P802.11w™/D1.~~01~~02

Draft Amendment to Standard for Information Technology - Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements -

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

Amendment <number> : Protected Management Frames

EDITORIAL NOTE - In this redline version of the draft, inserted text is shown as Inserted Text ***and deleted text is shown as*** ~~Deleted Text~~.

EDITORIAL NOTE - Editorial notes are distinguished like this. They are not part of the amendment and will be removed before it is published..

EDITORIAL NOTE - the amendment number will be inserted by IEEE-SA editorial staff during preparation for publication.

EDITORIAL NOTE- This revision of the amendment is based on the following (baseline) documents:

- *P802.11 REV-ma D9.0*
- *802.11k D6.0*
- *802.11r D4.0*
- 802.11n ~~D1~~D2.~~08~~0

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.

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. Editorial notes will not be carried over into future editions because the changes will be incorporated into the base standard.

EDITORIAL NOTE - The following table is not part of the amendment, and will be removed before the document is finalized.

Table 0—Change history

Draft version	Date	Contributions and motions applied
1.01	Jan. 17, 2007	Apply all editorial comment resolutions per 11-06-1729r11.
1.02	Mar. 8, 2007	Apply passed motions per 11-06-1759r4 and 11-07-100r3.

2. References

Insert the following new normative reference in alphabetical order:

NIST SP800-38B, Dworkin, M., "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", May 2005, <http://csrc.nist.gov/CryptoToolkit/modes/800-38_Series_Publications/SP800-38B.pdf>

3. Definitions

Change the following definition in alphabetical order:

Robust Management Frame: A management frame that is eligible for protection by the protected management traffic service.

4. Abbreviations and acronyms

Insert the following new abbreviations and acronyms in alphabetical order:

BIP	Broadcast/Multicast Integrity Protocol
~~DGTK~~IGTK	~~Disconnect~~ Integrity GTK
~~DHV~~	~~Disconnect Hash Value~~
~~IGTK~~	~~Integrity GTK~~
MMIE	Management MIC Information Element

5. General Description

5.1 General description of the architecture

5.2 Components of the IEEE 802.11 architecture

5.2.1 The independent BSS (IBSS) as an ad hoc network

5.2.2 STA membership in a BSSS is dynamic

5.2.3 Distribution system (DS) concepts

5.2.3.1 Extended service set (ESS): The large coverage network

5.2.3.2 RSNA

Insert at the end of the hashed item list in 5.2.3.2:

— Enhanced cryptographic encapsulation mechanisms for Robust Management frames

5.3 Logical service interfaces

~~***Insert at the end of the list of architectural services in 5.3, preserving the list order as follows:***~~

~~o) Unicast management frame confidentiality and integrity~~

~~p) Broadcast/Multicast management frame integrity~~

~~***EDITORIAL NOTE - 11ma D9.0 last entry is m; 11k added n.***~~

~~5.3.1 Station services (SS)~~

~~***Insert at the end of the list of SS, preserving the list order as follows:***~~

~~j) Unicast management frame confidentiality and integrity~~

~~k) Broadcast/Multicast management frame integrity~~

~~***EDITORIAL NOTE - 11ma D9.0 last entry is 'h'; 11k added 'i'.***~~

5.4 Overview of the services

5.4.1 Distribution of messages within a DS

5.4.2 Services that support the distribution service

5.4.2.4 Disassociation

Change the 3rd 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~~.~~ except when Robust Management frame protection is enabled and the disassociation message integrity check fails.

5.4.3 Access control and data confidentiality services

5.4.3.1 Authentication

5.4.3.2 Deauthentication

Change the 2nd paragraph of 5.4.3.2 as follows:

In an ESS, because authentication is a prerequisite for association, the act of deauthentication ~~shall cause~~ causes 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~~ is not be refused by either party except in the case when management frame protection is enabled. In the latter case, deauthentication will not occur if the ~~deauthentication~~ message integrity ~~check fails or, if the DHV~~ check fails. ~~When an AP sends a deauthentication notice to an associated STA, the association shall also be terminated~~.When an associated STA receives a deauthentication notice, validates the MIC in the ~~DHV~~M-MIE, and sends an 802.11 Acknowlegement frame, the association shall also be ~~deemed~~ terminated.

Change the 4th paragraph of 5.4.3.2 as follows:

In an RSNA, deauthentication also destroys any related PTKSA, group temporal key security association (GTKSA), station to station link master key security association (SMKSA), and station to station link 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.

EDITORIAL NOTE: The 11w 1.0 draft paragraphs did not match the 802.11ma D9.0 paragraph, so it has been adapted to match as close as feasible.

5.4.3.3 Data Confidentiality

5.4.3.4 Key management

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 robust management frame can determine which STA transmitted the MAC protocol data unit (MPDU) or MAC management protocol data unit (MMPDU). 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~~ frames, or unicast Robust Management frames, and Deauthenticate or Disassociate frames with Robust Management protection. The protocols do not guarantee data origin authenticity for broadcast/multicast data frames or broadcast/multicast Robust Management 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 robust management frame from another STA can detect whether the received data frame is an unauthorized retransmission. This mechanism is provided for STAs that use ~~CCMP~~ CCMP, ~~or~~ TKIP, or BIP.

Insert a new subclause 5.4.3.7 after 5.4.3.6 as follows:

5.4.3.7 Robust Management frame protection

Management frame protection is required in an RSNA to protect against forgery and eavesdropping on robust unicast management frames, and against forgery on robust broadcast/multicast management frames.

Management frame protection extends the CCMP data frame protection ~~protocol~~ to provide data confidentiality, replay protection, and data origin authenticity for robust unicast management frames, including Action frames, disassociate and deauthenticate frames.

Forgery protection for robust broadcast/multicast management ~~Action~~ frames is provided through the Broadcast/Multicast Integrity Protocol (BIP), using AES-128-CMAC for message integrity. ~~For robust broadcast/multicast management Action frames, the~~ The BIP protocol also provides replay protection~~, and specifies a mechanism for providing protection against forgery by an authenticated STA (insider attacks)~~. ~~Robust broadcast and multicast disassociate and deauthenticate management frames are protected against insider attacks.~~

EDITORIAL NOTE: The editor was further removed references to insider attack in the above paragraph, though this was missed in the adopted submission 11-06-1932r0. Since the removal of DHV, there is no longer the means to mitigate insider attacks.

Management frame protection protocols apply to Robust Management frames after the RSNA PTK key establishment for protection of unicast frames is completed and after the GTKs to protect broadcast/multicast frames that have been delivered. All management frames sent or received by a STA before keys are installed shall be unprotected.

5.5 Multiple logical address spaces

5.6 Differences between ESS and IBSS LANs

5.7 Reference Model

5.8 IEEE 802.11 and IEEE 802.1X

5.8.1 IEEE 802.11 usage of IEEE 802.1X

5.8.2 Infrastructure functional model overview

5.8.2.1 AKM operations with AS

Change the second paragraph of 5.8.2.1 as follows::

A 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.

— Derive a fresh pairwise transient key (PTK) from the PMK.

— Install the pairwise encryption and integrity keys into IEEE 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 Robust management frames is enabled, transport the integrity GTK (IGTK), and the IGTK sequence ~~number, and the Disconnect Hash Value (DHV)~~ number from Authenticator to the Supplicant and install these values in the STA and, in not already installed, in the AP.

— Validate 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 Robust Management Frame protection is enabled, the Authenticator ~~shall~~ also ~~use~~ uses the Group Key Handshake with all associated STAs to change the IGTK. The Authenticator ~~shall encrypt~~ encrypts the GTK, ~~IGTK,~~ and ~~DHV~~ IGTK values in the EAPOL-Key frame as described in 8.5.

Replace Figure 13 with the following figure, with the changes being the inclusion of “IGTK~~, DHV~~” in message 3 and in both the Supplicant and Authenticator boxes that begin with “Install”:

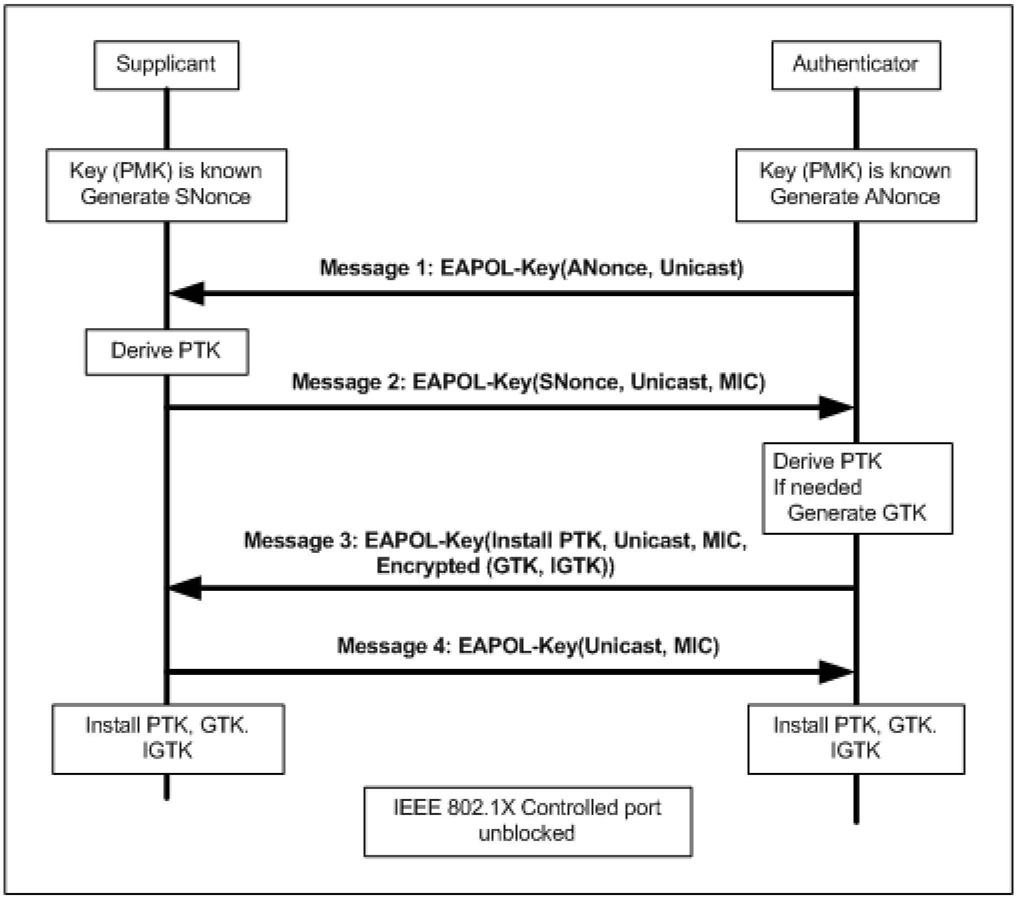


Figure 13—Establishing pairwise and group keys

Replace Figure 14 with the following figure, with the changes being the inclusion of "Encrypted IGTK" in Message 1, "IGTK~~, DHV~~" in the Authenticator box beginning with 'Generate" and "IGTK" in the Supplicant box beginning with "Install"; the update to the 2nd box on the right has intentially fixed from the 802.11ma D9.0 draft to correctly state "Encrypt GTK, IGTK with KEK":

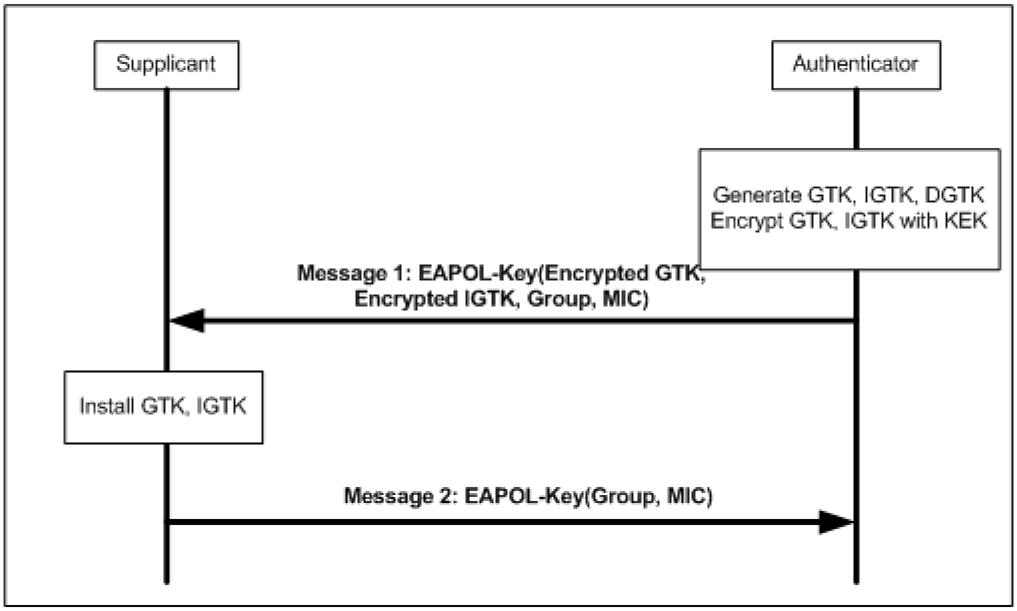

Supplicant
Authenticator
Generate GTK, IGTK, DGTK
Encrypt GTK, IGTK with KEK
Message 1: EAPOL-Key(Encrypted GTK,
Encrypted IGTK, Group, MIC)
Install GTK, IGTK
Message 2: EAPOL-Key(Group, MIC)

Figure 14—Delivery of subsequent group keys

5.8.2.2 Operations with PSK

Change the 3rd item in 5.8.2.2 as follows:

— The GTK ~~and GTK sequence number~~, IGTK, ~~GTK~~and their IGTK sequence number, ~~their associated~~ GTK sequence ~~numbers and per STA DHV~~ number are sent from the Authenticator to the Supplicant just as in the AS case. See Figure 13 and Figure 14.

5.8.3 IBSS functional model description

5.8.4 Authenticator-to-AS protocol

5.8.5 PMKSA caching

Insert a new subclause 5.8.6 after 5.8.5 as follows:

5.8.6 Protection of robust broadcast and multicast management Action frames

When Robust Management frame support has been enabled, all ~~robust~~ Robust broadcast/multicast management ~~Action~~ frames are submitted for encapsulation to the ~~Action frame~~ broadcast/multicast frame protection service as described in 11.7. This service shall protect the frame using BIP.

6. MAC service definition

6.1 Overview of MAC services

6.1.1 Data service

6.1.2 Security services

Change the text of 6.1.2 as follows:

Security services in IEEE 802.11 are provided by the authentication service and the TKIP, ~~and~~ CCMPand BIP mechanisms. The scope of the security services provided is limited to station-to-station data and robust management frame exchanges. 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 also provided for the MMPDU. 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 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 802.11 are as follows:

a) Data Confidentiality;

b) Authentication; and

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

BIP provides authentication (integrity) and access control for robust broadcast/multicast management frames.

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

The MAC sublayer security services provided by TKIP, ~~and~~ CCMP and BIP rely on information from non-layer-2 management or system entities. Management entities communicate information to TKIP and CCMP through a set of MAC sublayer management entity (MLME) interfaces and MIB attributres; 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.

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 within 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 show that the frame body must be one octet or longer to apply the encapsulation).

7.2 Format of individual frame types

7.3 Management frame body components

7.3.1 Fixed fields

7.3.1.1 Authentication algorithm number field

7.3.1.2 Authentication transaction sequence number field

7.3.1.3 Beacon interval field

7.3.1.4 Capability information field

7.3.1.5 Current AP address field

7.3.1.6 Listen interval field

7.3.1.7 Reason code field

Insert the following rows into Table 22 - Reason Codes before the "Reserved" entry and update the numbering appropriately:

Table 22—Reason Codes

Reason Code	Meaning
TBD	Invalid management group cipher
TBD	Robust management frame policy violation

EDITORIAL NOTE: The entry values are left as TBD for now, pending ANA assignment

7.3.2 Information Elements

7.3.2.25 RSN information element

Insert the following XXX rows (ignoring the header row) in Table 26 - Element IDs in the correct position to preserve ordering by the "Element ID" column and update the "Reserved" range of codes appropriately:

Table 26—Element IDs

Information Element	Element ID	Length (in octets)
Management MIC (see 7.3.2.51 (MMIE))	TBD	16

{EDITORIAL NOTE : TBD request to ANA for assignment of MMIE}.

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, and RSN Capabilities field, the PMK identifier (PMKID) count, ~~and~~ PMKID list. If dot11RSNAProtectedManagementFramesEnabled is set to TRUE, ~~and~~ a single management group cipher suite ~~selector~~selector is appended to the information element. See Figure 89. ~~All STAs implementing RSNA shall support this element.~~ The size of the RSN infomration element is limited by the size of an information element, which is 255 octets. Therefore, the number of pairwise cipher suited, AKM suites, and PMKIDs is limited.

Insert the following paragraph after the first paragraph of 7.3.2.25 as follows:

All STAs implementing RSNA shall support this element. Further, if dot11RSNAProtectedManagementFramesEnabled is TRUE, then the Robust Management frame protection bit in the RSN capabilities field shall be set to 1 and the Management Group Cipher Suite must be present in this information element.

Replace Figure 89 with the following figure, where a new field* Management Group Cipher *is inserted at the end and "Data" in serted in the 4th column to read "Group Data Cipher Suite":

Element ID	Length	Version	Group Data Cipher Suite	Pairwise Cipher Suite Count	Pairwise Cipher Suite List	AKM Suite Count	AKM Suite List	RSN Capabilities	PMKID Count	PMKID List	Management Group Suite
1	1	2	4	2	4*m	2	4*n	2	2	16*s	4

Figure 89—RSN Information Element format

Insert the following text before 7.3.2.25.1:

IEEE 802.1X authentication, CCMP pairwise and group key cipher suites (neither WEP-40, WEP-104, nor TKIP are allowed), robust management frame protection allowed and enforced with AES-128-CMAC as the broadcast/multicast management suite selector.

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 the data group key cipher suite

01 00, // pairwise key cipher suite count

00 0F AC 04, // CCMP as pairwise key cipher suite

01 00, // authentication count

00 0F AC 01 // IEEE 802.1X authentication

03 40 // Management frame protection is enabled and enforced

00 00 // No PMKIDs

00 0F AC 06, // AES-128-CMAC as the broadcast/multicast management cipher suite

7.3.2.25.26 Cipher suites

Change the 1st 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 data traffic.

Change the 3rd paragraph of 7.3.2.25.1 as follows:

The Pairwise Cipher Suite List field contains a series of cipher suite selectors that indicate the pairwise cipher suites contained in the RSN information element. The Management Group Cipher Suite field contains the cipher suite selector used by the BSS to protect broadcast/multicast management traffic.

Change Table 32 as follows:

Table 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 in an RSNA
00-0F-AC	5	WEP-104
00-0F-AC	TBD	AES-128-CMAC - default management group cipher suite in a BIP enabled RSNA
00-0F-AC	?-255	Reserved
Vendor OUI	Other	Vendor specific

Table 32—Cipher suite selectors

OUI	Suite Type	Meaning
Other	any	Reserved

EDITORIAL NOTE: Last assigned value is 5; should request ANA for value 6, but leave as TBD as noted above.

Insert the following paragraph after the third paragraph of 7.3.2.25.1:

When Robust Management frame protection is enabled, the negotiated pairwise cipher suite ~~shall be~~ is used to protect the unicast Robust Management frames and the management group cipher suite ~~shall be~~ is used to protect the broadcast/multicast Robust Management frames. Use of AES-128-CMAC is only valid as a management group cipher suite. CCMP is the only valid pairwise cipher suite used to protect unicast Robust Management frames.

Replace Table 33 with the following Table:

7.3.2.25.2 AKM suites

Table 33—Cipher suite usage

			Enabled Robust Management Frame Protection	
Cipher Suite Selector	**GTK**	**PTK**	**Unicast Robust management frames**	**Broadcast/multicast robust management frames**
Use group key	No	Yes	No	No
WEP-40	Yes	No	No	No
WEP-104	Yes	No	No	No
TKIP	Yes	Yes	No	No
CCMP	Yes	Yes	No	No
AES-128-CMAC	No	No	No	Yes

7.3.2.25.3 RSN capabilities

Replace Figure 91 with the following (change being the addition of bit 6 as AES-128-CMAC[#7], bit 7 as Robust Management frame protection and changing "Reserved" to be 8):

B0	B1	B2 – B3	B4 – B5	B6	B7	B8	B9	B10-15

Pre-Auth	No Pairwise	PTKSA Replay Counter	GTKSA Replay Counter	AES-128-CMAC	Robust Management frame protection	Reserved	PeerKey Enabled	Reserved

Figure 91—RSN Capabilities field format

EDITORIAL NOTE: TGr has already made the assignment of bit 6 for AES-128-CMAC which needs to be approved by IEEE 802.11 ANA. This update requests TGw usurp bit 7 for Robust Management frame ~~protection~~protection pending ANA assignment.

Insert after DashList item "Bits 6":

— Bit 7: Robust Management frame protection. ~~An AP and~~ A STA ~~with Robust Management frame protection both set~~ sets this bit to 1 to enable protection of ~~management~~ Robust Management frames. If an AP has set the `dot11RSNALegacyManagementFrames,` then that AP allows RSNA connections from non-AP STAs which do not provide Robust Management frame protection.

Change DashList item "Bits 7-8 and 10-15" as follows:

— Bits ~~7~~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.

—

EDITORIAL NOTE: 802.11ma ends with 7.3.2.35, TGr adds it through 48, TGn succeeds it through 50, TGw follows with 51

Insert at the end of subclause 7.3.2.50 the new subclause 7.3.2.51 as follows:

7.3.2.51 Management MIC information element

The Management MIC information element (MMIE) protects robust broadcast/multicast management frames from forgery and replay. It also provides message integrity for broadcast/multicast ~~Deauthenticate and Disassociate~~ Robust Management Frames. Figure 112wa shows the MMIE format.

	Element ID	Length	Key ID	Replay	MIC
Octets	1	1	2	6 ~~or 16~~	8

Figure 112wa—Management MIC information element format

EDITORIAL NOTE: The figure is numbered 112wa as TGn follows its own convention, 112 is the last figure in TGma D9.0 and TGr also has inserted figures after 112.

The value of the Element ID field ~~shall be~~ is TBD.

{EDITORIAL NOTE : TBD request to ANA for assignment}.

The Length field denotes the number of octets in the information element and ~~shall have~~ has a value of ~~either 16 or 26. Length 26 shall be used with protected Broadcast/Multicast Disassociate frames and protected Broadcast/Multicast Deauthenticate frames. Length 16 shall be used with other robust broadcast / multicast management frames~~16.

The Key ID field identifies the broadcast/multicast key used to compute the MIC. Bits 0-11 ~~shall define~~ defines a value in the range 0-4095. Bits 12 - 15 ~~shall be~~ are reserved and set to 0 on transmission and ignored on reception. ~~The current Disconnect GTK (DGTK) shall use Key ID value 0.~~

~~When the Length field is 16, the~~ The Replay field value ~~shall be~~ is 6 octets, interpreted as a 48-bit unsigned integer and used as a sequence number~~. The length value of 26 shall only be used within broadcast Disassociation or Deauthentication frames and then the Replay field value shall be 16 octets interpreted as a 128 bit key~~.

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

8. Security

8.1 Framework

8.1.1 Security methods

Insert the following sub-item at the end of 8.1.1:

— BIP, described in 8.3.4

8.1.2 RSNA equipment and RSNA capabilities

8.1.3 RSNA establishment

Insert sub-item '7' in the first item ('a') as follows:

7) If Robust Management frame protection is enabled, it programs the TK and pairwise cipher suite into the MAC for protection of robust unicast management frames. It also installs the IGTK, and IGTK sequence ~~counter and the DHV (for a non-AP STA) or the DGTK (for the AP)~~counter.

Insert sub-item '6' in the second item ('b') as follows:

6) If Robust Management frame protection is enabled, it protects the Robust management frames by programming the negotiated pairwise cipher suite and established ~~PTK and the~~ PTK, IGTK, and IGTK sequence ~~counter and DHV (for a non-AP STA) or DGTK (for the AP) into the MAC~~counter.

8.2 Pre-RSNA security methods

Change the title of 8.3 as follows:

8.3 RSNA ~~data~~ confidentiality and integrity protocols

8.3.1 Overview

Change the 1st paragraph of 8.3.1 as follows:

This standard defines two ~~three~~ RSNA data confidentiality and integrity protocols: TKIP, and ~~CCMP and~~ CCMP. This standard defines one integrity protocol: BIP. Implementation of CCMP shall be mandatory in all IEEE 802.11 devices claiming RSNA compliance. 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 protection of Robust management frames is enabled and is used to ~~protect~~ provide integrity protection for Robust broadcast/multicast management frames.

8.3.2 Temporal Key Integrity Protocol (TKIP)

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~~cipher and dot11RSNAProtectedManagementFramesEnabled is TRUE, Robust unicast management frames shall be protected with ~~CCMP by setting the management bit in the Nonce Flags of the Nonce construction to 1~~CCMP. ~~-~~A MAC implementation shall support CCMP for protecting management frames if CCMP and Robust Management Frame protection are both supported.

8.3.3.2 CCMP MPDU format

8.3.3.3 CCMP cryptographic encapsulation

8.3.3.3.1 PN processing

8.3.3.3.2 Construct AAD

Change Figue 135 with the following underlined updates as follows:

	FC (bits 4,5,6 = 0 in data MPDUs only) (bits ~~,~~11,12,~~13~~ = 0) (bit 13 = 0 in data MPDU only) (bits 14=1)	A1 or MMPDU DA	A2 or MMPDU SA	A3 or MMPDU BSSID	SC (bits 4-15=0)	A4	QC
Octets	2	6	6	6	2	6	2

Figure 135—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 sub-fields 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) ~~in a Data MPDU~~ masked to 0;

   5) The Protected Frame bit (bit 14) always set to 1.

b) A1 - MPDU Address 1 ~~or MMPDU DA~~ field

c) A2 - MPDU Address 2 ~~or MMPDU SA~~ field

d) A3 - MPDU Address 3 ~~or MMPDU BSSID~~ 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 Data MPDU.

g) QC - QoS Control field, if present, a 2-octet field that includes the MDSU priority. The QC TID 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).

EDITORIAL NOTE: ~~802.11ma includes the SC, A4 and QC descriptions that must be considered in this draft as well.~~ The SC field should be considered in an MMPDU as ~~well~~well so no modifications are made in there per CID , the A4 and Qos do not readily apply to management frames (yet).

8.3.3.3.3 Construct CCM nonce

Replace Figure 136 with the following figure "replacing "Priority Octet" with "Nonce Flag Octet", and addition of "Management" as bit 4 of this octet.":

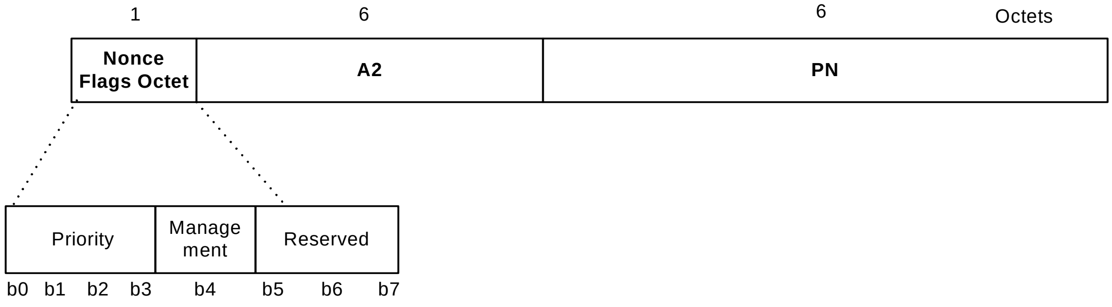


Figure 136—Nonce Construction

EDITORIAL NOTE: ANA must be requested for this bit (4). There has been issues editing the above Figure and thus is left as B4....readers should note that until ANA assigns this bit, it should read TBD, pending ANA assignment.

Change the ~~first hashed item description in~~ second paragraph 8.3.3.3.3 ~~by removing it and adding the following three items~~ as follows:

The Nonce field has an internal structure of the ~~Priority~~ Nonce Flags Octet || 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 o 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 field 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).
- 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)~~. The ~~; otherwise, if the Type field is not 00, then the Management ~~subfield ~~field of the Nonce ~~Flags field ~~flags octet shall be set to ~~0 if the Type field of the Frame Control field is 10 (Data frame).~~0.
- Bits 5 to 7 of the Nonce Flags field are reserved and shall be set to 0 on transmission and ignored on reception.
- 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.

8.3.3.3.4 Construct CCMP Header

8.3.3.3.5 CCM originator processing

Insert the following text at the end of 8.3.3.3.5:

A CCMP protected robust unicast management frame shall ~~use ~~be protected with the ~~same TK as a Data MPDU~~TK.

8.3.3.4 CCMP decapsulation

Change item 'c' as follows:

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

Insert the following paragraph at the end of 8.3.3.4:

When the received frame is a CCMP protected robust unicast 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 robust unicast management frame shall use the same TK as a Data MPDU or MMPDU

8.3.3.4.2 Decrypted CCMP MPDU

8.3.3.4.3 PN and replay detection

Change item 'e' as follows:

e) For each PTKSA, GTKSA, IGTKSA 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~~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 bullet after 'e' :

e1) If ~~management frame protection~~ dot11RSNAProtectedManagementFramesEnables is ~~enabled~~TRUE, the recipient shall maintain a single management frame replay counter and shall use the PN ~~recovered~~ from a received management frame to detect replayed management frames. A replayed frame occurs when the PN ~~extracted~~ from a received management frame is less than or equal to the current management frame replay counter value. The transmitter shall not reorder Robust Management frames.

Insert a new subclause 8.3.4 at the end of 8.3.3 as follows:

8.3.4 The Broadcast/Multicast integrity protocol

Broadcase/Multicast Integrity Protocol (BIP) provides data integrity and replay protection for ~~robust~~ Robust broadcast/multicast ~~management frames~~Management frames after successful completion of a 4-way Handshake. ~~It also provides source authentication for broadcast/multicast Disassociate and Deauthentication frames.~~

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 64 (8 octets).

BIP uses the Integrity GTK (IGTK) to compute the broadcast/multicast MMPDU MIC. The Authenticator distributes a new IGTK and the IGTK sequence number whenever it distributes a new GTK. The IGTK is identified by the MAC address of the STA transmitting it, plus a non-zero 12-bit key identifier that is encoded in the MMIE Key ID field.

~~BIP also distributes a per-supplicant disconnect hash value (DHV) computed from the DGTK whenever it distributes a new GTK. BIP uses the DGTK only for broadcast/multicast Disassociate and Deauthenticate messages. The DGTK is identified by the MAC address of the STA transmitting Disassociate and Deauthenticate messages.~~

8.3.4.2 BIP MMPDU format

Figure ~~137a~~ 137wa shows the BIP MMPDU.

IEEE 802.11 Header	Management Frame Body	Management MIC IE	FCS

Figure 137wa—BIP Base Mode Encapsulation

8.3.4.3 BIP AAD Construction

Like AES-CCMP, BIP uses an additional authentication data (AAD). The AAD shall be constructed from the IEEE 802.11 MMPDU 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 for the MMPDU, with the following exceptions:

a) FC - MMPDU 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 - MMPDU DA

c) A2 - MMPDU SA

d) A3 - MMPDU BSSID

e) SC - MMPDU Sequence Control field, with the sequence number field (bits 4-15 of the Sequence Control field) masked to zero. The Fragment number bits are not modified.

Figure ~~137b~~ 137wb depicts the format of the AAD. The length of the AAD is 22 octets.

	FC (bits 11, 12, 13 = 0)	A1	A2	A3	SC (Bits 4-15 = 0)
Octets:	2	6	6	6	2

Figure 137wb—~~AAD~~ wAAD Construction

8.3.4.4 BIP replay protection

~~BIP uses a different replay protection scheme, depending on the management frame type and format of the MMIE.~~

~~8.3.4.4.1 MMIE replay field as sequence number~~

~~When the MMIE Length field is 16, the MMIE Replay field represents a sequence number. An MMIE of length 16 shall only be used with broadcast/multicast action management frames. Broadcast/multicast action management frames shall only use MMIE of length 16.~~

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

~~In this case the~~ The transmitter shall insert a monotonically increasing value into the MMIE Replay field. The receiver shall maintain a 48-bit replay counter for each IGTK. The replay counter, provided in either the 4-way or Group Key handshakes, shall be set to the value provided by the Authenticator. ~~When a protected robust broadcast/multicast management frame subsequently arrives with an MMIE Length field of 16, the~~ The receiver shall interpret the MMIE Replay field as a 48 bit integer. It shall then compare this integer value against the replay counter for the IGTK identified by the MMIE Key ID field. If the integer value from the received MMIE Replay Field is less than or equal to the replay counter value for the IGTK, the receiver shall silently discard the frame and increment the `dot11RSNAStatsBIPReplays` counter.

~~8.3.4.4.2 MMIE replay field as current DGTK~~

~~When the MMIE Length field is 26, the MMIE Replay field contains the current DGTK of its sender. An MMIE of length 26 shall only be used with broadcast/multicast disassociation or deauthentication frames. Broadcast/multicast disassociation or deauthentication frames shall only use MMIE of length 26.~~

~~The transmitter shall insert the DGTK into the MMIE Replay field. Each receiver shall maintain its own 128-bit Disconnect Hash Value (DHV). The transmitter distributes the DHV when the pairwise security association is established. The DHV is derived by the transmitter as described in Section 8.5.1.3B.~~

~~The receiver identifies the DGTK and DHV by using the transmitter's MAC address.~~

~~Upon receipt of a protected broadcast/multicast Disassociate or Deauthenticate management frame, indicated by an MMIE Length field of 26, the receiver shall interpret the MMIE Replay field value as the DGTK. To determine if the received protected broadcast/multicast frame is a replay or forgery, the receiver shall compute:~~

~~C' := Truncate-128(SHA-256(TA || RA || DGTK))~~

~~and bitwise compare C' with the DHV for the TA in the received frame. If the two are not identical, the receiver shall silently discard the frame and increment the `dot11RSNAStatsBroadcastDHVMismatches` counter. If the two values are identical, the replay protection algorithm succeeds. In this case, the DHV and DGTK values shall be discarded immediately after completing the check.~~

8.3.4.5 BIP transmission

When a STA transmits a protected robust broadcast/multicast management frame it shall:

a) Select the appropriate key (~~IGTK or DGTK~~IGTK) for the frame and construct the MMIE (see 7.3.2.26) with the MIC field ~~muted~~ masked to zero.

b) Compute AAD as specified in 8.3.4.3.

c) Compute AES-128-CMAC over the concatenation of (AAD || Management Frame Body || MMIE)

d) Truncate the result and insert the result into the MMIE MIC field.

e) Compose the broadcast/multicast Robust Management frame as the IEEE 802.11 header, Management Frame Body, and MMIE.

f) Transmit the robust broadcast/multicast management frame.

8.3.4.6 BIP reception

When a STA receives a robust broadcast/multicast management frame protected by BIP, it shall:

a) Identify the appropriate key and associated state based on the MMIE Length field.

b) Execute the appropriate replay protection scheme defined in 8.3.4.4.

c) If the replay protection scheme succeeds, the receiver shall compute AAD for this management frame, as specified in 8.3.4.3 and compute the AES-~~CMAC~~128-~~64~~ CMAC over the concatenation of (AAD || Management Frame Body || MMIE). If the replay protection fails, the dot11RSNAStatsCMACReplays shall be incremented and the frame shall be discarded.

If the result does not match the value in the MMIE, then the receiver shall silently discard the frame and increment the `dot11RSNAStatsCMACICVErrors` counter. If BIP is enabled, broadcast management frames that are received without BIP protection shall be silently discarded.

8.4 RSNA security association management

8.4.1 Security associations

8.4.1.1 Security association definitions

Change the last sentence of the second paragraph as follows:

There are ~~four~~ six types of security associations supported by an RSN STA:

Insert the following item after the thirs hashed item (e.g. after GTKSA) in the list:

— IGTKSA: A result of a successful Group Key Handshake or successful 4-way Handshake.

EDITORIAL NOTE: Should we add to the end of the above “or FT 4-way Handshake, or a successful Fast BSS Transition method”?

8.4.1.1.1 PMKSA

8.4.1.1.2 PTKSA

8.4.1.1.3 GTKSA

Insert the following as a new subclause succeeding 8.4.1.1.3:

8.4.1.1.3A IGTKSA

~~The~~ When Robust Management frame is enabled, the IGTKSA is created by the 4-way Handshake or the Group Key Handshake and is unidirectional. In ~~an AP~~ a BSSID there is one IGTKSA, used for integrity protection of transmitted MMPDUs. An IGTKSA is created by the Supplicant's SME when Message 3 of the 4-Way Handshake is received or when Message 1 of the Group Key Handshake is received. The IGTKSA is created by the Authenticator's SME when the Authenticator SME changes the IGTK and has sent the IGTK to all STAs with which it has a PTKSA.

An IGTKSA consists of the following elements:

— Direction flag (whether the IGTK is used for transmit or receive)

~~— For the Authenticator, the IGTK~~

— KeyID

— For ~~each Supplicant~~the Authenticator, the ~~supplicant-specific DHV~~IGTK

— Authenticator MAC address.

8.4.1.2 Security association life cycle

8.4.1.2.1 Security association in an ESS

Change item ‘d’ 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 protect Robust management frames.

Change the last sentence of the last paragraph as follows:

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

8.4.2 RSNA selection

8.4.3 RSNA policy selection in an ESS

Insert the following text before 8.4.3.1:

An RSNA-capable AP may choose to accept, as set in the policy variable `dot11RSNALegacyManagementFrames`, RSNA STAs with or without the capability for management frame protection set in the RSNIE. A STA may choose not to associate with an AP that does not advertise protection of Robust management frames in the RSN capabilities. In the case where an RSNA STA tries to associate without Robust Management frame protection, the AP may reject the (re)association if `dot11RSNALegacyManagementFrames` is set to zero; in this instance, the AP shall return a (re)association response with a status code of TBD.

EDITORIAL NOTE: pending ANA assignments of new codes assigned per 7.3.1.7 additions.

8.4.4 RSNA policy selection in an IBSS

8.4.4.1 TSN policy selection in an IBSS

Insert a new subclause 8.4.4.2 as follows:

8.4.4.2 Robust Management frame policy selection in an IBSS

Robust management frame protection is valid only if RSNA is selected to protect data messages and `dot11RSNAProtectedManagementFramesEnabled` is set to ~~true~~TRUE. For unicast messages, Robust management frame protection uses the same cipher suite as for unicast data. The Management Group cipher suite advertised in the Beacons and Probe Responses is used. The same Management Group cipher suite shall be used by all STAs in the IBSS; a STA shall reject authentication or 4-way Handshake messages from a STA that advertises a different Management Group cipher suite from its own.

If the Robust Management frame protection bit of the RSN Capabilities field is set in the Beacons and Probe Responses received from a peer STA, then the local STA shall examine the Management Group Cipher Suite field in the RSN IE. An IBSS STA with Robust Management protection uses RSNA security association procedures, and in addition, includes the Management Group Cipher Suite field in the RSN IE, which is confirmed in the 4-Way Handshake.

8.4.5 RSN management of the IEEE 802.1X Controlled Port

8.4.6 RSNA authentication in an ESS

8.4.7 RSNA authentication in an IBSS

8.4.8 RSNA key management in an ESS

8.4.9 RSNA key management in an IBSS

Change the text of 8.4.9 as follows:

To establish a security association between two STAs in an IBSS, each STA's SME must have an accompanying IEEE 802.1X Authenticator and Supplicant. Each STA's SME initiates the 4-Way Handshake from the Authenticator to the peer STA's Supplicant (see 8.4.7). Two separate 4-Way Handshakes are conducted.

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 Robust Management frame protection is enabled, its own IGTK. Each Authenticator generates its own GTK and if Robust 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 Robust management frames protection is enabled, 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, AKMP, and if Robust Management frame protection is enabled, Management Group 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, AKMP, and if Robust Management frame protection is enabled, Management Group 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.

8.4.10 RSNA security association termination

Change first paragraph as follows:

When a non-AP STA SME receives a successful MLME Association or Reassociation confirm primitive 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 primitive 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~~, DGTKSA~~,, SMKSA, and any STKSA, and the AP’s SME shall delete the PTKSA, ~~DGTKSA~~ 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. 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.

EDITORIAL NOTE: TGr has also updated this paragraph, do we reconcile it here or let 802.11ma do the update?

Insert a new subclause 8.4.11 as follows:

8.4.11 Protection of unicast/broadcast/multicast management ~~Action~~ frames

When Robust Management frame protection is enabled, all transmissions of Robust management Action frames shall be protected. Unicast Action frames shall have integrity and confidentiality protection using pairwise keys. Multicast and broadcast ~~Action~~ Action, disassociate and deauthenticate frames (sent by the AP) shall be integrity protected only using BIP. ~~Broadcast and multicast Action frames (sent by the AP) may be converted to unicast frames by MUP.~~

NOTE- BIP does not provide protection against forgery by associated and authenticated non-AP STAs.

Protection of broadcast/multicast management Action frames shall be provided by a service in the MLME as described in 11.7.

8.5 Keys and key distribution

8.5.1 Key hierarchy

Change the first paragraph and its succeeding item list 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/multicast traffic~~

c) ~~Integrity GTK (IGTK)~~GTK, a hierarchy consisting of a single key to ~~provide integrity protection for robust broadcast~~ protect multicast and broadcast/multicast ~~management frames~~traffic

d) ~~Disconnect~~ Integrity GTK (~~DGTK~~IGTK), a hierarchy consisting of a single key to provide ~~source authentication~~ integrity protection for robust broadcast and multicast ~~de-authenticate and disassociate~~ management frames~~.~~

8.5.1.1 PRF

8.5.1.2 Pairwise key hierarchy

Change the description of "The temporal key (TK)" as follows:

— The temporal key (TK) is used for protecting both unicast data and Robust Management frames. The ~~temporal key (~~TK~~)~~ shall be computed as bits 256-283 (for CCMP) or bits 256-511 (for TKIP) of the PTK:

$$\mathrm{TK} \leftarrow \mathrm{L}(\mathrm{PTK}, 256, 128) \text{ or}$$

$$\mathrm{TK} \leftarrow \mathrm{L}(\mathrm{PTK}, 256, 256)$$

8.5.1.3 Group key hierarchy

Insert the following two new subclauses after 8.5.1.3 as follows:

8.5.1.3A Integrity Group Key hierarchy

The IGTK shall be initialized with a random value.

The Authenticator may update the IGTK for reasons such as:

a) The Disassociation or Deauthentication of a STA.

b) An event within the STA's SME which triggers a Group Key Handshake.

The EAPOL-Key state machines (8.5.5 and 8.5.6) configure the IGTK via the MLME-SETKEYS.request primitive.

The IGTK sequence counter is used to provide replay protection. ~~The DGTK is used to provide data origin authentication and integrity services for broadcast/multicast disassociate and deauthenticate management frames.~~

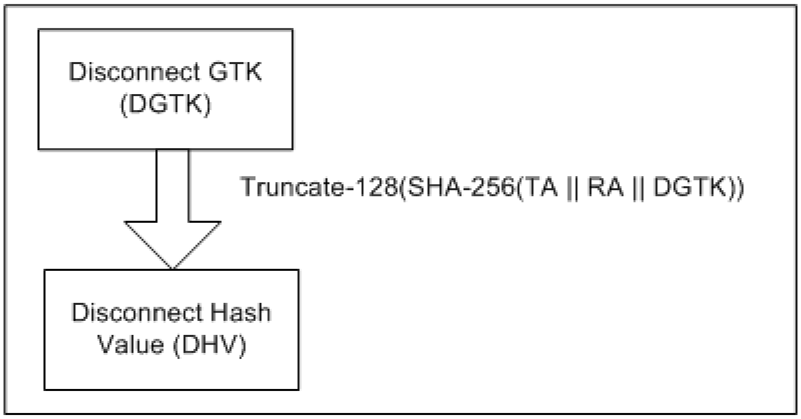
Note that a STA that has left the group can forge frames as an outsider until the IGTK is updated.

~~8.5.1.3B Disconnect Group Key Hierarchy~~

~~The DGTK shall be initialized with a random value.~~

~~Any Disconnect GTK (DGTK) may be re-initialized using a Group Key Handshake at a time interval configured into the AP to reduce the risk of data forgery if the DGTK is ever compromised.~~

~~The Disconnect Group Key Hierarchy uses SHA-256 to derive the DHV from the DGTK, as shown in Figure 106a. The DGTK is used to provide source authentication for broadcast and multicast Deauthenticate and Disassociate management frames.~~


Disconnect GTK (DGTK)
Truncate-128(SHA-256(TA || RA || DGTK))
Disconnect Hash Value (DHV)

~~Figure 106a—Disconnect Group Key Hierarchy~~

~~Where:~~

- ~~The DGTK shall be a random or pseudo-random value contributed by the IEEE 802.1X Authenticator~~
- ~~TA is the MAC address of the AP~~
- ~~RA is the MAC address of the STA~~
- ~~The Disconnect HashValue (DHV) shall be derived from the DGTK.~~

~~Note that the AP only needs to maintain the DGTK to generate the unique per STA DHV. The IGTK is used to provide integrity services to robust broadcast/multicast management frames.~~

8.5.2 EAPOL-Key Frames

Insert the following ~~rows~~ row into Table 62 - KDE before the "Reserved" entry and update the numbering appropriately:

Table 62—KDE

OUI	Data Type	Meaning
00-0F-AC	9	DHV KDE

Table 62—KDE

OUI	Data Type	Meaning
00-0F-AC	10	IGTK KDE

Table 62—KDE

OUI	Data Type	Meaning
00-0F-AC	9	IGTK KDE

EDITORIAL NOTE: 802.11ma has values assigned up through 8. Values assigned above are pending ANA request assignments. The Reserved field must be updated to be ~~11~~10-255.

Insert the following text and tables after Table 64:

~~The format of the DHV KDE is shown in Table 64.~~

Table 64A—DHV KDE format

DHV
16 octets

The format of the IGTK KDE is shown in Table 64B. The Sequence Number (PN) corresponds to the last PN used by the broadcast/multicast transmitter, to be used by the receiver as the initial value for the BIP replay counter.

Table 64B—IGTK KDE format

KeyID	PN	IGTK
2 octets	6 octets	16 octets

8.5.2.1 EAPOL-Key frame notation

Change the GTK notation as follows:

GTK[$N_{\underline{GTK}}$] is the GTK, with key identifier field set to $N_{\underline{GTK}}$. The key identifier specifies which index should be used for this GTK. Index 0 shall not be used for GTKs, except in mixed environments, as described in 8.5.1.

Insert the following text after the GTK notation:

IGTK[$N_{\underline{IGTK}}$] is the GTK, with key identifier field set to N_{IGTK}.
PN is the packet number provided by the IGTK KDE

~~DGTK~~	~~is the DHV KDE~~

8.5.3 4-Way Handshake

8.5.4 Group Key Handshake

Change the text of the first 3 paragraphs included the itemized list as follows:

The Authenticator uses the Group Key Handshake to send a new GTK, ~~IGTK,~~ and ~~DHV~~ 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_{GTK}], IGTK[N_{IGTK}]~~, DHV~~)
>
> Message 2: Supplicant ← Authenticator: EAPOL-Key(1,1,0,0,G,0,0,MIC,0,0,0)

Here, the following assumptions apply:

- Key RSC denotes the last frame sequence number sent using the GTK.
- ~~GTK[N_{GTK}] 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~~GTK[~~N_{IGTK}~~N_{GTK}] denotes the ~~IGTK~~ GTK encapsulated with its key identifier as defined in 8.5.2 using the KEK defined in 8.5.1.2 and associated IV.
- ~~DHV~~ IGTK[N_{IGTK}] denotes the ~~DHV~~ 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)
- IGTK, IGTK's key identifier, and sequence number (see 8.5.2)
- ~~DHV (see 8.5.2)~~

Change item 'c' in 8.5.4.1 as follows:

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

8.5.4.2 Group Key Handshake Message 2

8.5.4.3 Group Key Handshake implementation considerations

Change the second paragraph as follows:

The state machines in 8.5.5 and 8.5.6 change the GTK, ~~IGTK,~~ and ~~DHV~~ IGTK in use by the network. See Figure 152**.**

Change the last paragraph and its numbered list as follows:

The following steps occur:

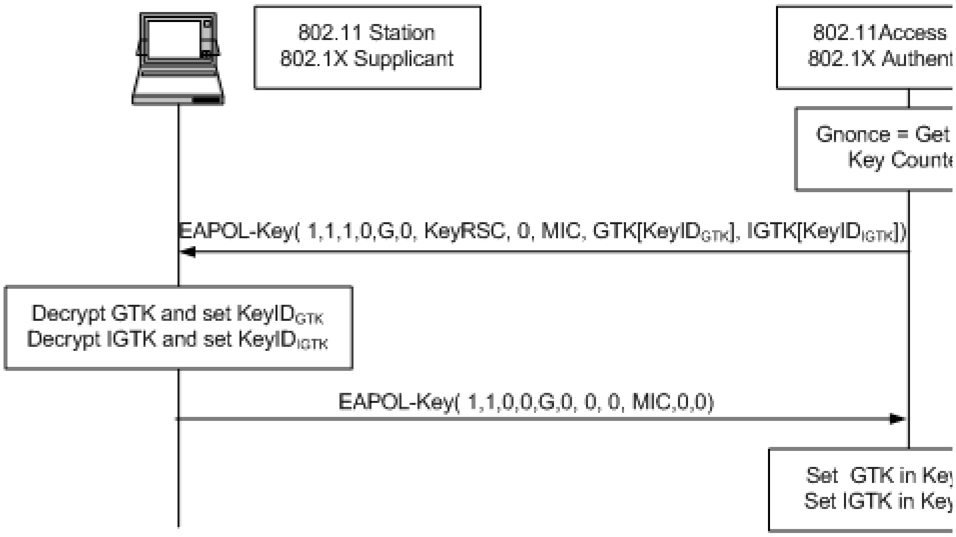
a) The Authenticator generates a new GTK, and IGTK~~, and DGTK. It derives a DHV from that DGTK~~. It encapsulates the GTK, ~~IGTK,~~ and ~~DHV~~ IGTK and sends an EAPOL-Key frame containing the GTK, ~~IGTK,~~ and ~~DHV~~ IGTK (Message 1), along with the last sequence number used with the GTK (RSC) and the last sequence number used with the IGTK (PN).

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

Replace Figure 152 with the following Figure, with the updated including IGTK ~~and DHV~~ on the first EAPOL-Key message and 2nd supplicant box, adding "0,0" on the second EAPOL-Key message and IGTK and new subscripts to the last Authenticator box:


802.11 Station
802.1X Supplicant
802.11Access
802.1X Authent
Gnonce = Get
Key Count
EAPOL-Key(1,1,1,0,G,0, KeyRSC, 0, MIC, GTK[KeyID_{GTK}], IGTK[KeyID_{IGTK}])
Decrypt GTK and set KeyID_{GTK}
Decrypt IGTK and set KeyID_{IGTK}
EAPOL-Key(1,1,0,0,G,0, 0, 0, MIC,0,0)
Set GTK in Key
Set IGTK in Key

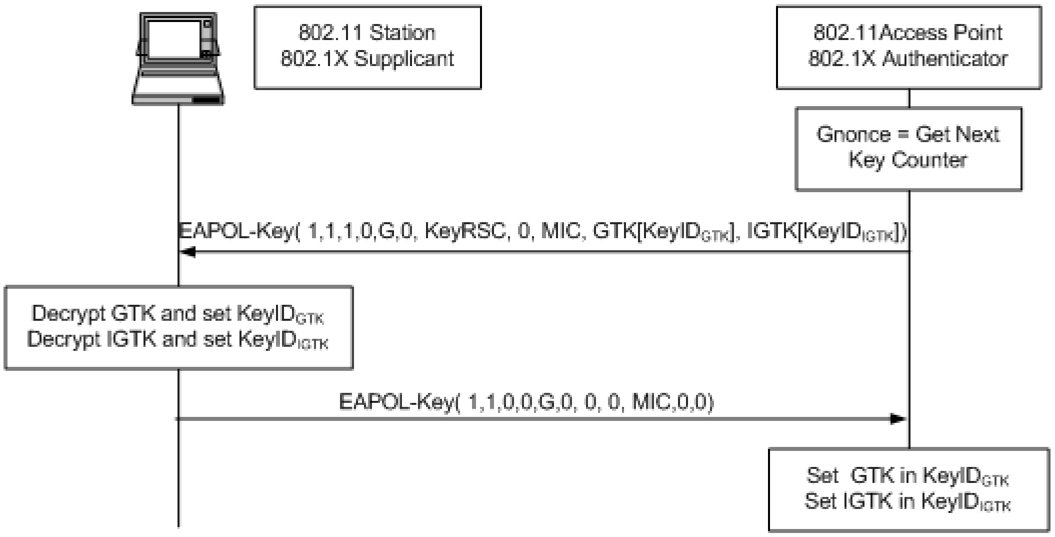

802.11 Station
802.1X Supplicant
802.11Access Point
802.1X Authenticator
Gnonce = Get Next Key Counter
EAPOL-Key(1,1,1,0,G,0, KeyRSC, 0, MIC, GTK[KeyID$_{GTK}$], IGTK[KeyID$_{IGTK}$])
Decrypt GTK and set KeyID$_{GTK}$
Decrypt IGTK and set KeyID$_{IGTK}$
EAPOL-Key(1,1,0,0,G,0, 0, 0, MIC,0,0)
Set GTK in KeyID$_{GTK}$
Set IGTK in KeyID$_{IGTK}$

Figure 152—Sample Group Key Handshake

8.5.5 RSNA Supplicant key management state machine

8.5.5.1 Supplicant state machine states

Replace Figure 153 with the following figure adding "IGTK[0...N] = 0" in the "AUTHENTICATION" box and "MLME-DeleteKeysRequest(IGTK[0...N])" to the "INITIALIZE" box:

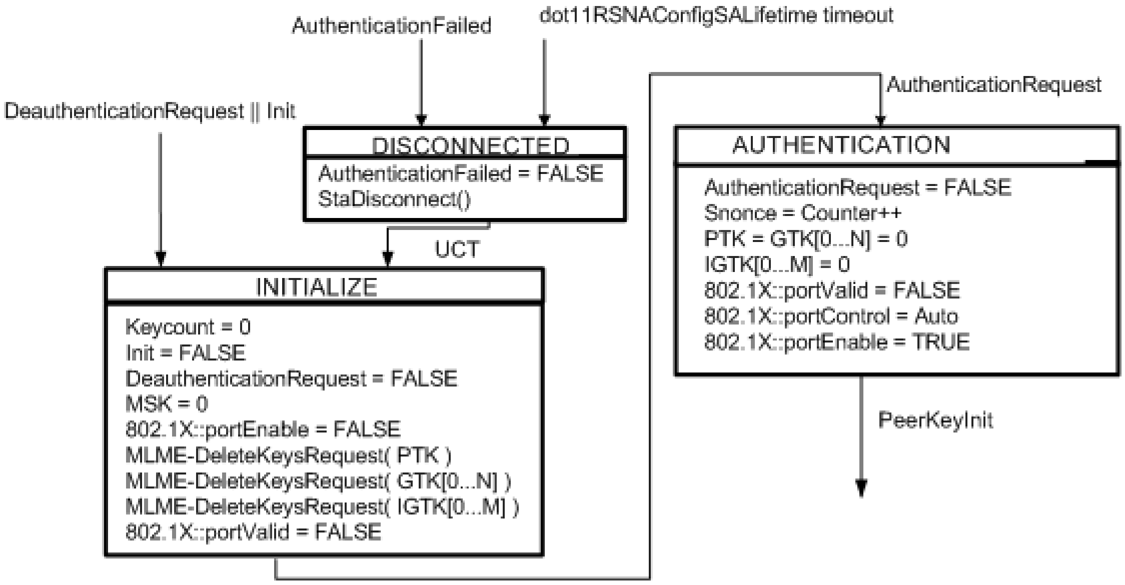

AuthenticationFailed
dot11RSNAConfigSALifetime timeout
AuthenticationRequest
DeauthenticationRequest || Init
DISCONNECTED
AuthenticationFailed = FALSE
StaDisconnect()
UCT
AUTHENTICATION
AuthenticationRequest = FALSE
Snonce = Counter++
PTK = GTK[0...N] = 0
IGTK[0...M] = 0
802.1X::portValid = FALSE
802.1X::portControl = Auto
802.1X::portEnable = TRUE
PeerKeyInit
INITIALIZE
Keycount = 0
Init = FALSE
DeauthenticationRequest = FALSE
MSK = 0
802.1X::portEnable = FALSE
MLME-DeleteKeysRequest(PTK)
MLME-DeleteKeysRequest(GTK[0...N])
MLME-DeleteKeysRequest(IGTK[0...M])
802.1X::portValid = FALSE

Figure 153—RSNA Supplicant key management state machine

8.5.5.2 Supplicant state machine variables

Insert the following ~~test~~ text immediately following the 'GTK[]' variable:

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

~~— DGTK - This variable represents the current DGTK.~~

8.5.5.3 Supplicant state machine procedures

Change the 'StaprocessEPOL-Key' item as follows:

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

StaProcessEAPOL-Key (S, M, A, I, K, RSC, ANonce, RSC, MIC, RSNIE, GTK[N_{GTK}], IGTK[N_{IGTK}],PN~~, DGTK~~)

```
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[$N_{GTK}$] ← Decrypt GTK) succeeds then
                if MLME-SETKEYS.request(NGTK, 0, RSC, GTK[NGTK]) fails then
                    invoke MLME-DEAUTHENTICATE.request
                endif
                else
                    State ← FAILED
                endif
        endif
        if IGTK then
            if (IGTK[$N_{IGTK}$] ← Decrypt IGTK) succeeds then
                if MLME-SETKEYS.request($N_{IGTK}$, 0, PN, IGTK[$N_{IGTK}$]) fails then
                    ~~invoke MLME-DEAUTHENTICATE.request~~
                ~~endif~~
                ~~else~~
                    ~~State ← FAILED~~
                ~~endif~~
            ~~endif~~
            ~~if DGTK then~~
                ~~if (DGTK ← Decrypt DGTK) succeeds then~~
                    ~~if MLME-SETKEYS.request(DGTK) fails then~~
                    invoke MLME-DEAUTHENTICATE.request
                endif
                else
                    State ← FAILED
                endif
            endif
            endif
        endif
else if KeyData = GTK then
        if State = MICOK then
            if (GTK[$N_{GTK}$] ← Decrypt GTK) succeeds then
                if MLME-SETKEYS.request($N_{GTK}$, T, RSC, GTK[$N_{GTK}$]) fails then
                    invoke MLME-DEAUTHENTICATE request
                endif
                else
                    State ← FAILED
                endif
                if (IGTK[$N_{IGTK}$] ← Decrypt IGTK) succeeds then
                    if MLME-SETKEYS.request(NIGTK, T, PN, IGTK[NIGTK]) fails then
                        ~~invoke MLME-DEAUTHENTICATE request~~
                    ~~endif~~
                    ~~else~~
                        ~~State ← FAILED~~
                    ~~endif~~
                    ~~if (DGTK ← Decrypt DGTK) succeeds then~~
                        ~~if MLME-SETKEYS.request(DGTK) fails then~~
                        invoke MLME-DEAUTHENTICATE request
                    endif
                    else
                        State ← FAILED
                    endif
                    else
                        State ← FAILED
                    endif
```

Change the second paragraph succeeding the pseudocode as follows:

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

Insert the following items at the end of 8.5.5.3:

- ~~**DecryptIGTK(x)** - Decrypt the IGTK from the EAPOL-Key frame.~~
- **~~DecryptDGTK~~DecryptIGTK(x)** - Decrypt the ~~DGTK~~ IGTK from the EAPOL-Key frame.

8.5.6 RSNA Authenticator key management state machines

Replace Figure 155 with the following Figure, with the updates being the insertion of "IGTK[M]" in the PTKINITNEGOTIATING box:

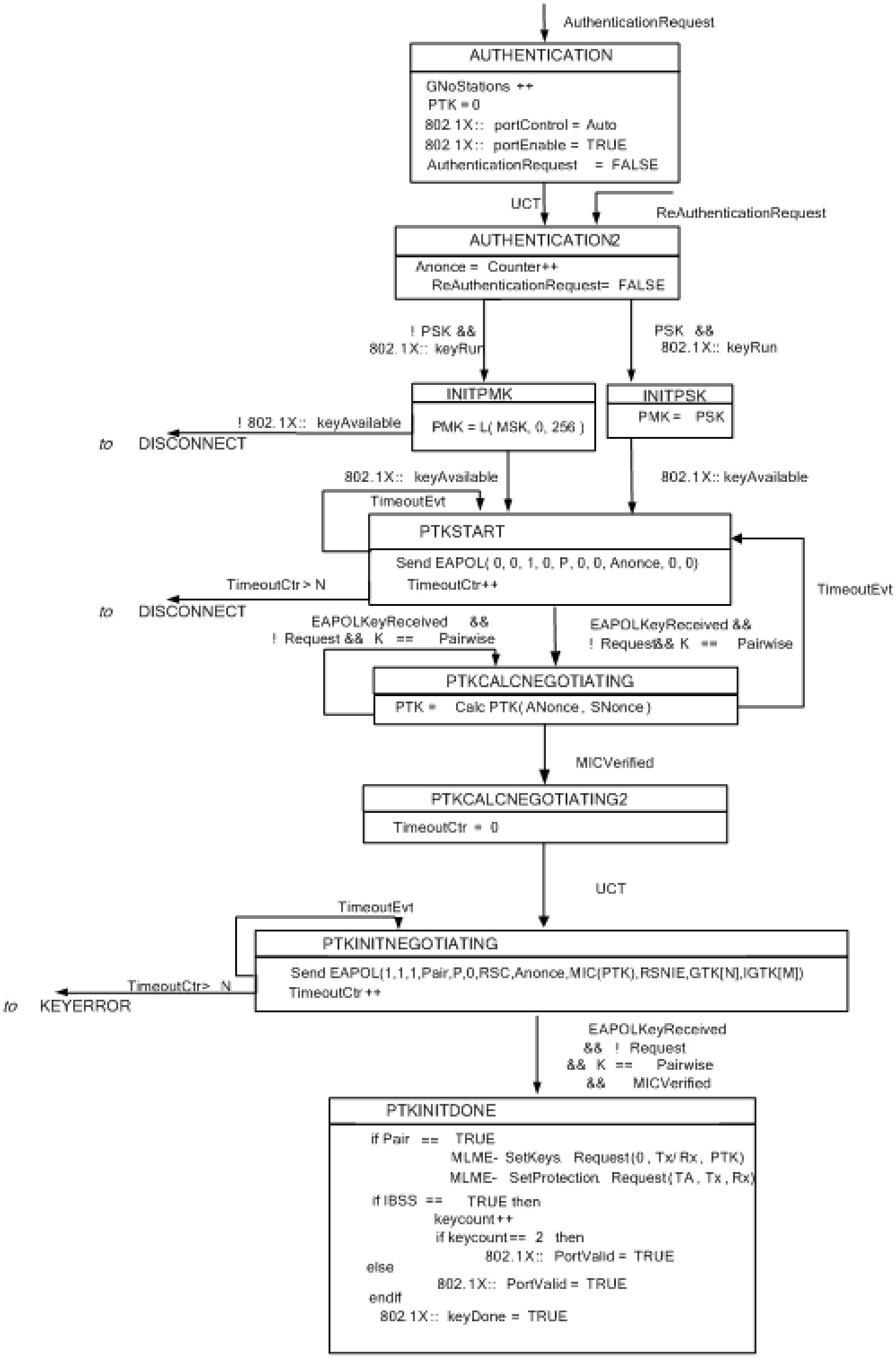


Figure 155—Authenticator state machines, part 1

8.6 Mapping EAPOL Keys to IEEE 802.11 keys

~~8.6.1 Mapping PTK to TKIP keys~~

~~*Change the 3rd and 4th paragraphs as follows:*~~

~~A STA shall use bits 128-191 of the temporal key as the Michael key for MSDUs or MMPDUs from the Authenticator's STA to the Supplicant's STA.~~

~~A STA shall use bits 192-255 of the temporal key as the Michael key for MSDUs or MMPDUs from the Supplicant's STA to the Authenticator's STA.~~

8.7 Per-frame pseudo-code

8.7.1 WEP frame pseudo-code

8.7.2 RSNA frame pseudo-code

Change the paragraph as follows:

STAs transmit protected MSDUs or MMPDUs to an RA when temporal keys are configured and an MLME.SETPROTECTION.request primitive has been invoked with ProtectType parameter Tx or Rx_Tx to that RA. STAs expect to receive protected MSDUs or MMPDUs from a TA when temporal keys are configured and an MLME.SETPROTECTION.request primitive has been invoked with ProtectType parameter Rx or Rx_Tx to that TA. MSDUs and MMPDUs that do not match these conditions are sent and are received without the benefit of encryption.

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 Bit 6 of RSNA Capability Field is set then
        if MMPDU has an individual RA and Protection for RA is off for Tx then
            transmit the MMPDU without protections
        else if (MMPDU has individual RA and Pairwise key exists for the MMPDU's RA) or (MMPDU has
        a multicast or broadcast RA and network type is IBSS and IBSS GTK exists for MMPDU's TA)
        then
            // If we find a suitable Pairwise or GTK for the mode we are in…
        if key is a null key then
            discard the entire MMPDU
        else
            // Note that it is assumed that no entry will be in the key
            // mapping table of a cipher type that is unsupported.
            Set the Key ID subfield of the IV field to zero.
                if cipher type of entry is AES-CCM then
                    Transmit the MMPDU, to be protected after fragmentation using AES-CCM
                else if  cipher type of entry is AES-128-CMAC then
                    Transmit the MMPDU with BIP
                endif
        endif
        else // Else we didn't find a key but we are protected, so handle the default key case or discard
            if IGTK entry for Key ID contains null then
                discard the MMPDU
            else if IGTK entry for Key ID is not null then
                Set the Key ID subfield of the IV field to the Key ID.
                if MMPDU has an individual RA and cipher type of entry is not TKIP then
                    discard the entire MMPDU
                else if cipher type of entry is AES-CCM then
                    Transmit the MMPDU, to be protected after fragmentation using AES-CCM
                endif
            endif
        endif
```

Insert a new subclause after 8.7.2.2:

8.7.2.2a Per-MPDU Tx pseudo-code for MMPDU

```
if dot11RSNAEnabled = TRUE and Bit 6 of RSNA Capability Field is set then
    if MMPDU is member of an MMPDU that is to be transmitted without protections then
        transmit the MPDU without protections
    else if MMPDU is to be protected using AES-CCM then
        Protect the MPDU using entry's key and AES-CCM
        Transmit the MPDU
    else
    // should not arrive here
    endif
endif
```

Insert a new subclause after 8.7.2.3 :

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

```
if dot11RSNAEnabled = TRUE and Bit 6 of RSNA Capability Field is set then
    if  the Protected Frame subfield of the Frame Control Field is zero then
        if Protection for TA is off for Rx then
            Receive the unencrypted MPDU without protections
        else
            Discard the frame body without indication to LLC
        endif
    else if Protection is true for TA then
        if ((MPDU has individual RA and Pairwise key exists for the MPDU's TA) or (MPDU
        has a broadcast/multicast RA and network type is IBSS and IBSS GTK exists for
        MPDU's RA)) then
```

```
            if key is null then
                discard the frame body
            else if entry has an AES-CCM key then
                decrypt frame using AES-CCM key
                discard the frame if the integrity check fails and increment
                dot11RSNAStatsCCMPDecryptErrors
            else if entry has a AES-128-CMAC key then
                check integrity of the frame using AES-128-CMAC key
                discard the frame if the ICV fails and increment dot11CMACICVErrors
            else
                discard the frame body
            endif
        else if GTK for the Key ID does not exist then
            discard the frame body
        else if GTK for the Key ID is null then
            discard the frame body
        else if the GTK for the Key ID is a CCM key then
            decrypt frame using AES-CCM key
            discard the frame if the integrity check fails and increment
            dot11RSNAStatsCCMPDecryptErrors
        else if the IGTK for the Key ID is a AES-128-CMAC key then
            integrity check the frame using AES-128-CMAC decryption
            discard the frame if the ICV fails and increment dot11CMACICVErrors
        endif
    else
        discard the frame body
    endif
endif
```

Insert a new subclause after 8.7.2.4 :

8.7.2.4a Per-MMPDU Rx pseudo-code

```
if dot11RSNAEnabled = TRUE and Bit 6 of RSNA Capability Field is set then
    if the frame was not protected then
    Receive the MMPDU unprotected
    else// Have a protected MMPDU
        if Pairwise key is an AES-CCM key then
            Accept the MMPDU if its MPDUs had sequential PNs (or if it consists of only
            one
            MPDU), otherwise discard the MMPDU as a replay attack and increment
            dot11RSNAStatsCCMPReplays
        else if  Pairwise key is a AES-128-CMAC key then
            Accept the MMPDU if its MPDUs had sequential PNs (or if it consists of only
            one
            MPDU), otherwise discard the MMPDU as a replay attack and increment
            dot11RSNAStatsCMACReplays
        endif
    endif
endif
```

9. MAC Sublayer functional description

10. Layer Management

11. MLME

EDITORIAL NOTE: TGn has added sections up through 11.17, TGw succeeds with 11.18.

Insert at the end of Clause 11 a new subclause as follows:

11.18 Broadcast and multicast Frame procedures

When Robust Management frame protection is enabled, the MLME shall provide an encapsulation service for robust broadcast/multicast management frames. All Robust management frames shall be submitted to this service for encapsulation and transmission.

The broadcast/multicast frame protection service shall take the following actions:

- ~~All broadcast/multicast deauthentication and disassociation frames shall be protected using BIP with the Broadcast/Multicast Disconnect Hash value mechanism.~~
- For Robust broadcast/multicast ~~action~~ action, disassociate and deauthenticate management frames, the broadcast/multicast management frame shall be encapsulated and protected using BIP.

The broadcast/multicast frame protection service shall be used by other services internal to the MLME layer and shall take as an input the body of the frame as described in 7.2.3.12. The destination address is assumed to be the broadcast/multicast address. The service shall return a status result indicating acceptance of the frame for processing or rejection of the frame.

Annex A

A.1 Introduction

A.2 Abbreviations and special symbols

A.3 Instructions for completing the PICs proforma

A.4 PICS proforma - IEEE Std 802.11, 2006 Edition

A.4.1 Implementation identification

A.4.2 Protocol summary

A.4.3 IUT configuration

A.4.4 MAC protocol

A.4.4.1 MAC protocol capabilities

Insert the following row to end of table in A.4.4.1 as a subset of the RSN ~~with the appropriate number for 'XX' as follows~~:

PCX 34.1.~~XX~~10	Protection Management Frame	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	~~PCS~~PCX34.1:O	Yes No

EDITORIAL NOTE: The entry value is shown as PCX 34.1.10 but its final value is pending ANA assignment

Annex D

(normative)

ASN.1 encoding of the MAC and PHY MIB

Insert the following at the end of the `Dot11StationConfigEntry` ***in Annex D:***

```
dot11RSNAProtectedManagementFramesEnabled       TruthValue
dot11RSNABcastProtectedManagementFramesEnabled        TruthValue
```

Insert the following after the dott11RSNAStats TABLE entries in Annex D:

```
--**********************************************************
--* Robust Management frame protection MIBs
--**********************************************************

dot11RSNAProtectedManagementFramesEnabled       OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
               "This variable indicates whether or not this STA
                Protects unicast Management Frames."
                  DEFAULT { TRUE }
                  ::= { dot11StationConfigEntry ~~27 ~~TBD }
```

EDITORIAL NOTE: The entry value is left as TBD for now, pending ANA assignment

```
dot11RSNABcastProtectedManagementFramesEnabled        OBJECT-TYPE
               SYNTAX TruthValue
               MAX-ACCESS read-write
               STATUS current
               DESCRIPTION
               "This variable indicates whether or not this STA
               protects broadcast/multicast management Frames.
                  DEFAULT { TRUE }
                  ::= { dot11StationConfigEntry ~~28 ~~TBD }

~~Dot11RSNAProtectedManagementBroadcastPolicyOBJECT_TYPE~~
               ~~SYNTAX INTEGER { bip(1), mut(2) }~~
               ~~MAX-ACCESS read-write~~
               ~~STATUS current~~
               ~~DESCRIPTION~~
               ~~"This variable sets the policy of an access point for~~
               ~~the transmission of broadcast or multicast management~~
                ~~action frames. The variable is disregarded in non-AP~~
                ~~STAs and IBSS STAs"~~
```

EDITORIAL NOTE: The entry value is left as TBD for now, pending ANA assignment

```
dot11RSNALegacyManagementFrames                 OBJECT-TYPE
      SYNTAX TruthValue
      MAX-ACCESS read-write
      STATUS current
      DESCRIPTION
            "This variable indicates whether or not this STA      sup-
            ports robust RSNA non-AP STAs which do not provide robust
            management frames protection.”
```

```
                DEFAULT { FALSE }
                ::= { dot11StationConfigEntry ~~29~~ TBD}
```

EDITORIAL NOTE: The entry value is left as TBD for now, pending ANA assignment

```
dot11RSNAStatsCMACICVErrors               OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of received MPDUs discarded by the CMAC integ-
            rity checking algorithm."
                ~~DEFAULT { FALSE }~~
                ::= { ~~dot11StatsEntry~~ 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."
                ~~DEFAULT { FALSE }~~
                ::= { ~~dot11StatsEntry~~ dot11RSNAStatsEntry 12 }

~~dot11RSNAStatsTKIPHdrErrors~~ dot11RSNAStatsBIPReplays OBJECT-TYPE
             SYNTAX Counter32
             MAX-~~ACCESS~~ ACESS read-only
         ~~    STATUS current~~
             ~~DESCRIPTION~~STATUS current
            ~~"Counts the number of TKIP header errors in robust manage-~~
            ~~ment frames."~~
            ~~DEFAULT     {FALSE}~~
    ~~::= { dot11RSNAStatsEntry 13}~~DESCRIPTION:

~~dot11RSNAStatsBIPReplays OBJECT-TYPE~~
         ~~    SYNTAX Counter32~~
         ~~    MAX-ACESS read-only~~
         ~~    STATUS current~~
~~    DESCRIPTION:~~
            "The number of received BIP frames discarded due to dupli-
            cate or old sequence numbers"
             ::= {dot11RSNAStatsEntry 14}

~~dot11RSNAStatsBroadcastDHVMismatches OBJECT-TYPE~~
         ~~    SYNTAX Counter32~~
         ~~    MAX-ACESS read-only~~
         ~~    STATUS current~~
         ~~DESCRIPTION:~~
            ~~"The number of received broadcast disassociate or deauthenticate frames discarded due to invalid DGTKs"~~
         ~~::= {dot11RSNAStatsEntry 15}~~
--**********************************************************
--* End of Robust Management Frame MIB
--**********************************************************
```

Append to the dot11RSNAStatsEntry Sequence the following:

```
dot11RSNAStatsCMACICVErrors               Counter32,
dot11RSNAStatsCMACReplays                 Counter32,
~~dot11RSNAStatsTKIPHdrErrors~~dot11RSNAStatsBIPReplaysCounter32~~,~~
~~dot11RSNAStatsBIPReplays~~                  ~~Counter32,~~
~~dot11RSNAStatsBroadcastDHVMismatches~~      ~~Counter32~~
```