IEEE P802.11
Wireless LANs

IEEE 802.1X Pre-Authentication

Date: May 14, 2002

Authors: **Bernard Aboba**
Microsoft
One Microsoft Way, Redmond, WA 98052-6399
Phone: +1 425-706-6605
E-mail: bernarda@microsoft.com

Abstract

This paper describes the architectural implications of IEEE 802.1X pre-authentication. Starting from a threat model, and an explicit set of goals and objectives, this paper describes some of the issues surrounding IEEE 802.1X pre-authentication. These include advertisement of capabilities, integration between with the 802.1X and 802.11 state machines, encapsulation of 802.1X pre-authentication data frames, secure ciphersuite negotiation, authentication and integrity protection of management frames and key establishment. For each issue, potential solutions are enumerated and evaluated, and relationships with other aspects of the problem are described. The overall evaluation is that IEEE 802.1X pre-authentication appears both feasible and desirable.

1. Introduction

1.1. Overview of the architecture

Allowing IEEE 802.1X authentication to occur prior to association has a considerable effect on the Robust Security Network (RSN) architecture. This document examines the implications of IEEE 802.1X pre-authentication, analyzing the design tradeoffs and recommending solutions. Problems addressed by this specification include:

- Capabilities advertisement. This addresses how IEEE 802.1X pre-authentication, ciphersuite support, etc. are advertised. Assuming support for protected management frames, it is not necessary for Beacons or Probe Request/Response messages to be authenticated and integrity protected. This is discussed in Section 2.
- Secure state machine interlock. This addresses how IEEE 802.1X is securely integrated within the 802.11 state machine. Since the original IEEE 802.11 specification supports pre-authentication, the existing 802.11 state machine can be utilized without modification. This is discussed in Section 3.
- Low roaming latency. With IEEE 802.1X pre-authentication, it is possible for STAs to authenticate prior to association. This enables a reduction in the period of connectivity loss during roaming in some, though not all situations. In an RSN-capable wireless LAN, two types of pre-

authentication are possible. In “unassociated pre-authentication”, STA A pre-authenticates to STA B while it is unassociated to any STA (State 1 in the 802.11 state machine). In “associated pre-authentication” STA A pre-authenticates to STA B while it is both authenticated and associated to STA C (State 3 in the 802.11 state machine). The implications of each approach, including security vulnerabilities, are described in Section 4.

- Secure ciphersuite negotiation. This problem relates to how the desired ciphersuite may be selected, and how both the available ciphersuites and the selection can be determined to be authentic. This paper explores several approaches to secure ciphersuite negotiation, including support within EAP, support within the 4-way key handshake, or support within an authenticated Association/Reassociation exchange. This is discussed in Section 5.
- Protected control and management traffic. This problem relates to how associations are established and terminated, and how to secure control frames as well as Association Request/Response, Reassociation Request/Response, Disassociate, and Deauthenticate frames. Deriving keying material prior to association makes this possible, and improves resistance to denial of service attacks. This paper describes two approaches to protection of management and control traffic, one involving use of the existing TKIP and AES ciphers, and another approach involving addition of a message integrity check (MIC). Issues with the current design of TKIP and AES are described. This is discussed in Section 6.
- Key establishment and sychronization. This problem relates to how key state is established between STAs, the key hierarchy, and how keys are guaranteed to be fresh. The architecture in this document supports secure key derivation as well as synchronized activation of keys between two STAs. This is accomplished using a 4-way key handshake in concert with an authenticated association/reassociation exchange. This is discussed within Section 7.

1.2. Threat model

In order to understand whether security objectives have been met, and evaluate alternative proposals, a threat model is required. The threat model for an RSN is described below.

IEEE 802.11 is used to transmit data, authentication and control/management traffic over wireless LANs. Therefore the data, authentication and control/management traffic is vulnerable to attack. Examples of attacks include:

[1] An adversary attempting to acquire confidential data and identities by snooping data packets.

[2] An adversary attempting to modify packets containing data, authentication or control/management messages.

[3] An adversary attempting to inject packets into an 802.11 conversation, including data, authentication or control/management traffic.

[4] An adversary attempting to hijack an 802.11 conversation, including data, authentication or control/management traffic.

[5] An adversary attempting to deny service to 802.11 stations or access points.

[6] An adversary attempting to disrupt the security negotiation process, in order to weaken the authentication, or gain access to user passwords. This includes submission of inauthentic capability advertisements, disruption of the ciphersuite negotiation, or disruption of the IEEE 802.1X authentication conversation.

[7] An adversary attempting to impersonate a legitimate 802.11 Station or Access point.

1.3. Security Protocol

To address the above threats, an RSN MUST provide confidentiality, data origin authentication, integrity, and replay protection on a per-packet basis for data traffic. This is accomplished through the introduction of two new ciphers: TKIP and a cipher based on AES. Confidentiality services are important for IEEE 802.11 data traffic since wireless LANs are inherently vulnerable to snooping.

Per-packet data origin authentication, integrity and replay protection is required for control and management traffic. Confidentiality is not a requirement for control or management traffic, since this traffic does not ordinarily provide information valuable to an attacker. Negotiation of the ciphersuite and authentication methods MUST be authenticated and integrity protected so as to prevent subversion of these negotiations.

Requirements for EAP authentication methods used with IEEE 802.1X and 802.11 include the following:

- Mutual authentication. Mutual authentication of the communication endpoints MUST be provided.
- Key derivation. Authentication methods MUST derive keys in order to enable per-packet authentication, integrity and replay protection as well as confidentiality. The key derivation MUST be accomplished in a manner capable of providing a Pairwise Master Key (PMK) to both the Supplicant and Authenticator.
- Dictionary attack resistance. The authentication method SHOULD provide resistance against offline dictionary attack. Where password authentication is used, users are notoriously prone to selection of poor passwords. Without dictionary attack protection, it is easy for an attacker snooping authentication traffic at a popular location to gather a large number of authentication exchanges, and successfully obtain a substantial fraction of the passwords used in those

exchanges via an offline dictionary attack. Given the steadily declining prices of computing power, successful dictionary attacks can now be mounted at minimal expense.

- Support for fast reconnect. Since IEEE 802.1X pre-authentication permits a STA to authenticate to multiple STAs while associating to only one STA, it potentially increases the load on the backend authentication server, if present. In order to improve scalability, it is highly desirable for EAP methods used with 802.1X and 802.11 to support "fast reconnect", enabling caching of authentication credentials and shortening of the authentication conversation.
- Protected EAP conversation. An important objective of the RSN architecture is to provide protection for secure negotiations, including protected ciphersuite and authentication negotiation, as well as secure communication of the result of the authentication conversation. As is described in Section 5, protected ciphersuite negotiation can be provided via protection of 802.11 management traffic (Association/Reassociation). Within this specification, completion of the protected Association/Reassociation exchange also serves to signal key activation, implying that "unassociated" IEEE 802.1X pre-authentication (described in Section 4) is carried out without the protection of 802.11 ciphersuites. As a result, EAP authentication methods used with 802.11 SHOULD provide for the authentication, integrity and replay protection of the EAP conversation, including the Identity, Nak and Notification types, and success and failure indications.

These requirements apply both to authentication within an ESS and an IBSS.

1.4. The Robust Security Network

A Robust Security Network provides a number of additional security features not present in the basic IEEE 802.11 architecture. These features notably include:

- enhanced authentication mechanisms for both APs and STAs;
- key management algorithms;
- protection of management and control frames;
- secure ciphersuite negotiation;
- dynamic, association-specific cryptographic keys; and
- enhanced data encapsulation mechanisms, known as TKIP and AES Privacy.

An RSN makes use of protocols above the IEEE 802.11 MAC sub layer to provide the authentication and key management. This provides added flexibility by allowing authentication and key management functionality to be updated without requiring modifications to the IEEE 802.11 MAC sub layer.

An RSN introduces several new components into the IEEE 802.11 architecture that are not present in non-RSN systems.

The first new component is the ***802.1X port access entity (PAE)***. Within an RSN, IEEE 802.1X pre-authentication is used to establish authentication and key state prior to association. This is accomplished as the result of a conversation between the 802.1X Supplicant PAE and the 802.1X Authenticator PAE. However, as an RSN only employs IEEE 802.1X for the purposes of pre-authentication, the 802.11 state machine determines which frames may be accepted in which states, and therefore when used with 802.11, there is no notion of 802.1X controlled and uncontrolled ports.

A second (optional) component is the ***backend Authentication Server*** (AS). The AS is an entity that resides in the DS that may participate in the authentication of STAs (including APs) in the ESS. The backend authentication server may authenticate STAs and APs—or it may provide material that the RSN elements can use to authenticate each other. The AS communicates with the Authenticator on each STA, enabling the STA to be authenticated to the ESS and *vice versa*. Mutual authentication of both the ESS and the STA is an important goal of the RSN.

Figure 1 depicts some of the relationships among these components.

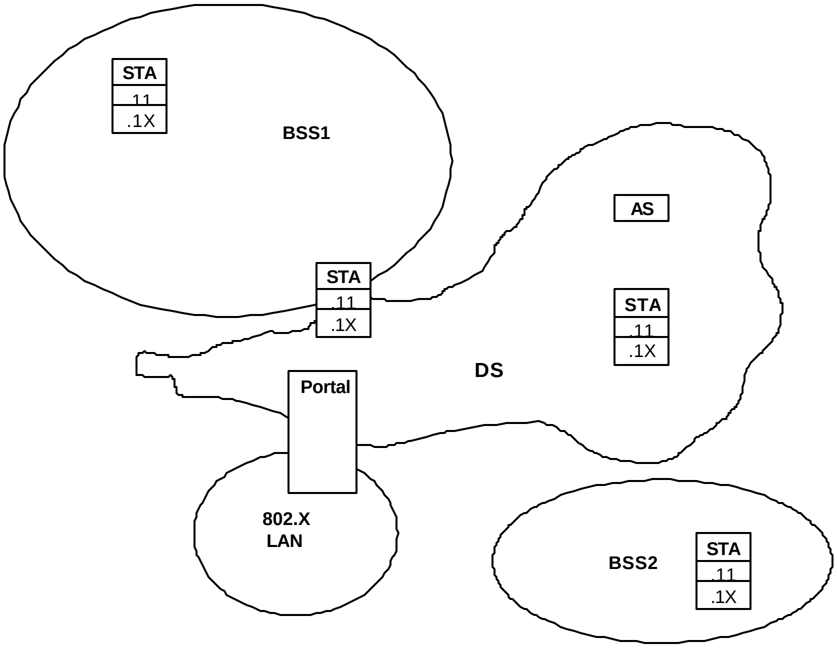

Figure 1: A robust security network (RSN)

2. RSN capability advertisement

STAs determine that RSN is available via an information element contained within the Beacon, and Probe Response frames. The Beacon and Probe Response frames advertise what the AP is capable of doing, and the protected association or reassociation request frame contains what the station is requesting for its association. The Probe Response message contains what an IBSS station is capable of doing. Advertisement of RSN capability is assumed to imply and require support for both “unassociated” and “associated” IEEE 802.1X Pre-authentication. Pre-authentication is described in Section 4.

This document does not propose that Beacons or Probe Request/Response frames be protected for several reasons:

- Since Beacons and Probe Request/Response frames may be sent and received prior to authentication, dynamic keying material may not exist with which to protect these frames.
- Since Beacons are broadcast frames, the default key would need to be used to provide protection. This provides no assurance against spoofing of Beacon frames by STAs that have already authenticated and obtained the default key.
- Since STAs use Beacon and Probe Response frames in order to discover the existence and capabilities of other STAs, encrypting Beacons and Probe Responses would creates a circular dependency: it is not possible to pre-authenticate without learning the capabilities of the peer STA, and the capabilities cannot be learned until pre-authentication is complete and keys are available.
- By including the information present in Beacons and Probe Responses within the protected Association/Reassociation exchange, it is possible to confirm the authenticity of Beacons and Probe Responses, without having to protect those messages.

Not protecting Beacons and Probe Responses does introduce potential security vulnerabilities:

- By not encrypting Beacons and Probe Responses, an attacker can determine the capabilities of RSN wireless LANs. This is useful, for example, in discovering networks with known security vulnerabilities, so that they can be attacked. Turning off Beaconing does not help very much, since attackers can still monitor Probe Response messages, which contain the same information. As a result, the best defense against this vulnerability is to make sure that the RSN configuration is secure. This includes avoiding use of both WEP and RSN on the same wireless LAN.
- Were a spoofed Beacon or Probe Response to omit advertisement of RSN capability, a STA might be fooled into believing that an RSN-capable AP lacked RSN capability, thereby negotiating a lower level of security. Since non-RSN wireless LANs do not support protected management frames, forged Beacons or

Probe Responses would not be detected. To address this vulnerability, it is recommended that after a suitable transition period, RSN-capable APs SHOULD be configured to accept only Association/Reassociation frames indicating RSN support, and RSN-capable STAs SHOULD be configured by default to ONLY associate with APs that advertise RSN capabilities.

RSN Information Element

The RSN Information Element contains a list of authentication and unicast cipher suite selectors, a single multicast cipher suite selector and whether unicast keys are supported. No additional capabilities are included in order to determine the algorithms used for protection of management frames. All STAs implementing RSN shall support this element:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Element ID |    Length     |           Version             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Multicast ciphersuite                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Unicast suite count      |    Authentication suite count |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Unicast suite 1                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        Unicast suite n                        ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Authentication suite 1                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Authentication suite n                     ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2 RSN Element format

Element ID

The Element ID field is a single octet. The RSN capability advertisement element ID is 37 decimal (0x25 Hex)

Length

The length field is one octet. It represents the length of the information element following normal IEEE 802.11 information element rules.

Version field

The version field is two octets. It represents the the version number of the RSN.

- It is expected that the station and AP/station may support a range of versions but they must support a contiguous range of versions.

- The AP and station shall advertise the highest version it supports.
- A station shall request for the highest version it supports that is lower or equal to the version the AP/station is advertising.
- If the AP/station is advertising a lower version than the station supports the station shall not authenticate with or associate to the AP/station.
- If the version from a station is outside the range the AP/station supports, the AP/station shall send an authenticated disassociation frame and/or a deauthenticate message (authenticated or unauthenticated, depending on whether authentication and key state has been previously established) to the station. Otherwise the AP/station shall adapt to the version specified by the station.

Version 1 specifies the following requirements:

1. RSN information element. An AP/station supporting RSN shall put the RSN information element in Beacons and Probe Responses. A station supporting RSN shall put the RSN information element in authenticated association/reassociation requests and responses.
2. TKIP encryption cipher. An AP and station shall support TKIP encryption.
3. Michael integrity check. An AP and station shall support the Michael integrity check.
4. Key updates using EAPOL-Key descriptor from this document.

A suite selector has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     OUI                       |   Suite       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The order of the OUI field follows the ordering convention for MAC addresses from IEEE 802.11 7.1.1. For example, for an OUI of 010203 then the OUI field will appear as follows:

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Address + 0  | Address + 1   | Address + 2   |
|  (0x01)       |   (0x02)      |   (0x03)      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Table 1 – Authentication Suite Selectors

OUI	Type	Meaning
00:00:00	0	None
00:00:00	1	Unspecified authentication over 802.1X: default for RSN
00:00:00	2	Pre-shared Key over 802.1X
00:00:00	3-255	Reserved
Other	Any	Vendor Specific

The authentication suite selector value 00:00:00:1 “Unspecified authentication over IEEE 802.1X” implies support for both “unassociated” and “associated” IEEE 802.1X pre-authentication. This is the assumed default when this information is not supplied. A station shall ignore any values it does not recognize.

The authentication suite selector value 00:00:00:2 “Pre-shared key over 802.1X” is used when a pre-shared key is used with 802.1X.

In IBSS mode RSN only supports 00:00:00:0 “None”. This means that RSN encryption and integrity is supported but authentication and key management is not supported.

Note: The inclusion of different Authentication types allows the simplification of the User Interface. It allows the pre-shared key UI to be enabled/disabled on stations depending on the configuration of the AP so users are only asked for the information that is required for any particular scenario. Only one of “Unspecified authentication over 802.1X” or “Pre-shared key over 802.1X” is allowed in an RSN information element, i.e. both authentication suit selectors cannot be in an RSN information element at the same time.

Table 2 – Cipher Suite Selectors

OUI	Type	Meaning
00:00:00	0	None
00:00:00	1	WEP
00:00:00	2	TKIP
00:00:00	3	Reserved for AES cipher: default for RSN
00:00:00	4-255	Reserved
Other	Other	Vendor Specific

The cipher suite selector 00:00:00:3 “AES” is the implied default cipher suite value when this information is not supplied.

The cipher suite selector 00:00:00:1 “WEP” is only valid for multicast cipher suite and should only be used for non-RSN legacy support due to the reduction in security by using WEP.

The cipher suite selector 00:00:00:0 “None” is only valid for the Access Point and only valid for the unicast cipher suite. It shall only be used if a unicast cipher cannot be used by the Access Point, i.e. the multicast cipher and keys are to be used for unicast traffic as well as multicast/broadcast traffic (i.e. when the AP only supports default keys). An RSN AP shall use “None” to inform all stations that it will not be using Pairwise keys for unicast traffic and cannot be used in combination with another unicast cipher suite.

Note: A station shall also support a single Pairwise key, since Group keys shall not use index 0, Pairwise keys can always be implemented as default key 0 on the station.

Note: A station may choose not to associate to APs that does not support a unicast cipher for security policy reasons. A station shall ignore any values it does not recognize.

When the information element is used in an association request message or Probe Response for IBSS stations no authentication suite and only one unicast cipher suite is allowed.

Non-RSN capable stations shall not use the RSN information element.

APs shall not advertise RSN information element unless RSN is supported and enabled.

APs shall not advertise unsupported configurations and will send a Dissasociation Notification (Reason code 1) and a Deauthenticate to a STA requesting an unsupported configuration.

Example information elements:

1. 802.1X, AES for unicast and multicast, WEP stations are not supported.

```
        25,
        02,
        01, 00,                          // Version 1
```

2. 802.1X authentication, No unicast cipher suite and WEP for multicast cipher suite, WEP stations are supported.

```
25,
0C,
01, 00,                              // Version 1
00, 00, 00, 01                       // Multicast WEP
01, 00
00, 00, 00, 00                       // Unicast None
```

3. State machine

This section presents the concepts and terminology involved in integration of the IEEE 802.1X and IEEE 802.11 state machines. Specific terms are defined in the terminology section. Illustrations convey the relationship between IEEE 802.1X concepts and implementation within IEEE 802.11. The architectural descriptions are not intended to represent any specific physical implementation of IEEE 802.1X, 802.11 or the backend authentication server.

3.1. How wireless LAN systems are different

While many of the concepts presented within IEEE 802.1X remain valid, wireless LAN systems differ fundamentally from the wired networks for which IEEE 802.1X was developed. As a result, when IEEE 802.1X concepts need to be reformulated for use with wireless networks. Principal differences include:

- ***Supplicant initiation.*** Within a wired network, either the Supplicant or the Authenticator can initiate 802.1X authentication. The Supplicant initiates by sending an EAPOL-Start message; the Authenticator initiates by sending an EAP-Request/Identity. However, within IEEE 802.11, authentication and association is always initiated by the STA, and as a result, 802.1X pre-authentication may only be initiated by the Supplicant. As a result, RSN Authenticators MUST NOT send unsolicited 802.1X data frames to Supplicants. Since Supplicants can only receive 802.1X data frames from Authenticators to whom they had previously sent an EAPOL-Start frame, unsolicited 802.1X data frames shall be silently discarded. Among other things, this eliminates the potential for situations in which both Supplicant and Authenticator initiate 802.1X authentication.
- ***Shared media.*** Wireless LANs are shared media, and therefore the point-to-point connectivity assumed by IEEE 802.1X is not available. This implies that a cryptographic security association needs to be established between the Supplicant and Authenticator in order to create a one-to-one relationship.
- ***No controlled and uncontrolled ports.*** IEEE 802.1X assumes that a port exists prior to the initiation of the conversation between the Supplicant and Authenticator. However in an RSN, 802.1X pre-authentication occurs prior to association, and so acceptance of frames is governed by the 802.11 state machine, and the concept of IEEE 802.1X uncontrolled and controlled ports does not apply.
- ***Extended authentication requirements.*** IEEE 802.1X was developed for use with wired media where physical security may be assumed and security services such as per-packet confidentiality, authentication and integrity protection may not be

required. As a result, 802.1X does not require use of EAP methods supporting mutual authentication or key derivation. However, for use with IEEE 802.11, rogue access points are a concern, and per-packet confidentiality, integrity, authentication and replay protection is a requirement. As a result, when used with IEEE 802.11, the requirements for EAP methods are considerably more stringent.

- ***Need for key management and synchronization.*** Since IEEE 802.1X was developed for wired networks, key management and synchronization techniques were not well developed. However, in IEEE 802.11 dynamic key derivation is a requirement, as is synchronization of key installation between the Supplicant and the Authenticator.
- ***Non-negligible latency and packet loss***. 802.1X assumes the low latency and packet loss characteristic of wired networks. However, within wireless LANs latency may be substantial, particularly on the edge of the coverage area, and low packet loss cannot be assumed. STAs and APs may lose connectivity for substantial periods of time, and as a result, it is possible for 802.1X endpoints to lose synchronization. As a result, it is necessary to develop mechanisms to ensure state synchronization between the Supplicant and Authenticator.
- ***Increased scope of security threats***. On wired LANs, the 802.1X threat model centers on attackers gaining physical access to the wired network. On wireless LANs, attackers may act at a distance. On wired LANs the threat model centers on data frame vulnerabilities; on wireless LANs it is also necessary to protect Management and Control traffic. As a result, on wireless LANs the scope of the security threats is considerably greater, and articulation of the threat model is particularly important.

3.2.Roaming model

IEEE 802.11 enables authentication to be performed prior to association, allowing the STA to authenticate to multiple APs, while associating with only one. This provides support for “make before break” roaming, allowing STAs to limit connectivity interruptions resulting from authentication.

This is particularly important for IEEE 802.1X authentication, which can require a substantial number of round-trips. For example, when certificate authentication is used, conversations of 10+ round-trips are common. Where a backend authentication server is utilized, such initial authentication conversations can take a considerable time (hundreds of ms) to complete. While it is possible to shorten subsequent authentication conversations via “fast reconnect”, where the backend authentication server is located far from the Authenticator, the latencies involved may still be substantial.

With pre-authentication, as long as the IEEE 802.1X conversation can be completed prior to Association/Reassociation, no additional delays will result, as long as sufficient time is available for pre-authentication. While some additional processing is required to support protected management frames, it is not expected that this will contribute substantially to the overall time required to complete Association/Reassociation.

The amount of time available for pre-authentication depends on the degree of coverage overlap as well as the velocity of the STA. In general, where “fast reconnect” is supported, only modest coverage overlap is required to permit pre-authentication to complete in time, provided that the STA is moving at a rate of speed characteristic of a human on foot or bicycle. Where the velocity is characteristic of a moving vehicle, sufficient time may not be available for pre-authentication, without enabling the STA to discover APs whose Beacon messages it cannot hear yet. Were such support to be provided (such as via Candidate Access Router (CAR) discovery), “associated” pre-authentication, discussed in Section 4, could be used without modification.

3.3. Relationships among services

A STA keeps two state variables for each STA with which direct communication via the wireless medium is needed:

- Authentication state: The values are unauthenticated and authenticated.
- Association state: The values are unassociated and associated.

These two variables create three local states for each remote STA:

- State 1: Initial start state, unauthenticated, unassociated.
- State 2: Authenticated, not associated.
- State 3: Authenticated and associated.

The relationships between the stations state variables and the services are given in Figure 3 below:

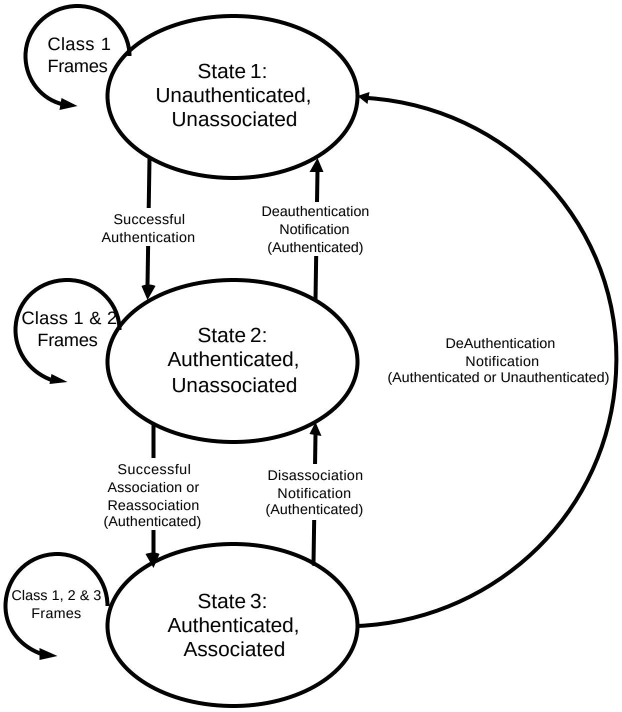
Class 1 Frames
State 1: Unauthenticated, Unassociated
Successful Authentication
Deauthentication Notification (Authenticated)
Class 1 & 2 Frames
State 2: Authenticated, Unassociated
DeAuthentication Notification (Authenticated or Unauthenticated)
Successful Association or Reassociation (Authenticated)
Disassociation Notification (Authenticated)
Class 1, 2 & 3 Frames
State 3: Authenticated, Associated

Figure 3 – Relationship between state variables and services

The current state existing between the source and destination determines the IEEE 802.11 frame types that may be exchanged between that pair of STAs. The state of the sending

STA given by Figure 1 is with respect to the intended receiving STA. The allowed frame types are grouped into classes and the classes correspond to the station state. In state 1, only Class 1 frames are allowed. In state 2, either Class 1 or Class 2 frames are allowed. In State 3, all frames are allowed (Classes 1, 2 and 3). Within RSN, 802.11 authentication frames are not used. Rather, authentication is accomplished via sending and receiving IEEE 802.1X frames.

The frame classes are defined as follows:

a) Class 1 frames (permitted from within States 1,2, and 3:

1) Control frames
   i. Request to send (RTS)
   ii. Clear to send (CTS)
   iii. Acknowledgment (ACK)
   iv. Contention-Free (CF)-End+ACK
   v. CF-End

2) Management frames
   i. Probe request/response
   ii. Beacon
   iii. Authentication: successful authentication enables a station to exchange Class 2 frames. Unsuccessful authentication leaves the STA in State 1.
   iv. Deauthentication:
   - Within RSN, Deauthentication messages are authenticated using the key material derived during IEEE 802.1X authentication. While by default an RSN-enabled station SHOULD silently discard Deauthentication messages that are unauthenticated or fail authentication, an RSN station MAY process unauthenticated Deauthentication messages if explicitly configured to do so. Since this exposes the station to denial of service attacks based on spoofed Deauthentication messages, this capability should be enabled with care.
   - A valid Deauthentication notification when in State 2 or State 3 changes the STA's state to State 1. The STA shall become authenticated again prior to sending Class 2 frames.
   v. Announcement traffic indication message (ATIM)

3) Data frames
   i. Data: Data frames with frame control (FC) bits "To DS" and "From DS" both false. IEEE 802.1X data frames sent the FC bits "To DS" and "From DS" both false are classified as Class 1 frames.

b) Class 2 frames (if and only if authenticated; allowed from within States 2 and 3 only):

1) Management frames:

i. Association request/response.
   - Within RSN, Association request and response messages MUST be authenticated and integrity protected using the key material derived during 802.1X authentication. When RSN is enabled, Stations MUST silently discard association request or response messages which are unauthenticated, or which fail authentication.
   - Successful association enables Class 3 frames.
   - Unsuccessful, or unauthenticated association leaves the STA in state 2.

ii. Reassociation request/response.
   - Within RSN, Reassociation request and response messages MUST be authenticated and integrity protected using the key material derived during 802.1X authentication. When RSN is enabled, Stations MUST silently discard reassociation request or response messages which are unauthenticated, or which fail authentication.
   - Successful reassociation enables Class 3 frames.
   - Unsuccessful or unauthenticated reassociation leaves the STA in state 2 (with respect to the STA that was sent the reassociation message). Reassociation frames shall only be sent if the sending STA is already associated in the same ESS.

iii. Dissassociation.
   - An authenticated Dissassociation when in State 3 changes a station’s state to State 2. The station shall become associated again if it wishes to utilize the DS.
   - Within RSN, Disassociation Notifications MUST be authenticated and integrity protected using the key material derived during 802.1X authentication. When RSN is enabled, Stations MUST silently discard Dissassociation Notifications which are unauthenticated, or which fail authentication.

c) Class 3 frames (if and only if associated, allowed only from within State 3):

1. Data frames.
   - Data subtypes: Data frames allowed. That is, either the “To DS” or “From DS” FC bits may be set to true to utilize DSSs. IEEE 802.1X data frames with either the “To DS” or “From DS” FC bits set to true are classified as Class 3 frames. These frames MUST have the FC “WEP” bit set.
2. Management frames.
   - Deauthentication. A non-discarded Deauthentication notification when in State 3 implies disassociation as well, changing the STA’s state from 3 to 1. The station

shall become authenticated again prior to another association.

3. Control frames:
   - PS-Poll

If STA A receives an Association or Reassociation Request from STA B that is not authenticated with STA A, STA A shall send a deauthenticate frame to STA B. Where RSN is enabled, at STA A's discretion, prior to sending the deauthenticate frame, it MAY initiate IEEE 802.1X authentication with STA B, and once complete, use the newly created key material in order to send an authenticated deauthenticate frame to STA B. Otherwise, STA A MAY send an unauthenticated deauthenticate frame to STA B.

If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is authenticated but not associated with STA A, STA A shall send a disassociation frame to STA B. When RSN is enabled, the disassociation frame MUST be authenticated using key material derived during IEEE 802.1X authentication. RSN-capable STAs receiving disassociation frames that are unauthenticated or which fail authentication MUST silently discard these frames.

If STA A receives a Class 3 frame with a unicast address in the Address 1 field from STA B that is not authenticated with STA A, STA A shall send an unauthenticated deauthentication frame to STA B. Since STA B is not authenticated it cannot have established key state with STA A and there is no way to authenticate the deauthentication frame. It is generally infeasible for STA B to queue the Class 3 frames, then initiate IEEE 802.1X authentication, and once complete, to dequeue and process the Class 3 frame. This is because the latency involved in IEEE 802.1X authentication might require STA B to queue a large number of data frames.

(The use of the word "receive" in this subclause refers to a frame that meets all of the filtering criteria specified in Clause 8 and 9).

4. IEEE 802.1X

Within an RSN, IEEE 802.1X is used for pre-authentication, so that the allowable frames are determined by the IEEE 802.11 state machine, discussed in Section 3. As a result, the IEEE 802.1X notion of controlled and uncontrolled ports does not apply to RSN wireless LANs.

For the purposes of describing the operation of IEEE 802.1X pre-authentication within IEEE 802.11, a STA may serve in one of two roles.

a) Authenticator. The STA configured to enforce authentication and authorization adopts the Authenticator role;

b) Supplicant. The STA configured to access the services offered by the Authenticator system adopts the Supplicant role.

Only the Authenticator and Supplicant are required to complete an authentication exchange. It is possible for a STA to adopt the Supplicant role in some authentication exchanges, and the Authenticator role in others. An example of the latter might occur when a STA acts in the role of a Supplicant in a BSS, but as either the Supplicant or the Authenticator in an IBSS.

In addition to the two required components, an optional component may be present:

c) Backend authentication server. The backend authentication server, which is typically located within the DS, performs the authentication function necessary to check the credentials of the Supplicant on behalf of the Authenticator, and indicates whether or not the Supplicant is authorized to access the Authenticator's services. Note that the backend authentication server is optional, and that this document does not specify use of an authentication, authorization and accounting protocol for communication between the Authenticator and the backend authentication server.

A Port Access Entity (PAE) operates the Algorithms and Protocols associated with the authentication mechanisms for a given STA.

In the Supplicant role, the PAE is responsible for responding to requests from an Authenticator for information that will establish its credentials. The PAE that performs the Supplicant role in an authentication exchange is known as the Supplicant PAE.

In the Authenticator role, the PAE is responsible for communication with the Supplicant, and for submitting the information received from the Supplicant to the backend authentication server in order for the credentials to be checked, and for authorization state to be determined.

The PAE that performs the Authenticator role in an authentication exchange is known as the Authenticator PAE. Within IEEE 802.1X pre-authentication, the Authenticator PAE controls movement from State 1 (unauthenticated, unassociated) to State 2 (authenticated, unassociated), based on the outcome of authentication and key establishment.

4.1. IEEE 802.1X pre-authentication

IEEE 802.1X data frames of any Class can be used for pre-authentication or re-authentication. IEEE 802.1X data frames with both the "From DS" and "To DS" FC bits false are Class 1 frames, and thus may be sent within any state. IEEE 802.1X data frames with either the "From DS" or "To DS" FC bits true are Class 3 frames and may only be sent within State 3.

This specification supports two forms of IEEE 802.1X pre-authentication:

a. Unassociated pre-authentication. Here the STA is not associated to any STA. Since the STA is not yet associated (States 1 or 2), Class 1 IEEE

802.1X data frames are sent, with the "To DS" and "From DS" FC bits set to false. With this form of pre-authentication, it is necessary for the STA to listen and send on the radio channel of the STA that it wishes to authenticate to. Since keys have not yet been activated, IEEE 802.1X data frames sent in unassociated pre-authentication have the "WEP" FC bit set to false.

b. Associated pre-authentication. Here the STA A is authenticated and associated to STA C, but desires to pre-authenticate to STA B. Since the STA is associated, it may send Class 3 IEEE 802.1X data frames, with the "To DS" or "From DS" FC bits set to true (Class 3). With this form of pre-authentication, it is not necessary for the STA to send or receive on an alternate channel. As a result, connectivity interruptions are minimized and interactions with power save are simplified. Since keys have been activated, IEEE 802.1X data frames sent in associated pre-authentication have the "WEP" FC bit set to true.

Class 1 IEEE 802.1X data frames, sent within States 1 or 2, require that the sending STA be tuned to the same radio channel as the receiving STA. For a STA that is already authenticated and associated to one STA, but wishing to pre-authenticate to another STA, it can be difficult to switch radio channels long enough to complete a potentially lengthy pre-authentication without risking wholesale packet loss, even if power-saving mode and associated queueing is utilized.

This problem can be avoided by utilizing Class 3 IEEE 802.1X data frames. Since Class 3 data frames can be sent within State 3, they may originate from, or be destined to the DS, and thus may have the "From DS" or "To DS" FC bits set to true. This allows a STA in State 3 to pre-authenticate to another STA via the DS without having to be tuned to the same radio channel.

As an example, suppose that STA A has authenticated and associated with STA B. Through active or passing scanning, STA A detects the presence of STA C, and wishes to pre-authenticate to it. This can be accomplished by having STA A tune to the radio channel of STA C, followed by an exchange of Class 1 IEEE 802.1X data frames between STA A and STA C. However, since STA A is in State 3 with respect to STA B, it is also possible for STA A to exchange Class 3 IEEE 802.1X data frames with STA C, with STA B relaying these frames back and forth between the WM and the DS.

Note that IEEE 802.1X does not prohibit forwarding of IEEE 802.1X frames destined to a unicast MAC address, only frames destined to a non-forwardable multicast MAC address. IEEE 802.1X does not require filtering of IEEE 802.1X frames by Ethertype.

Reference model

This specification presents the architectural view, emphasizing the separation of the system into four major parts: the MAC of the data link layer, the PHY, IEEE 802.1X, and Upper Layer authentication protocols. The layers and sub layers described in this standard are shown in Figure 4.

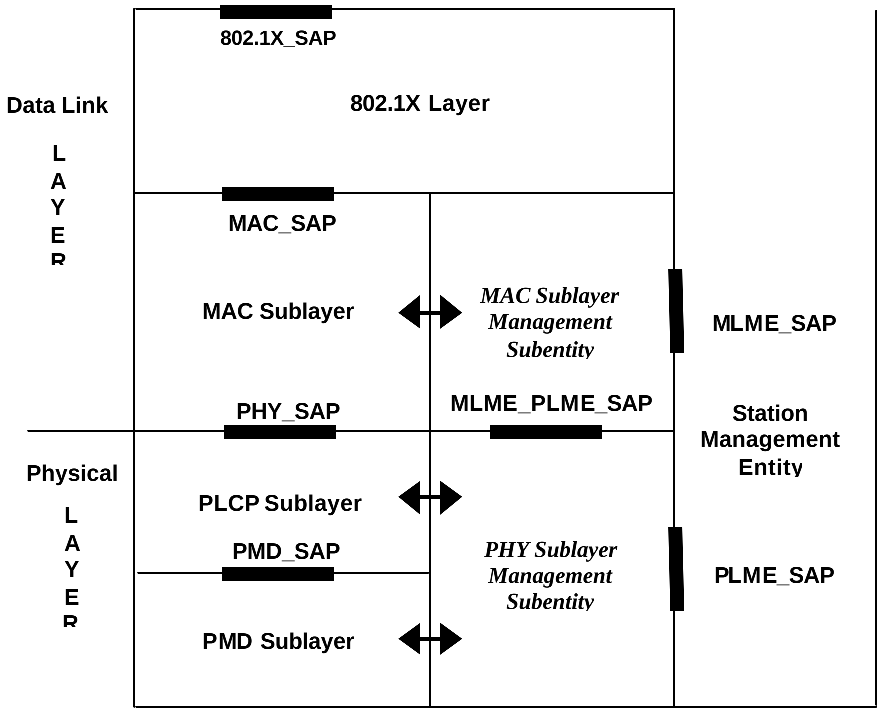

Figure 4 – Reference Model

ESS Authentication

Both 802.1X pre-association and the association process are driven by the station. In the ESS case the station MLME will chose APs that it may want to associate to and will request the 802.1X Supplicant to pre-authenticate to those 802.1X Authenticators or APs. The 802.1X Supplicant starts the authentication process by sending an EAPOL-Start frame to the Authenticator. The MLME in the AP on receiving the Authentication.Indication will request the Authenticator to start the authentication process by sending an EAP-Request/Identity message to the Supplicant. The 802.1X messages are sent as 802.11 data frames to the Authenticator. During the 802.1X authentication process, the Supplicant and Authenticator obtain keys.

Once pre-authentication has been completed, the STA selects the AP to associate to, and the station MLME sends an authenticated IEEE 802.11 Association Request frame to the AP. The AP will then send an authenticated IEEE 802.11 association response message back to the station. When the Supplicant completes association, it installs the keys into the encryption/integrity engine for use by the association.

If the 802.11 Supplicant does an 802.1X re-authentication after initial 802.1X authentication the 802.1X messages are sent as encrypted data messages if key mapping keys are used.

The EAPOL-Key message is used to exchange information between the Supplicant and the Authenticator for the keying process. There is a single Pairwise key between the Supplicant and Authenticator produced by the 4-way handshake. The Pairwise key is used to transfer Group key updates and may be used as a Pairwise transient key. Group key updates use two key indexes to mitigate the loss in the ongoing data transmissions while keys are being distributed and applied at the Supplicants.

IBSS Authentication

The authentication process is driven by the Supplicant. In the IBSS case the MLME will chose stations that it may want to authenticate to and send an 802.1X EAPOL-Start frame to the Authenticator. The MLME in the station on receiving a SSN MLME-Authentication.Indication will request the Authenticator to start the authentication process by sending an EAP-Request/Identity message to the Supplicant. The data messages are sent with the FromDS and ToDS bits set to 0 and they are always sent unencrypted since no keys are available. During the authentication process, the Supplicant obtains keys.

The EAPOL-Key message is used to exchange information between the Supplicant and the Authenticator for the keying process. There is a single Pairwise key between the Supplicant and Authenticator produced by the 4-way handshake. The Pairwise key is used to transfer Group key updates and may be used as a Pairwise transient key.

4.2. Security issues

While enabling Class 3 IEEE 802.1X data frames to be forwarded to and from the DS solves a number of problems, it also introduces several potential security vulnerabilities:

a. An unauthenticated STA on the WM can attempt to pre-authenticate to an AP reachable via the DS.
b. An authenticated STA on the WM can attempt to spoof an IEEE 802.1X data frame originating from an AP MAC address, sent to an authenticated STA on the WM.
c. A host on the DS can spoof an IEEE 802.1X data frame originating from the MAC address of an authenticated STA on the WM.

Attack a is not feasible, since prior to authentication, a STA may only send Class 1 data frames with “From DS” and “To DS” FC bits set to false. Thus, an AP receiving a Class 3 IEEE 802.1X data frame from an unauthenticated STA with the “From DS” or “To DS” MUST silently discard the frame.

Attack b is also not feasible. Since IEEE 802.1X pre-authentication is always initiated by the STA, not by the AP, a STA receiving an unsolicited IEEE 802.1X data frame from an AP MUST silently discard the frame. Furthermore, APs MUST preclude an authenticated STA from changing its MAC address once authentication and key state have been established.

In attack c, a DS host may attempt a denial of service by sending an EAPOL-Logoff frame to the AP, with a source MAC address of the STA on the WM. This attack can be prevented by an AP that implements anti-spoofing precautions. While the EAPOL-Logoff would be expected to arrive on the WM where the STA is attached, instead it arrives on the DS. Alternatively, the DS host could send an EAP Failure packet to the STA, originating from the AP's MAC address. In this case, the AP receives on the DS a packet sourced from one of its own MAC addresses. In both these cases, basic anti-spoofing functionality can preclude an attack.

5. Authenticated ciphersuite negotiation

Within this specification, it is assumed that RSN-capable wireless LANs support protected ciphersuite and authentication negotiation. This includes the ability to verify the authenticity and integrity of capabilities advertised in the Beacon and Probe Response.

Several mechanisms for this are available, including:

- Support within EAP.
- Support within the 4-way key handshake.
- Support within a protected Association/Reassociation exchange.

Support for authenticated ciphersuite negotiation within EAP is attractive since it guarantees that the ciphersuite will be securely determined prior to the exchange of management frames and the initiation of the 4-way handshake. This is convenient in that it guarantees that the 4-way handshake, and associated derivation of the Pairwise Transient Key (PTK) and Group Transient Key (GTK) are based on the correctly selected ciphersuite. This prevents a rogue AP spoofing a Beacon or Probe Response from causing an incorrect 4-way handshake to be run, deriving incorrect keys that will later cause management frame authentication to fail. Since not all EAP methods support protected ciphersuite negotiation, so that if this approach is taken, ciphersuite negotiation probably needs to be supported as an EAP extension method, which could then be used alongside any existing EAP authentication technique.

Since the 4-way handshake results in the derivation of the PTK and GTK, which are dependent on the selected ciphersuite, it can be argued that the ciphersuite should be verified within the 4-way handshake. Since the 4-way handshake occurs prior to association/reassociation, protected ciphersuite negotiation occurring at this point in the conversation protects against the derivation of incorrect keys, and does not require extensions to EAP. On the other hand, inclusion of protected ciphersuite negotiation within the 4-way handshake complicates its design.

Since the association/reassociation messages are authenticated and integrity protected, it is possible to use these messages to confirm the authenticity of capabilities advertised within the Beacon and Probe Response, as follows:

- The capabilities received by the STA in the Beacon or Probe Response are included within the Association/Reassociation Request.
- The capabilities sent by the AP in its Beacon or Probe Response are included in the Association/Reassociation Response.

- In order to authenticate and integrity protect the capabilities, the MIC calculation is run over the capabilities.
- By comparing the capabilities that were sent and received, it is possible for both the STA and AP to determine whether Beacons and Probe Responses have been tampered with, and if so, what changes were made.

The advantage of this approach is that it does not require extensions to EAP, and does not require adding features to the 4-way key exchange. On the other hand, since Association/Reassociation occurs *after* both 802.1X pre-authentication and the 4-way handshake, this increases the vulnerability to rogue Beacons or Probe Responses. As a result, if this approach is chosen there are some implications:

- It is necessary to ensure that authentication keys used in management frame authentication are independent of the chosen ciphersuite, so that a rogue AP spoofing Beacon or Probe Response messages cannot cause management frame authentication to fail.
- It may be necessary to re-run the 4-way handshake when management frames fail authentication.

6. Management frame authentication

6.1. Authenticated and unauthenticated management frames

In RSN-enabled Wireless LANs, management frames including Association Request/Response, Reassociation Request/Response, and Disassociation MUST be authenticated and integrity protected using key material established during IEEE 802.1X authentication. As a result, STAs receiving messages of these types which are unauthenticated or fail authentication MUST silently discard them. Deauthentication messages may also be authenticated and integrity protected, provided that key material is available. However, this is not always possible.

For example, consider what happens when, after STA A authenticates to STA B, STA B subsequently discards the authentication and key state for STA A, sending a Disassociate or Deauthenticate frame. If STA A was disconnected at the time, it will not receive the frame, and may not be aware that STA B has discarded its authentication and key state.

As a result, STA A may consider itself to be in State 2, (in which case it may send an Association or Reassociation Request to STA B), or in State 3 (in which case it may send a Class 3 data frame to STA B). On receiving these frames, STA B will send a Deauthenticate frame to STA A. However, since STA B no longer maintains keying material for STA A, the Deauthenticate frame will be unauthenticated.

While RSN-capable STAs MAY send unauthenticated Deauthenticate frames, RSN-capable STAs receiving such messages SHOULD silently discard them by default. Where STA A believes itself to be in State 2, and has received and discarded the Deauthenticate frame after sending an Association or Reassociation Request to STA B, it will resend the Request to STA B, and will subsequently time out. Where STA A believes itself to be in State 3, and has received and discarded the Deauthenticate frame after sending an Class 3

Data Frame to STA B, no ACK will be received, and therefore STA A will also eventually time out. As a result, STA A will eventually delete its authentication and key state with respect to STA B and return to State 1.

6.2. Detecting forged Beacon or Probe Response frames

In order to confirm that the information advertised within the Beacon and Probe Response frames is authentic, it is necessary to echo this information within authenticated Association/Reassociation Request frames. This enables the associating STAs to determine whether unauthentic Beacons or Probe Responses have been received.

As a result, the authenticated Association/Reassociation Request frame contains the RSN Information Element, representing the selected ciphersuites and authentication method. The authenticated Association/Reassociation Response frame also contains the RSN Information Element, echoing the RSN IE included within Beacon and Probe Response frames.

On receiving an authenticated Association/Reassociation Response frame, the STA shall compare the RSN IE included within that frame with the RSN IE received in the corresponding Beacon and/or Probe Response. If the two differ, then the STA shall assume that it has received a forged Beacon or Probe Response. If the STA would have selected different security parameters based on the RSN IE included within the authenticated Association/Reassociation Response, then the STA sends an authenticated Reassociation Request included the updated security parameters.

6.3. Approaches to management frame authentication

For authenticating management frames, including Association request/response, Reassociation request/response, Disassociation, and Deauthenticate, several approaches are possible. These include:

1) Use of RSN ciphers, such as TKIP or AES to authenticate, integrity and replay protect and encrypt the management frames.
2) Addition of an Authenticator Information Element (IE) to management frames.

Every thing else being equal, use of the AES and TKIP ciphers is preferable, since this allows the STA to make best use of hardware support for cryptographic operations, if available. Use of the RSN ciphers also simplifies the key hierarchy, since the existing RSN keying material can be used for securing management and control frames as well as data.

However, there are also significant disadvantages to this approach. While STAs learn of the available ciphersuites through Beacons and Probe Request/Response messages, the ciphersuite is negotiated in the protected Association or Reassociation exchange. Since management frames are sent prior to the completion of this exchange, it is not possible for the STA to select the appropriate ciphersuite with which to protect management frames, based on this method. It is possible for the STA to select an EAP method that provides for protected ciphersuite negotiation, thereby selecting the appropriate ciphersuite during the IEEE 802.1X authentication process. However, not all EAP methods support protected ciphersuite negotiation, so this is not universally applicable.

Note that the Pairwise Temporal Key (PTK) and Groupwise Temporal Key (GTK) are derived during the 4-way key handshake, which does not include verification of the selected ciphersuites. This means that if the keys used to authenticate management frames depend on the selected ciphersuite, then it is possible for a forged Beacon or Probe Response to cause management frame authentication to fail. As a result, use of selected ciphersuites to protect management frames introduces a potential vulnerability. As we will describe, this implies that the key hierarchies associated with all ciphersuites need to derive the management frame authentication keys in the same way.

Furthermore, RSN ciphersuites such as TKIP and AES are oriented towards protection of MSDUs, not MPDUs. In order to adequately protect Management frames, it is necessary for the Message Integrity Check (MIC) to cover the entire Management frame, including the Frame Control, Duration, DA, SA, BSSID, Sequence Control and Frame Body fields. Typically, the integrity protection within RSN ciphers will only cover the SA and DA as well as the MSDU. As a result, in order to adequately protect management frames, it would be necessary to encapsulate them as MSDUs, and then decapsulate and process them after decryption.

Addition of an Authenticator Information Element to management frames avoids many of these issues. Since management frames only require authentication, integrity and replay protection, if management frame authentication keys are derived the same way for all ciphersuites, it is not necessary to negotiate the ciphersuite prior to sending authenticated management frames. Since management frame authentication is independent of the ciphersuite used to protect data, the MIC used for management frame authentication may be defined independently of the MICs used in integrity protection and authentication of data frames. This allows the Authenticator Information Element can be defined to cover the desired fields of the Management frame header. By default within RSN version 1, all STAs MUST support the HMAC-SHA1 algorithm for authentication and integrity protection of management frames.

Disadvantages of the Authenticator Information Element approach are that the MIC calculation will typically need to be done in software, and therefore will not be able to leverage hardware acceleration, if available.

6.4. Control frame authentication

Control frames include the PS-Poll, Request to Send (RTS), Clear to Send (CTS), Acknowledgment (ACK), CF-END, and CF-End+CF-Ack. Control frames are Class 1 frames, and so they may be sent within any state. This implies that, like Beacons and Probe Request/Responses, they may be sent prior to authentication and key derivation. As a result, it is only possible to authentication and integrity protect control frames sent in States 2 and 3.

Many of the same considerations apply to protection of control frames as apply to protection of management frames. However, control frames may be sent at higher data rates than management frames. As a result, the performance penalty from handling protection in software is much higher. Therefore, use of the native RSN AES and TKIP ciphers is more attractive for protection of control frames than addition of an Authenticator Information Element.

Unfortunately the TKIP and AES ciphers apply to MSDUs, not MPDUs. As a result, they do not encrypt, authenticate, or integrity protect the 802.11 header. Since management and control frames represent MPDUs, not MSDUs, the TKIP and AES ciphers as defined cannot be used to protect these frames. This is a flaw in the design of both the TKIP and AES ciphers, and if not fixed, will expose RSN implementations to denial of service attacks.

It is therefore recommended that IEEE 802.11 Tgi consider application of TKIP and AES to MPDUs, rather than MSDUs.

6.5. Authenticator Information Element

The Authenticator Information Element is defined as follows:

```
0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Element ID |    Length     |            Algorithm          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Replay Counter                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Replay Counter                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             MIC...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                             MIC...                            ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Element ID

The Element ID field is a single octet. The Authenticator element ID is 38 decimal (0x26 Hex)

Length

The length field is one octet. It represents the length of the information element following normal IEEE 802.11 information element rules.

Algorithm

The algorithm field is two octets. It represents the algorithm to be used in computing the Message Integrity Check. In RSN version 1, only the HMAC-SHA1 algorithm is supported (algorithm 0x0001) and MUST be implemented. Since only a single MIC algorithm is supported for the Authenticator IE within RSN version 1, STAs can assume that the algorithm is available and it need not be negotiated.

Replay counter

The replay counter is 16 octets (64 bits). It represents a monotonically increasing counter which may start at any value. Before the replay counter wraps, authentication and key state MUST be re-established, so that the Authenticator Information Element cannot be replayed. A 64-bit NTP timestamp MAY be used as the replay counter.

MIC

The MIC field is of variable length, determined by the algorithm specified in the algorithm field. For the HMAC-SHA1 algorithm, the length of the MIC field is 20 octets (160 bits). The MIC field is calculated over the frame control, Duration, DA, SA, BSSID, Sequence Control, and Frame Body fields within the management frame. This includes the Authenticator IE present as the last information element within the Frame Body, with all of its fields filled in (element ID, length, algorithm, replay counter) with the exception of the MIC field itself, which is set to zero.

6.6. Unicast frames

Unicast management frames include Association Request/Response, Reassociation Request/Response, Disassociation, and Deauthenticate frames. For these frames, the MIC is calculated using Authenticator Information Element transmit and receive MIC keys, derived as part of the Pairwise Transient Key (PTK) hierarchy. The derived Authenticator IE MIC keys are the same for any ciphersuite, so as to ensure that management frame authentication is not dependent on the negotiated ciphersuite.

6.7. Multicast/broadcast frames

The Disassociation and Deauthenticate frames may be broadcast. When these frames are sent to a broadcast destination, the MIC is calculated using the Authenticator Information Element MIC key, derived as part of the Group Transient Key (GTK) hierarchy. The derived Authenticator IE MIC group keys are the same for any ciphersuite, so as to ensure that management frame authentication is not dependent on the negotiated ciphersuite.

Note that the use of group keys permits any STA with knowledge of the GMK to forge broadcast Disassociation or Deauthenticate frames. As a result, STAs MAY silently discard broadcast Disassociation or Deauthenticate frames, even if they are successfully authenticated. STAs taking this conservative approach will timeout rather than immediately acting on the Disassociation or Deauthenticate frame, so that performance will be affected.

7. Key management

7.1. Establishing and discarding authentication and key state

Within this specification, the STA establishes key state on another STA through successfully completing IEEE 801.X authentication with that STA. Successful authentication includes establishment of key state via a 4-way handshake. IEEE 802.1X pre-authentication is always initiated by the STA, and is never initiated by the AP.

It is assumed that a STA conforming to this specification will either completely maintain or discard authentication and key state. That is, once IEEE 802.1X authentication is complete, the established key state, including the Pairwise Master Key (PMK), Pairwise Transient Key (PTK), Group Master Key (GMK), Groupwise Transient Key (GTK) and IV, will remain stored by the STA until *all* the authentication and key state is discarded. As a result, the 4-way handshake is considered an integral part of the authentication process, and is not re-run prior to the Association or Reassociation exchange.

To maintain the integrity of the derive keys, STAs MUST NOT discard portions of the authentication and key state. For example, it is forbidden for the STA to discard the IV, PTK or GTK while retaining the authentication state, PMK and GMK. This ensures that when a STA associates or reassociates to a STA with which it had previously authenticated, that either all the authentication and key material remains valid, or the STA will need to authenticate again.

Where the TKIP or AES ciphers are encrypted, the authenticity and integrity of protected data frames shall be verified. This SHOULD include determining that protected IEEE 802.1X data frames originate from an authenticated STA on the WM. This includes determining that the STA has not changed its MAC address since establishing authentication and key state, so as to prevent spoofing.

7.2. Key activation

In IEEE 802.1X pre-authentication, there is an important distinction between when keys are derived and when they are enabled for use with the selected ciphersuite. While IEEE 802.1X is used to derive keying material, subsequent exchanges determine when the keying material is loaded into the integrity/confidentiality engine. The 4-way handshake confirms that the Authenticator and Supplicant have the same PMK, and that the PMK is fresh. The 4-way handshake is initiated as part of authenticating a Supplicant and an Authenticator but it shall be initiated if a data integrity failure occurs.

IEEE 802.1X messages are only encrypted using the Pairwise key because if Group keys are used to encrypt 802.1X messages there is an initialization problem with stations after the first association. The 802.1X EAPOL-Key descriptor containing the Group key is encrypted with the Group key when it is sent to the new station.

Once keys have been activated, IEEE 802.1X data frames are sent with the “WEP” FC bit set to true. Thus the timing of the activation determines when IEEE 802.1X data frames may be protected. There are two alternatives for this: use of the 4-way handshake for key activation, or use of the authenticated Association/Reassociation exchange.

Where the STA supports key mapping keys, it may conclude multiple IEEE 802.1X authentication and 4-way key handshake conversations, each resulting in derivation of a key mapping key for a TA/RA combination. On completion of the 4-way handshake, the STA may load the derived key material into the key mapping tables, or may store them for future use. As a result, where the STA supports key mapping keys, the 4-way handshake may be used to determine when the keys are activated.

However, where the STA only supports default keys, loading the derived key material immediately would result in an inability to decrypt traffic sent by STAs using the original default key. As a result, STAs supporting only default keys must wait to load keying material until after completion of the authenticated Association/Reassociation exchange.

Thus, if it is desired to enable STAs that only support default keys to participate in an RSN, then the completion of the authenticated Association/Reassociation exchange will need to signal activation of the derived keys, rather than completion of the 4-way handshake. If it is possible to require support for key mapping keys, then the 4-way handshake can be used instead.

Sections 7.2.1, 7.2.2 and 7.2.3 analyze the implications of the two approaches.

7.2.1. Activation via the 4-way handshake

If the 4-way handshake controls activation, IEEE 802.1X data frames are sent with the "WEP" FC bit set in both States 2 and 3, since within these states both authentication and the 4-way handshake have been completed. It is also possible to set the "WEP" FC bit on the last frame of the authentication conversation within State 1 (the EAP Success/Failure message), since by this time the authentication and 4-way handshake have been completed.

7.2.2. Activation via the Association/Reassociation exchange

In this approach, key activation occurs as a result of completing the protected Association/Reassociation exchange (State 3). As a result, Class 1 IEEE 802.1X data frames sent within States 1 or 2 will have the "WEP" FC bit set to false. IEEE 802.1X data frames sent within State 3, including both Class 1 and Class 3 frames, will have the "WEP" FC bit set to true.

Since data frames with the "WEP" FC bit set to true are not be sent until the STA has entered State 3 (authenticated, associated), the authenticated association/reassociation exchange governs the activation of keys, and the sending of secured data frames, as follows:

- On sending a successful authenticated Association or Reassociation Response, the sending STA will activate the derived keys for use by the integrity/confidentiality engine.
- On receiving a successfully verified authenticated Association or Reassociation Response, the receiving STA will activate the derived keys for use by the integrity/confidentiality engine.

7.2.3. Summary

Since activation via the Association/Reassociation exchange only enables protection of frames sent within State 3, the 4-way handshake approach is more secure, since it also enables protection of frames sent within State 2 and the tail end of State 1. For example, when the 4-way handshake is used for key activation, EAP Success and Failure messages may have the "WEP" FC bit set to true in any State. This is possible because in State 1 the 4-way key handshake will complete prior to sending EAP Success and Failure messages. Since keys are active upon entering State 2, the "WEP" FC bit may also be set to true for frames sent in this state.

Therefore, use of the 4-way handshake for key activation enables protection of the EAP Success and Failure messages within any state. This is helpful since these frames would otherwise be sent in the clear, even if the chosen EAP method is capable of protecting them. This is because IEEE 802.1X "manufactures" cleartext EAP Success and Failure messages on receiving an Access Accept/Reject from the backend authentication server. As a result, even if the EAP method protects EAP Success and Failure messages, 802.1X will throw the protected messages away, and replace them with cleartext messages.

Note that the selected EAP method does not support protection of the EAP conversation, then security vulnerabilities remain, regardless of how the keys are activated. Even with the 4-way handshake approach, without a protected EAP method, it is possible to spoof messages of type Identity, Nak and Notification sent from within State 1. In State 1, only EAP Success or Failure messages can be sent with the "WEP" FC bit turned on, since this is only enabled once the 4-way handshake has concluded.

7.3. Key hierarchy

There are two key hierarchies:

1. Pairwise key hierarchy
2. Group key hierarchy

The Pairwise key hierarchy takes a Pairwise Master Key and generates a Pairwise transient key which is used to obtain the EAPOL-Key MIC and Encryption keys, the Authenticator Information element MIC transmit and receive keys, and the Pairwise data MIC and encryption keys. Pairwise keys are used between a single Supplicant and a single Authenticator.

The Group key hierarchy takes a Group Master Key and generates a Group Transient key which is used to obtain Group data MIC and encryption keys, as well as the Group

Authenticator Information Element MIC keys. Group Keys are used between a single Authenticator and all Supplicants authenticated to that Authenticator.

The following functions are used in the following section:

PRF	Pseudo-random function defined in Section **Error! Reference source not found.**.
L (I, F, L)	Take from I starting from the left, bit F for L bits moving to the right using 7.1.1 bit convention from IEEE 802.11.

The terms AA (Authenticator Address) and SA (Supplicant Address) are used. In an ESS network the AA is the AP wireless MAC address and SA is the station MAC address. In an IBSS the AA is the station (who has been chosen as the Authenticator) MAC address, and other stations MAC address will be the SA.

Rekey

In addition to authentication, IEEE 802.1X may used by 802.11 in order to rekey MAC keys, using the EAPOL-Key frame. This may occur when a given time period has expired, when a pre-set byte or packet count is reached, or when the IV space of the selected ciphersuite has been exhausted, since security is compromised when IVs are re-used.

Since both the TKIP and AES ciphers support a large IV space, in most situations reauthentication and associated key update will occur before rekey is required. Therefore, when the TKIP or AES ciphersuites are selected, rekey of the Pairwise Key hierarchy will typically only in exceptional circumstances, such as detection of an attack on the TKIP cipher. As a result, in normal operation with the TKIP and AES ciphersuites, the primary use for the EAPOL-Key frame is to update the Group Key hierarchy.

TKIP and AES key hierarchies

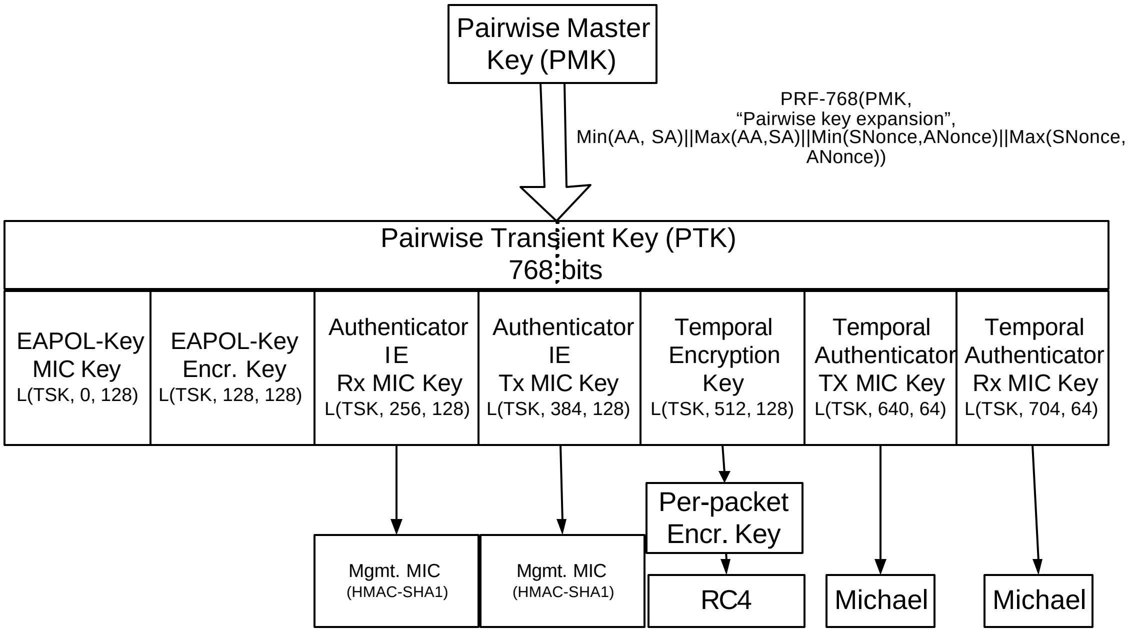
Pairwise Master Key (PMK)
PRF-768(PMK, “Pairwise key expansion”, Min(AA, SA)||Max(AA,SA)||Min(SNonce,ANonce)||Max(SNonce, ANonce))
Pairwise Transient Key (PTK) 768 bits
EAPOL-Key MIC Key L(TSK, 0, 128)
EAPOL-Key Encr. Key L(TSK, 128, 128)
Authenticator IE Rx MIC Key L(TSK, 256, 128)
Authenticator IE Tx MIC Key L(TSK, 384, 128)
Temporal Encryption Key L(TSK, 512, 128)
Temporal Authenticator TX MIC Key L(TSK, 640, 64)
Temporal Authenticator Rx MIC Key L(TSK, 704, 64)
Per-packet Encr. Key
Mgmt. MIC (HMAC-SHA1)
Mgmt. MIC (HMAC-SHA1)
RC4
Michael
Michael

Figure 5—Complete TKIP Pairwise Key Hierarchy

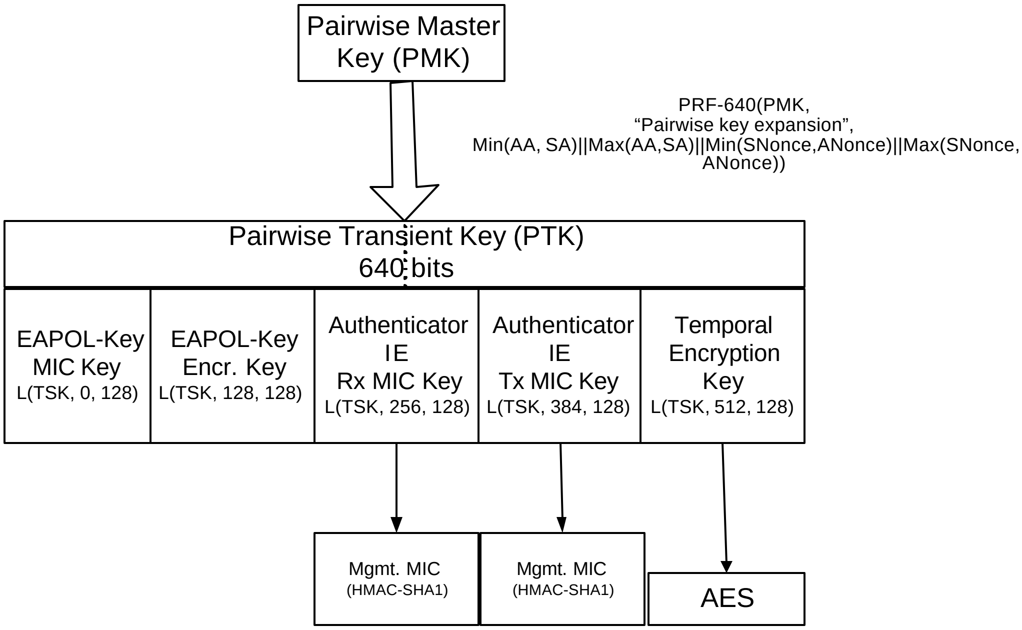
Pairwise Master Key (PMK)
PRF-640(PMK, “Pairwise key expansion”, Min(AA, SA)||Max(AA,SA)||Min(SNonce,ANonce)||Max(SNonce, ANonce))
Pairwise Transient Key (PTK) 640 bits
EAPOL-Key MIC Key L(TSK, 0, 128)
EAPOL-Key Encr. Key L(TSK, 128, 128)
Authenticator IE Rx MIC Key L(TSK, 256, 128)
Authenticator IE Tx MIC Key L(TSK, 384, 128)
Temporal Encryption Key L(TSK, 512, 128)
Mgmt. MIC (HMAC-SHA1)
Mgmt. MIC (HMAC-SHA1)
AES

Figure 6—Complete AES Pairwise Key Hierarchy

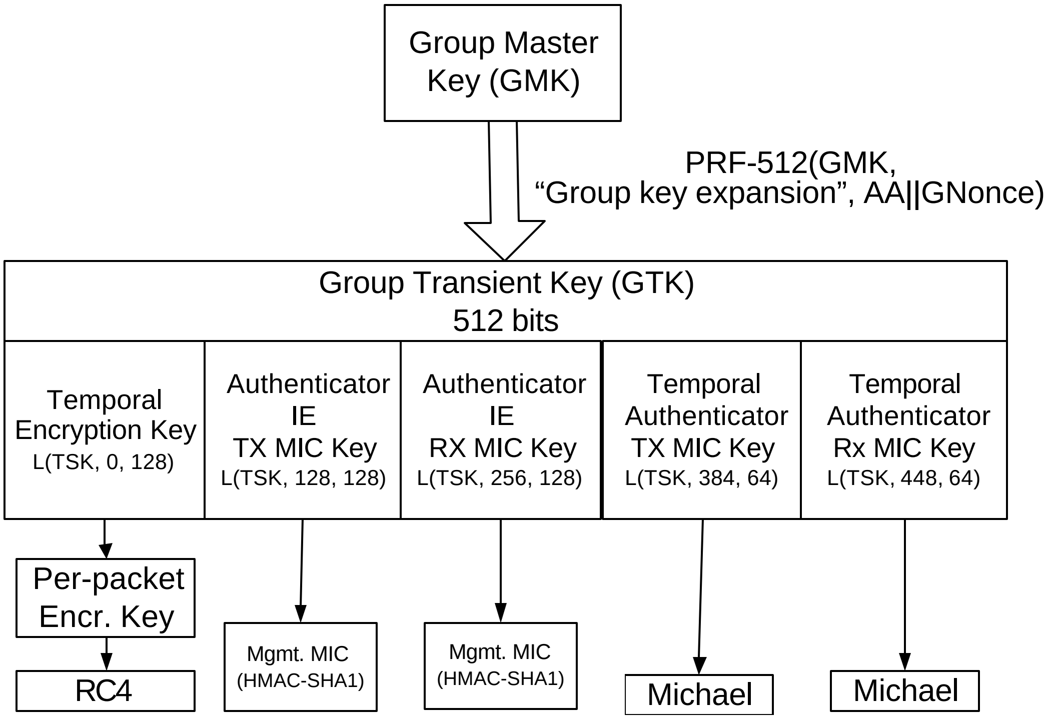

Figure 7—Complete TKIP Group Key Hierarchy

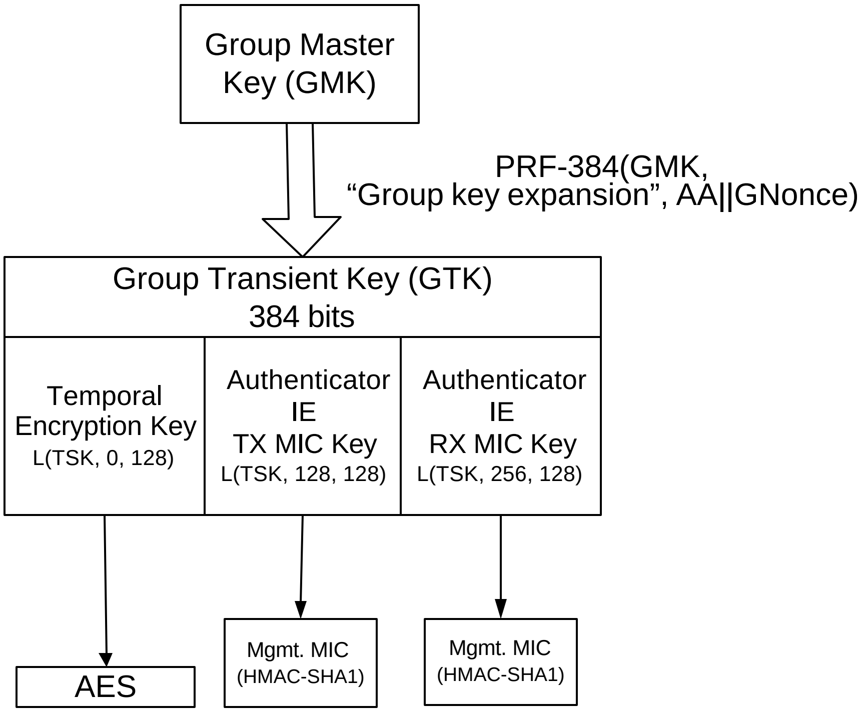

Figure 8—Complete AES Group Key Hierarchy

Pairwise master key (PMK)

The Pairwise Master Key is generated as a result of authentication between the Supplicant and Authentication Server involved. The EAP authentication method shall generate a 256 bit key that is used for the PMK, and the backend authentication server will ensure that this is distributed to the AP using attributes specified in Appendix A. An EAP authentication method normally has a Master Key generated by the authentication and the PMK should be derived from or protected by the Master Key. This key generation is normally carried out independently and simultaneously on the backend authentication server and the Supplicant, based on information that was communicated between the backend authentication server and the Supplicant during authentication. In general, the mechanism for generating the PMK will be dependent on the EAP method that is selected, as well as the mechanism used to transmit the PMK between the backend authentication server and the AP as well as between the Supplicant and backend authentication server.

Group master key (GMK)

The Group Master Key (GMK) for the Group key hierarchy should be initialized using a cryptographically secure random number. If this is not possible it shall be initialized to the first PMK the Group key master receives (since there is no need to send broadcast traffic unless there is at least one station associated), but the following rules shall then be applied:

1. The GMK should be updated periodically from another current PMK.

2. The GMK shall be changed when the AP deletes the association state for the station whose PMK is being used as the GMK.

Nonce Generation

All stations contain a global Key Counter which is 256 bits in size. It should be initialized at system boot up time to

PRF-256(Random number, “Init Counter”, Local Mac Address || Time)

The Local Mac Address should be AA on the Authenticator and SA on the Supplicant.

Random number should be the best possible random number possible and 256 bits in size. Time should be the current time (from NTP or another time in NTP format). This initialization is to ensure that different initial Key Counter values occur across system restarts whether a real-time clock is available or not. The Key Counter must be incremented (all 256 bits) each time a value is used as a nonce. The Key Counter must not be allowed to wrap to the initialization value.

Pairwise transient keys

Pairwise TKs are derived from the Pairwise MK using a PRF with AA, SA, SNonce and ANonce as inputs. The size of the PRF computation shall be taken as the size specified by EAPOL-Key Key Length plus the size of the EAPOL-Key MIC Key, the size of the EAPOL-Key Encryption Key, and the size of the Authenticator IE MICs, i.e. 32+16+16 +

16 + 16 = 96 octets (768 bits) for TKIP and 16+16+16+16+16 = 80 octets (640 bits) for AES.

PTK = PRF-640/768 (PMK, “Pairwise key expansion”, Min(AA, SA)

|| Max(AA, SA) || Min(SNonce, ANonce) || Max(SNonce, ANonce))

AA and SA are concatenated in integer order i.e. the lower MAC address is concatenated first, followed by the higher MAC address. SNonce is a nonce sent by the Supplicant and ANonce is a nonce sent by the Authenticator to the Supplicant. They are concatenated in integer order i.e. the smaller nonce is concatenated first, followed by the larger nonce.

The Min/Max of the MAC addresses and Nonces are done so the PRF is independent on whether it is run on the Authenticator or Supplicant. AA and DA are part of the PRF input so that the inputs are unique to each station pair.
ANonce is a nonce taken from the Key Counter on the Authenticator whenever a new Pairwise TK is derived. ANonce is used so the inputs to PRF are different for each TK set. If a station re-associates to the same AP, a different ANonce value is used for the derivation of a new TK set.

SNonce is a nonce taken from the Key Counter on the Supplicant; its value is taken when a PTK is instantiated and is sent to the PTK Authenticator.

A PTK is normally derived once for an authentication session. A Supplicant or an Authenticator may use the 4-way handshake to change the PTK. The only time this is specified in this document is when a data integrity failure occurs.

Note: A different ANonce shall be used for every 4-way handshake.

Group transient keys

The Authenticator may derive new Group Transient Keys when it wants to update the Group encryption/integrity keys. The Key Counter is used and incremented whenever a Group Transient Key (GTK) is derived. GTKs are derived from the GMK using a PRF with AA and GNonce as inputs. The size of the PRF computation shall be taken as the size specified by the cipher suite, i.e. 32 + 16 + 16= 64 octets (512 bits) for TKIP and 16 + 16 + 16 = 48 octets (384 bits) for AES.

GTK = PRF-384/512 (GMK, “Group key expansion”, AA || GNonce)

GNonce is a value taken from the Key Counter on the Authenticator; its value is taken when a GTK is instantiated and is sent by the GTK Authenticator.
A Group key update may occur for a number of reasons:

1. A station disassociating or deauthenticating may trigger a Group key update otherwise the disassociated/deauthenticated station can still read broadcast traffic from the network.
2. A data integrity failure shall trigger a Group key update.

4-way handshake

The Authentication Server and Key Management system does not need to rekey for IV exhaustion. The IV space for TKIP is 2^{48} and 2^{47} for AES. This is considered large enough that 802.1X authentication will occur before the IV space is exhausted.

Rather, the 4-way handshake is used to confirm that the Authenticator and Supplicant have the same PMK, that the PMK is live and to derive a fresh PTK. It is also used to tell the Supplicant whether to install the encryption/integrity into the data encryption/integrity engine. The handshake is initiated as part of authenticating a Supplicant and an Authenticator but it shall be initiated if a data integrity failure occurs. The handshake is

Authenticator -> Supplicant: ANonce
Supplicant -> Authenticator: SNonce,
    MIC(EAPOL-Key MIC Key(PTK(ANonce, SNonce)), EAPOL-Key message)
Authenticator -> Supplicant: Install, ANonce,
    MIC(EAPOL-Key MIC Key(PTK(ANonce, SNonce)), EAPOL-Key message)
Supplicant -> Authenticator:
    MIC(EAPOL-Key MIC Key(PTK(ANonce, SNonce)), EAPOL-Key message)

MIC(X, Y) where X is the key and Y is the data that is MICed. Y is the EAPOL-Key as defined in the section on EAPOL-Key MIC. X is the EAPOL-Key MIC key which is taken from the PTK as defined in Figure 5—Complete TKIP Pairwise Key Hierarchy.

ANonce is a nonce from the Authenticator.

SNonce is a nonce from the Supplicant.

Install is true if the Pairwise data encryption and integrity key should be installed in the encryption/integrity engine.

The above messages are sent as EAPOL-Key messages.

The Supplicant can trigger a 4-way handshake by sending an EAPOL-Key message with the Request bit set to 1.

Note: While the MIC calculation is the same in each direction the Ack bit is different in each direction (It is set in messages from the Authenticator and not set in messages from the Supplicant). 4-way handshake requests from the Supplicant have the Request bit set. The Authenticator and Supplicant must check these bits to stop reflection attacks.
The first message is from the Authenticator, contains a nonce and does not contain an integrity check. The Supplicant on receiving the message generates a nonce and then derives a PTK. The Supplicant then sends a message to the Authenticator containing its nonce, with an integrity check using the EAPOL-Key MIC Key from the PTK.
The Authenticator takes the Supplicant nonce and derives the PTK and then checks the integrity check. The Authenticator then sends the third message to the Supplicant containing information whether to install a PTK into the encryption/integrity engine and

an integrity check from the EAPOL-Key MIC Key. The Supplicant then sends the last message to confirm to the Authenticator that the key has been installed if required.
If the Authenticator does not receive a reply to its messages, it should retry three times at one seconds intervals and then disassociate/deauthenticate the station. If the station does not receive the initial message when it expects to, it should disassociate and deauthenticate and try another AP/station.

Note: The timeout should be larger than the short retry timeout.

Note: The Authenticator should ignore EAPOL-Key messages it is not expecting in reply to messages it has sent or EAPOL-Key messages with the Ack bit set. This stops an attacker from sending the first message to the supplicant who responds to the Authenticator. The Supplicant on calculating a new PTK should hold it in temporary storage until the 3rd message is received, after validating the EAPOL-Key MIC using the EAPOL-Key MIC Key from temporary storage, the EAPOL-Key Encryption key can be used to initialized the RC4 engine used to decrypt the Group key EAPOL-Key messages, the EAPOL-Key MIC key must be saved to validate EAPOL-Key messages received in the future. The EAPOL-Key MIC key should be initialized to a random number so attackers cannot send EAPOL-Key messages during initialization. The encryption/integrity keys are configured into the encryption/integrity engine once the reply to the 3rd message is sent. So only the EAPOL-Key MIC key needs to be saved per association. The keeping of the new PTK in temporary storage is so an attacker cannot interfere with normal communication between the Supplicant and Authenticator. An attacker can interfere with a 4-way handshake during the processing of a 4-way handshake. The Authenticator should use the replay counter and Key information field to filter most re-transmit and invalid messages but it is possible for an attacker to mimic an Authenticator that reset during a 4-way handshake. In this case the Authenticator can spot that it is receiving messages that it did not initiate but the 4-way handshake state is incorrect. In this situation the Authenticator will disassociate the station but it should detect and log these occurrences.

Group key update

The Group key update sends a new Group Transient key to the Supplicant. It may be initiated as the final stage of authenticating a Supplicant if the Authenticator is the GTK Authenticator, it shall be initiated if a data integrity failure occurred on the GTK; when a Supplicant disassociates or deauthenticates or on a management event.

Authenticator -> Supplicant: Key Index, Enc(GTK),
    MIC(EAPOL-Key MIC Key(PTK(ANonce, SNonce)), EAPOL-Key message)
Supplicant -> Authenticator:
    MIC(EAPOL-Key MIC Key(PTK(ANonce, SNonce)), EAPOL-Key message)

Key Index is the index in the encryption/integrity engine that the Authenticator wants the key installed

Enc(GTK): The Group transient key is encrypted using the EAPOL-Key encryption key obtained from the PTK which is derived in the 4-way handshake.
MIC(X, Y) where X is the key and Y is the data that is MICed. Y is the EAPOL-Key as defined in the section on EAPOL-Key MIC. X is the EAPOL-Key MIC key which is taken from the PTK as defined in Figure 5—Complete TKIP Pairwise Key Hierarchy

A Group key update can be triggered by the Supplicant by sending an EAPOL-Key message with the Request bit set.

An Authenticator shall do a 4-way handshake before a Group key update if both are required to be done.

Note: The Supplicant does not require the GNonce but the Authenticator should send the Nonce it used to derive the GTK to help with interoperable issues.

Supplicant Request for key update

The Supplicant can request for a key update by sending an EAPOL-Key message with the Request bit set. This is used when the MAC detects a data integrity attack.

Use of the EAPOL-Key Replay Counter

The EAPOL-Key Replay Counter is to help the Supplicant and Authenticator discard invalid messages. The replay counter should be initialized to 0 on association or re-association. The Supplicant when replying to a message from the Authenticator should use the replay counter in the message from the Authenticator. The Authenticator should use this to ignore invalid messages such as late messages from the Supplicant. The Supplicant should also keep track of the replay counter for messages from the Authenticator and ignore messages with invalid replay counter. The local replay counter that is used to check incoming messages should not be updated until the EAPOL-Key MIC is checked and is valid. This means that the Supplicant does not update the replay counter from the first message in the 4-way handshake where no MIC exists in the message, so the Supplicant must allow for the re-transmission of the first message when checking for the replay counter of the third message,. The Supplicant has a replay counter for when it sends request EAPOL-Key messages to the Authenticator and the Authenticator should check this replay counter on receiving Request messages.

EAPOL-Key encoding

The various exchanges described above are encoded using the EAPOL-Key as follows:

EAPOL-Key (S, M, A, T, N, K, ANonce/SNonce, GNonce, MIC, GTK)

Parameters are:

S: Initial Key exchange is complete. This is the EAPOL-Key Information Secure bit.

M: MIC is available in message. This should be set in all messages except the first 4-way handshake message. This is the EAPOL-Key Information Key MIC bit.

A: Response is required to this message. Used when the receiver should respond to this message. This is the EAPOL-Key Information Key Ack bit.

T: Tx/Rx for Group key and Install/Not install for Pairwise key. This is the EAPOL-Key Information Tx/Rx Flag bit.

N: Key Index. Specifies which index should be used for this Group Key. Index 0 shall not be used for Group keys. This is the EAPOL-Key Information key index bits.

K: Key type - P (Pairwise), G (Group). This is the EAPOL-Key Information Key Type bit.

ANonce/SNonce/GNonce: Authenticator/Supplicant/Group Nonce. This is the EAPOL-Key Key Nonce field.

MIC: Integrity check which is generated using the EAPOL-Key MIC Key. This is the EAPOL-Key MIC field.

GTK: Group temporal key which is encrypted using the EAPOL-Key Encryption Key. This is the EAPOL-Key Material field.

EAPOL-Key messages

EAPOL-Key messages are used for two different exchanges:

- 4-way handshake to confirm the PMK at the Supplicant and Authenticator are the same and is live.
- Updating the Group Transient key at the Supplicant.

Annex A. IEEE 802.11/AAA Architecture Description (non-normative)

This Annex provides an overview of the usage of AAA protocols in 802.11 Enhanced Security systems, and provides examples of how the Pairwise Master Key (PMK) can be derived and transported.

Authentication Server and Key management system

This specification does not require use of a backend authentication server, nor does it require use of any authentication, authorization or accounting (AAA) protocol. As a result, any AAA protocol, including RADIUS, Diameter or COPS may be used. This section provides an overview of the AAA authentication and key management process which is used regardless of the AAA protocol that is employed.

There are three logical entities in the authentication and key management system, the Supplicant, Authenticator and Authentication Server. The Authenticator and backend authentication server communicate via the DS. The Supplicant and Authenticator communicate via the WM for in "disconnected" pre-authentication, and via the WM and DS for "connected" pre-authentication. The Supplicant and backend authentication server communicate indirectly using the AAA protocol with the Authenticator acting as a pass-through.

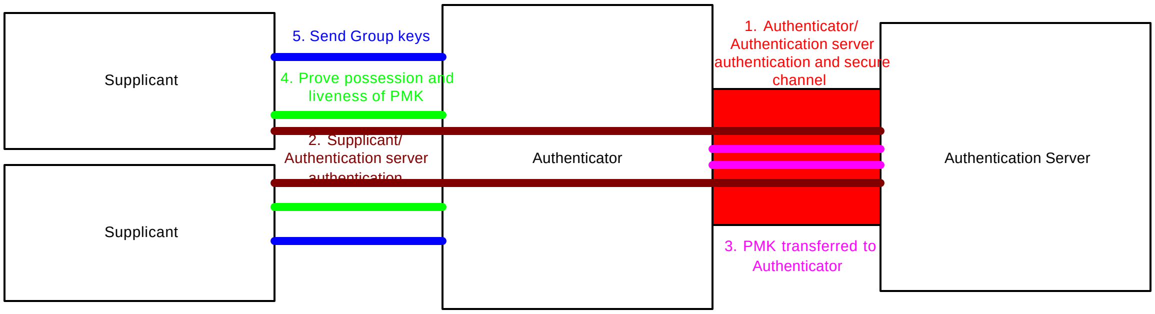
Supplicant
Supplicant
5. Send Group keys
4. Prove possession and liveness of PMK
2. Supplicant/ Authentication server authentication
Authenticator
1. Authenticator/ Authentication server authentication and secure channel
3. PMK transferred to Authenticator
Authentication Server

Figure 9 – Relationship between Supplicant, Authenticator and Backend Authentication Server

As part of the AAA exchange, the following operations occur:

1. The Authenticator and Authentication Server mutually authenticate. Where RADIUS is used, this is accomplished via a shared secret (described in RFC 2865 and 2866) and/or IPsec (described in RFC 3162). For Diameter, authentication is accomplished using TLS or IPsec. Within RADIUS, replay protection is provided via IPsec where this is employed or via the Request Authenticator which must be globally and temporally unique. Within Diameter, replay protection is provided by IPsec or TLS as well as by inclusion of Event-Timestamp and Nonce AVPs.
2. The Supplicant and Authentication Server mutually authenticate and generate a Master Key. The authentication is carried over the mutually authenticated channel created between the Authenticator and the Authentication Server.

3. A Pairwise master key (PMK) is generated for use between the Supplicant and Authentication Server. The PMK may be generated from the master key that is obtained from the Supplicant/Authentication Server authentication, as it is within EAP-TLS (RFC 2716), or it may be generated by some other means (e.g. Randomly chosen by the Authentication Server) and protected in transport from the Authentication Server to the Supplicant using material derived from the master key.
4. The Authentication Server transports the PMK to the Authenticator over the mutually authenticated channel created between the Authenticator and Authentication Server. In order to protect the PMK from compromise, it is encrypted in some fashion. For RADIUS, this can be accomplished using the shared secret as described in RFC 2548, or alternatively, IPsec ESP with non-null transform can be used as described in RFC 3162. With Diameter, protection can be provided via IPsec ESP with non-null transform, with TLS, or via use of CMS. When a pre-shared key is used, it is used directly as the PMK and this step is skipped.
5. A 4-way handshake occurs between the Supplicant and Authenticator to confirm the existence of the PMK, to confirm that the knowledge of PMK is current, and to derive the Pairwise transient key from the PMK. EAPOL-Key messages are used to carry out this exchange. However, with IEEE 802.1X pre-authentication, the 4-way handshake is not used to install the encryption and integrity keys into the encryption/integrity engine if required nor to confirm installation of the keys. Rather, this is accomplished via the authenticated association/reassociation exchange.
6. The Group Transient key is sent from the Authenticator to the Supplicant to allow the Supplicants to transmit and receive broadcast messages and optionally to be used to send unicast packets to the Authenticator. EAPOL-Key messages are used to carry out this exchange.

Since the Supplicant/Authentication Server authentication is carried over the Authenticator/Authentication server mutually authenticated channel supporting replay protection, and since the EAP conversation between the Supplicant and Authentication Server is required to be authenticated, replay and integrity protected on a per-packet basis, the Authentication Server can guarantee that the Authenticator it is communicating with is the same Authenticator that the Supplicant is communicating with.

The Authenticator/Authentication Server authentication protocol is not specified here but the protocol must meet the following requirements:

1. Per-packet mutual authentication and replay protection between the Authenticator and Authentication Server.
2. Support for tunneling of EAP authentication between the Supplicant and Authentication Server.
3. Generation and transport of the PMK from the Authentication Server to the Authenticator for use in communication with the Supplicant.

RFC 2548 Attributes

If the protocol between the Authenticator or AP and Authentication Server is RADIUS then the MS-MPPE-Recv-Key (vendor-id = 17) attribute (See RFC2548 Section 2.4.3) MAY be used to transport the Pairwise Master Key (PMK). The PMK and any derived keys shall not be used any longer than the Session-Timeout attribute + the IEEE 802.1X reAuthMax*txPeriod values.

> Note: If the Radius Session-Timeout attribute is not in the Radius Accept message the PMK lifetime is infinite.