the Reason Code "Robust Management frame policy violation."
0	0	1	1	The STA shall not exchange data with the peer STA and shall reject security association attempts from the peer STA with the Reason Code "Robust Management frame policy violation."
0	0	1	0	The STA may establish a security association with the peer STA.
1	0 or 1	0	1	The STA shall not establish a security association with the peer STA and shall reject security association attempts from the peer STA with the Status Code "Robust Management frame policy violation" because the peer STA is advertising an invalid setting. The STA shall not exchange data with the peer STA.
0	1	1	0 or 1	The peer STA shall not establish a security association with the peer STA and shall reject security association attempts from the STA with the Status Code "Robust Management frame policy violation" because the STA is advertising an invalid setting.

8.4.6 RSNA authentication in an ESS

8.4.6.1 Preauthentication and RSNA key management

Change the sixth paragraph of 8.4.6.1 as follows:

The result of preauthentication may be a PMKSA, if the IEEE 802.1X authentication completes successfully. The AKM shall be set to 00-0F-AC:1 in the PMKSA that results from preauthentication. If preauthentication produces a PMKSA, then, when the Supplicant's STA associates with the preauthenticated AP, the Supplicant can use the PMKSA with the 4-Way Handshake.

8.4.6.2 Cached PMKSAs and RSNA key management

Change the second paragraph in 8.4.6.2 as follows:

If a non-AP STA in an ESS has determined it has a valid PMKSA with an AP to which it is about to (re)associate, it includes the PMKID for the PMKSA in the RSN information element in the (Re)Association Request. Upon receipt of a (Re)Association Request with one or more PMKIDs, an AP checks whether its Authenticator has retained a PMK for the PMKIDs, whether the AKM in the cached PMKSA matches the AKM in the (Re)Association Request, and whether the PMK is still valid. If so, it shall assert~~s~~ possession of the PMK by beginning the 4-Way Handshake after association has completed; otherwise it shall begin~~s~~ a full IEEE 802.1X authentication after association has completed.

8.4.9 RSNA key management in an IBSS

Change the second through fourth paragraphs of 8.4.9 as follows:

The 4-Way Handshake is used to negotiate the pairwise cipher suites, as described in 8.4.4. The IEEE 802.11 SME configures the temporal key portion of the PTK into the IEEE 802.11 MAC. Each Authenticator uses the KCK and KEK portions of the PTK negotiated by the exchange it initiates to distribute its own GTK and if Management Frame Protection is enabled, its own IGTK. Each Authenticator generates its own GTK and if Management Frame Protection is enabled, its own IGTK, and uses either the 4-Way Handshake or the Group Key Handshake to transfer the GTK and if Management Frame Protection is negotiated, the IGTK, to other STAs with whom it has completed a 4-Way Handshake. The pairwise key used between any two STAs shall be the pairwise key from the 4-Way Handshake initiated by the STA with the highest MAC address.

A STA joining an IBSS is required to adopt the security configuration of the IBSS, which includes the group cipher suite, pairwise cipher suite, ~~and~~ AKMP, and if Management Frame Protection is enabled, Group Management Cipher Suite (see 8.4.4). The STA shall not set up a security association with any STA having a different security configuration. The Beacon and Probe Response frames of the various STAs within an IBSS must reflect a consistent security policy, as the beacon initiation rotates among the STAs.

A STA joining an IBSS shall support and advertise in the Beacon frame the security configuration of the IBSS, which includes the group cipher suite, advertised pairwise cipher suite, ~~and~~ AKMP, and if Management Frame Protection is enabled, Group Management Cipher Suite (see 8.4.4). The STA may use the Probe Request frame to discover the security policy of a STA, including additional unicast cipher suites the STA supports. A STA shall ignore Beacon frames that advertise a different security policy. If enabled, Management Frame Protection shall only be used as a required feature (MFPR) in an IBSS.

8.4.10 RSNA security association termination

Change the text in 8.4.10 as follows:

When a non-AP STA SME receives a successful ~~MLME-Association-or-Reassociation-confirm~~ MLME-ASSOCIATE.confirm or MLME-REASSOCIATE.confirm primitive that is not part of a Fast BSS Transition or receives or invokes an MLME Disassociation or Deauthentication primitive, it will delete some security associations. Similarly, when an AP SME

— receives an ~~MLME-Association-or-Reassociation-indication~~ MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive from a non-AP STA that has not negotiated Management Frame Protection, or

— receives an MLME-ASSOCIATE.indication or MLME-REASSOCIATE.indication primitive from a non-AP STA that has negotiated Management Frame Protection that a) has resulted in an MLME Association or Reassociation Response that is successful, and b) ~~that~~ is not part of a Fast BSS Transition, or

— receives or invokes an MLME Disassociation or Deauthentication primitive,

it will delete some security associations. In the case of an ESS the non-AP STA's SME shall delete the PTKSA, GTKSA, IGTKSA, SMKSA, and any STKSA, and the AP's SME shall delete the PTKSA, and invoke an STSL application teardown procedure for any of its STKSAs. An example of an STSL application teardown procedure is described in 11.7.3. In the case of an IBSS, the STA's SME shall delete the PTKSA and the receive GTKSA and IGTKSA. Once the security associations have been deleted, the SME then invokes MLME-DELETEKEYS.request primitive to delete all temporal keys associated with the deleted security associations.The IEEE 802.1X Controlled Port returns to being blocked. As a result, all data frames are unauthorized before invocation of an MLME-DELETEKEYS.request primitive.

Insert the following two new subclauses (8.4.11 and 8.4.12) after 8.4.10 as follows:

8.4.11 Protection of Robust Management frames

This subclause defines rules that shall be followed by STAs that implement Management Frame protection and have dot11RSNAEnable set to TRUE.

A STA with dot11RSNAProtectedManagementFramesEnabled set to FALSE shall transmit and receive unprotected unicast Robust Management frames to and from any associated STA, and shall discard protected unicast Robust Management frames received from any associated STA.

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE and dot11RSNAUnprotectedManagementFramesAllowed set to TRUE shall transmit and receive unprotected unicast Robust Management frames to and from any associated STA that advertised MFPC = 0, and shall discard protected unicast Robust Management frames received from any associated STA that advertised MFPC = 0.

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE and dot11RSNAUnprotectedManagementFramesAllowed set to TRUE shall transmit and receive protected unicast Robust Management frames to and from any associated STA that advertised MFPC = 1, and shall discard unprotected unicast Robust Action frames received from any STA that advertised MFPC = 1, and it shall discard received unprotected unicast Disassociation and Deauthentication frames from a STA that advertised MFPC = 1 after the PTK and IGTK have been installed. The receiver shall process unprotected unicast Disassociation and Deauthentication frames before the PTK and IGTK are installed.

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE and dot11RSNAUnprotectedManagementFramesAllowed set to FALSE shall transmit and receive protected unicast Robust Action frames to and from any STA, shall not transmit unprotected unicast Robust Action frames to any STA, and shall discard unprotected unicast Robust Action frames received from a STA after the PTK and IGTK have been installed. The receiver shall process unprotected unicast Disassociation and Deauthentication frames before the PTK and IGTK are installed.

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE shall protect transmitted group addressed Robust Management frames using the Group Management Cipher suite.

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE shall discard group addressed Robust Management frames received from any associated STA that advertised MFPC = 1 if the frames are unprotected or if a matching IGTK is not available.

A STA with dot11RSNAUnprotectedManagementFramesAllowed set to FALSE shall discard received group addressed Robust Management frames that are unprotected or for which a matching IGTK is not available.

A STA with dot11RSNAProtectedManagementFramesEnabled set to FALSE shall transmit group addressed Robust Management frames unprotected, and shall ignore the protection on received group addressed Robust Management frames.

NOTE—BIP does not provide protection against forgery by associated and authenticated non-AP STAs. A STA that has left the group can successfully forge management frames until the IGTK is updated.

Protection of group addressed Robust Management frames shall be provided by a service in the MLME as described in 11.12.

Robust Management frame protection cannot be applied until the PTK and IGTK has been established with the STA. A STA shall not transmit Robust Action frames until it has installed the PTK for the peer STA, or in the case of group addressed frames, has installed the IGTK. The STA shall discard any Robust Action frames received before the PTK and IGTK are installed. Action frames with "No" in the "Robust" column in Table 7-24 shall not be protected.

8.4.12 Robust Management frame Selection Procedure

A STA with dot11RSNAProtectedManagementFramesEnabled set to TRUE shall negotiate Robust Management frame protection with a STA that advertised MFPC = 1.

8.5 Keys and key distribution

8.5.1 Key hierarchy

Change the first paragraph in 8.5.1 as follows:

RSNA defines ~~two~~ the following key hierarchies:

a) Pairwise key hierarchy, to protect unicast traffic

b) GTK, a hierarchy consisting of a single key to protect multicast and broadcast traffic

NOTE—Pairwise key support with TKIP or CCMP allows a receiving STA to detect MAC address spoofing and data forgery. The RSNA architecture binds the transmit and receive addresses to the pairwise key. If an attacker creates an MPDU with the spoofed TA, then the decapsulation procedure at the receiver will generate an error. GTKs do not have this property.

c) Integrity GTK (IGTK), a hierarchy consisting of a single key to provide integrity protection for broadcast and multicast Robust Management frames

8.5.1.1 PRF

Insert the following paragraph at the end of 8.5.1.1:

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, the KDF specified in 8.5.1.5.2 shall be used instead of the PRF construction defined here.

8.5.1.2 Pairwise key hierarchy

Change the first paragraph of 8.5.1.2 as follows:

Except when pre-authentication is used, the~~The~~ pairwise key hierarchy utilizes PRF-384 or PRF-512 to derive session-specific keys from a PMK, as depicted in Figure 8-20. The PMK shall be 256 bits. The pairwise key hierarchy takes a PMK and generates a PTK. The PTK is partitioned into KCK, KEK, and temporal keys, which are used by the MAC to protect unicast communication between the Authenticator's and Supplicant's respective STAs. PTKs are used between a single Supplicant and a single Authenticator.

Insert the following at the end of 8.5.1.2:

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, HMAC-SHA-256 is used to calculate the PMKID, and the PMK identifier is defined as

PMKID = Truncate-128(HMAC-SHA-256(PMK, "PMK Name" || AA || SPA))

NOTE—When the PMKID is calculated for the PMKSA as part of RSN preauthentication, the AKM has not yet been negotiated. In this case, the HMAC-SHA1-128 based derivation is used for the PMKID calculation.

8.5.1.3 Group key hierarchy

Change the first paragraph of 8.5.1.3 as follows:

The GTK shall be a random number. The following is an example method for deriving a random GTK. Any other pseudo-random function, such as that specified in 8.5.1.1, could also be used.

Insert a new subclause (8.5.1.3a) after 8.5.1.3 as follows:

8.5.1.3a Integrity group key hierarchy

The Authenticator shall select the IGTK as a random value each time it is generated.

The Authenticator may update the IGTK for any reason, including:

a) The disassociation or deauthentication of a STA.
b) An event within the STA's SME that triggers a Group Key Handshake.

The IGTK is configured via the MLME-SETKEYS.request primitive, see 10.3.17. IGTK configuration is described in the EAPOL-Key state machines, see 8.5.5 and 8.5.6.

The IPN is used to provide replay protection.

8.5.1.4 PeerKey key hierarchy

Insert the following two new paragraphs at the end of 8.5.1.4:

When the negotiated AKM is 00-0F-AC:5 or 00-0F-AC:6, HMAC-SHA-256 is used to calculate the SMKID, and an SMK identifier is defined as

SMKID = Truncate-128(HMAC-SHA-256(SMK, "SMK Name" || PNonce || MAC_P || INonce || MAC_I))

8.5.2 EAPOL-Key frames

Change item iii) in Key Information field list item b1) as follows:

iii) The value 3 shall be used for all EAPOL-Key frames to and from a STA when the negotiated AKM is 00-0F-AC:3, ~~or~~ 00-0F-AC:4, 00-0F-AC:5, or 00-0F-AC:6. This value indicates the following:

—AES-128-CMAC is the EAPOL-Key MIC. AES-128-CMAC is defined by FIPS SP800-38B.[3] The output of the AES-128-CMAC shall be 128 bits.

—The NIST AES key wrap is the EAPOL-Key encryption algorithm used to protect the Key Data field. IETF RFC 3394 defines the NIST AES key wrap algorithm.

Replace Table 8-2 with the following table:

Table 8-2—Cipher suite key lengths

Cipher suite	CCMP	TKIP	WEP-40	WEP-104	BIP
Key length (octets)	16	32	5	13	16

Insert Data Type 9 into Table 8-4 and update the reserved row as follows:

Table 8-4—KDE

OUI	Data Type	Meaning
00-0F-AC	9	IGTK KDE
00-0F-AC	~~9~~10–255	Reserved

Insert the following text and Figure 8-32a before Table 8-6 (before the paragraph starting "The following EAPOL-Key frames are used to implement the three different exchanges"):

The format of the IGTK KDE is shown in Figure 8-32a. The IPN corresponds to the last packet number used by the broadcast/multicast transmitter, and is used by the receiver as the initial value for the BIP replay counter.

KeyID	IPN	IGTK
2 octets	**6 octets**	**(Length – 12) octets**

Figure 8-32a—IGTK KDE format

8.5.2.1 EAPOL-Key frame notation

Insert the following text before the notation for PMKID:

IGTK[M] is the IGTK, with key identifier field set to M.

IPN is the current IGTK replay counter value provided by the IGTK KDE

8.5.3 4-Way Handshake

8.5.3.2 4-Way Handshake Message 2

Insert a new item c) in the fourth paragraph lettered list in 8.5.3.2 as follows:

c) If Management Frame Protection is being negotiated, the AP initializes the SA Query Transaction-Identifier to an implementation specific non-negative integer value, valid for the current pairwise security association.

8.5.3.3 4-Way Handshake Message 3

Change the entry for Key Data in 8.5.3.3 as follows:

Key Data = For PTK generation, the AP's Beacon/Probe Response frame's RSN information element, and, optionally, a second RSN information element that is the Authenticator's pairwise cipher suite assignment, and, if a group cipher has been negotiated, the encapsulated GTK and the GTK's key identifier (see 8.5.2), and if Management Frame Protection is negotiated, the IGTK KDE. For STK generation Initiator RSN IE, Lifetime of SMK is used.

8.5.3.6 Sample 4-Way Handshake

Replace Figure 8-33 with the following figure:

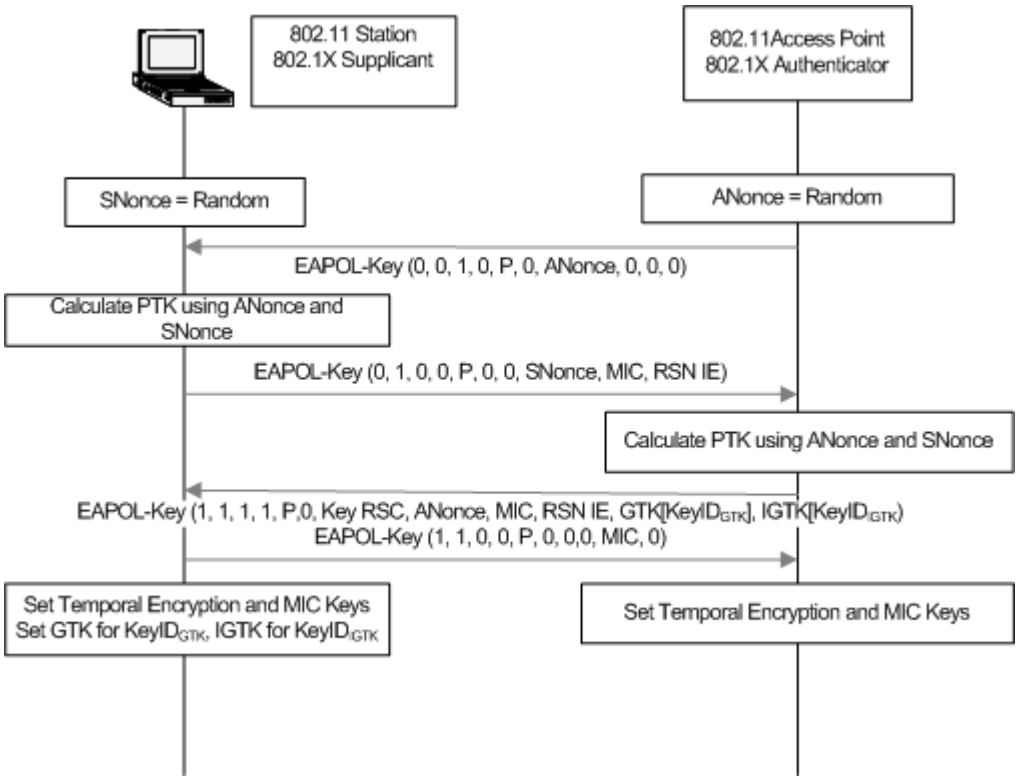

Figure 8-33—Sample 4-Way Handshake

Change list item e) in 8.5.3.6 as follows:

e) The Authenticator sends an EAPOL-Key frame containing ANonce, the RSN information element from its Beacon or Probe Response messages, MIC, whether to install the temporal keys, ~~and~~ the encapsulated GTK, and if Management Frame Protection is negotiated, the IGTK.

8.5.4 Group Key Handshake

Change the first three paragraphs in 8.5.4, including the dashed list, as follows:

The Authenticator uses the Group Key Handshake to send a new GTK, and, if Management Frame Protection is negotiated, a new IGTK to the Supplicant.

The Authenticator may initiate the exchange when a Supplicant is disassociated or deauthenticated.

Message 1: Authenticator → Supplicant: EAPOL-Key(1,1,1,0,G,0,Key RSC,0, MIC, GTK[N], IGTK[M])

Message 2: Supplicant → Authenticator: EAPOL-Key(1,1,0,0,G,0,0,MIC,0)

Here, the following assumptions apply:

— Key RSC denotes the last frame sequence number sent using the GTK.

— GTK[N] denotes the GTK encapsulated with its key identifier as defined in 8.5.2 using the KEK defined in 8.5.1.2 and associated IV.

— IGTK[M], when present, denotes the IGTK encapsulated with its key identifier as defined in 8.5.2 using the KEK defined in 8.5.1.2 and associated IV.

— The MIC is computed over the body of the EAPOL-Key frame (with the MIC field zeroed for the computation) using the KCK defined in 8.5.1.2.

8.5.4.1 Group Key Handshake Message 1

Change the description for Key Data in 8.5.4.1 as follows:

Key Data = encrypted, encapsulated

—GTK and the GTK's key identifier (see 8.5.2)

—When present, IGTK, IGTK's key identifier, and IPN (see 8.5.2)

Change list item c) in 8.5.4.1 as follows:

c) Uses the MLME-SETKEYS.request primitive to configure the temporal GTK and, when present, IGTK into its IEEE 802.11 MAC.

8.5.4.4 Sample Group Key Handshake

Change the second paragraph in 8.5.4.4 as follows:

The state machines in 8.5.5 and 8.5.6 change the GTK and, when present, IGTK in use by the network. See Figure 8-34**.**

Change the last paragraph in 8.5.4.4 as follows:

The following steps occur:

a) The Authenticator generates a new GTK and when Management Frame Protection has been negotiated, a new IGTK. It encapsulates the GTK and, as necessary, the IGTK, and sends an EAPOL-Key frame containing the GTK and IGTK (Message 1), along with the last sequence number used with the GTK (RSC) and the last IPN used with the IGTK.

b) On receiving the EAPOL-Key frame, the Supplicant validates the MIC, decapsulates the GTK, and, when present, the IGTK, and uses the MLME-SETKEYS.request primitive to configure the GTK, PN, IGTK, RSC, and IPN in its STA.

c) The Supplicant then constructs and sends an EAPOL-Key frame in acknowledgment to the Authenticator.

d) On receiving the EAPOL-Key frame, the Authenticator validates the MIC. If the GTK, and, if present, the IGTK ~~is~~ are not already configured into IEEE 802.11 MAC, after the Authenticator has delivered the GTK and IGTK to all associated STAs, it uses the MLME-SETKEYS.request primitive to configure the GTK and IGTK into the IEEE 802.11 STA.

Replace Figure 8-34 with the following figure:

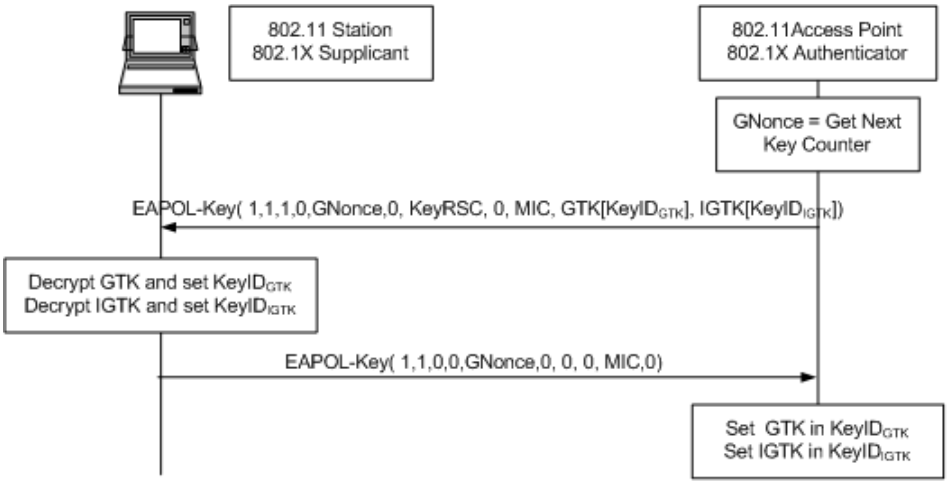

Figure 8-34—Sample Group Key Handshake

8.5.5 RSNA Supplicant key management state machine

Replace Figure 8-35 with the following figure:

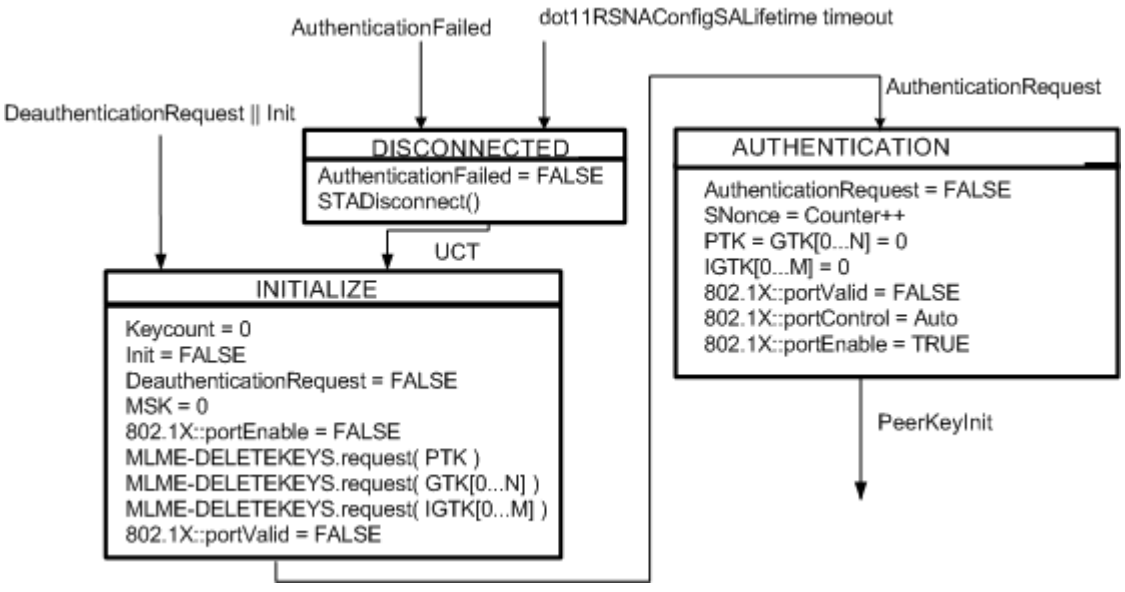

Figure 8-35—RSNA Supplicant key management state machine

8.5.5.2 Supplicant state machine variables

Insert the following dashed list item immediately following the GTK[] variable:

— IGTK[]—This variable represents the current IGTKs for each group management key index.

8.5.5.3 Supplicant state machine procedures

Change the StaProcessEAPOL-Key item in 8.5.5.3 as follows:

— **StaProcessEAPOL-Key** - The Supplicant invokes this procedure to process a received EAPOL-Key frame. The pseudo-code for this procedure is as follows:

StaProcessEAPOL-Key (*S, M, A, I, K*, *RSC*, *ANonce*, RSC, *MIC*, RSNIE, *GTK[N]*, *IGTK[M], IPN*)
TPTK ← PTK
TSNonce ← 0
PRSC ← 0
UpdatePTK ← 0
State ← UNKNOWN
if M = 1 **then**
  if Check MIC(*PTK, EAPOL-Key frame*) fails **then**
    State ← FAILED
  else
    State ← MICOK
  endif
endif
if *K* = *P* **then**
  if *State* ≠ FAILED **then**
    if *PSK* exists **then**—PSK is a preshared key
      PMK ← *PSK*
    else
      PMK ← L(MSK, 0, 256)
    endif
    TSNonce ← *SNonce*
    if *ANonce* ≠ *PreANonce* **then**
      TPTK ← Calc PTK(PMK, *ANonce, TSNonce*)
      PreANonce ← *ANonce*
    endif
    if State = MICOK **then**
      PTK ← *TPTK*
      UpdatePTK ← *I*
      if *UpdatePTK* = 1 **then**
        if no *GTK* **then**
          PRSC ← RSC
        endif


```
                    if MLME-SETKEYS.request(0, TRUE, PRSC, PTK) fails then
                        invoke MLME-DEAUTHENTICATE.request
                    endif
                    MLME.SETPROTECTION.request(TA, Rx)
                endif
                if GTK then
                    if (GTK[N] ← Decrypt GTK) succeeds then
                        if MLME-SETKEYS.request(N, 0, RSC, GTK[N]) fails then
                            invoke MLME-DEAUTHENTICATE.request
                        endif
                    else
                        State ← FAILED
                    endif
                endif
                if IGTK then
                    if (IGTK[M] ← Decrypt IGTK) succeeds then
                        if MLME-SETKEYS.request(M, 0, IPN, IGTK[M]) fails then
                            invoke MLME-DEAUTHENTICATE.request
                        endif
                    else
                        State ← FAILED
                    endif
                endif
            endif
        else if KeyData = GTK then
            if State = MICOK then
                if (GTK[N] ← Decrypt GTK) succeeds then
                    if MLME-SETKEYS.request(N, T, RSC, GTK[N]) fails then
                        invoke MLME-DEAUTHENTICATE request
                    endif
                    else
                        State ← FAILED
                    endif
                    if (IGTK[M] ← Decrypt IGTK) succeeds then
                        if MLME-SETKEYS.request(M, T, IPN, IGTK[M]) fails then
                            invoke MLME-DEAUTHENTICATE request
                        endif
                        else
                            State ← FAILED
                        endif
                    else
                        State ← FAILED
                    endif
```

```
            endif
            if A = 1 && State ≠ Failed then
                Send EAPOL-Key(0,1,0,0,K,0,0,TSNonce,MIC(TPTK),RSNIE)
            endif
            if UpdatePTK = 1 then
                MLME-SETPROTECTION.request(TA, Tx_Rx)
            endif
            if State = MICOK && S = 1 then
                MLME-SETPROTECTION.request(TA, Tx_Rx)
                if IBSS then
                    keycount++
                    if keycount = 2 then
                    802.1X::portValid ← TRUE
                    endif
                else
                    802.1X::portValid ← TRUE
                endif
            endif
        endif
```

Change the second paragraph following the pseudo-code as follows:

When processing 4-Way Handshake Message 3, the GTK and IGTK are ~~is~~ decrypted from the EAPOL-Key frame and installed. The PTK shall be installed before the GTK and IGTK.

Insert the following dashed list item at the end of 8.5.5.3:

— **DecryptIGTK(x)**—Decrypt the IGTK from the EAPOL-Key frame.

8.5.6 RSNA Authenticator key management state machine

Replace Figure 8-37 with the following figure:

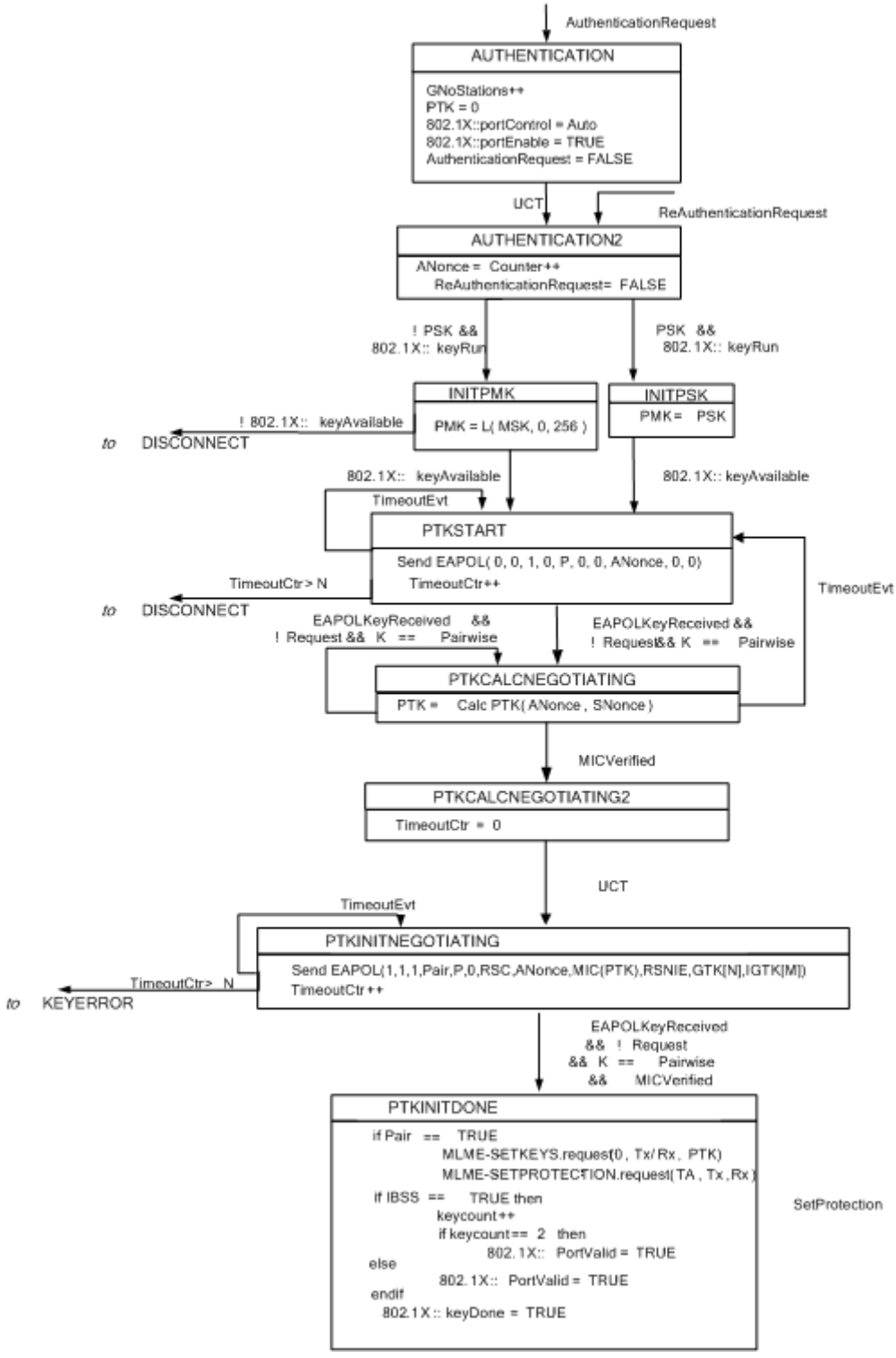

Figure 8-37—Authenticator state machines, part 1

Replace Figure 8-40 with the following figure:

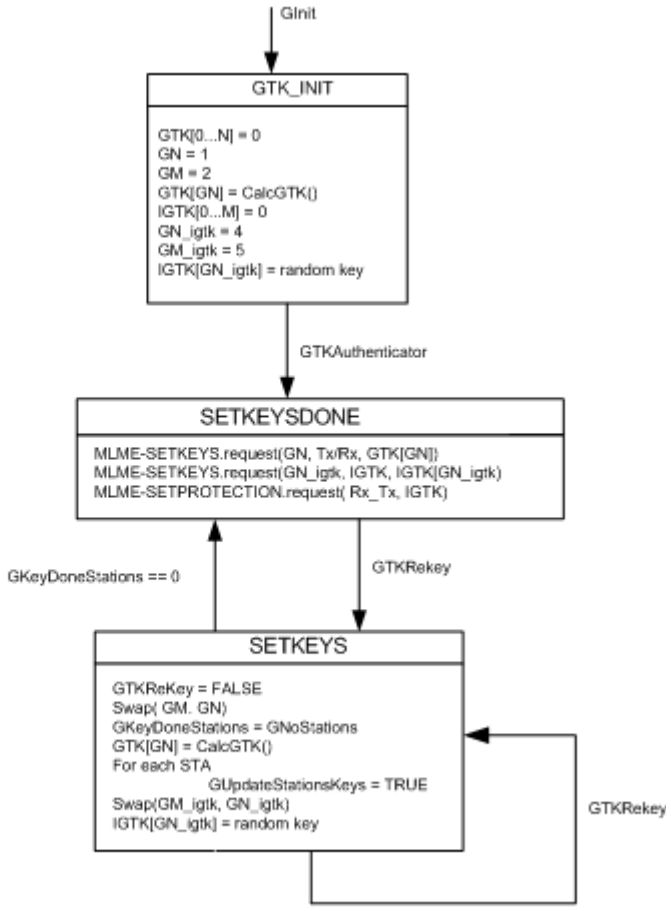

Figure 8-40—Authenticator state machines, part 4

8.6 Mapping EAPOL keys to IEEE 802.11 keys

8.6.3 Mapping PTK to CCMP keys

Change the second paragraph in 8.6.3 as follows:

A STA shall use the temporal key as the CCMP key for ~~MSDUs~~ MPDUs between the two communicating STAs.

Insert a new subclause (8.6.6a) after 8.6.6 as follows:

8.6.6a Mapping IGTK to BIP Keys

See 8.5.1.3a for the definition of the IGTK key. A STA shall use bits 0–127 of the IGTK as the AES-128-CMAC key.

8.7 Per-frame pseudo-code

8.7.2 RSNA frame pseudo-code

Change the text in 8.7.2 as follows:

STAs transmit protected MSDUs and Robust Management frames to an RA when temporal keys are configured and an MLME-SETPROTECTION.request primitive has been invoked ~~for transmit~~ with ProtectType parameter Tx or Rx_Tx to that RA. STAs expect to receive protected MSDUs and Robust Management frames from a TA when temporal keys are configured and an MLME-SETPROTECTION.request primitive has been invoked ~~for receive~~ with ProtectType parameter Rx or Rx_Tx from that TA. MSDUs and Robust Management frames that do not match these conditions are sent in the clear and are received in the clear.

8.7.2.1 Per-MSDU Tx pseudo-code

Insert a new subclause after 8.7.2.1 as follows:

8.7.2.1a Per-MMPDU Tx pseudo-code

```
if ((dot11RSNAEnabled = TRUE) and (frame is a Robust Management frame)) then
    if ((dot11RSNAProtectedManagementFramesEnabled = FALSE) then
        Transmit the MMPDU without protection
    else // dot11RSNAProtectedManagementFramesEnabled = TRUE
        if (dot11RSNAUnprotectedManagementFramesAllowed = TRUE) then
            if (MMPDU has an individual RA) then
                if (peer STA advertised MFPC = 1) then
                    if (Pairwise key exists for the MMPDU's RA) then
                        // Note that it is assumed that no entry in the key
                        // mapping table will be of an unsupported cipher.
                        Transmit the MMPDU, to be protected after fragmentation
                        // see 8.7.2.2a
                    else if (Robust Action frame) then
                        // pairwise key was not found
                        Discard the MMPDU and generate an MLME.confirm primitive to
                        notify the SME that the MMPDU was not delivered
                    else // Disassociation or Deauthentication
                        Transmit the MMPDU without protection
                    endif
                else // (peer STA didn't advertised MFPC = 1)
                    Transmit the MMPDU without protection
                endif
            else // MMPDU has a multicast/broadcast RA
                if (IGTK exists) then
                    // if we find a suitable IGTK
                    Transmit the MMPDU with protection // See 8.7.2.2a
                else if (MMPDU is Disassociate ||Deauthenticate ||(not a Robust Action frame))
                then
```

```
                    Transmit the MMPDU without protection
                else
                    Discard the MMPDU and generate an MLME.confirm primitive to notify
                    the SME that the MMPDU was undeliverable
                endif
            endif
        else // dot11RSNAUnprotectedManagementFramesAllowed = FALSE
            if (MMPDU has an individual RA) then
                if (peer STA advertised MFPC = 1) then
                    if (Pairwise key exists for the MMPDU's RA) then
                        // Note that it is assumed that no entry in the key
                        // mapping table will be of an unsupported cipher.
                        Transmit the MMPDU, to be protected after fragmentation
                        // see 8.7.2.2a
                    else if (Robust Action frame) then
                        // pairwise key was not found
                        Discard the MMPDU and generate an MLME.confirm primitive to
                        notify the SME that the MMPDU was not delivered
                    else // FrameControlSubType is Disassociation or Deauthentication
                        Transmit the MMPDU without protection
                    endif
                else // peer STA didn't advertise MFPC = 1
                    Discard the MMPDU and generate an MLME.confirm primitive to notify
                    the SME that the MMPDU was not delivered
                endif
            else // MMPDU has a multicast/broadcast RA
                if (IGTK exists) then
                    // if we find a suitable IGTK
                    Transmit the MMPDU with protection // See 8.7.2.2a
                else if (MMPDU is Disassociate || Deauthenticate || (not a Robust Action
                frame)) then
                    Transmit the MMPDU without protection
                else
                    Discard the MMPDU and generate an MLME.confirm primitive to notify
                    the SME that the MMPDU was undeliverable
                endif
            endif
        endif
    endif
else // (dot11RSNAEnabled = FALSE) or (not a Robust Management Frame)
    Use 8.7.2.1 to transmit the frame
endif
```


Insert a new subclause (8.7.2.2a) after 8.7.2.2:

8.7.2.2a Per-MPDU Tx pseudo-code for MMPDU

```
if ((dot11RSNAEnabled = TRUE) then
    if (MPDU is member of an MMPDU that is to be transmitted without protection) then
        Transmit the MPDU without protection
     else if (MPDU has an individual RA) then
        Protect the MPDU using entry's TK and selected cipher from RSN IE
        Transmit the MPDU
    else
        // MPDU has a multicast/broadcast RA
        Protect the MPDU using IGTK and BIP
        Transmit the MPDU
     endif
endif
```

Insert a new subclause (8.7.2.3a) after 8.7.2.3:

8.7.2.3a Per-MPDU Rx pseudo-code for an MMPDU

```
if ((dot11RSNAEnabled = TRUE) and (frame is a Robust Management frame)) then
    if ((dot11RSNAProtectedManagementFramesEnabled = FALSE) then
        if (Protected Frame subfield of the Frame Control field is set to 1) then
            Discard the frame
        else
            Receive the MMPDU
        endif
    else // dot11RSNAProtectedManagementFramesEnabled = TRUE
        if (dot11RSNAUnprotectedManagementFramesAllowed = TRUE) then
            if (STA with frame TA advertised MFPC = 0) then
                if (Protected Frame subfield of the Frame Control field is set to 1) then
                    Discard the frame
                else
                    Make frame available for further processing
                endif
            else // STA with frame TA advertised MFPC = 1
                if (MMPDU has an individual RA) then
                    if (Pairwise key does not exist) then
                      if (frame is a Disassociation or Deauthentication) then
                        if (Protected Frame subfield of the Frame Control field is set to 0) then
                            Make the MPDU available for further processing
                        else // encrypted
                            Discard the frame
```

```
            endif
          else // frame is not a Disassociation or Deauthenticate
            Discard the frame
          endif
        else if (security association has an AES-CCM key) then
          if (Protected Frame subfield of the Frame Control field is set to 0) then
            //unprotected frame
            Discard the frame
          else // frame is encrypted
            if (PN is not sequential) then
                Discard the MPDU as a replay
                Increment dot11RSNAStatsCCMPReplays
            else
                Decrypt frame using AES-CCM key
                if (the integrity check fails) then
                    Discard the frame
                    Increment dot11RSNAStatsCCMPDecryptErrors
                else
                    Make the MPDU available for further processing
                endif
            endif
          endif
        else // key for some other cipher—for future expansion
        endif
      else // MMPDU has a multicast/broadcast RA
        if (IGTK does not exist) then
          if (Disassociation or Deauthentication) then
            Make frame available for further processing
          else
            Discard the frame
          endif
        else // IGTK exists
          if (MMIE is not present) then
            Discard the frame
          else // MMIE is present
            if (AES-128-CMAC IGTK) then
                if (IPN is not valid) then
                    Discard the frame as a replay
                    Increment dot11RSNAStatsCMACReplay
                else if (integrity check fails) then
                    Discard the frame
                    Increment dot11RSNAStatsCMACICVError
                else
```



```
                        Make frame available for further processing
                    endif
                else // some other kind of key—for the future
                endif
            endif
        endif
    endif
else // dot11RSNAUnprotectedManagementFramesAllowed = FALSE
    if (MMPDU has an individual RA) then
        if (peer STA advertised MFPC = 1) then
            if (Pairwise key exists for the MMPDU's RA) then
                if (security association has an AES-CCM key) then
                    if (Protected Frame subfield of the Frame Control field is set to 0)
                then
                        if (frame is a Disassociation or Deauthentication) then
                            Make the MPDU available for further processing
                        else // encrypted
                            Discard the frame
                        endif
                    else // frame is encrypted
                        if (PN is not sequential) then
                            Discard the MPDU as a replay
                            Increment dot11RSNAStatsCCMPReplays
                        else
                            Decrypt frame using AES-CCM key
                            if (the integrity check fails) then
                                Discard the frame
                                Increment dot11RSNAStatsCCMPDecryptErrors
                            else
                                Make the MPDU available for further processing
                            endif
                        endif
                    endif
                else // key for some other cipher—for future expansion
                endif
            else if (Robust Action frame) then
                Discard the frame
            else if (Deauthenticate || Disassociate || (not a Robust Action frame))
            then
                Make frame available for processing
            else
                Discard the frame
```

```
                    endif
                  else // peer STA didn't advertise MFPC = 1
                    Discard the frame
                  endif
                else // MMPDU has a multicast/broadcast RA
                  if (IGTK exists) then
                    if (MMIE is not present) then
                      Discard the frame
                    else // MMIE is present
                      if (AES-128-CMAC IGTK) then
                        if (PN is not valid) then
                          Discard the frame as a replay
                          Increment dot11RSNAStatsCMACReplay
                        else if (security association has an AES-128-CMAC IGTK) then
                          Discard the frame
                          Increment dot11RSNAStatsCMACICVError
                        else
                          Make frame available for further processing
                        endif
                      else // some other kind of key—for the future
                      endif
                    endif
                  else // IGTK does not exist
                    if (Disassociation or Deauthentication) then
                      Make frame available for further processing
                    else
                      Discard the frame
                    endif
                  endif
                endif
              endif
            endif
          else // (dot11RSNAEnabled = FALSE) or (not a Robust Management Frame)
            Use 8.7.2.3 to receive the frame
          endif
```


Insert a new subclause (8.7.2.5) after 8.7.2.4:

8.7.2.5 Per-MMPDU Rx pseudo-code

if (*dot11RSNAEnabled* = TRUE) **then**

if (*dot11RSNAProtectedManagmentFramesEnabled* = TRUE) **then**

if (the MPDU was not protected) **then**

Receive the MMPDU unprotected

Make the MMPDU available to higher layers

else //Have a protected MMPDU

if ((MMPDU has individual RA) **and** (security association has an AES-CCM key)) **then**

if (the MPDU has only one MPDU or multiple MPDUs with sequential PNs) **then**

Receive the MMPDU protected

Make the MMPDU available to higher layers

else

Discard the MMPDU as a replay

Increment *dot11RSNAStatsRobustMgmtCCMPReplays*

endif

else if ((MPDU has group addressed RA) **and** (security association has an AES-128-CMAC IGTK)) **then**

Receive the MMPDU

Make the MMPDU available to higher layers

else

if (any other cipher exists) **then**

Process the frame using other cipher

else

Discard the frame

endif

endif

endif

endif

endif

10. Layer Management

10.3 MLME SAP interface

10.3.17 SetKeys

10.3.17.1 MLME-SETKEYS.request

10.3.17.1.2 Semantics of the service primitive

Change the KeyID and Key Type entries in the SetKeyDescriptor of 10.3.17.1.2 as follows:

Name	Type	Valid range	Description
Key ID	Integer	0–3 shall be used with WEP, TKIP, and CCMP, 4–5 with BIP, and 6–4095 are reserved	Key identifier
Key Type	Integer	Group, Pairwise, Peerkey, IGTK	Defines whether this key is a group key, pairwise key, ~~or~~ PeerKey, or Integrity Group key.

10.3.18 DeleteKeys

10.3.18.1 MLME-DELETEKEYS.request

10.3.18.1.2 Semantics of the service primitive

Change the Key Type entry in the DeleteKeyDescriptor of 10.3.18.1.2 as follows:

Name	Type	Valid range	Description
Key Type	Integer	Group, Pairwise, Peerkey, IGTK	Defines whether this key is a group key, pairwise key, ~~or~~ PeerKey, or Integrity Group key.

10.3.22 SetProtection

10.3.22.1 MLME-SETPROTECTION.request

10.3.22.1.2 Semantics of the service primitive

Change the Key Type entry in the Protectlist of 10.3.22.1.2 as follows:

Name	Type	Valid range	Description
Key Type	Integer	Group, Pairwise, ~~or~~ Peerkey, IGTK	Defines whether this key is a group key, pairwise key, ~~or~~ PeerKey, or Integrity Group key.

Insert the following new subclauses (10.3.39 through 10.3.44) at the end of 10.3.38 as follows:

10.3.39 SA Query support

10.3.39.1 MLME-SAQuery.request

10.3.39.1.1 Function

This primitive requests that a SA Query Request frame be sent to a specified peer STA to which the STA is associated.

10.3.39.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.request(
                        PeerSTAAddress,
                        TransactionIdentifier
                    )
```

Name	Type	Valid Range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 7.4.9.1	The Transaction Identifier to identify the SA Query Request and Response transaction

10.3.39.1.3 When generated

This primitive is generated by the SME to request that a SA Query Request frame be sent to a specified peer STA with which the STA is associated.

10.3.39.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a SA Query Request frame. The STA then attempts to transmit this to the peer STA with which it is associated.

10.3.39.2 MLME-SAQuery.confirm

10.3.39.2.1 Function

This primitive reports the result of a SA Query procedure.

10.3.39.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SAQuery.confirm(
PeerSTAAddress,
TransactionIdentifier
)

Name	Type	Valid Range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 7.4.9.1	The Transaction Identifier to identify the SA Query Request and Response transaction

10.3.39.2.3 When generated

This primitive is generated by the MLME as a result of the receipt of a valid SA Query Response frame.

10.3.39.2.4 Effect of receipt

On receipt of this primitive, the SME may use the response as a sign of liveness of the peer STA.

10.3.39.3 MLME-SAQuery.indication

10.3.39.3.1 Function

This primitive indicates that a SA Query Request frame was received from a STA.

10.3.39.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-SAQuery.indication(
PeerSTAAddress,
TransactionIdentifier
)

Name	Type	Valid range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 7.4.9.1	The Transaction Identifier to identify the SA Query Request and Response transaction

10.3.39.3.3 When generated

This primitive is generated by the MLME when a valid SA Query Request frame is received.

10.3.39.3.4 Effect of receipt

On receipt of this primitive the SME operates according to the procedure in 11.3.

10.3.39.4 MLME-SAQuery.response

10.3.39.4.1 Function

This primitive is generated in response to an MLME-SAQuery.indication requesting a SA Query Response frame be sent to a STA.

10.3.39.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-SAQuery.response(
                        PeerSTAAddress,
                        TransactionIdentifier
                      )
```

Name	Type	Valid Range	Description
PeerSTA Address	MAC Address	Any valid individual MAC Address	Specifies the address of the peer MAC entity for the SA Query
TransactionIdentifier	2 octets	As defined in 7.4.9.1	The Transaction Identifier to identify the SA Query Request and Response transaction

10.3.39.4.3 When generated

This primitive is generated by the SME, in response to an MLME-SAQuery.indication, requesting a SA Query Response frame be sent to a STA.

10.3.39.4.4 Effect of receipt

On receipt of this primitive, the MLME constructs a SA Query Response frame. The STA then attempts to transmit this to the STA indicated by the PeerSTAAddress parameter.

10.3.40 Protected Extended Channel Switch Announcement

The following MLME primitives support the signaling of Protected Extended Channel Switch Announcement.

10.3.40.1 MLME-PDEXTCHANNELSWITCH.request

10.3.40.1.1 Function

This primitive requests that a Protected Extended Channel Switch Announcement frame be sent by an AP.

10.3.40.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDEXTCHANNELSWITCH.request (
Mode,
RegulatoryClass,
ChannelNumber,
ChannelSwitchCount,
VendorSpecificInfo
)

Name	Type	Valid range	Description
Mode	Integer	0,1	Channel switch mode, as defined for the Extended Channel Switch Announcement element.
RegulatoryClass	Integer	As defined in Annex J	Specifies the new regulatory class.
ChannelNumber	Integer	As defined in Annex J	Specifies the new channel number.
ChannelSwitch-Count	Integer	0–255	Specifies the number of TBTTs until the channel switch event, as described for the Extended Channel Switch Announcement element.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.40.1.3 When generated

This primitive is generated by the SME to request that a Protected Extended Channel Switch Announcement frame be sent to a non-AP STA that is associated to the AP.

10.3.40.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs and transmits a Protected Extended Channel Switch Announcement frame.

10.3.40.2 MLME-PDEXTCHANNELSWITCH.confirm

10.3.40.2.1 Function

This primitive reports the result of a request to switch channel.

10.3.40.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDEXTCHANNELSWITCH.confirm (
ResultCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID PARAMETERS or UNSPECIFIED FAILURE	Reports the result of an extended channel switch request.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.40.2.3 When generated

This primitive is generated by the MLME when a protected extended channel switch request completes. Possible unspecified failure causes include an inability to schedule an extended channel switch announcement.

10.3.40.2.4 Effect of receipt

The SME is notified of the results of the extended channel switch procedure.

10.3.40.3 MLME-PDEXTCHANNELSWITCH.indication

10.3.40.3.1 Function

This primitive indicates that a Protected Extended Channel Switch Announcement frame was received from an AP.

10.3.40.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDEXTCHANNELSWITCH.indication (
Peer MAC Address,
Mode,
RegulatoryClass,
ChannelNumber,
ChannelSwitchCount,
VendorSpecificInfo
)

Name	Type	Valid range	Description
PeerMAC Address	MACAddress	Any valid individual MAC Address	The address of the peer MAC entity from which the Extended Channel Switch Announcement frame was received.
Mode	Integer	0,1	Channel switch mode, as defined for the Channel Switch Announcement element.
RegulatoryClass	Integer	As defined in Annex J	Specifies the new regulatory class.
ChannelNumber	Integer	As defined in Annex J	Specifies the new channel number.
ChannelSwitch-Count	Integer	0–255	Specifies the number of TBTTs until the channel switch event, as described for the Extended Channel Switch Announcement element.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.40.3.3 When generated

This primitive is generated by the MLME when a valid Protected Extended Channel Switch Announcement frame is received.

10.3.40.3.4 Effect of receipt

On receipt of this primitive, the SME decides whether to accept the switch request.

10.3.40.4 MLME-PDEXTCHANNELSWITCH.response

10.3.40.4.1 Function

This primitive is used to schedule an accepted extended channel switch.

10.3.40.4.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-PDEXTCHANNELSWITCH.response     (
                                     Mode,
                                     RegulatoryClass,
                                     ChannelNumber,
                                     ChannelSwitchCount,
                                     VendorSpecificInfo
                                     )
```


Name	Type	Valid range	Description
Mode	Integer	0,1	Channel switch mode, as defined for the Channel Switch Announcement element.
RegulatoryClass	Integer	As defined in Annex J	Specifies the new regulatory class.
ChannelNumber	Integer	As defined in Annex J	Specifies the new channel number.
ChannelSwitch-Count	Integer	0–255	Specifies the number of TBTTs until the channel switch event, as described for the Extended Channel Switch Announcement element.
Vendor-SpecificInfo	A set of informa-tion elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.40.4.3 When generated

This primitive is generated by the SME to schedule an accepted Protected extended channel switch request.

10.3.40.4.4 Effect of receipt

On receipt of this primitive, the MLME schedules the extended channel switch. The actual channel switch is at the appropriate time through the MLME-PLME interface using the PLME-SET primitive of the dot11CurrentFrequency MIB attribute.

10.3.41 Protected DSE Power Constraint Announcement

The following MLME primitives support the signaling of DSE power constraint to dependent STAs.

10.3.41.1 MLME-PDDSETPC.request

10.3.41.1.1 Function

This primitive requests that a Protected DSE power constraint frame be sent by an enabling STA.

10.3.41.1.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-PDDSETPC.request          (
                               RequesterSTAAddress,
                               ResponderSTAAddress,
                               DSELocalPowerConstraint,
                               VendorSpecificInfo
                               )
```

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiates the enablement process.
DSELocalPower-Constraint	Integer	0–255	Specifies the Local power constraint, as described in the DSE Power Constraint frame (see 7.4.7.9).
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.41.1.3 When generated

This primitive is generated by the SME to request that a Protected DSE Power Constraint Announcement frame be sent to a dependent STA.

10.3.41.1.4 Effect of receipt

Upon receipt of this primitive, the MLME constructs a Protected DSE Power Constraint Announcement frame. The enabling STA then schedules this frame for transmission.

10.3.41.2 MLME-PDDSETPC.confirm

10.3.41.2.1 Function

This primitive reports the results of a request to send a Protected DSE Power Constraint Announcement frame.

10.3.41.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDSETPC.confirm (
ResultCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, or TIMEOUT	Indicates the result of MLME-PDDSETPC.request.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.41.2.3 When generated

This primitive is generated by the MLME when a Protected DSE power constraint announcement completes.

10.3.41.2.4 Effect of receipt

The SME is notified of the results of the Protected DSE power constraint procedure.

10.3.41.3 MLME-PDDSETPC.indication

10.3.41.3.1 Function

This primitive indicates that a Protected DSE Power Constraint Announcement frame was received from an enabling STA.

10.3.41.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDSETPC.indication (
RequesterSTAAddress,
ResponderSTAAddress,
DSELocalPowerConstraint,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiated the enablement process.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that is the enabling STA.
DSELocalPower-Constraint	Integer	0–255	Specifies the Local power constraint, as described in the DSE Power Constraint frame (see 7.4.7.9).
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.41.3.3 When generated

This primitive is generated by the MLME when a valid Protected DSE Power Constraint Announcement frame is received.

10.3.41.3.4 Effect of receipt

On receipt of this primitive, the SME performs the DSE power constraint procedure (see 11.11.5).

10.3.41.4 MLME-PDDSETPC.response

10.3.41.4.1 Function

This primitive is used to report the result of the DSE power constraint procedure.

10.3.41.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDSETPC.response (
RequesterSTAAddress,
ResponderSTAAddress,
ResultCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peerMAC entity that initiates the enabling process.
ResultCode	Enumeration	SUCCESS, REFUSED	Reports the result of a DSE power constraint procedure.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.41.4.3 When generated

This primitive is generated by the SME to schedule a response to Protected DSE power constraint announcement.

10.3.41.4.4 Effect of receipt

On receipt of this primitive, the MLME schedules the transmission of a Protected DSE power constraint result to the enabling STA that sent the protected DSE power constraint announcement.

10.3.42 Protected Enablement

This mechanism supports the process of establishing an enablement relationship with a peer MAC entity.

10.3.42.1 MLME-PDENABLEMENT.request

10.3.42.1.1 Function

This primitive requests enablement with a specified peer MAC entity.

10.3.42.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDENABLEMENT.request (
RequesterSTAAddress,
ResponderSTAAddress,
EnablementTimeLimit,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiates the enablement process.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity of the enabling STA.
EnablementTime-Limit	Integer	>= 1	Specifies a time limit (in TU) after which the enablement process will be terminated.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.42.1.3 When generated

This primitive is generated by the SME for a STA to establish enablement with a specified peer MAC entity in order to permit Protected Dual of Public Action frames to be exchanged between the two STAs. During the enablement procedure, the SME can generate additional MLME-PDENABLEMENT.request primitives.

10.3.42.1.4 Effect of receipt

This primitive initiates an enablement procedure. The MLME subsequently issues a MLME-PDENABLEMENT.confirm that reflects the results.

10.3.42.2 MLME-PDENABLEMENT.confirm

10.3.42.2.1 Function

This primitive reports the results of a protected enablement attempt with a specified peer MAC entity.

10.3.42.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDENABLEMENT.confirm (
RequesterSTAAddress,
ResponderSTAAddress,
ResultCode,
EnablementIdentifier,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiated the enablement process. This value must match the RequesterSTAAddress parameter specified in the corresponding MLME-PDENABLEMENT.request.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peerMAC entity with which the enablement process was attempted.This value must match the ResponderSTAAddress parameter specified in the corresponding MLME-PDENABLEMENT.request.
ResultCode	Enumeration	SUCCESS, INVALID_ PARAMETERS, TIMEOUT, TOO_MANY_ SIMULTANEOUS_ REQUESTS, REFUSED	Indicates the result of MLME-PDENABLEMENT.request.
Enablement-Identifier	Integer	0–65535	Specifies the dependent enablement identifier.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.42.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-PDENABLEMENT.request for enablement with a specified peer MAC entity.

10.3.42.2.4 Effect of receipt

The SME is notified of the results of the protected enablement procedure.

10.3.42.3 MLME-PDENABLEMENT.indication

10.3.42.3.1 Function

This primitive indicates receipt of a request from a specific peer MAC entity to establish a protected enablement relationship with the STA processing this primitive.

10.3.42.3.2 Semantics of the service primitive

The primitive parameters are as follows:

```
MLME-PDENABLEMENT.indication (
                             RequesterSTAAddress,
                             ResponderSTAAddress,
                             VendorSpecificInfo
                             )
```


Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiated the enablement process.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that is the enabling STA.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.42.3.3 When generated

This primitive is generated by the MLME as a result of the receipt of a Protected enablement request from a specific peer MAC entity.

10.3.42.3.4 Effect of receipt

The SME is notified of the receipt of this protected enablement request.

10.3.42.4 MLME-PDENABLEMENT.response

10.3.42.4.1 Function

This primitive is used to send a protected response to a specified peer MAC entity that requested enablement with the STA that issued this primitive.

10.3.42.4.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDENABLEMENT.response
(
RequesterSTAAddress,
ResponderSTAAddress,
ResultCode,
EnablementIdentifier,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that initiated the enablement process.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peerMAC entity that is the enabling STA.
ResultCode	Enumeration	SUCCESS, REFUSED	Indicates the result response to the enablement request from the peer MAC entity.
Enablement-Identifier	Integer	0–65535	Specifies the dependent enablement identifier.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.42.4.3 When generated

This primitive is generated by the SME of a STA as a response to an MLME-PDENABLEMENT.indication primitive.

10.3.42.4.4 Effect of receipt

This primitive initiates transmission of a Protected Dual of Public Action response to the specific peer MAC entity that requested enablement.

10.3.43 Protected Deenablement

10.3.43.1 MLME-PDDEENABLEMENT.request

10.3.43.1.1 Function

This primitive requests that the enablement relationship with a specified peer MAC entity be invalidated.

10.3.43.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDEENABLEMENT.request (
RequesterSTAAddress,
ResponderSTAAddress,
ReasonCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that requests the deenablement process.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity that becomes deenabled in the process.
ReasonCode	As defined in frame format	As defined in 7.4.7.4	Specifies the reason code for initiating the deenablement process.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.43.1.3 When generated

This primitive is generated by the SME for a STA to invalidate enablement with a specified peer MAC entity in order to prevent the exchange of Protected Dual of Public Action frames between the two STAs. During the deenablement procedure, the SME can generate additional MLME-PDDEENABLEMENT.request primitives.

10.3.43.1.4 Effect of receipt

This primitive initiates a protected deenablement procedure. The MLME subsequently issues a MLME-PDDEENABLEMENT.confirm that reflects the results.

10.3.43.2 MLME-PDDEENABLEMENT.confirm

10.3.43.2.1 Function

This primitive reports the results of a Protected deenablement attempt with a specified peer MAC entity.

10.3.43.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDEENABLEMENT.confirm (
RequesterSTAAddress,
ResponderSTAAddress,
ResultCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity that initiated the deenablement process. This value must match the RequesterSTAAddress parameter specified in the corresponding MLME-PDDEENABLEMENT.request.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the deenablement process was attempted. This value must match the ResponderSTA-Address parameter specified in the corresponding MLME-PDDEENABLEMENT.request.
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS	Indicates the result of MLME-PDDEENABLEMENT.request.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.43.2.3 When generated

This primitive is generated by the MLME as a result of an MLME-PDDEENABLEMENT.request to invalidate the protected enablement relationship with a specified peer MAC entity.

10.3.43.2.4 Effect of receipt

The SME is notified of the results of the deenablement procedure.

10.3.43.3 MLME-PDDEENABLEMENT.indication

10.3.43.3.1 Function

This primitive reports the invalidation of a protected enablement relationship with a specified peer MAC entity.

10.3.43.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PDDEENABLEMENT.indication (
RequesterSTAAddress,
ResponderSTAAddress,
ReasonCode,
VendorSpecificInfo
)

Name	Type	Valid range	Description
RequesterSTA-Address	MACAddress	Any valid individual MAC address	Specifies the address of the peer MAC entity with which the enablement relationship was invalidated.
Responder-STAAddress	MACAddress	Any valid individual MAC address	Specifies the address of the MAC entity with which the enablement relationship was invalidated.
ReasonCode	As defined in frame format	As defined in 7.4.7.4	Specifies the reason the deenablement procedure was initiated.
Vendor-SpecificInfo	A set of information elements	As defined in 7.3.2.26	Zero or more information elements.

10.3.43.3.3 When generated

This primitive is generated by the MLME as a result of the invalidation of an enablement relationship with a specific peer MAC entity.

10.3.43.3.4 Effect of receipt

The SME is notified of the invalidation of the specific enablement relationship.

10.3.44 Vendor Specific Public Action

This set of primitives supports the signaling of Vendor Specific Public Action frames between peer SMEs.

10.3.44.1 MLME-PVSPECIFIC.request

10.3.44.1.1 Function

This primitive requests transmission of a Vendor Specific Public Action frame to a peer entity.

10.3.44.1.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PVSPECIFIC.request (
Peer MAC Address,
OUI,
Vendor Specific Content
)

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity to which the Vendor Specific Public Action frame is sent.
OUI	3 octets	00-00-00 - FF-FF-FF	A public value assigned by the IEEE to identify the entity that has defined the content of the particular Vendor Specific Public Action.
Vendor-SpecificInfo	A set of information elements and vendor-specific fields	A set of information elements and vendor-specific fields	A set of information elements as defined in 7.3.2. A set of vendor-specific fields specifying the required fields for the Vendor Specific Public Action frame.

10.3.44.1.3 When generated

This primitive is generated by the SME to request that a Vendor Specific Public Action frame be sent to a peer entity.

10.3.44.1.4 Effect of receipt

On receipt of this primitive, the MLME constructs a Vendor Specific Public Action frame containing the set of information elements and vendor-specific fields. The STA then attempts to transmit the frame to the peer entity.

10.3.44.2 MLME-PVSPECIFIC.confirm

10.3.44.2.1 Function

This primitive reports the result of a request to send a Vendor Specific Public Action frame to the peer entity.

10.3.44.2.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PVSPECIFIC.confirm (
ResultCode
)

Name	Type	Valid range	Description
ResultCode	Enumeration	SUCCESS, INVALID_PARAMETERS, TIMEOUT, TRANSMISSION_FAILURE, UNSPECIFIED_FAILURE	Indicates the result of the corresponding MLME-PVSPECIFIC.request

10.3.44.2.3 When generated

This primitive is generated by the MLME when the request to transmit a Vendor Specific Public Action frame completes and indicates the results of the request.

10.3.44.2.4 Effect of receipt

On receipt of this primitive, the SME evaluates the ResultCode.

10.3.44.3 MLME-PVSPECIFIC.indication

10.3.44.3.1 Function

This primitive indicates that a Vendor Specific Public Action frame has been received from a peer entity.

10.3.44.3.2 Semantics of the service primitive

The primitive parameters are as follows:

MLME-PVSPECIFIC.indication (
Peer MAC Address,
OUI,
Vendor Specific Content
)

Name	Type	Valid range	Description
Peer MAC Address	MACAddress	Any valid individual MAC address	The address of the peer MAC entity that sent the Vendor Specific Public Action frame.
OUI	3 octets	00-00-00 - FF-FF-FF	A public value assigned by the IEEE to identify the entity that has defined the content of the particular Vendor Specific Public Action.
Vendor-SpecificInfo	A set of information elements and vendor-specific fields	A set of information elements and vendor-specific fields	A set of information elements as defined in 7.3.2. A set of vendor-specific fields specifying the required fields for the Vendor Specific Public Action frame.

10.3.44.3.3 When generated

This primitive is generated by the MLME when a valid Vendor Specific Public Action frame is received.

10.3.44.3.4 Effect of receipt

On receipt of this primitive, the Vendor Specific Content can be made available for SME processes.

11. MLME

11.3 STA authentication and association

11.3.1 Authentication and deauthentication

11.3.1.1 Authentication—originating STA

Change the last paragraph in 11.3.1.1 as follows:

If the requested authentication mechanism is other than FT authentication, the STA's SME ~~shall~~ may delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) before invoking MLME-AUTHENTICATE.request primitive.

11.3.1.2 Authentication—destination STA

Change the second to the last paragraph in 11.3.1.2 as follows:

If the requested authentication mechanism is other than FT authentication, the STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using the MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-AUTHENTICATE.indication primitive if and only if Management Frame Protection had not been negotiated when the PTKSA(s) were created.

11.3.2 Association, reassociation, and disassociation

11.3.2.2 AP association procedures

Insert item b1) after list item b) in 11.3.2.2:

b1) If the STA is associated, has a valid security association, and has negotiated Management Frame Protection, the AP shall reject the Association Request with status code "Association request rejected temporarily; try again later." The AP shall not modify any association state for the non-AP STA, and shall include in the Association Response a Timeout Interval IE with Timeout interval type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA. Following this, if the AP is not already engaging in an SA Query with the STA, the AP shall issue one MLME-SAQuery.request primitive to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQUERY.confirm is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed. The STA shall insert the TransactionIdentifier field value in the SA Query Request frame, and increment the value by 1 for each subsequent SA Query Request frame, rolling over the value to 0 after the maximum allowed value is reached. An AP may interpret reception of a valid protected frame as an indication of a successfully completed SA Query. If an MLME-SAQUERY.confirm with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the AP shall allow the association process to be started without starting an additional SA Query procedure.

Change the last paragraph of 11.3.2.2 and add a new last paragraph as follows:

The STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-ASSOCIATE.indication primitive that results in a successful MLME-ASSOCIATE.response primitive.

In case of a failed SA Query procedure, if the AP receives an Association frame from the STA with which it has an existing SA, then the AP shall send a protected Disassociation frame to this STA prior to terminating the old SA, with Reason Code "Previous Authentication no longer valid."

11.3.2.4 AP reassociation procedures

Insert item b1) after list item b) in 11.3.2.4 as follows:

b1) If the STA is associated and has a valid security association, has negotiated management frame protection, and the reassociation is not a part of a Fast BSS Transition, the AP shall reject the Reassociation Request with status code "Association request rejected temporarily; Try again later." The AP shall not modify any association state for the non-AP STA, and shall include in the Reassociation Response a Timeout Interval IE with type set to 3 (Association Comeback time), specifying a comeback time when the AP would be ready to accept an association with this STA. Following this, if the AP is not in an ongoing SA Query with the STA, the AP shall issue one MLME-SAQuery.request primitive to the STA every dot11AssociationSAQueryRetryTimeout TUs until a matching MLME-SAQUERY.confirm is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure have passed. The STA shall insert the TransactionIdentifier in SA Query Request, and increment this by 1 for each subsequent SA Query Request, and rolling over to 0 after the maximum allowed value in this field. An AP may interpret reception of a valid protected frame as an indication of a successfully completed SA Query. If an MLME-SAQUERY.confirm with an outstanding transaction identifier is not received within dot11AssociationSAQueryMaximumTimeout period, the AP shall allow the association process to be started without starting additional SA Query procedure.

Change the last paragraph of 11.3.2.4 and add a new last paragraph as follows:

Except when the association is part of a fast BSS transition, the STA's SME shall delete any PTKSA and temporal keys held for communication with the indicated STA by using MLME-DELETEKEYS.request primitive (see 8.4.10) upon receiving a MLME-ASSOCIATE.indication primitive that results in a successful MLME-REASSOCIATE.response primitive.

In case of a failed SA Query procedure, if the AP receives a Reassociation frame from the STA with which it has an existing SA, then the AP shall send a protected Disassociate frame to this STA prior to terminating the old SA, with Reason Code "Previous Authentication no longer valid."

11.11 DSE procedures

11.11.1 General

Insert a new sentence at the end of the fourth paragraph of 11.11.1 as follows:

A fixed STA is a registered STA that broadcasts its registered location and is restricted from enabling other STAs (see 11.11.30. An enabling STA is a registered STA that broadcasts its registered location, and regulatory authorities permit it to enable operation of unregistered STAs (see 11.11.4). A dependent STA is an unregistered STA that operates under the control of an enabling STA (see 11.11.5). When Management Frame Protection is negotiated, stations shall use Protected Dual of Public Action frames instead of unicast Public Action frames for DSE procedures.

Insert two new subclauses (11.12 and 11.13) at the end of Clause 11 as follows:

11.12 Broadcast and multicast Robust Management frame procedures

When Management Frame Protection is negotiated, the MLME shall provide an encapsulation service for group addressed Robust Management frames. All group addressed Robust Management frames shall be submitted to this service for encapsulation and transmission.

The group addressed frame protection service shall take the following actions:

— Management Frame Protection for multicast/broadcast shall be set using the MLME-SETPROTECTION.request primitive with the Protectlist including a Key Type value of IGTK. A non-AP STA shall also set the Protect Type value to Rx. In an IBSS, STAs shall set the ProtectType value to Rx_Tx. An AP shall set the Protect Type value to Tx.

— The IGTK shall be installed using the MLME-SETKEYS.request primitive with the value IGTK for the Key Type field in the Key Descriptor element.

— All group addressed Robust Management frames shall be encapsulated and protected using BIP (see 8.3.4).

11.13 SA Query procedures

If d`ot11RSNAProtectedManagementFramesEnabled` is true, then the STA shall support the SA Query procedure.

To send an SA Query Request frame to a peer STA, the STA's SME shall issue an MLME-SAQuery.request primitive. A STA that supports the SA Query procedure and receives an SA Query Request frame shall respond with an SA Query Response frame when all of the following are true: the receiving STA is currently associated to the sending STA, and no pending MLME-ASSOCIATE.request or MLME-REASSOCIATE.request primitives are outstanding for the STA that receives the SA Query indication.

If a non-AP STA that has an SA with its AP for an association that negotiated Management Frame Protection receives an unprotected Deauthentication or Disassociation frame with reason code 6 or 7 from the AP, the non-AP STA may use this as an indication that there may be a mismatch in the association state between itself and the AP. In such a case, the non-AP STA may initiate the SA Query procedure with the AP to verify the validity of the SA by issuing one MLME-SAQuery.request primitive every dot11AssociationSAQueryRetryTimeout until a matching MLME-SAQuery.confirm is received or dot11AssociationSAQueryMaximumTimeout TUs from the beginning of the SA Query procedure has passed. If the AP replies to the SA Query request with a valid SA Query response that has a matching transaction identifier, the non-AP STA may continue to use the SA. If no valid SA Query response is received, the non-AP STA may destroy the SA and move into State 1 with the AP.

11A. Fast BSS Transition

11A.2 Key holders

11A.2.2 Authenticator key holders

Change the second item in the second dashed list of 11A.2.2 as follows:

— The R1KH shall derive and distribute the GTK and IGTK to all connected STAs.

11A.4 FT initial mobility domain association

11A.4.2 FT initial mobility domain association in an RSN

Replace Figure 11A-2 with the following figure:

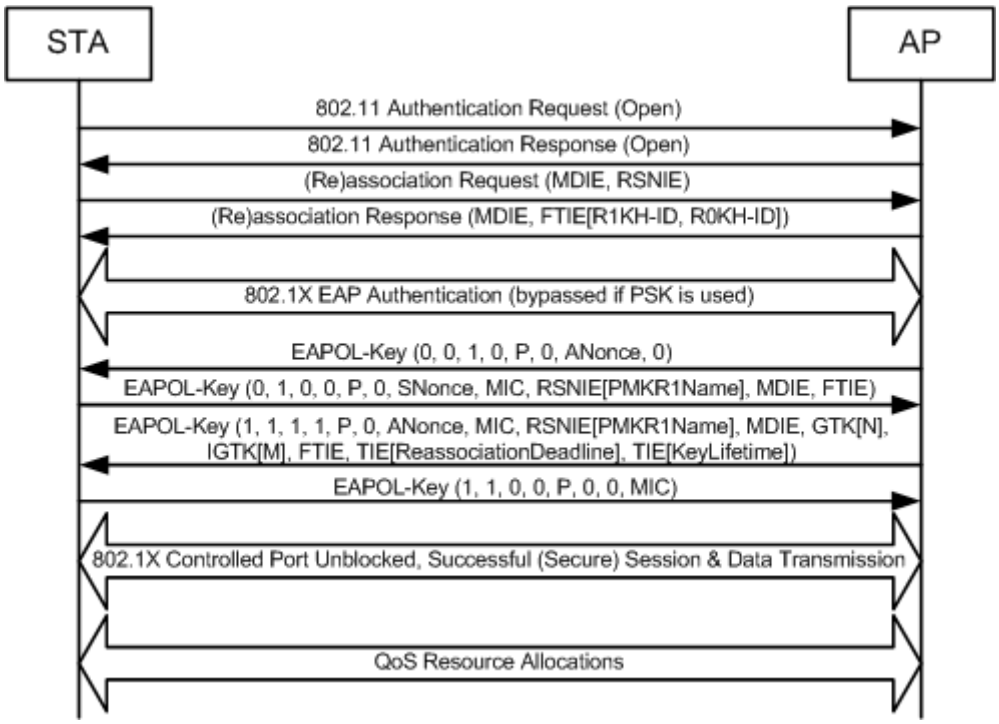

Figure 11A-2—FT Initial Mobility Domain Association in an RSN

Change the third message of the twelfth paragraph in 11A.4.2 as follows:

R1KH → S1KH: Data(EAPOL-Key(1, 1, 1, 1, P, 0, 0, ANonce, MIC, RSNIE[PMKR1Name], MDIE, GTK[N], IGTK[M], FTIE, TIE[ReassociationDeadline], TIE[KeyLifetime])

11A.5 FT protocol

11A.5.2 Over-the-air FT protocol authentication in an RSN

Replace Figure 11A-4 with the following figure:

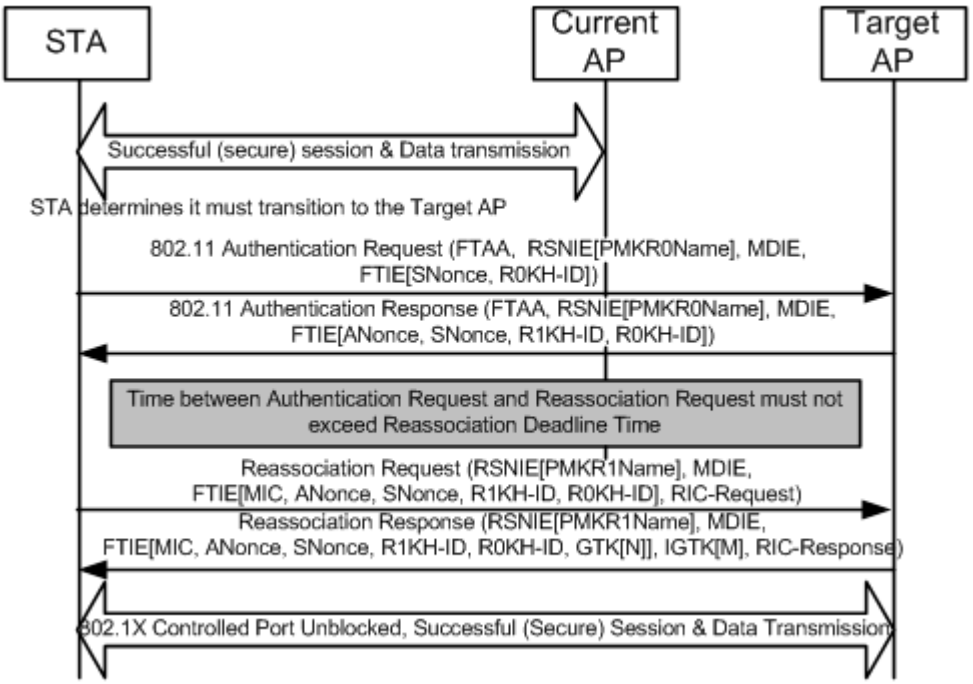

Figure 11A-4—Over-the-air FT Protocol in an RSN

11A.5.3 Over-the-DS FT Protocol authentication in an RSN

Replace Figure 11A-5 with the following figure:

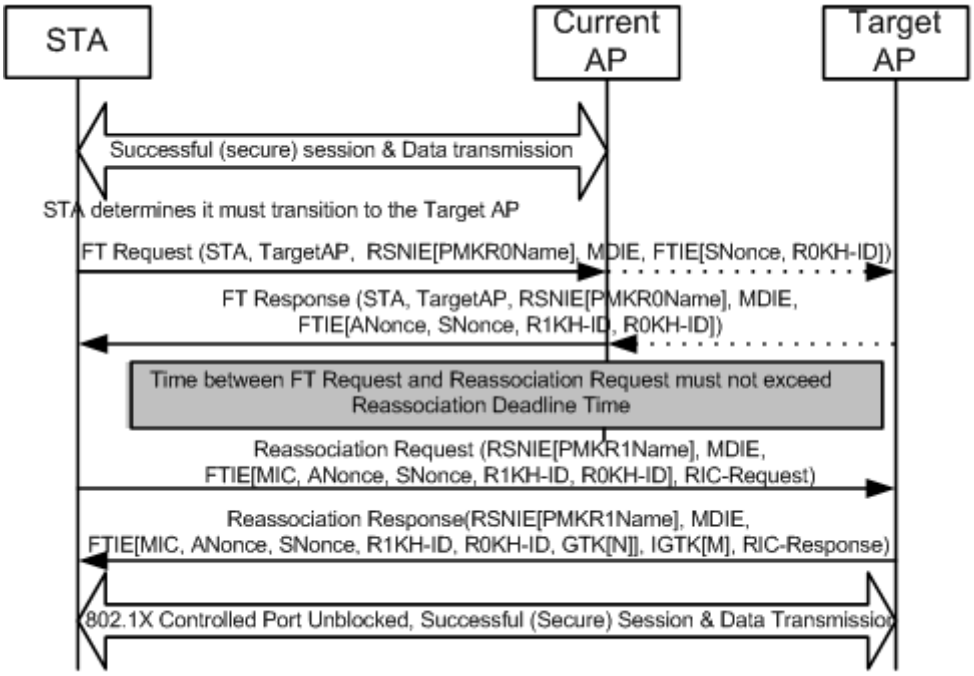

Figure 11A-5—Over-the-DS FT Protocol in an RSN

11A.6 FT Resource Request Protocol

11A.6.2 Over-the-air fast BSS transition with resource request

Replace Figure 11A-9 with the following figure:

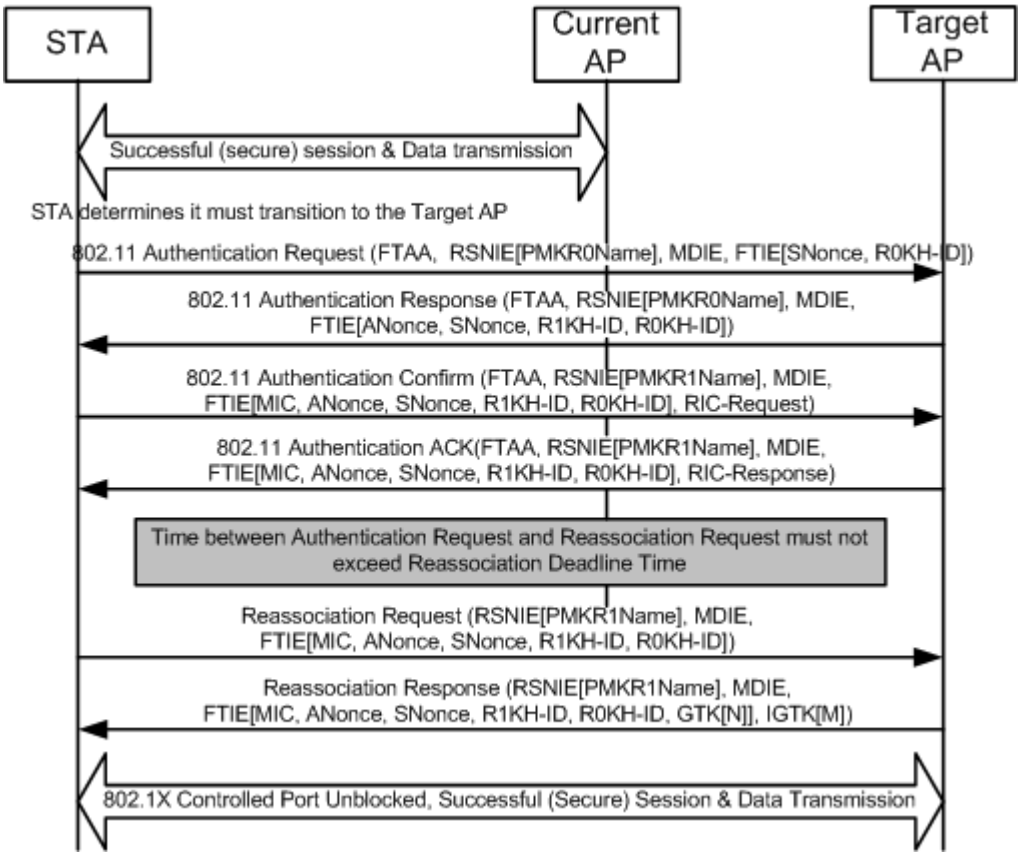

Figure 11A-9—Over-the-air FT Resource Request Protocol in an RSN

Change the twelfth paragraph in 11A.6.2 as follows:

In an RSN, on successful completion of the FT authentication exchange of the FT Resource Request Protocol, the PTKSA has been established and proven live. The key replay counter shall be initialized to zero and the subsequent EAPOL-Key frames (e.g., GTK and IGTK updates) shall use the key replay counter to ensure they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame from the STA within the reassociation deadline timeout value.

11A.6.3 Over-the-DS fast BSS transition with resource request

Change the tenth paragraph in 11A.6.3 as follows:

In an RSN, on successful completion of the FT Confirm/Acknowledgement frame exchange, the PTKSA has been established and proven live. The key replay counter shall be initialized to zero and the subsequent EAPOL-key frames (e.g., GTK and IGTK updates) shall use the key replay counter to ensure they are not replayed. The PTKSA shall be deleted by the target AP if it does not receive a Reassociation Request frame

from the STA within the reassociation deadline timeout value. Resource request procedures are specified in 11A.11.

11A.7 FT reassociation

11A.7.1 FT reassociation in an RSN

Change the second message of the second paragraph in 11A.7.1 as follows:

Target AP → STA: Reassociation Response(RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], GTK[N], IGTK[M], RIC-Response)

11A.8 FT authentication sequence

11A.8.5 FT authentication sequence: contents of fourth message

Change the third dashed item of the fourth paragraph in 11A.8.5 as follows:

— When this message of the authentication sequence appears in a Reassociation Response frame, the Optional Parameter(s) field in the FTIE may include ~~a~~ the GTK and IGTK subelements. If a GTK or an IGTK are ~~is~~ included, the Key field of the subelement shall be encrypted using KEK and the NIST AES key wrap algorithm. The Key field shall be padded before encrypting if the key length is less than 16 octets or if it is not a multiple of 8. The padding consists of appending a single octet 0xdd followed by zero or more 0x00 octets. When processing a received message, the receiver shall ignore this trailing padding. Addition of padding does not change the value of the Key Length field. Note: The length of the encrypted Key field can be determined from the length of the GTK or IGTK subelement.

11A.9 FT security architecture state machines

11A.9.3 R1KH state machine

Replace Figure 11A-14 with the following figure:

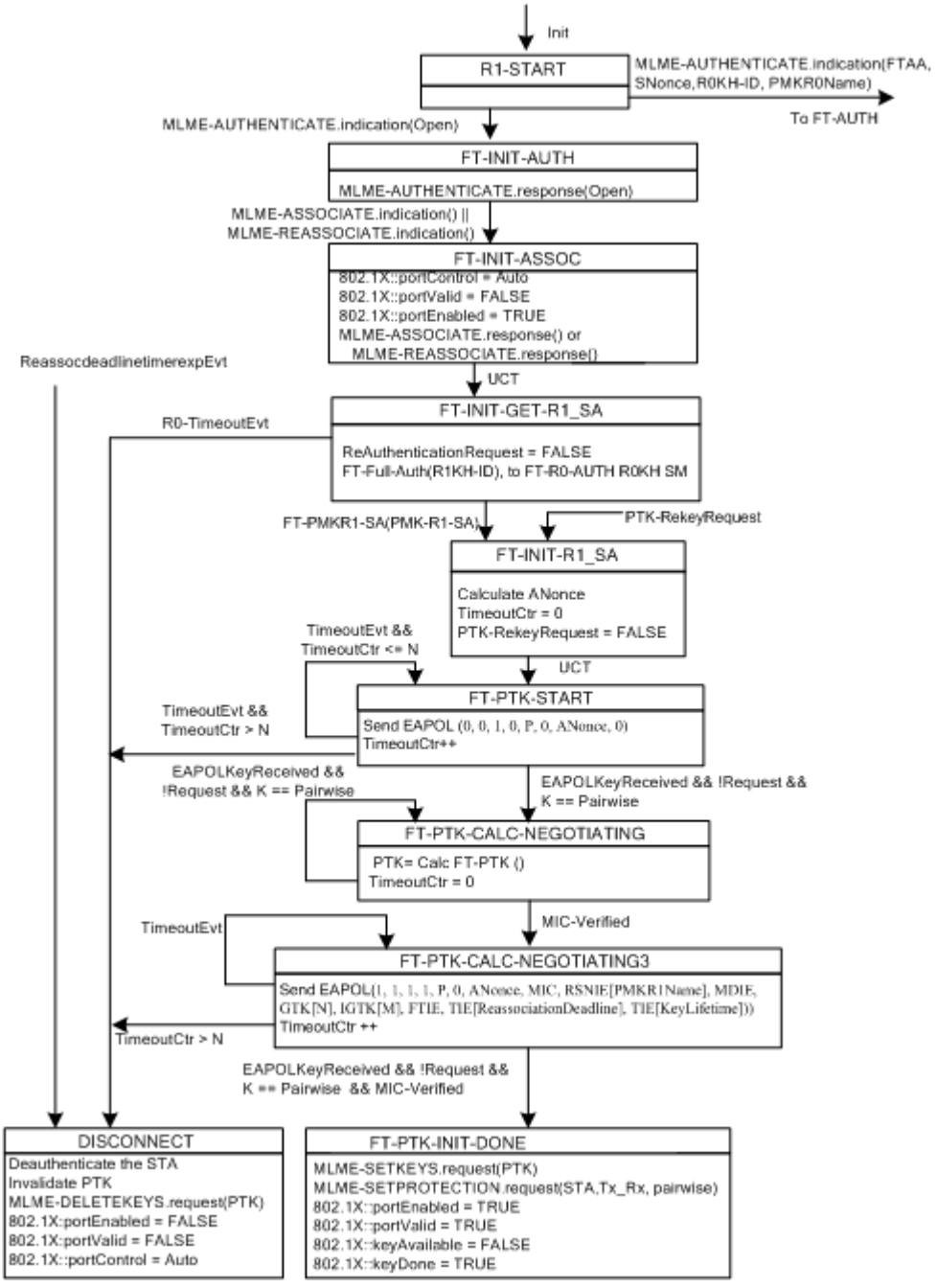

Figure 11A-14—R1KH state machine, including portions of the SME (part 1)

Replace Figure 11A-15 with the following figure:

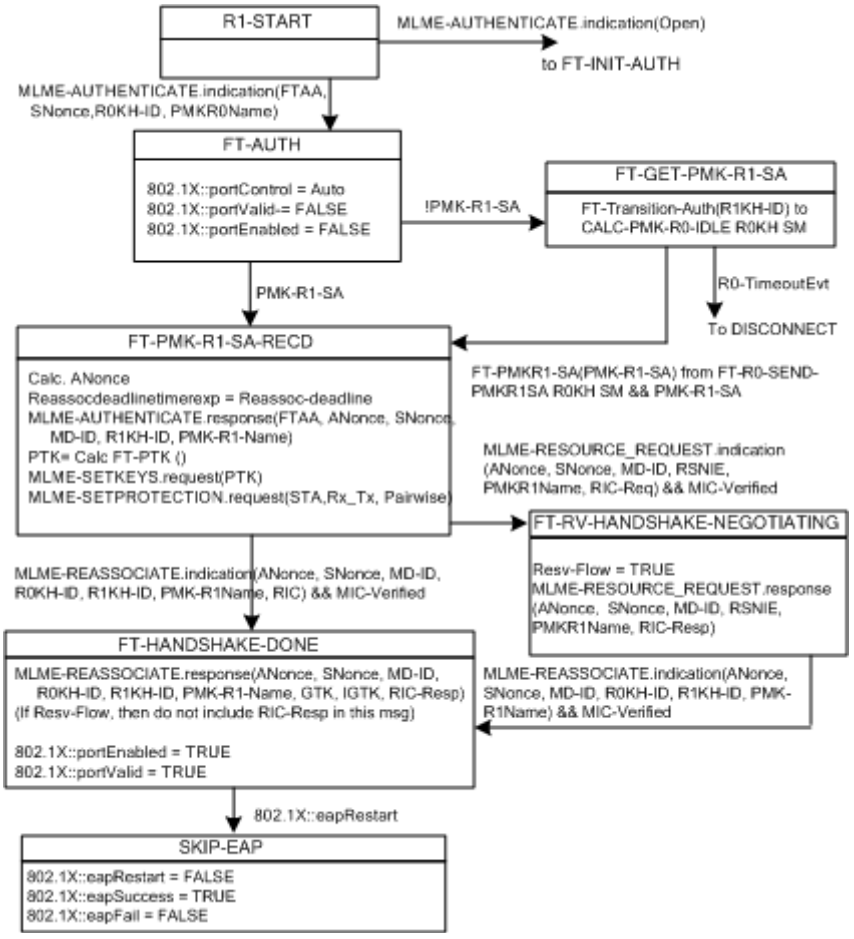

Figure 11A-15—R1KH state machine, including portions of the SME (part 2)

Annex A

(normative)

Protocol Implementation Conformance Statement (PICS) proforma

A.4 PICS proforma—IEEE Std 802.11-2007

A.4.4 MAC protocol

A.4.4.1 MAC protocol capabilities

Change row entry PC34 of the table in A.4.4.1 as follows:

* PC34	Robust security network association (RSNA)	7.2.2, 7.3.1.4, 5.4.3.3, 8.7.2.1, 8.7.2.2, 8.7.2.3, 8.7.2.4, 11.3.1, 11.3.2, 8.3.3	O	Yes ☐ No ☐

Change row entry PC34.1.3 of the table in A.4.4.1 as follows:

*PC34.1.3	Authentication key management (AKM) suite list	7.3.2.25, 8.3.1	PC34.1: M	Yes ☐ No ☐

Insert the following rows after PC34.1.9 in the table in A.4.4.1 as follows:

*PC 34.1.10	Management Frame Protection	7.3.1.11, 7.4.2, 7.1.3.1.9, 7.3.2.25.3, 8.3.2.1.1, 8.3.2.1.2, 8.3.2.2, 8.3.2.3.4, 8.3.3.3.2, 8.3.3.3.5, 8.3.3.4.1, 8.3.3.4.3, 8.4.3, 8.7.2.1a, 8.7.2.2a, 8.7.2.3a, 8.7.2.5	PC34:O	Yes ☐ No ☐
*PC 34.1.10.1	BIP	8.3.4, 11	PC34.1.10:M	Yes ☐ No ☐
PC 34.1.10.1.1	Management MIC IE	7.3.2.55	PC34.1.10.1:M	Yes ☐ No ☐
PC 34.1.11	AKM: IEEE 802.1X authentication with SHA-256 PRF	7.3.2.25, 8.5	PC34:O	Yes ☐ No ☐
PC 34.1.12	AKM: PSK with SHA-256 PRF	7.3.2.25, 8.5	PC34:O	Yes ☐ No ☐

Insert the following row after PC35.6 in the table in A4.4.1:

PC36	SA Query Procedure	7.4.9, 11.3	PC34.1.10:M	Yes ☐ No ☐

Annex D

(normative)

ASN.1 encoding of the MAC and PHY MIB

Change the `Dot11StationConfigEntry` ***in Annex D by inserting the new entries at the end of the existing entries as follows:***

```
Dot11StationConfigEntry ::=
     SEQUENCE {
        dot11StationID                              MacAddress,
        dot11MediumOccupancyLimit                      INTEGER,
        dot11CFPollable                             TruthValue,
        dot11CFPPeriod                                 INTEGER,
        dot11CFPMaxDuration                            INTEGER,
        dot11AuthenticationResponseTimeOut          Unsigned32,
        dot11PrivacyOptionImplemented               TruthValue,
        dot11PowerManagementMode                       INTEGER,
        dot11DesiredSSID                           OCTET STRING,
        dot11DesiredBSSType                            INTEGER,
        dot11OperationalRateSet                    OCTET STRING,
        dot11BeaconPeriod                              INTEGER,
        dot11DTIMPeriod                                INTEGER,
        dot11AssociationResponseTimeout             Unsigned32,
        dot11DisassociateReason                        INTEGER,
        dot11DisassociateStation                    MacAddress,
        dot11DeauthenticateReason                      INTEGER,
        dot11DeauthenticateStation                  MacAddress,
        dot11AuthenticateFailStatus                    INTEGER,
        dot11AuthenticateFailStation                MacAddress,
        dot11MultiDomainCapabilityImplemented       TruthValue,
        dot11MultiDomainCapabilityEnabled           TruthValue,
        dot11CountyString                          OCTET STRING,
        dot11SpectrumManagementImplemented          TruthValue,
        dot11SpectrumManagementRequired             TruthValue,
        dot11RSNAOptionImplemented                  TruthValue,
        dot11RSNAPreauthenticationImplemented       TruthValue,
        dot11RegulatoryClassesImplemented           TruthValue,
        dot11RegulatoryClassesRequired              TruthValue,
        dot11QoSOptionImplemented                   TruthValue,
        dot11ImmediateBlockAckOptionImplemented     TruthValue,
        dot11DelayedBlockAckOptionImplemented       TruthValue,
```

```
        dot11DirectOptionImplemented                TruthValue,
        dot11APSDOptionImplemented                  TruthValue,
        dot11QAckOptionImplemented                  TruthValue,
        dot11QBSSLoadOptionImplemented              TruthValue,
        dot11QueueRequestOptionImplemented          TruthValue,
        dot11TXOPRequestOptionImplemented           TruthValue,
        dot11MoreDataAckOptionImplemented           TruthValue,
        dot11AssociateinNQBSS                       TruthValue,
        dot11DLSAllowed                             TruthValue,
        dot11AssociateStation                       MacAddress,
        dot11AssociateID                               INTEGER,
        dot11AssociateFailStation                   MacAddress,
        dot11AssociateFailStatus                       INTEGER,
        dot11ReassociateStation                     MacAddress,
        dot11ReassociateID                             INTEGER,
        dot11ReassociateFailStation                 MacAddress,
        dot11ReassociateFailStatus                     INTEGER,
        dot11RadioMeasurementCapable                TruthValue,
        dot11RadioMeasurementEnabled                TruthValue,
        dot11RRMMeasurementProbeDelay                  INTEGER,
        dot11RRMMeasurementPilotPeriod                 INTEGER,
        dot11RRMLinkMeasurementEnabled              TruthValue,
        dot11RRMNeighborReportEnabled               TruthValue,
        dot11RRMParallelMeasurementsEnabled         TruthValue,
        dot11RRMRepeatedMeasurements                TruthValue,
        dot11RRMBeaconPassiveMeasurementEnabled     TruthValue,
        dot11RRMBeaconActiveMeasurementEnabled      TruthValue,
        dot11RRMBeaconTableMeasurementEnabled       TruthValue,
        dot11RRMBeaconMeasurementReportingConditionsEnabled
                                                    TruthValue,
        dot11RRMFrameMeasurementEnabled             TruthValue,
        dot11RRMChannelLoadMeasurementEnabled       TruthValue,
        dot11RRMNoiseHistogramMeasurementEnabled    TruthValue,
        dot11RRMStatisticsMeasurementEnabled        TruthValue,
        dot11RRMLCIMeasurementEnabled               TruthValue,
        dot11RRMLCIAzimuthEnabled                   TruthValue,
        dot11RRMTransmitStreamCategoryMeasurementEnabled
                                                    TruthValue,
        dot11RRMTriggeredTransmitStreamCategoryMeasurementEnabled
                                                    TruthValue,
        dot11RRMAPChannelReportEnabled              TruthValue,
        dot11RRMMIBEnabled                          TruthValue,
        dot11RRMMaxMeasurementDuration              Unsigned32,
```



```
        dot11RRMNonOperatingChannelMaxMeasurementDuration
                                                    Unsigned32,
        dot11RRMMeasurementPilotTransmissionInformationEnabled
                                                    TruthValue,
        dot11RRMMeasurementPilotCapability          Unsigned32,
        dot11RRMNeighborReportTSFOffsetEnabled      TruthValue,
        dot11RRMRCPIMeasurementEnabled              TruthValue,
        dot11RRMRSNIMeasurementEnabled              TruthValue,
        dot11RRMBSSAverageAccessDelayEnabled        TruthValue,
        dot11RRMBSSAvailableAdmissionCapacityEnabled TruthValue,
        dot11RRMAntennaInformationEnabled           TruthValue,
        dot11FastBSSTransitionImplemented           TruthValue,
        dot11LCIDSEImplemented                      TruthValue,
        dot11LCIDSERequired                         TruthValue,
        dot11DSERequired                            TruthValue,
        dot11ExtendedChannelSwitchEnabled           TruthValue,
        dot11RSNAProtectedManagementFramesEnabled   TruthValue,
        dot11RSNAUnprotectedManagementFramesAllowed  TruthValue,
        dot11AssociationSAQueryMaximumTimeout       Unsigned32,
        dot11AssociationSAQueryRetryTimeout          Unsigned32
    }
```

Insert the following after the `dot11ExtendedChannelSwitchEnabled` ***MIB definition:***

```
--**********************************************************
--* Management Frame Protection MIBs
--**********************************************************

dot11RSNAProtectedManagementFramesEnabled       OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This variable indicates whether or not this STA
            enables Management Frame Protection."
      DEFVAL { FALSE }
               ::= { dot11StationConfigEntry 88}



dot11RSNAUnprotectedManagementFramesAllowed     OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This variable indicates whether or not this STA supports
```

```
            RSNA STAs which do not provide Robust Management frames pro-
            tection."
      DEFVAL { TRUE }
                ::= { dot11StationConfigEntry 89}

--*******************************************************
--* SA Query Procedure MIBs
--*******************************************************
dot11AssociationSAQueryMaximumTimeout          OBJECT-TYPE
      SYNTAX Unsigned32 (1...4294967295)
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This attribute specifies the number of time units (TUs)
            that an AP can wait, from the scheduling of the first SA
            Query Request to allow association process to be started
            without starting additional SA Query procedure if a success-
            ful SA Query Response is not received."
      DEFVAL { 1000 }
                ::= { dot11StationConfigEntry 90}

dot11AssociationSAQueryRetryTimeout            OBJECT-TYPE
      SYNTAX Unsigned32 (1...4294967295)
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This attribute specifies the number of time units (TUs)
            that an AP waits between issuing two subsequent MLME-
            SAQUERY.request primitives."
      DEFVAL { 201 }
                ::= { dot11StationConfigEntry 91}
```

Insert at the end of the `dot11RSNAStatsEntry SEQUENCE` ***the following entries:***

```
Dot11RSNAStatsEntry ::=
    SEQUENCE {
          dot11RSNAStatsIndex                       Unsigned32,
          dot11RSNAStatsSTAAddress                  MacAddress,
          dot11RSNAStatsVersion                     Unsigned32,
          dot11RSNAStatsSelectedPairwiseCipher      OCTET STRING,
          dot11RSNAStatsTKIPICVErrors               Counter32,
          dot11RSNAStatsTKIPLocalMICFailures        Counter32,
          dot11RSNAStatsTKIPRemoteMICFailures       Counter32,
          dot11RSNAStatsCCMPReplays                 Counter32,
          dot11RSNAStatsCCMPDecryptErrors           Counter32,
          dot11RSNAStatsTKIPReplays                 Counter32,
          dot11RSNAStatsCMACICVErrors               Counter32,
```



```
            dot11RSNAStatsCMACReplays                   Counter32,
            dot11RSNAStatsRobustMgmtCCMPReplays         Counter32,
            dot11RSNABIPMICErrors                       Counter32 }
```

Insert at the end of the `dot11RSNAStatsTKIPReplays OBJECT-TYPE` ***definition the following new*** `dot11RSNAStatsEntry` ***definitions:***

```
dot11RSNAStatsCMACICVErrors              OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of received MPDUs discarded by the CMAC integ-
            rity check algorithm."
                ::= { dot11RSNAStatsEntry 11 }

dot11RSNAStatsCMACReplays           OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of received MPDUs discarded by the CMAC replay
            errors."
                ::= { dot11RSNAStatsEntry 12 }


dot11RSNAStatsRobustMgmtCCMPReplays OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of received Robust Management frame MPDUs dis-
            carded due to CCMP replay errors"
             ::= {dot11RSNAStatsEntry 13}

dot11RSNABIPMICErrors OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of received MMPDUs discarded due to BIP MIC
            errors"
             ::= {dot11RSNAStatsEntry 14}
```

Annex H

(informative)

RSNA reference implementations and test vectors

Insert the following two new clauses (H.8 and H.9) at the end of Annex H:

H.8 Test vectors for AES-128-CMAC

Test vectors for AES-128-CMAC are in Annex D.1 of NIST SP-800-38B

H.9 Management Frame Protection test vectors

H.9.1 BIP with broadcast Deauthentication frame

Unprotected broadcast Deauthentication frame (without FCS):
c0 00 00 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00 09 00 02 00

FC=c0 00
DUR=00 00
DA=ff ff ff ff ff ff
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=09 00
Reason Code: 02 00

IGTK: 4e a9 54 3e 09 cf 2b 1e ca 66 ff c5 8b de cb cf

BIP AAD (FC | A1 | A2 | A3): c0 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00

Management Frame Body: 02 00

MMIE (with MIC=0): 4c 10 04 00 04 00 00 00 00 00 00 00 00 00 00 00 00

AES-128-CMAC(AAD | Management Frame Body | MMIE): 48 df bf a7 b8 27 88 72

Protected broadcast Deauthentication frame (without FCS):

c0 00 00 00 ff ff ff ff ff ff 02 00 00 00 00 00 02 00 00 00 00 00 09 00 02 00 4c 10 04 00 04 00 00
00 00 00 48 df bf a7 b8 27 88 72

FC=c0 00 (note: Protected flag is _not_ set)
DUR=00 00
DA=ff ff ff ff ff ff
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=09 00
Reason Code: 02 00

MMIE: 4c 10 04 00 04 00 00 00 00 00 48 df bf a7 b8 27 88 72

(KeyID = 04 00 (= 4), Seq# = 04 00 00 00 00 00, MIC = 48 df bf a7 b8 27 88 72)

H.9.2 CCMP with unicast Deauthentication frame

Plaintext unicast Deauthentication frame (without FCS):
c0 00 00 00 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 60 00 02 00

FC=c0 00
DUR=00 00
DA=02 00 00 00 01 00
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=60 00
Reason Code: 02 00

CCMP TK: 66 ed 21 04 2f 9f 26 d7 11 57 06 e4 04 14 cf 2e

CCM flags: 59 (Adata: 1, M: 011, L: 001)
Nonce = Nonce Flags | A2 | PN
= 10 (Management)
02 00 00 00 00 00
00 00 00 00 00 01
l(m) = 00 02

AAD = FC | A1 | A2 | A3 | SC | A4 | QC
= c0 40 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 00 00
AAD blocks:
00 16 c0 40 02 00 00 00 01 00 02 00 00 00 00 00
02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Encrypted Deauthentication frame (without FCS):
c0 40 00 00 02 00 00 00 01 00 02 00 00 00 00 00 02 00 00 00 00 00 60 00 01 00 00 20 00 00 00 00
1d 07 ca fd 04 09 bb 8b af ef

FC=c0 40 (note: Protected bit set)
DUR=00 00
DA=02 00 00 00 01 00
SA=02 00 00 00 00 00
BSSID=02 00 00 00 00 00
SEQ=60 00
CCMP Header: 01 00 00 20 00 00 00 00 (PN=1, ExtIV=1)
Encrypted Data: 1d 07

MIC: ca fd 04 09 bb 8b af ef