Robust Security Network (RSN) Fast BSS Transition (FT)

White Paper

September 2008
Version 2.02

Author:
Devin Akin, CTO
The CWNP Program
Devin@cwnp.com

Technical Reviewer:

David Coleman, CWNE
AirSpy Training, Inc.
David@AirSpy.com

Introduction

The 802.11 standard describes the generic process for 802.1X/EAP authentication and key management processes, which includes both generation and distribution of Master Session, Pairwise Master (and Transient), and Group Master (and Temporal) keys. For a detailed description of 802.11 Authentication and Key Management (AKM), refer to a whitepaper by the same name by Devin Akin, May 2005 at the CWNP Learning Center at .cwnp.com.

This whitepaper describes specific features found in the 802.11 standard that are designed to aid wireless client stations in fast BSS transition (FT). Applications such as wireless Voice over Wi-Fi (VoWiFi) are being deployed at a surprising rate, and interoperable FT mechanisms should be deployed by all vendors as a benefit to end users. Most vendors, with notable exceptions, now support Opportunistic Key Caching (also called Proactive Key Caching or Opportunistic PMK Caching). The IEEE 802.11r-2008 amendment was ratified on July 15, 2008, giving the industry a standard around which to build products and interoperability certifications.

It is the goal of this whitepaper to explain what FT mechanisms are offered by the 802.11 standard and what additional mechanisms, though not specified by the standard, are deployed across a wide vendor base in today's market. Additionally, intra-controller and inter-controller scenarios will be discussed.

Some useful definitions are:

Pairwise Master Key (PMK) – The highest order key used within this standard. The PMK may be derived from a key generated by an Extensible Authentication Protocol (EAP) method or may be obtained directly from a preshared key (PSK).

Pairwise Master Key Security Association (PMKSA) - The context resulting from a successful IEEE 802.1X authentication exchange between the peer and Authentication Server (AS) or from a preshared key (PSK).

PMK Identifier (PMKID) – A number referring to 1) a cached PMKSA that has been obtained through preauthentication with the target AP or 2) a cached PMKSA from an EAP authentication, or 3) a PMKSA derived from a PSK for the target AP.

Pairwise Transient Key (PTK) - A value that is derived from the pairwise master key (PMK), Authenticator address (AA), Supplicant address (SPA), Authenticator nonce (ANonce), and Supplicant nonce (SNonce) using the pseudo-random function (PRF) and that is split up into as many as five keys, i.e., temporal encryption key, two temporal message integrity code (MIC) keys, EAPOL-Key encryption key (KEK), EAPOL-Key confirmation key (KCK).

Pairwise Transient Key Security Association (PTKSA) - The context resulting from a successful 4-Way Handshake exchange between the peer and Authenticator.

Preauthentication – The act of authenticating with an AP to which the STA is currently not associated while the STA is already associated with an AP. If the authentication is left until reassociation time, this may impact the speed with which a STA can reassociate between APs, limiting BSS-transition mobility performance. The use of preauthentication takes the overhead of the authentication service out of the time-critical reassociation process.

Robust Security Network (RSN) – A security network that allows only the creation of robust security network associations (RSNAs). An RSN can be identified by the indication in the RSN information element (IE) of Beacon frames that the group cipher suite specified is not wired equivalent privacy (WEP).

4-Way Handshake – A pairwise key management protocol defined by the 802.11 standard. This handshake confirms mutual possession of a pairwise master key (PMK) by two parties and distributes a group temporal key (GTK).

In this document, we will insert an italicized 802.11 standard section, and then discuss the relevant parts immediately below it. We often highlight parts of the italicized sections in green for emphasis. Our discussions will include screenshots of WLAN protocol analyzer traces.

How PMKSAs and PMKIDs are formed and used

When an RSNA-enabled client station successfully 802.1X/EAP authenticates to an RSNA-enabled AP (or WLAN controller), and before the 4-Way Handshake, a PMKSA is formed. The PMKSA may be cached at each node if PMK Caching is enabled on the client device and the AP. Each PMKSA has a unique identifier called a PMKID. The following 802.11 section outlines when and where the PMKID is used.

7.3.2.25.4 PMKID

The PMKID Count and List fields shall be used only in the RSN information element in the (Re)Association Request frame to an AP. The PMKID Count specifies the number of PMKIDs in the PMKID List field. The PMKID list contains 0 or more PMKIDs that the STA believes to be valid for the destination AP. The PMKID can refer to

a) A cached PMKSA that has been obtained through preauthentication with the target AP
b) A cached PMKSA from an EAP authentication
c) A PMKSA derived from a PSK for the target AP

NOTE - A STA can choose not to insert a PMKID in the PMKID List field if the STA does not want to use that PMKSA.

The section above specifies that Association Request and Reassociation Request frames may carry the PMKID in their RSN Information Element (IE). RSN IEs are included in specific 802.11 frames and carries security information about the BSS. From reading this section, we also find out:

- 802.1X/EAP authentications produce PMKSAs
- PMKSAs can be derived from a Preshared Key (PSK)
- STAs and APs may both cache PMKSAs (along with their identifier)
- STAs can choose not to use the PMKID in the (Re)Association Request frame.

If you're wondering what is in a PMKSA, section 8.4.1.1.1 below has your answer.

8.4.1.1.1 PMKSA

When the PMKSA is the result of a successful IEEE 802.1X authentication, it is derived from the EAP *authentication and authorization parameters provided by the AS. This security association is bidirectional. In other words, both parties use the information in the security association for both sending and receiving.*

The PMKSA is created by the Supplicant's SME when the EAP authentication completes successfully or the PSK is configured. The PMKSA is created by the Authenticator's SME when the PMK is created from the keying information transferred from the AS or the PSK is configured. The PMKSA is used to create the PTKSA. PMKSAs are cached for up to their lifetimes. The PMKSA consists of the following elements:
— PMKID, as defined in 8.5.1.2. The PMKID identifies the security association.
— Authenticator MAC address.
— PMK
— Lifetime, as defined in 8.5.1.2.
— AKMP.
— All authorization parameters specified by the AS or local configuration. This can include parameters such as the STA's authorized SSID.

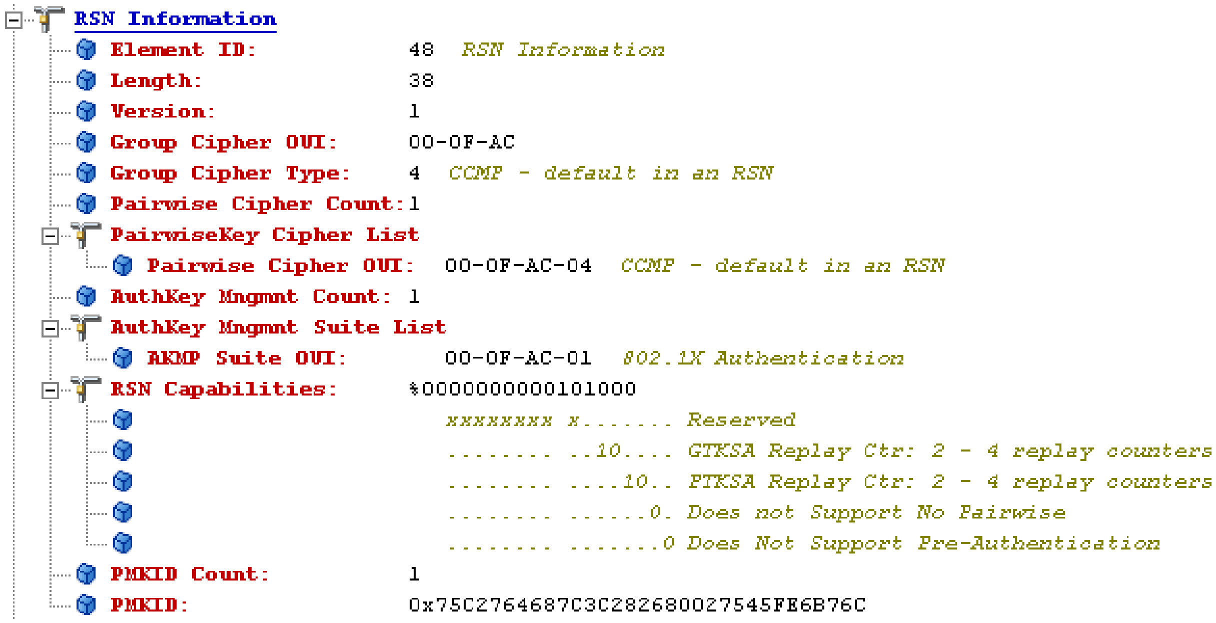
Definition: SME, 802.11-1999 (R2003), Section 10.1

> *In order to provide correct MAC operation, a station management entity (SME) shall be present within each STA. The SME is a layer-independent entity that may be viewed as residing in a separate management plane or as residing "off to the side." The exact functions of the SME are not specified in this standard, but in general this entity may be viewed as being responsible for such functions as the gathering of layer-dependent status from the various layer management entities, and similarly setting the value of layer-specific parameters. SME would typically perform such functions on behalf of general system management entities and would implement standard management protocols.*

Notice the items highlighted in green in this section. It is of particular importance to understand that the PMKSA contains the PMK, the Authenticator MAC address, and the PMKID. Sometimes "PMK" and "PMKSA" are referred to interchangeably by vendors and various documentation, but they are actually different entities within the 802.11 standard.

In the following figure, you see the RSN IE of an Association Request frame, which includes the PMKID Count and PMKID List (called just "PMKID" in this analyzer). You can see in this Association Request frame that only one PMKID is in the list. In the next section, 8.4.1.2.1, you will see that 1 or more PMKIDs may be in the PMKID List, but from practical experience, we can tell you that there's almost always just one in the list. The PMKID listed in the PMKID List is the one cached at the client station specifically for the access point radio (BSSID) to which it wishes to authenticate in the (Re)Association Request frame.

FIGURE 1 (OmniPeek) – Association Request Frame (RSN IE)



```
RSN Information
  Element ID:              48   RSN Information
  Length:                  38
  Version:                 1
  Group Cipher OUI:        00-0F-AC
  Group Cipher Type:       4   CCMP - default in an RSN
  Pairwise Cipher Count:1
  PairwiseKey Cipher List
    Pairwise Cipher OUI:   00-0F-AC-04   CCMP - default in an RSN
  AuthKey Mngmnt Count: 1
  AuthKey Mngmnt Suite List
    AKMP Suite OUI:        00-0F-AC-01   802.1X Authentication
  RSN Capabilities:        %0000000000101000
                              xxxxxxxx x....... Reserved
                              ........ ..10.... GTKSA Replay Ctr: 2 - 4 replay counters
                              ........ ....10.. PTKSA Replay Ctr: 2 - 4 replay counters
                              ........ ......0. Does not Support No Pairwise
                              ........ .......0 Does Not Support Pre-Authentication
  PMKID Count:             1
  PMKID:                   0x75C2764687C3C282680027545FE6B76C
```

Let's take a look at 8.4.1.2.1 now. Notice in the second section, it says, "it can include one or more PMKIDs in the RSN information element of its (Re)Association Request frame." Figure 1 illustrates a PMKID Count set to 1.

8.4.1.2.1 Security association in an ESS

A STA roaming within an ESS establishes a new PMKSA by one of three schemes:

— *In the case of (re)association followed by IEEE 802.1X or PSK authentication, the STA repeats the same actions as for an initial contact association, but its Supplicant also deletes the PTKSA when it roams from the old AP. The STA's Supplicant also deletes the PTKSA when it disassociates/deauthenticates from all basic service set identifiers (BSSIDs) in the ESS.*

— *A STA (AP) can retain PMKs for APs (STAs) in the ESS to which it has previously performed a full IEEE 802.1X authentication. If a STA wishes to roam to an AP for which it has cached one or more PMKSAs, it can include one or more PMKIDs in the RSN information element of its (Re)Association Request frame. An AP whose Authenticator has retained the PMK for one or more of the PMKIDs can skip the 802.1X authentication and proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake. If none of the PMKIDs of the cached PMKSAs matches any of the supplied PMKIDs, then the Authenticator shall perform another IEEE 802.1X authentication. Similarly, if the STA fails to send a PMKID, the STA and AP must perform a full IEEE 802.1X authentication*

— *A STA already associated with the ESS can request its IEEE 802.1X Supplicant to authenticate with a new AP before associating to that new AP. The normal operation of the DS via the old AP provides the communication between the STA and the new AP. The STA's IEEE 802.11 management entity delays reassociation with the new AP until IEEE 802.1X authentication completes via the DS. If IEEE 802.1X authentication completes successfully, then PMKSAs shared between the new AP and the STA will be cached, thereby enabling the possible usage of reassociation without requiring a subsequent full IEEE 802.1X authentication procedure.*

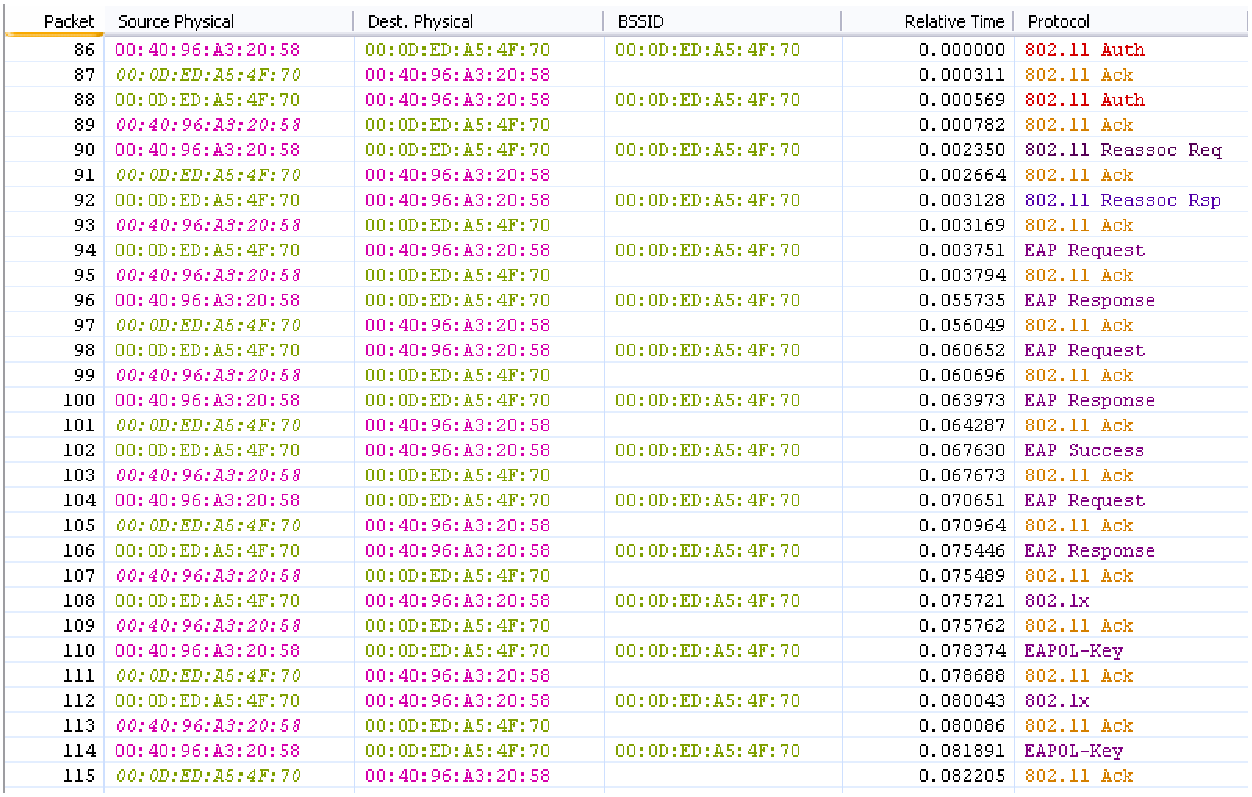
NOTE: *It is possible for more than one PMKSA to exist. As an example, a second PMKSA may come into existence through PMKSA caching. A STA might leave the ESS and flush its cache. Before its PMKSA expires in the AP's cache, the STA returns to the ESS and establishes a second PMKSA from the AP's perspective.*

Section 8.4.1.2.1 above tells us that a STA roaming within an ESS may establish a new PMKSA by any one of three methods:

1) Full 802.1X/EAP authentication
2) PMK caching
3) Preauthentication via the Distribution System

Each of the main points is highlighted in green in the section above. Let's address all three scenarios. First, when a STA initially authenticates to an ESS, it must perform a full 802.1X/EAP authentication. This process can be very time-consuming depending on the EAP type and network load conditions. Some EAP authentication exchanges may take over 1 full second to complete. Figure 2 illustrates a full 802.1X/LEAP (the shortest, fastest wireless-useable EAP type) reassociation that takes 82.2 ms under optimal lab conditions where only one client device is present on the channel and the RADIUS authentication server is integrated into the AP (which is under zero load). Notice the quoted EAP processing time from Cisco Systems' application note Cisco Fast Secure Roaming following Figure 2.

FIGURE 2 (OmniPeek) – 802.1X/EAP Authentication Sequence



Packet	Source Physical	Dest. Physical	BSSID	Relative Time	Protocol
86	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.000000	802.11 Auth
87	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.000311	802.11 Ack
88	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.000569	802.11 Auth
89	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.000782	802.11 Ack
90	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.002350	802.11 Reassoc Req
91	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.002664	802.11 Ack
92	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.003128	802.11 Reassoc Rsp
93	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.003169	802.11 Ack
94	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.003751	EAP Request
95	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.003794	802.11 Ack
96	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.055735	EAP Response
97	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.056049	802.11 Ack
98	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.060652	EAP Request
99	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.060696	802.11 Ack
100	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.063973	EAP Response
101	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.064287	802.11 Ack
102	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.067630	EAP Success
103	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.067673	802.11 Ack
104	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.070651	EAP Request
105	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.070964	802.11 Ack
106	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.075446	EAP Response
107	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.075489	802.11 Ack
108	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.075721	802.1x
109	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.075762	802.11 Ack
110	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.078374	EAPOL-Key
111	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.078688	802.11 Ack
112	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.080043	802.1x
113	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.080086	802.11 Ack
114	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.081891	EAPOL-Key
115	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.082205	802.11 Ack

Cisco LEAP Authentication Prior to Fast Secure Roaming
A Cisco LEAP client using Cisco IOS Software version 12.2(8)JA or earlier needs to perform a full Cisco LEAP reauthentication each time it roams. A Cisco LEAP reauthentication requires:

- *A minimum of 100ms*
- *An average of ~600ms*
- *Up to 1.2seconds +*

The timeframes above are in addition to the channel-scanning portion of the L2 roam. Cisco LEAP authentication takes this much time because it requires three roundtrips to a Remote Authentication Dial-In User Service (RADIUS) server using the following process:

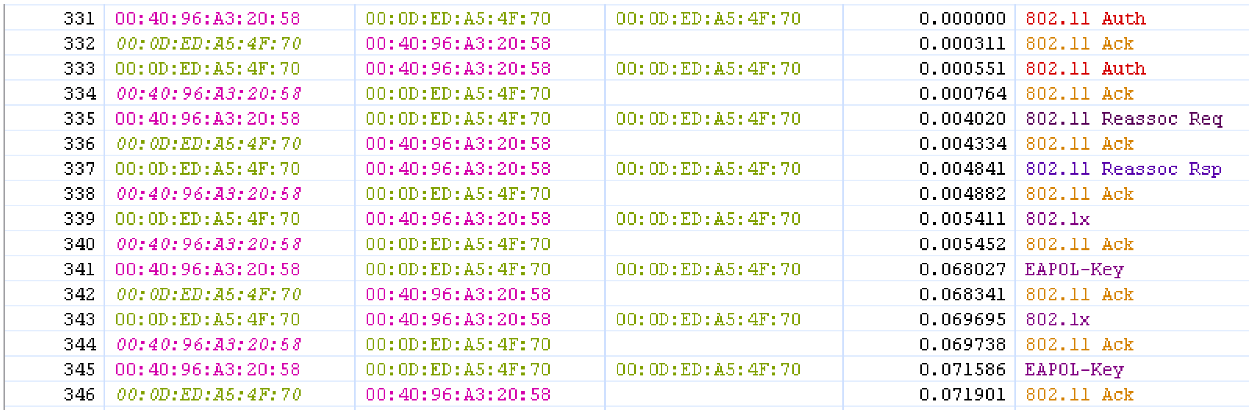
- *Client sends identity, Cisco Secure Access Control Server (ACS) or RADIUS Server sends challenge*
- *Client sends challenge response, Cisco Secure ACS sends success*
- *Client sends challenge, Cisco Secure ACS sends challenge response*

In addition to network transit times, each of these roundtrip transactions requires time-consuming cryptographic calculations, hence the total times quoted above.

Cisco Fast Secure Roaming
Application Note, Page 12
Bruce McMurdo, Author
©2004 Cisco Systems

Second, PMK Caching (often called “fast/secure roam-back”) is the process of keeping a PMKSA for a period of time after successfully forming an authentication. When a station wishes to roam to another AP, it looks up the BSSID of that AP in its cache hoping to find a PMKID associated with it. If it finds a cached PMKID, then by definition it has a cached PMKSA because the PMKID is part of the PMKSA. If the station wishes to include the PMKID in the (Re)Association Request frame, then it may place it in the RSN IE PMKID List field. Using this feature is optional and up to the discretion of the client station manufacturer. The point of placing the PMKID into the RSN IE of the (Re)Association Request frame is that if the AP also has that particular PMKSA cached, an 802.1X/EAP authentication may be skipped and the authentication may proceed directly to the 4-Way Handshake. Going only through the 4-Way Handshake is many times faster than a full 802.1X/EAP authentication because communication with the RADIUS server is not part of the process. Reassociations with 4-Way Handshakes are together typically below 100 ms, fast enough for VoWiFi and other time-sensitive applications. Figure 3 illustrates an 802.11 4-Way Handshake following a reassociation that takes place in 71.9 ms (measured from the beginning of the first Open System Authentication frame to the ACK following the final EAPoL-Key frame.

FIGURE 3 (OmniPeek) – Reassociation Using a Cached PMKSA



331	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.000000	802.11 Auth
332	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.000311	802.11 Ack
333	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.000551	802.11 Auth
334	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.000764	802.11 Ack
335	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.004020	802.11 Reassoc Req
336	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.004334	802.11 Ack
337	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.004841	802.11 Reassoc Rsp
338	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.004882	802.11 Ack
339	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.005411	802.1x
340	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.005452	802.11 Ack
341	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.068027	EAPOL-Key
342	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.068341	802.11 Ack
343	00:0D:ED:A5:4F:70	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	0.069695	802.1x
344	00:40:96:A3:20:58	00:0D:ED:A5:4F:70		0.069738	802.11 Ack
345	00:40:96:A3:20:58	00:0D:ED:A5:4F:70	00:0D:ED:A5:4F:70	0.071586	EAPOL-Key
346	00:0D:ED:A5:4F:70	00:40:96:A3:20:58		0.071901	802.11 Ack

Third, preauthentication with an AP via the Ethernet network infrastructure enables a STA to stay on-channel while authenticating with nearby APs through the AP to which it is currently connected. Preauthentication via the network infrastructure is also addressed in other 802.11 sections, such as 8.4.6 below. The note found in section 8.4.6 clearly shows us that a STA must perform preauthentication over the wired network infrastructure through its current AP.

8.4.6 RSNA authentication in an ESS

NOTE: *A roaming STA's IEEE 802.1X Supplicant may initiate preauthentication by sending an EAPOL-Start message via its old AP, through the DS, to a new AP.*

Preauthentication for Fast BSS Transition

Let's take a look at the details of how Preauthentication works in the 802.11 standard. The 802.11-2007 standard, which includes the 802.11i amendment, gives plenty of detail in section 8.4.6.1.

8.4.6.1 Preauthentication and RSNA key management

A STA shall not use preauthentication except when pairwise keys are employed. Preauthentication shall not be used unless the new AP advertises the preauthentication capability in the RSN information element. When preauthentication is used, then

a) Authentication is independent of roaming.
b) The STA's Supplicant may authenticate with multiple APs at a time.

NOTE: *Preauthentication can be useful as a performance enhancement, as reassociation will not include the protocol overhead of a full reauthentication when it is used.*

Preauthentication uses the IEEE 802.1X protocol and state machines with EtherType 88-C7, rather than the EtherType 88-8E. Only IEEE 802.1X frame types EAP-Packet and EAPOL-Start are valid for preauthentication.

NOTE: *Some IEEE 802.1X Authenticators may not bridge IEEE 802.1X frames, as suggested in C.1.1 of IEEE P802.1X-REV. Preauthentication uses a distinct EtherType to enable such devices to bridge preauthentication frames.*

A STA’s Supplicant can initiate preauthentication when it has completed the 4-Way Handshake and configured the required temporal keys. To effect preauthentication, the STA’s Supplicant sends an IEEE 802.1X EAPOL-Start message with the DA being the BSSID of a targeted AP and the RA being the BSSID of the AP with which it is associated. The target AP shall use a BSSID equal to the MAC address of its Authenticator.

As preauthentication frames do not use the IEEE 802.1X EAPOL EtherType field, the AP with which the STA is currently associated need not apply any special handling. The AP and the MAC in the STA shall handle these frames in the same way as other frames with arbitrary EtherType field values that require distribution via the DS.

An AP’s Authenticator that receives an EAPOL-Start message via the DS may initiate IEEE 802.1X authentication to the STA via the DS. The DS will forward this message to the AP with which the STA is associated. The result of preauthentication may be a PMKSA, if the IEEE 802.1X authentication completes successfully. If preauthentication produces a PMKSA, then, when the Supplicant's STA associates with the preauthenticated AP, the Supplicant can use the PMKSA with the 4-Way Handshake.

Successful completion of EAP authentication over IEEE 802.1X establishes a PMKSA at the Supplicant. The Authenticator has the PMKSA when the AS completes the authentication, passes the keying information (the authentication, authorization, and accounting [AAA] key, a portion of which is the PMK) to the Authenticator, and the Authenticator creates a PMKSA using the PMK. The PMKSA is inserted into the PMKSA cache. Therefore, if the Supplicant and Authenticator lose synchronization with respect to the PMKSA, the 4-Way Handshake will fail. In such circumstances, the MIB variable dot11RSNAStats-4WayHandshakeFailures shall be incremented.

A STA’s Supplicant may initiate preauthentication with any AP within its present ESS with preauthentication enabled regardless of whether the targeted AP is within radio range. Even if a STA has preauthenticated, it is still possible that it may have to undergo a full IEEE 802.1X authentication, as the AP’s Authenticator may have purged its PMKSA due to, for example, unavailability of resources, delay in the STA associating, etc.

While this text in this section is very straight forward, it would be helpful to have graphics representing some of these steps. The next section will provide the graphics.

Preauthentication over the Distribution System

Figure 4 attempts to illustrate details outlined in section 8.4.6.1 above regarding preauthentication via the distribution system (DS).

FIGURE 4 – Preauthentication (Conceptual)

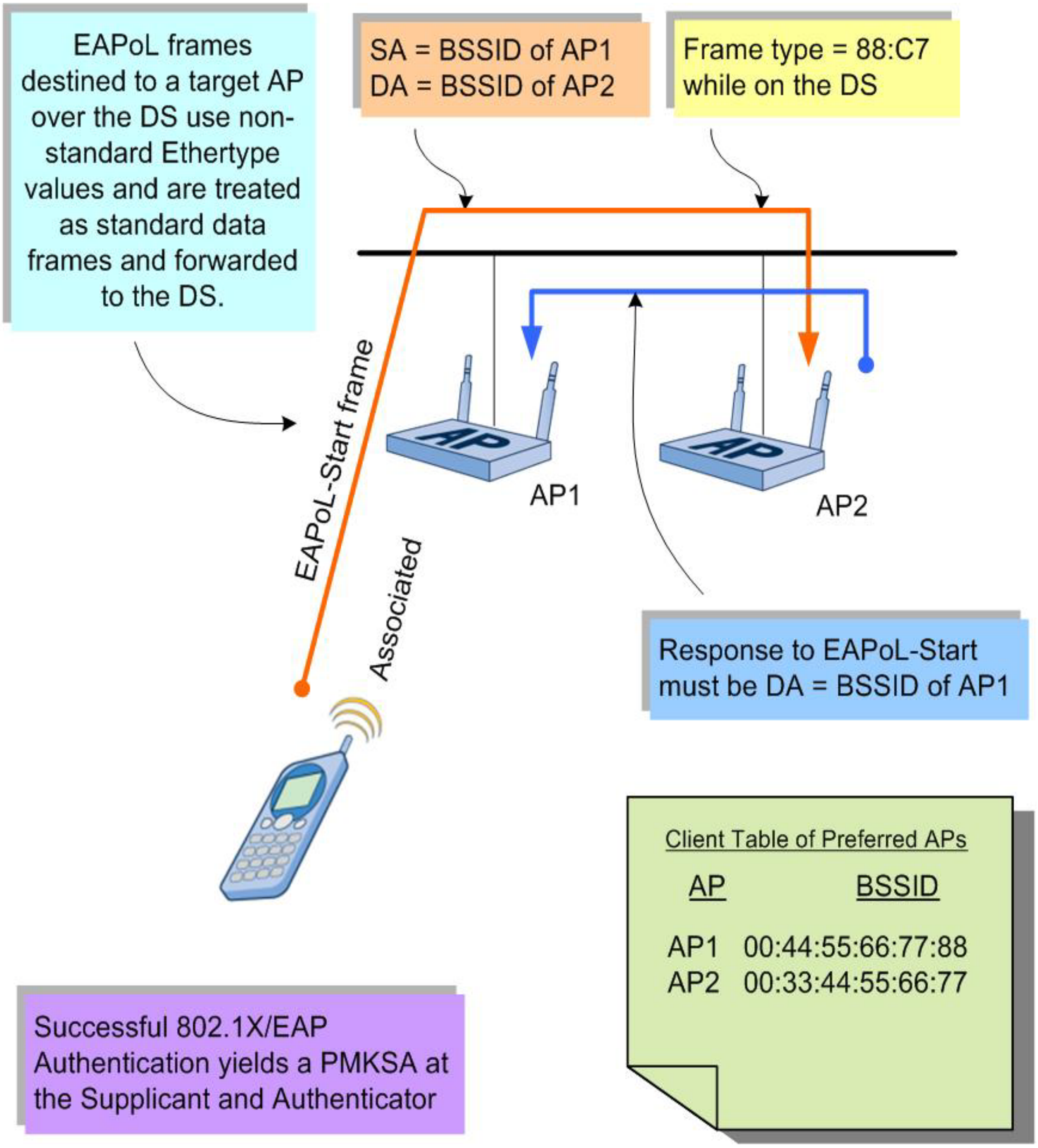
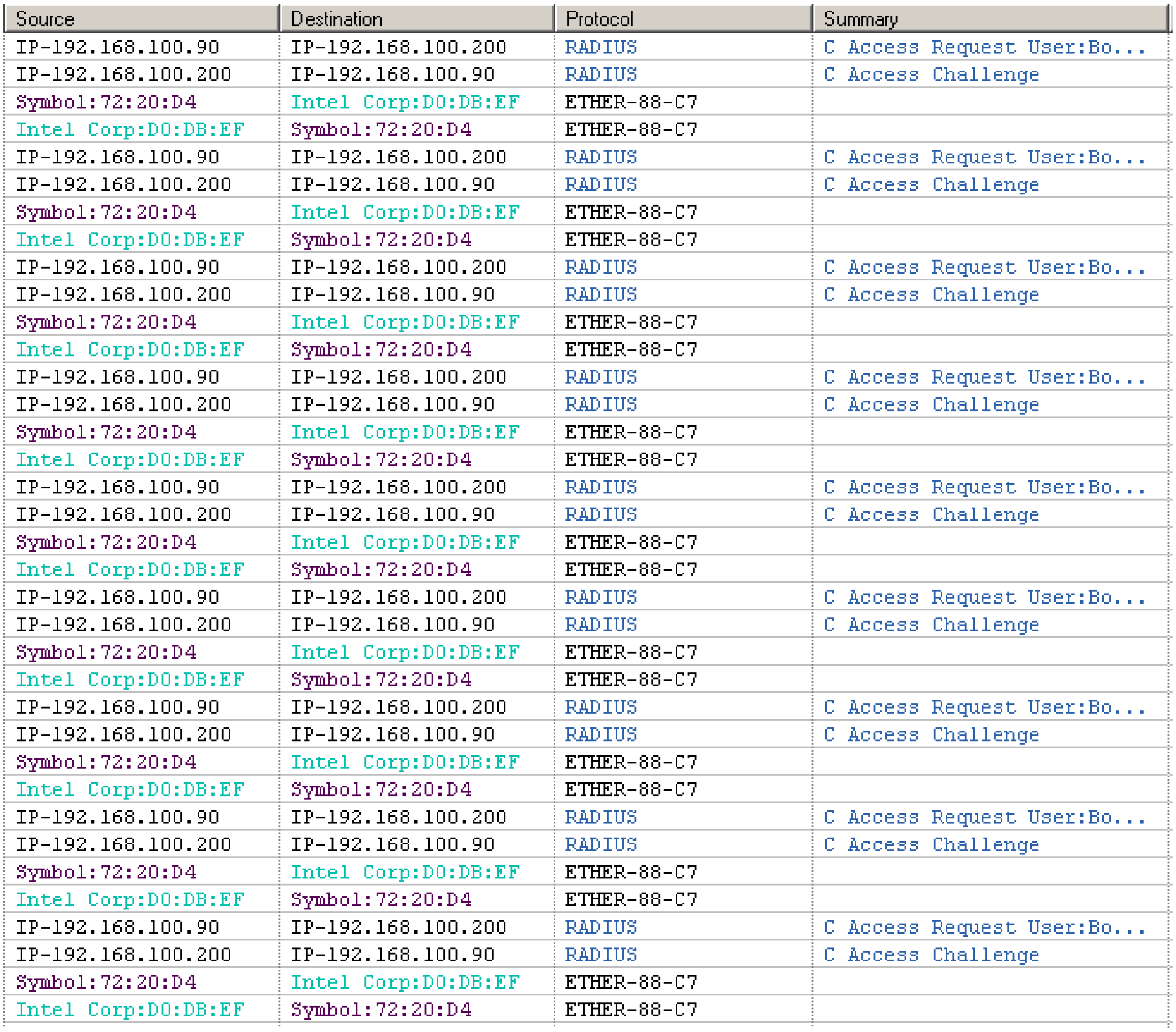

EAPoL frames destined to a target AP over the DS use non-standard Ethertype values and are treated as standard data frames and forwarded to the DS.
SA = BSSID of AP1
DA = BSSID of AP2
Frame type = 88:C7 while on the DS
EAPoL-Start frame
Associated
AP
AP1
AP
AP2
Response to EAPoL-Start must be DA = BSSID of AP1
Client Table of Preferred APs
AP
BSSID
AP1 00:44:55:66:77:88
AP2 00:33:44:55:66:77
Successful 802.1X/EAP Authentication yields a PMKSA at the Supplicant and Authenticator

Figure 5 shows preauthentication while it's happening on an Ethernet protocol analyzer over the Ethernet. Not all frames in the exchange are shown in this screenshot.

FIGURE 5 (OmniPeek) – Preauthentication Frame Exchange and RADIUS Authentication



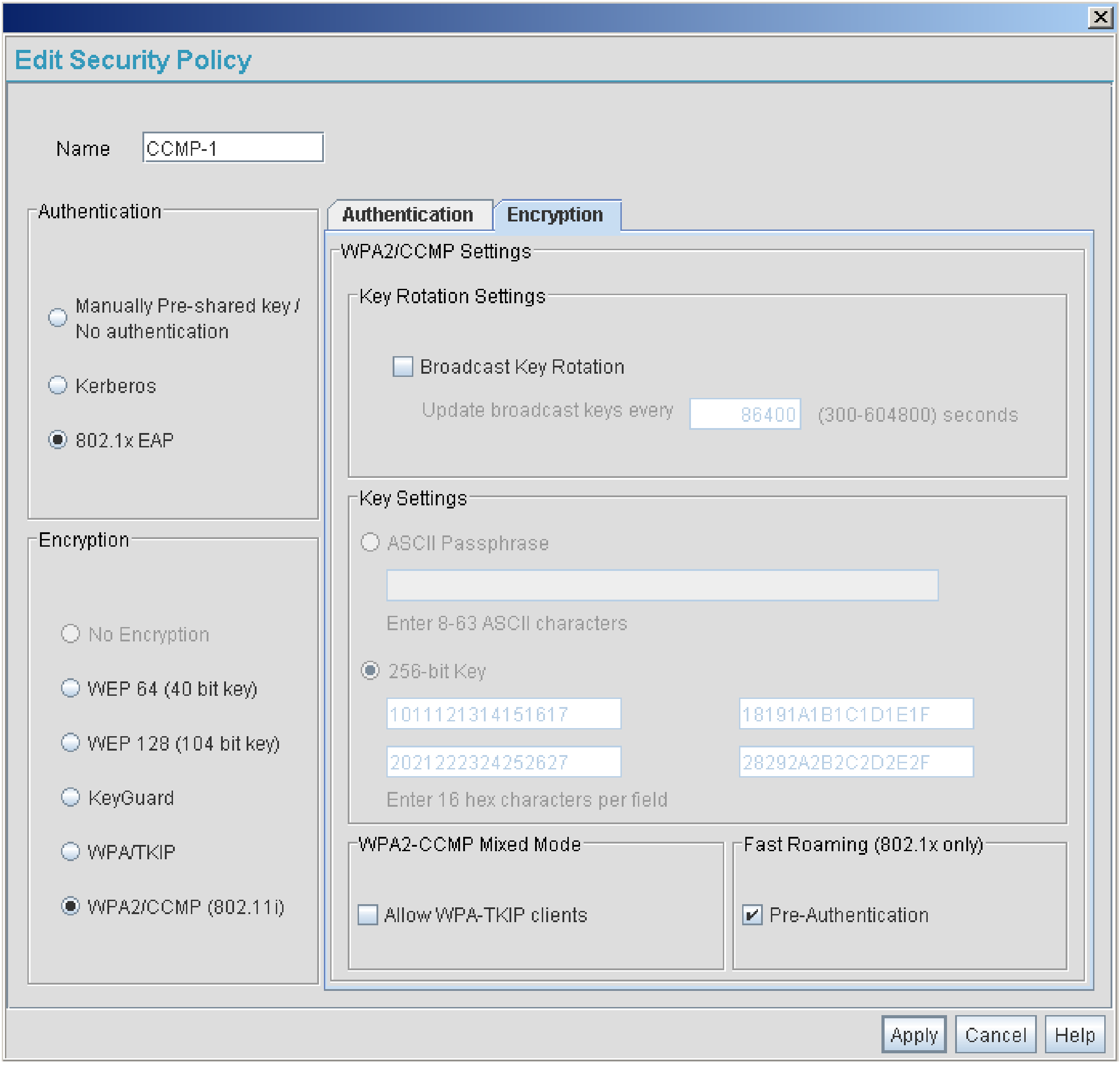
Source	Destination	Protocol	Summary
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	
IP-192.168.100.90	IP-192.168.100.200	RADIUS	C Access Request User:Bo...
IP-192.168.100.200	IP-192.168.100.90	RADIUS	C Access Challenge
Symbol:72:20:D4	Intel Corp:D0:DB:EF	ETHER-88-C7	
Intel Corp:D0:DB:EF	Symbol:72:20:D4	ETHER-88-C7	

When a STA has previously performed a full 802.1X/EAP authentication with an AP (whether during association, reassociation, or preauthentication), a PMKSA exists both in the STA and the AP (provided the cache in each has not been dumped). This is true for each STA/AP authentication that has taken place in the ESS. There are rules regarding use of cached PMKSAs, and these are found in 802.11 section 8.4.6.2, discussed in a later section.

Preauthentication Configuration on Infrastructure Devices

Let’s take a look at an autonomous access point that supports Preauthentication. Notice the checkbox at the bottom right enabling/disabling Preauthentication.

FIGURE 6 (Motorola AP-5131 Autonomous AP) – Preauthentication Feature



Edit Security Policy

Name: CCMP-1

Authentication
- ○ Manually Pre-shared key / No authentication
- ○ Kerberos
- ● 802.1x EAP

Encryption
- ○ No Encryption
- ○ WEP 64 (40 bit key)
- ○ WEP 128 (104 bit key)
- ○ KeyGuard
- ○ WPA/TKIP
- ● WPA2/CCMP (802.11i)

WPA2/CCMP Settings

Key Rotation Settings
- ☐ Broadcast Key Rotation
- Update broadcast keys every 86400 (300-604800) seconds

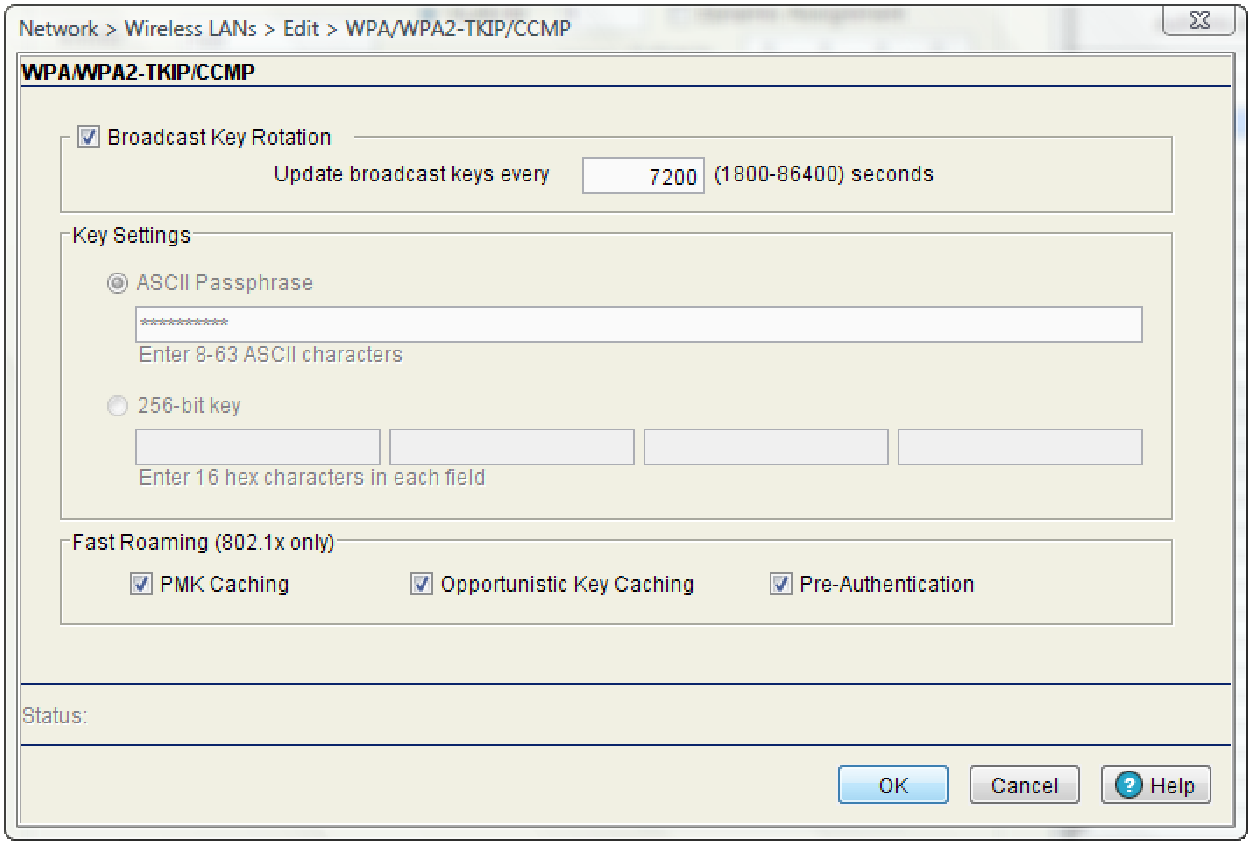
Key Settings
- ○ ASCII Passphrase
- Enter 8-63 ASCII characters
- ● 256-bit Key
- 1011121314151617 18191A1B1C1D1E1F
- 2021222324252627 28292A2B2C2D2E2F
- Enter 16 hex characters per field

WPA2-CCMP Mixed Mode
- ☐ Allow WPA-TKIP clients

Fast Roaming (802.1x only)
- ☑ Pre-Authentication

Preauthentication was designed for use by autonomous APs but works equally well between WLAN controllers to solve the same problem. Notice in figure 7 that Preauthentication is available in some controllers for the purpose of fast/secure roaming between controllers.

FIGURE 7 (Motorola WS5100 WLAN Controller) – Preauthentication Feature



Network > Wireless LANs > Edit > WPA/WPA2-TKIP/CCMP

WPA/WPA2-TKIP/CCMP

☑ Broadcast Key Rotation

Update broadcast keys every 7200 (1800-86400) seconds

Key Settings

● ASCII Passphrase

****

Enter 8-63 ASCII characters

○ 256-bit key

Enter 16 hex characters in each field

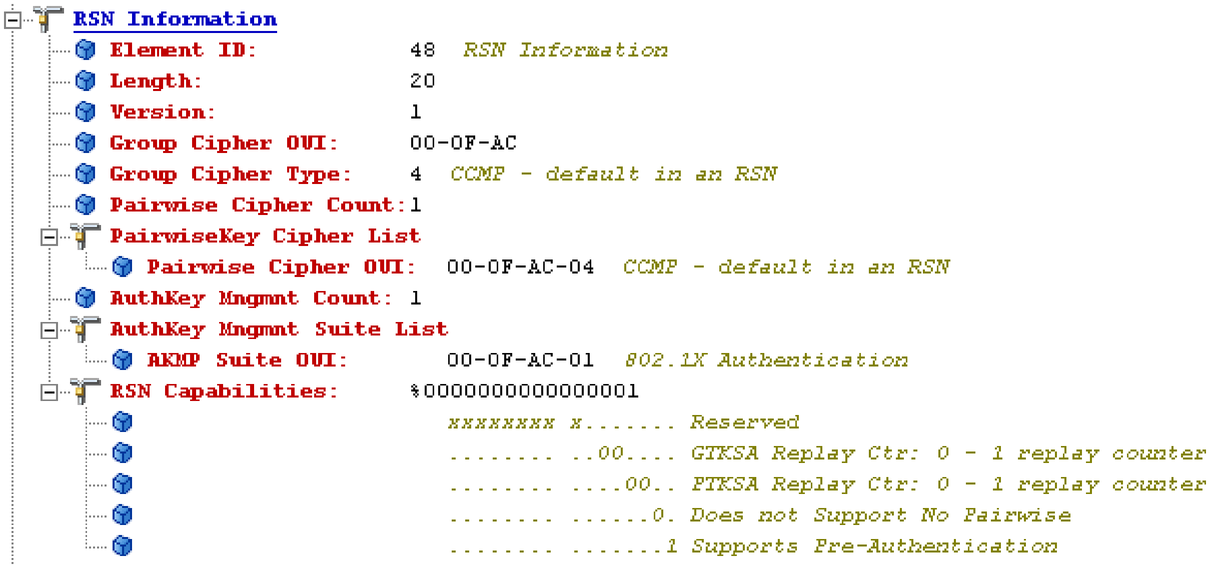
Fast Roaming (802.1x only)

☑ PMK Caching ☑ Opportunistic Key Caching ☑ Pre-Authentication

Status:

Let’s take a look at preauthentication with a WLAN protocol analyzer. In the RSN Capabilities portion of the RSN IE, Preauthentication is supported, as is illustrated with the 1.

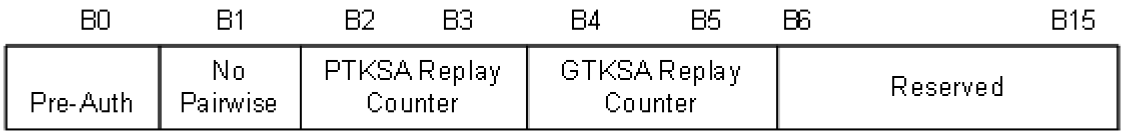
FIGURE 8 (OmniPeek) – Preauthentication Enabled



```
RSN Information
  Element ID:              48  RSN Information
  Length:                  20
  Version:                 1
  Group Cipher OUI:        00-0F-AC
  Group Cipher Type:       4  CCMP - default in an RSN
  Pairwise Cipher Count:1
  PairwiseKey Cipher List
    Pairwise Cipher OUI:   00-0F-AC-04  CCMP - default in an RSN
  AuthKey Mngmnt Count:  1
  AuthKey Mngmnt Suite List
    AKMP Suite OUI:        00-0F-AC-01  802.1X Authentication
  RSN Capabilities:        %0000000000000001
                             xxxxxxxx x....... Reserved
                             ........ ..00.... GTKSA Replay Ctr: 0 - 1 replay counter
                             ........ ....00.. PTKSA Replay Ctr: 0 - 1 replay counter
                             ........ ......0. Does not Support No Pairwise
                             ........ .......1 Supports Pre-Authentication
```

7.3.2.25.3 RSN Capabilities

The length of the RSN Capabilities field is 2 octets. The format of the RSN Capabilities field is as illustrated in Figure 46tc and described after the figure.



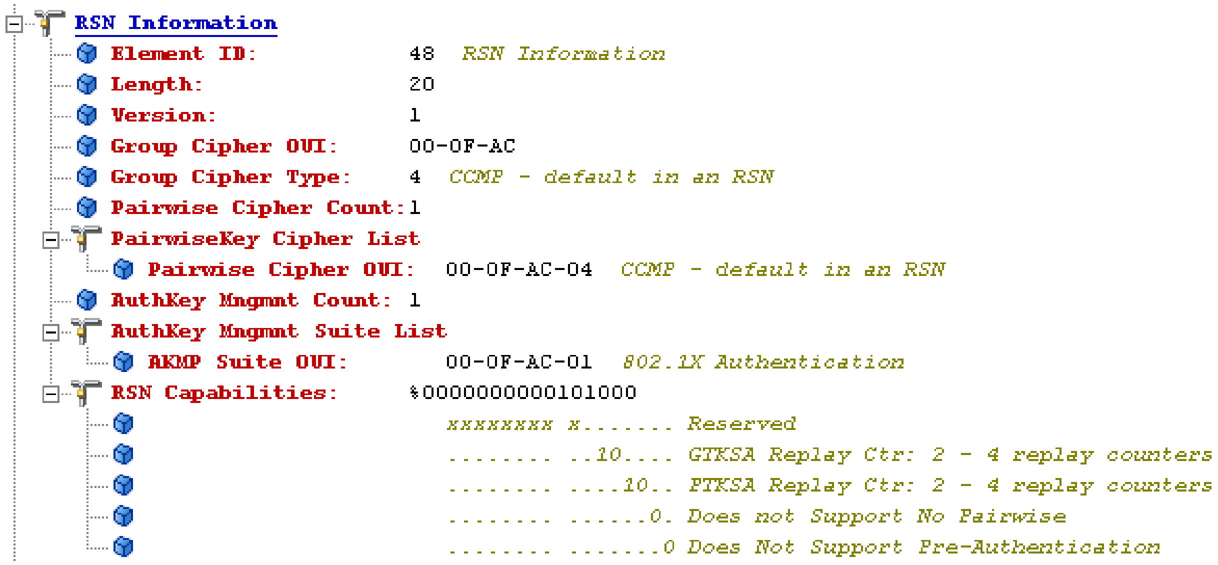
B0	B1	B2 B3	B4 B5	B6 B15
Pre-Auth	No Pairwise	PTKSA Replay Counter	GTKSA Replay Counter	Reserved

Figure 46tc—RSN Capabilities field format

— Bit 0: Pre-Authentication. An AP sets the Pre-Authentication subfield of the RSN Capabilities to 1 to signal it supports preauthentication (see 8.4.6.1) and sets the subfield to 0 when it does not support preauthentication. A non-AP STA sets the Pre-Authentication subfield to 0.

In contrast, figure 9 shows an RSN IE from an AP that has preauthentication disabled.

FIGURE 9 (OmniPeek) – Preauthentication Disabled



```
RSN Information
    Element ID:               48  RSN Information
    Length:                   20
    Version:                  1
    Group Cipher OUI:         00-0F-AC
    Group Cipher Type:        4  CCMP - default in an RSN
    Pairwise Cipher Count:1
    PairwiseKey Cipher List
        Pairwise Cipher OUI:  00-0F-AC-04  CCMP - default in an RSN
    AuthKey Mngmnt Count: 1
    AuthKey Mngmnt Suite List
        AKMP Suite OUI:       00-0F-AC-01  802.1X Authentication
    RSN Capabilities:         %0000000000101000
                                  xxxxxxxx x....... Reserved
                                  ........ ..10.... GTKSA Replay Ctr: 2 - 4 replay counters
                                  ........ ....10.. PTKSA Replay Ctr: 2 - 4 replay counters
                                  ........ ......0. Does not Support No Pairwise
                                  ........ .......0 Does Not Support Pre-Authentication
```

Some preauthentication details

First, understand that Preauthentication is optional. Some vendors support it, others do not. Preauthentication must be both supported and enabled on the APs and client devices in order to use it in the ESS.

Another important point is focused around the last paragraph in 8.4.6.1 above which says, "*A STA's Supplicant may initiate preauthentication with any AP within its present ESS with preauthentication enabled regardless of whether the targeted AP is within radio range.*" This logically raises the following questions:

1. "How would my client device know of APs outside its radio range?"
2. "How would it benefit my client device to know about radios outside of its radio range?"

Let's start with an example of a vendor who has implemented a feature to address this question in both autonomous APs and controllers. Cisco Systems offers a proprietary key management scheme called "Cisco Fast Secure Roaming (FSR)." As part of FSR when used with autonomous APs, wireless domains (think of a domain as a group) of ≤30 APs are formed. APs to which clients associate send a list of all APs in the domain (which should be the APs in the immediate physical area) to each authenticated client device as they authenticate. See below for the reference.

- *Access points store a maximum list of 30 adjacent access points. This list is aged out over a one-day period.*
- *When a client associates to an access point, the associated access point sends the adjacent access point list to the client as a directed unicast packet.*

Cisco Fast Secure Roaming
Application Note, Page 12
Bruce McMurdo, Author
©2004 Cisco Systems

Since the 802.11 standard gives the implementer broad (vague) latitude in implementing this feature, Cisco's method of having the authenticator inform the client devices of its surrounding APs is as good as any other method currently available. Any vendor can choose to have their autonomous APs or controllers push such a list of nearby APs to client devices as they authenticate. This feature is simply an addition to the standard that enhances the functionality without negating interoperability. Fortunately, these "neighbor reports" are part of the 802.11r-2008 amendment. When this feature is present, in whatever form, it is a good thing. Why? Well, that leads to the second question we asked above.

By preauthenticating to all nearby APs over the network infrastructure (through its current wireless connection to an AP), a wireless client can assure fast/secure roaming to nearby APs. The primary scenario where the value of this feature is likely to break down is in high-speed client roaming between autonomous RSN APs. For example, suppose a forklift was moving through a warehouse at 20 mph (32 km/h) where the network was designed with small cells using autonomous APs. In a case like this, you must remember that preauthentication is up to the client to initiate, each AP (whether autonomous or lightweight) is viewed as a separate AP, and each "preauthentication" is a full 802.1X/EAP authentication. The 802.1X/EAP process can take as much as one full second (depending on the EAP type) and therefore the process of preauthentication simply is not fast enough, even when handled in advance of a roam, to keep the client device reassociating via only the 4-Way Handshake.

Preauthentication Conclusion

The whole Ethertype scenario is pretty ugly, as you may have concluded already, and was the primary reason why the 802.11r task group was launched. Preauthentication is not scalable because it requires every AP to remember PMKSAs for all stations that could possibly roam to it, but simply haven't yet. Each station has to authenticate to each access point, instead of the infrastructure as a whole, and therefore is inefficient. Stations have to guess which AP they may hand off to, and therefore how many APs to preauthenticate with. The network infrastructure has no input on this matter, so the administrator cannot tune it (e.g. when there are more or less channels). Every optimization that exists applies more readily to Opportunistic Key Caching (discussed later) than it does to preauthentication. For example, since a controller has only one authenticator, it doesn't make sense to have the station authenticate with every AP individually.

PMKSA Caching Rules

8.4.6.2 Cached PMKSAs and RSNA key management

A STA can retain PMKSAs it establishes as a result of previous authentication. The PMKSA cannot be changed while cached. The PMK in the PMKSA is used with the 4-Way Handshake to establish fresh PTKs.

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

If both sides assert possession of a cached PMKSA, but the 4-Way Handshake fails, both sides may delete the cached PMKSA for the selected PMKID. If a STA roams to an AP with which it is preauthenticating and the STA does not have a PMKSA for that AP, the STA must initiate a full IEEE 802.1X EAP authentication.

The section highlighted in green above is worth paying particular attention to. This shows that it is the STA's decision as to whether or not it initiates fast roaming. The decision is based on whether the STA has a PMKSA corresponding to the BSSID (the MAC address of the AP's radio). The STA does not have to use the PMKID of a cached PMKSA, but if it does, the AP lets the STA know that the PMKID included in the (Re)Association Request frame was acceptable by starting the 4-Way Handshake.

Section 8.4.1.2.1 (page 4 above) says, "*An AP whose Authenticator has retained the PMK for one or more of the PMKIDs can skip the 802.1X authentication and proceed with the 4-Way Handshake. The AP shall include the PMKID of the selected PMK in Message 1 of the 4-Way Handshake.*" Section 8.5.3.1.4 (shown below) illustrates the contents of Message 1 in the 4-Way Handshake with the PMKID highlighted in green. Figure 10 illustrates a frame decode of Message 1 in the 4-Way Handshake with the same information highlighted in blue.

8.5.3.1 4-Way Handshake Message 1

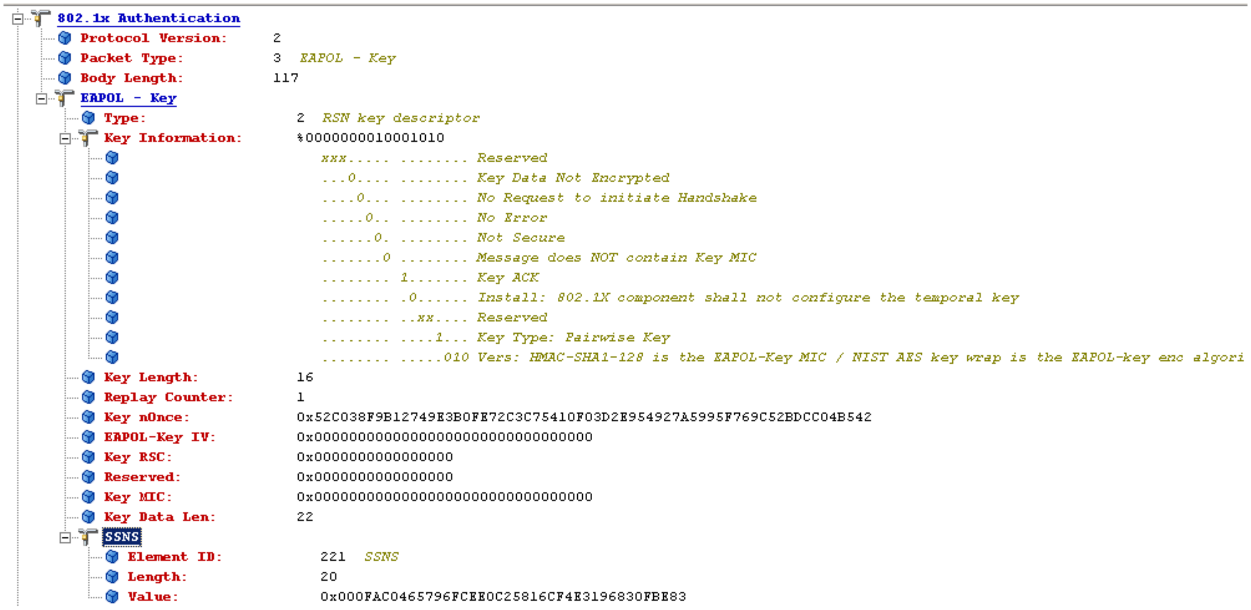
Message 1 uses the following values for each of the EAPOL-Key frame fields:
Descriptor Type = N – see 8.5.2
Key Information:
Key Descriptor Version = 1 (RC4 encryption with HMAC-MD5) or 2 (NIST AES key wrap with HMAC-SHA1-128)
Key Type = 1 (Pairwise)
Install = 0
Key Ack = 1
Key MIC = 0
Secure = 0
Error = 0
Request = 0
Encrypted Key Data = 0
Reserved = 0 – unused by this protocol version
Key Length = Cipher-suite-specific; see Table 20f
Key Replay Counter = n – to allow Authenticator to match the right Message 2 from Supplicant

Key Nonce = ANonce
EAPOL-Key IV = 0
Key RSC = 0
Key MIC = 0
Key Data Length = 22
Key Data = PMKID for the PMK being used during this exchange

The Authenticator sends Message 1 to the Supplicant at the end of a successful IEEE 802.1X authentication, after PSK authentication is negotiated, when a cached PMKSA is used, or after a STA requests a new key. On reception of Message 1, the Supplicant determines whether the Key Replay Counter field value has been used before with the current PMKSA. If the Key Replay Counter field value is less than or equal to the current local value, the Supplicant discards the message. Otherwise, the Supplicant

a) Generates a new nonce SNonce.
b) Derives PTK.
c) Constructs Message 2.

FIGURE 10 (OmniPeek) – Message 1 of the 4-Way Handshake



```
802.1x Authentication
  Protocol Version:      2
  Packet Type:           3  EAPOL - Key
  Body Length:           117
  EAPOL - Key
    Type:                  2  RSN key descriptor
    Key Information:       %0000000010001010
                             xxx..... ........ Reserved
                             ...0.... ........ Key Data Not Encrypted
                             ....0... ........ No Request to initiate Handshake
                             .....0.. ........ No Error
                             ......0. ........ Not Secure
                             .......0 ........ Message does NOT contain Key MIC
                             ........ 1....... Key ACK
                             ........ .0...... Install: 802.1X component shall not configure the temporal key
                             ........ ..xx.... Reserved
                             ........ ....1... Key Type: Pairwise Key
                             ........ .....010 Vers: HMAC-SHA1-128 is the EAPOL-Key MIC / NIST AES key wrap is the EAPOL-key enc algori
    Key Length:            16
    Replay Counter:        1
    Key nOnce:             0x52C038F9B12749E3B0FE72C3C75410F03D2E954927A5995F769C52BDCC04B542
    EAPOL-Key IV:          0x00000000000000000000000000000000
    Key RSC:               0x0000000000000000
    Reserved:              0x0000000000000000
    Key MIC:               0x00000000000000000000000000000000
    Key Data Len:          22
    SSNS
      Element ID:            221  SSNS
      Length:                20
      Value:                 0x000FAC0465796FCEE0C25816CF4E3196830FBE83
```

Client Configuration of PMK Caching and Preauthentication

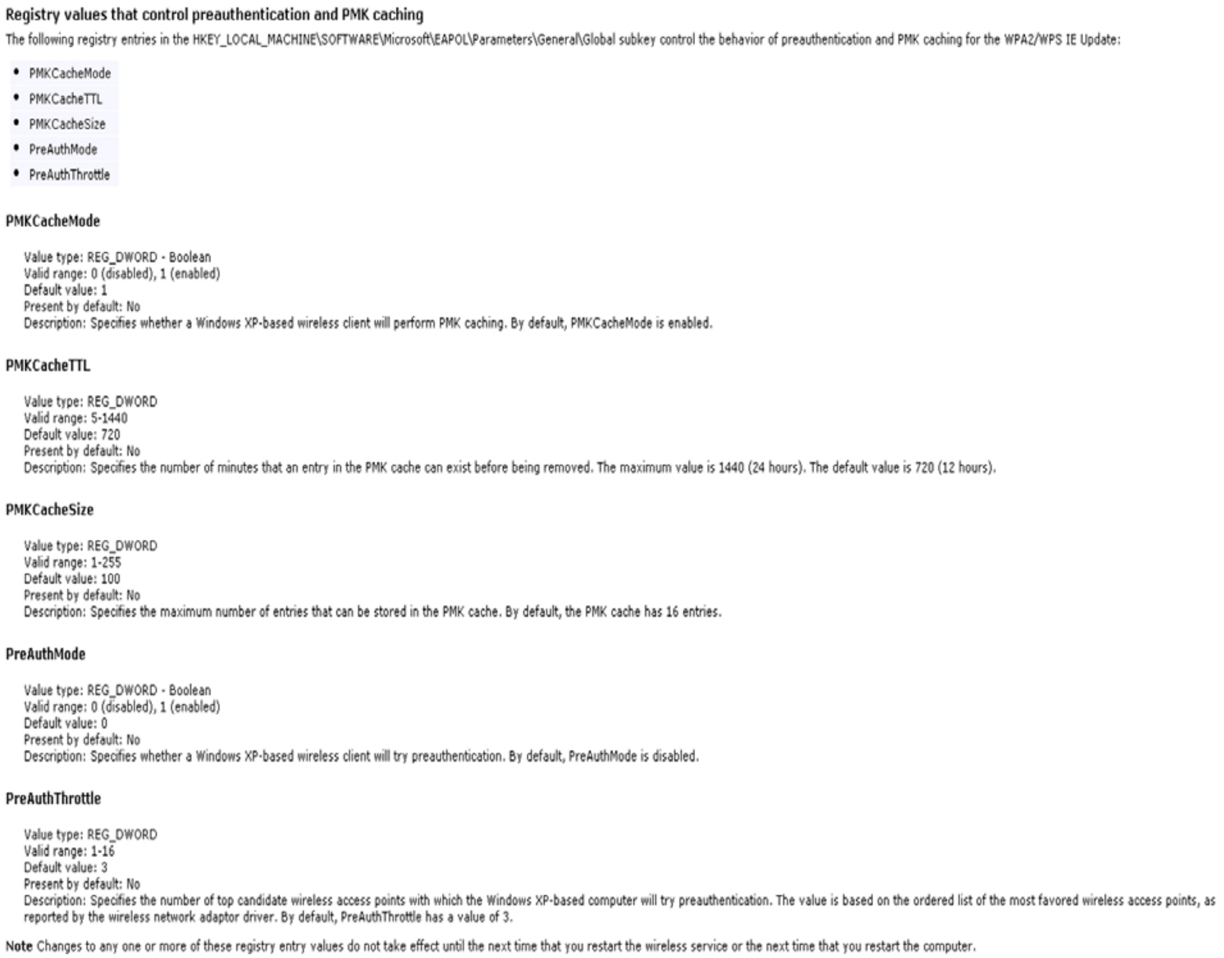
The latest generation of vendor client utilities range greatly in their configuration parameters for PMK Caching and Preauthentication, and some have none at all. In cases like that, it is sometimes advantageous to let the operating system's integrated client utilities or a full-featured third party supplicant (like Juniper's Odyssey Access Client) handle these matters. Let's look at Microsoft's WPA2-compliant Wireless Zero Configuration (WZC) Service as part of the Windows XP operating system with service pack 2 and the WPA2/WPS IE update. Figure 11 illustrates WZC's configuration parameters for:

- **PMKCacheMode -** PMK Cache enable/disable
- **PMKCacheTTL -** How long the PMK will be cached

- **PMKCacheSize -** How many PMKs can be stored in the PMK cache
- **PreAuthMode -** Preauthentication enable/disable
- **PreAuthThrottle** - To how many candidate APs the client will attempt to preauthenticate

Many client adapter drivers allows WZC to control the wireless radio's operation. In cases where the vendor's client utilities do not have adequate PMK Caching and Preauthentication configuration parameters, and they are needed, then letting the WZC client utility control the radio card would be beneficial. It should be noted, however, that many published security risks exist with the WZC therefore many government agencies and corporations maintain security policies restricting the use of the WZC.

FIGURE 11 (Microsoft, Article ID 893357)



Registry values that control preauthentication and PMK caching

The following registry entries in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\EAPOL\Parameters\General\Global subkey control the behavior of preauthentication and PMK caching for the WPA2/WPS IE Update:

- PMKCacheMode
- PMKCacheTTL
- PMKCacheSize
- PreAuthMode
- PreAuthThrottle

PMKCacheMode

Value type: REG_DWORD - Boolean
Valid range: 0 (disabled), 1 (enabled)
Default value: 1
Present by default: No
Description: Specifies whether a Windows XP-based wireless client will perform PMK caching. By default, PMKCacheMode is enabled.

PMKCacheTTL

Value type: REG_DWORD
Valid range: 5-1440
Default value: 720
Present by default: No
Description: Specifies the number of minutes that an entry in the PMK cache can exist before being removed. The maximum value is 1440 (24 hours). The default value is 720 (12 hours).

PMKCacheSize

Value type: REG_DWORD
Valid range: 1-255
Default value: 100
Present by default: No
Description: Specifies the maximum number of entries that can be stored in the PMK cache. By default, the PMK cache has 16 entries.

PreAuthMode

Value type: REG_DWORD - Boolean
Valid range: 0 (disabled), 1 (enabled)
Default value: 0
Present by default: No
Description: Specifies whether a Windows XP-based wireless client will try preauthentication. By default, PreAuthMode is disabled.

PreAuthThrottle

Value type: REG_DWORD
Valid range: 1-16
Default value: 3
Present by default: No
Description: Specifies the number of top candidate wireless access points with which the Windows XP-based computer will try preauthentication. The value is based on the ordered list of the most favored wireless access points, as reported by the wireless network adaptor driver. By default, PreAuthThrottle has a value of 3.

Note Changes to any one or more of these registry entry values do not take effect until the next time that you restart the wireless service or the next time that you restart the computer.

Opportunistic Key Caching

Microsoft supports a feature in Windows XP (with service pack 2 and the WPA2/WPS IE update) called Opportunistic Key Caching (OKC). This feature is also supported by some WLAN infrastructure vendors such as Colubris, Aruba Networks, and Motorola in their controller products and by Juniper Networks in their Wi-Fi supplicant software called Odyssey Access Client. In this section, we will discuss how OKC works in a Split-MAC architecture.

Refer to figure 11 for this explanation. When authenticating to the network for the first time, STA-1 performs a full 802.1X/EAP authentication which yields ***PMKSA1*** (which includes ***pmkid1***) on STA-1 and AP-1. When STA-1 wishes to roam from AP-1 to AP-2, it calculates an Opportunistic PMKID, ***pmkid2***, based on ***pmkid1***, the BSSID of AP-2 (Authenticator Address - AA), and its own MAC address (Supplicant Address –SPA). The IEEE 802.11 standard states:

A PMK identifier is defined as:

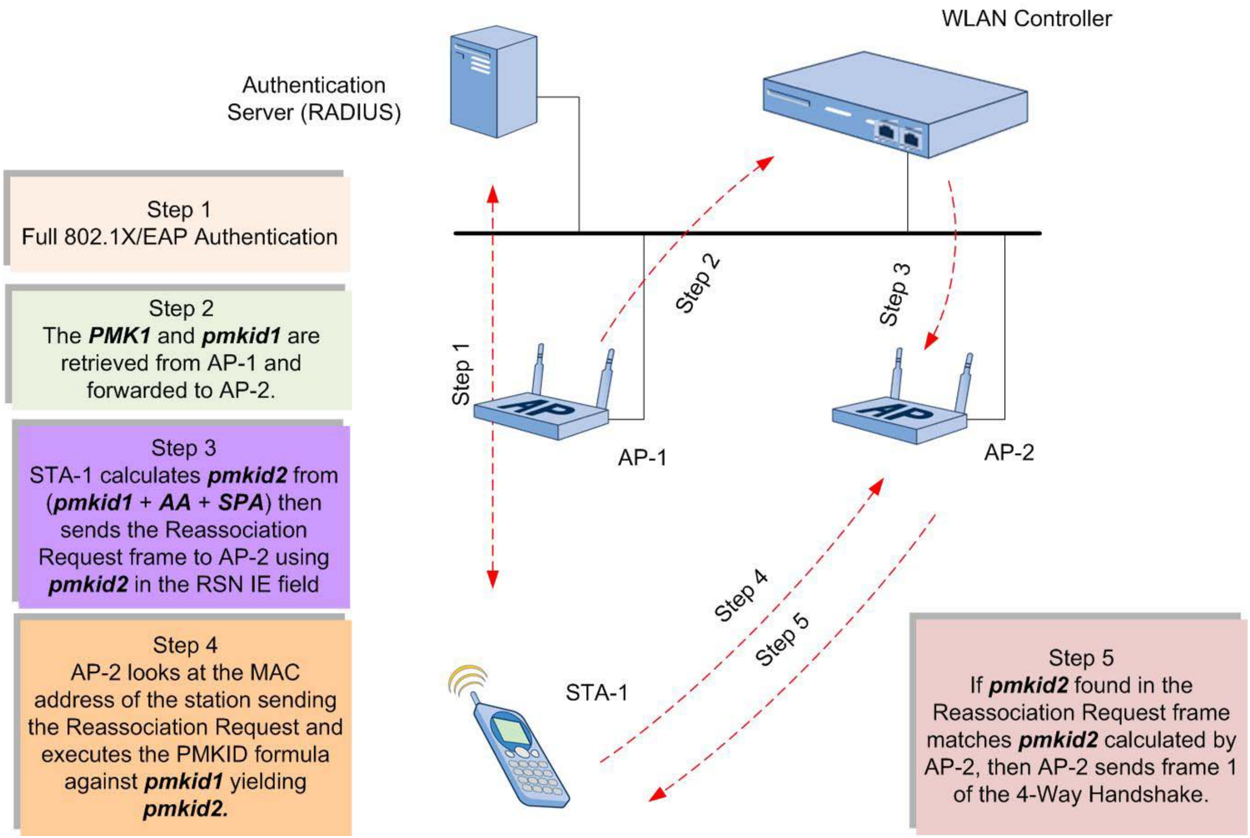
PMKID = HMAC-SHA1-128(PMK, "PMK Name" || AA || SPA)

In a Split-MAC architecture, the controller has access to all PMKSAs from all connected access points. The controller immediately retrieves ***PMK1*** and ***pmkid1*** from AP-1 and forwards them to other nearby access points. Nearby access points receiving ***PMK1*** and ***pmkid1*** from the controller use it to derive PTKs, but use the PMKID formula shown above to derive ***pmkid2***.

NOTE: It's important to note that some vendors' access points do not store PMKSAs, but rather this is done in software on the controller only.

When STA-1 sends a Reassociation Request frame to AP-2 with ***pmkid2*** in the RSN IE, AP-2 calculates ***pmkid2*** using ***pmkid1 + AA + SPA***. If ***pmkid2*** is acceptable to AP-2, it will then send frame 1 of the 4-Way Handshake. This practice of building all future PMKIDs based on the initial PMKSA allows FT throughout the ESS almost indefinitely provided the first PMKSA expires neither on the AP (or controller) nor the STA. OKC is the most widely implemented fast BSS transition method currently available. Now that the 802.11r-2008 amendment is ratified, WLAN controller vendors should quickly move toward standardization of the FT process.

FIGURE 11 (Conceptual) - Opportunistic Key Caching in a Split-MAC Architecture



Step 1
Full 802.1X/EAP Authentication

Step 2
The ***PMK1*** and ***pmkid1*** are retrieved from AP-1 and forwarded to AP-2.

Step 3
STA-1 calculates ***pmkid2*** from (***pmkid1*** + ***AA*** + ***SPA***) then sends the Reassociation Request frame to AP-2 using ***pmkid2*** in the RSN IE field

Step 4
AP-2 looks at the MAC address of the station sending the Reassociation Request and executes the PMKID formula against ***pmkid1*** yielding ***pmkid2.***

Step 5
If ***pmkid2*** found in the Reassociation Request frame matches ***pmkid2*** calculated by AP-2, then AP-2 sends frame 1 of the 4-Way Handshake.

NOTE: WZC with the WPA2/WPS IE update may not always behave predictably. Even if a station has a PMKSA cached for a particular AP to which it wants to roam does not mean it has to use a PMKID in the Reassociation Request frame – it's optional. As an added security advantage, a STA or an AP can, ***at any time***, decide to invalidate a cached PMKSA. This forces a STA to perform a new full 802.1X/EAP authentication.

We feel that it's important for large volume, enterprise-class client radio vendors such as Intel, Broadcom, and Atheros to offer additional configuration parameters such as those found in WZC, at least until there is a ratified FT standard, so that the consumer has a choice of client utilities for environments that cannot tolerate authentication latency during roaming.

Inter-controller Handoff Protocols

Many vendors build their own protocols for handling unique functionality such as inter-controller handoffs. Motorola, the example used above, supports both Preauthentication and OKC, but clients typically only support one of these features at a time. That means that if OKC is used, FT will occur within controllers but not between them unless security state information is shared within a controller cluster. It also means that if Preauthentication is used, the outstanding benefits of OKC may go unused. Another example of how a major Wi-Fi vendor is handling Inter-controller handoffs is detailed below.

A mobility group is a group of WLCs that acts as one virtual WLC by sharing key client, AP, and RF information. The WLC is able to make decisions based on the data from the entire mobility group domain rather than simply from its own connected APs and clients. The mobility group forms a mesh of *authenticated tunnels between the WLCs in the mobility group, allowing any WLC to directly contact other WLCs in the group.*

Cisco Systems
Enterprise Mobility 4.1 Design Guide
October 2007

The proof is in the pudding with proprietary Inter-controller handoff methods, so your mileage may vary. We recommend performing a pilot (proof-of-concept test) before implementing any WLAN network that requires FT functionality. Vendors will likely migrate toward a standards-based approach to FT (both for intra- and inter-controller BSS transitions) once there is a ratified amendment to the 802.11 standard.

Fast BSS Transition Primer– IEEE 802.11r-2008 amendment

The IEEE 802.11r-2008 amendment, ratified July 15, 2008, specifies extensions to the IEEE 802.11-2007 standard for providing mechanisms for Fast BSS Transition.

Some useful definitions are:

Mobility Domain – A set of BSSs, within the same ESS, identified by a Mobility Domain Identifier.

Fast BSS Transition – A STA movement from one BSS in one ESS to another BSS within the same ESS, that minimizes the amount of time that data connectivity is lost between STA and the DS.

Pairwise Master Key R0 (PMK-R0) – The key at the first level of the Fast BSS Transition key hierarchy

Pairwise Master Key R1 (PMK-R1) – A key at the second level of the Fast BSS Transition key hierarchy.

Pairwise Master Key R0 Key Holder (R0KH) – The component of RSNA key management of the Authenticator that is authorized to derive and hold the PMK-R0, derive the PMK-R1s, and distribute the PMK-R1s to the R1KHs.

Pairwise Master Key R0 Key Holder Identifier (R0KH-ID) – An identifier that names the holder of the PMK-R0 Key in the Authenticator

Pairwise Master Key R1 Key Holder (R1KH) – The component of RSNA key management of the Authenticator that receives a PMK-R1 from the R0KH, holds the PMK-R1, and derives the PTKs.

Pairwise Master Key R1 Key Holder Identifier (R1KH-ID) – An identifier that names the holder of a PMK-R1 key in the Authenticator.

Pairwise Master Key S0 Key Holder (S0KH) – The component of RSNA key management of the Supplicant that derives and holds the PMK-R0, derives the PMK-R1s, and provides the PMK-R1s to the S1KH.

Pairwise Master Key S0 Key Holder Identifier (S0KH-ID) – An identifier that names the holder of the PMK-R0 in the Supplicant.

Pairwise Master Key S1 Key Holder (S1KH) – The component of RSNA key management in the Supplicant that receives a PMK-R1 from the S0KH, holds the PMK-R1, and derives the PTKs.

Pairwise Master Key S1 Key Holder Identifier (S1KH-ID) – An identifier that names the holder of the PMK-R1 in the Supplicant.

PMKR0Name – An identifier that names the PMK-R0.

PMKR1Name – An identifier that names a PMK-R1.

PTKName – An identifier that names the PTK.

Resource Information Container (RIC) – A sequence of Information Elements that include resource request and response parameters.

The processes for a fast BSS transition (FT) vary based on whether the association:

- Is the initial mobility domain association or an FT reassociation
- Is an over-the-air or over-the-DS FT reassociation
- Is an RSN or non-RSN network (we only discuss RSNs in this whitepaper)
- Is requesting resources from the target AP (typically for QoS purposes)

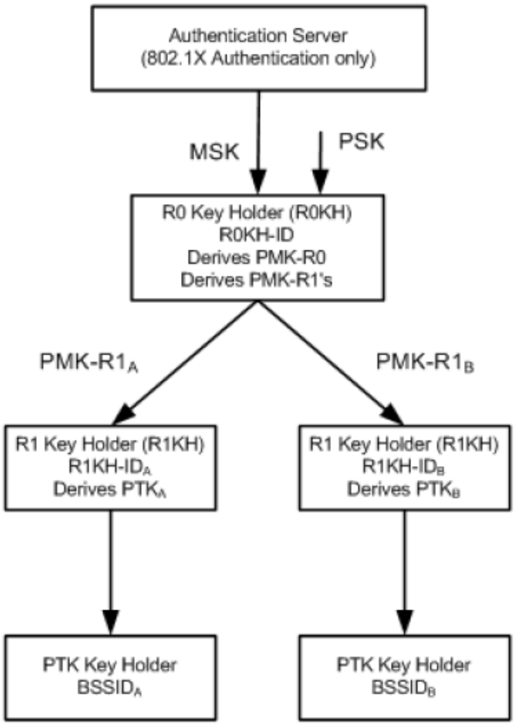
Throughout the next section, we're going to see the term "FT" followed by other terms. This often means that the original item has been modified in some way to accommodate fast BSS transition. For example, "FT Association Request" means that an Association Request frame has been modified with new FT information.

FT Key Architecture

The IEEE 802.11r amendment uses a three-tier key architecture. The Master Session Key (MSK) is exported from the 802.1X/EAP authentication. When 802.1X/EAP is used, the MSK is sent in a RADIUS attribute to the Authenticator (typically a controller) encrypted with the RADIUS shared secret. In this way, the Supplicant and Authenticator will hold the MSK. When a Preshared Key (PSK) is used for authentication, the PSK is used as the MSK by all stations in the ESS.

In a Split-MAC architecture, both the Supplicant and Authenticator use the MSK to derive the PMK-R0 and subsequently the PMK-R1 for each lightweight AP. The Authenticator (R0KH) uses a secure channel (not specified by the 802.11r amendment) to send each unique PMK-R1 key to the appropriate AP (R1KH). The Supplicant then performs an FT 4-Way Handshake with the initial AP to develop the PTKSA for that AP. Figure 12 illustrates the general format of FT Key Architecture.

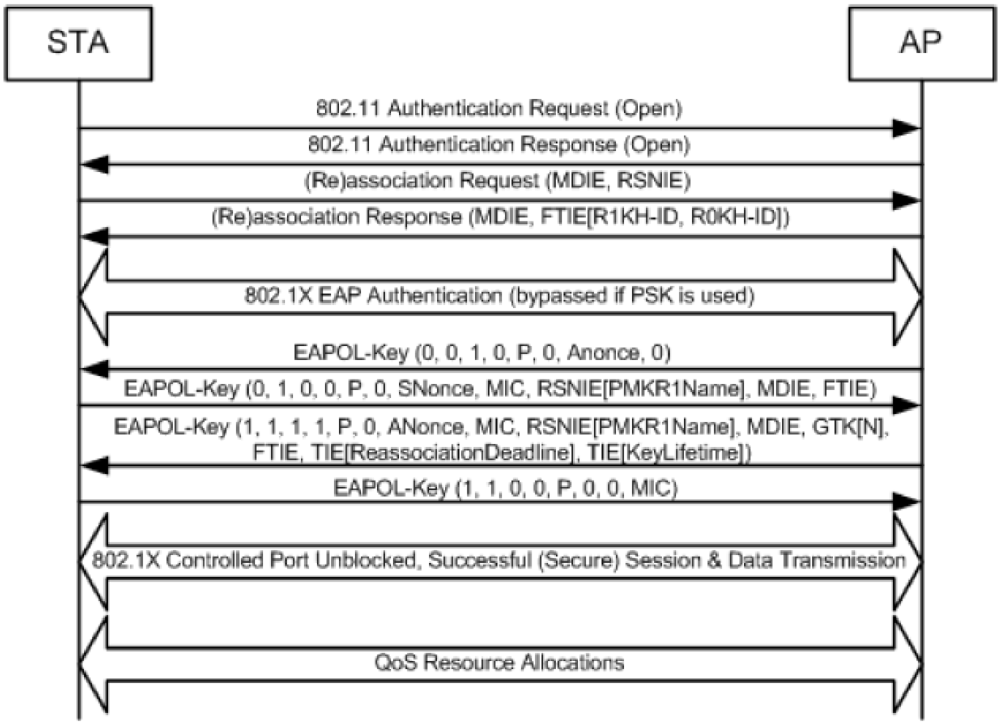
FIGURE 12 – FT Key Architecture



Fast BSS Transition (FT) Initial Mobility Domain Association

When a STA initially joins a mobility domain, it uses Open System Authentication and FT Association Request/Response frames (a 4-frame exchange). The 802.1X/EAP mutual authentication exchange between the Supplicant and AS and distribution of the MSK to the Authenticator by the AS follows. Once the PMK-R0 and PMK-R1 keys are derived at the R0KH device and unique PMK-R1 keys are distributed to R1KHs, then an FT 4-Way Handshake can be used to develop a PTKSA at the R1KH and S1KH. Once the PMK-R1 keys are distributed, there is no need to go through this initial process again within the same mobility domain. Figure 13 illustrates FT initial Mobility Domain Association with FT 4-Way Handshake.

FIGURE 13 – FT Initial Mobility Domain Association with FT 4-Way Handshake


STA
AP
802.11 Authentication Request (Open)
802.11 Authentication Response (Open)
(Re)association Request (MDIE, RSNIE)
(Re)association Response (MDIE, FTIE[R1KH-ID, R0KH-ID])
802.1X EAP Authentication (bypassed if PSK is used)
EAPOL-Key (0, 0, 1, 0, P, 0, Anonce, 0)
EAPOL-Key (0, 1, 0, 0, P, 0, SNonce, MIC, RSNIE[PMKR1Name], MDIE, FTIE)
EAPOL-Key (1, 1, 1, 1, P, 0, ANonce, MIC, RSNIE[PMKR1Name], MDIE, GTK[N], FTIE, TIE[ReassociationDeadline], TIE[KeyLifetime])
EAPOL-Key (1, 1, 0, 0, P, 0, 0, MIC)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission
QoS Resource Allocations

Over-the-Air Fast BSS Transition in an RSN

An AP must announce its specific support for over-the-air fast BSS transition in the Mobility Domain Information Element (MDIE) in Beacons, Probe Responses, and (Re)Association Responses. A Supplicant wishing to associate to an FT-enabled Authenticator must have matching MDIE information in its Authentication and (Re)Association Requests.

When roaming within a mobility domain using over-the-air FT, STAs use FT Authentication Request/Response and FT Reassociation Request/Response frames as part of a 4-frame reassociation exchange. These four frames contain the appropriate information to build a PTKSA between the target (new) AP and the Supplicant. No 802.1X/EAP or 4-Way Handshake is necessary to unlock the 802.1X controlled port. Figure 14 illustrates over-the-air FT in an RSN.

FIGURE 14 – Over-the-Air FT Protocol in an RSN

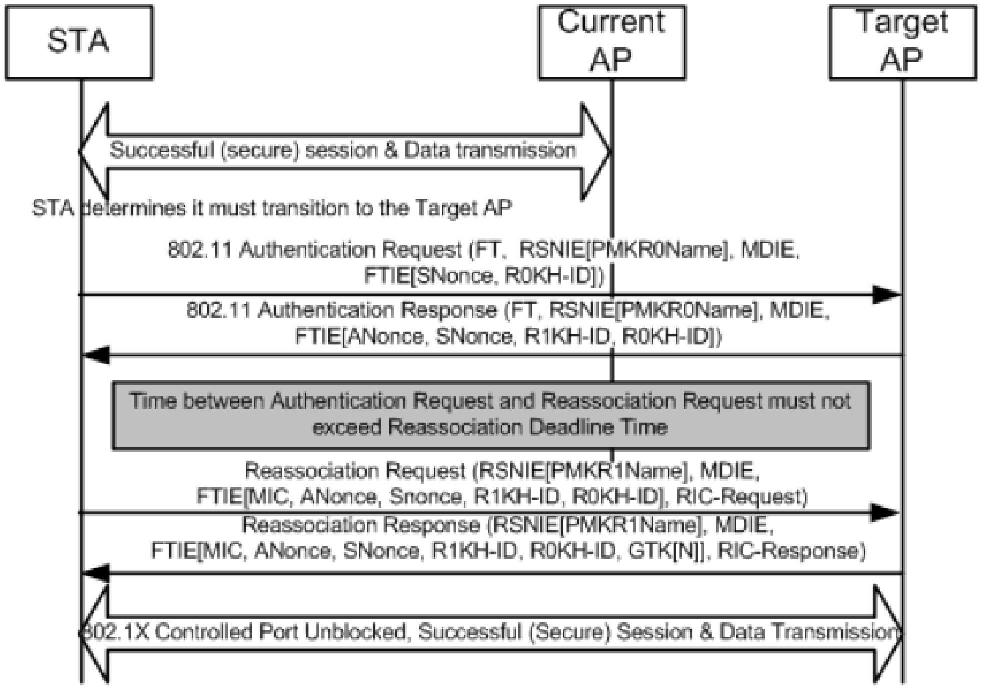
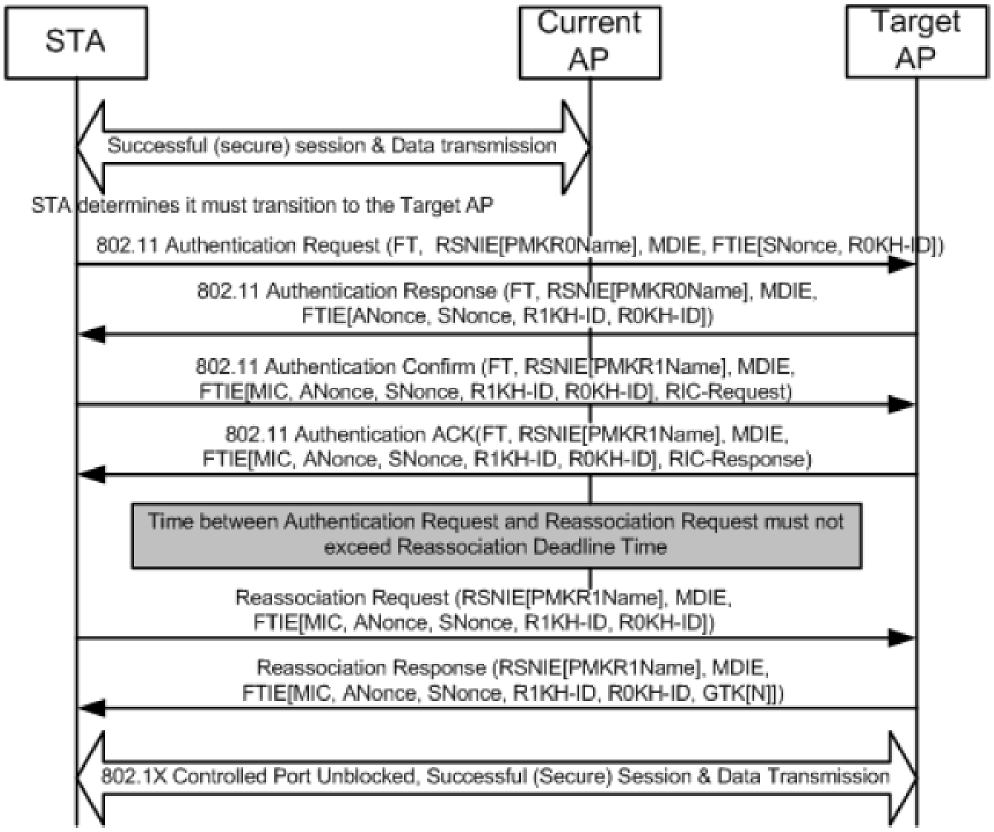

STA
Current AP
Target AP
Successful (secure) session & Data transmission
STA determines it must transition to the Target AP
802.11 Authentication Request (FT, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])
802.11 Authentication Response (FT, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])
Time between Authentication Request and Reassociation Request must not exceed Reassociation Deadline Time
Reassociation Request (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, Snonce, R1KH-ID, R0KH-ID], RIC-Request)
Reassociation Response (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], RIC-Response)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Figure 15 illustrates over-the-air FT in an RSN with optional Resource Request. Notice that the optional Resource Request protocol requires two additional frames: FT Authentication Confirm and FT Authentication ACK.

When using the resource request procedure, the STA has the option to request a resource allocation at the target AP. To request resources, the STA creates a Resource Information Container (RIC) and inserts it in an appropriate request message to the target AP. The request message is sent to the target AP either directly (over-the-air) or via the current AP (over-the-DS). In an RSNA, resource requests and responses are exchanged only after the establishment of the PTK, and are protected by message integrity checks.

This means that the 802.11 Authentication Request/Response exchange in Figure 15 is sufficient for the Authenticator and Supplicant to create a PTKSA. Notice that the ANonce, SNonce, and PMKR0Name are exchanged during this frame exchange. The R1KH of the target AP uses the value of PMKR0Name and other information in the frame to calculate PMKR1Name. If the target AP does not have the key identified by PMKR1Name, it may retrieve that key from the R0KH identified by the STA. Upon receiving a new PMK-R1 key for a STA, the target AP deletes the prior PMK-R1 SA and PTKSAs derived from the prior PMK-R1 key. The STA and the target AP compute the PTK and PTKName using the PMK-R1, PMKR1Name, ANonce, and SNonce.

FIGURE 15 – Over-the-Air FT Protocol in an RSN (with Resource Request)


STA
Current AP
Target AP
Successful (secure) session & Data transmission
STA determines it must transition to the Target AP
802.11 Authentication Request (FT, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])
802.11 Authentication Response (FT, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])
802.11 Authentication Confirm (FT, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
802.11 Authentication ACK(FT, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Response)
Time between Authentication Request and Reassociation Request must not exceed Reassociation Deadline Time
Reassociation Request (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID])
Reassociation Response (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]])
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

Over-the-DS Fast BSS Transition in an RSN

When roaming within a mobility domain using over-the-DS FT, STAs use FT (Action Frame) Request/Response and FT Reassociation Request/Response frames as part of a 4-frame reassociation exchange. These four frames contain the appropriate information to build a PTKSA between the target (new) AP and the Supplicant. No 802.1X/EAP or 4-Way Handshake is necessary to unlock the 802.1X controlled port. Figure 16 illustrates over-the-DS FT in an RSN.

Figure 17 illustrates over-the-DS FT in an RSN with optional Resource Request. Notice that the optional Resource Request protocol requires two additional frames: FT Authentication Confirm and FT Authentication ACK.

FIGURE 16 – Over-the-DS FT Protocol in an RSN

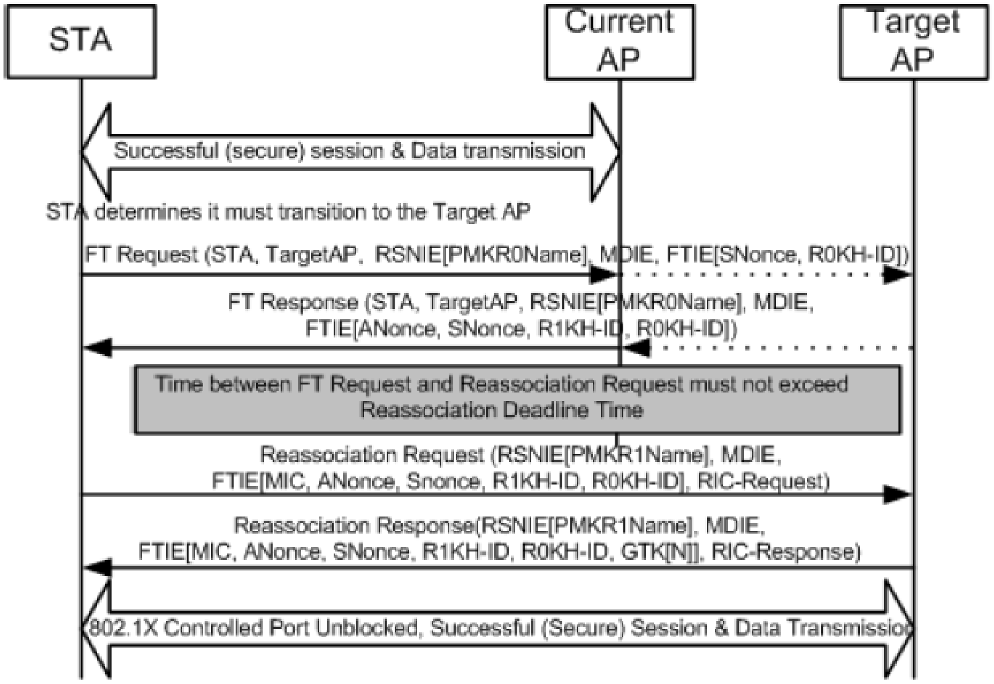
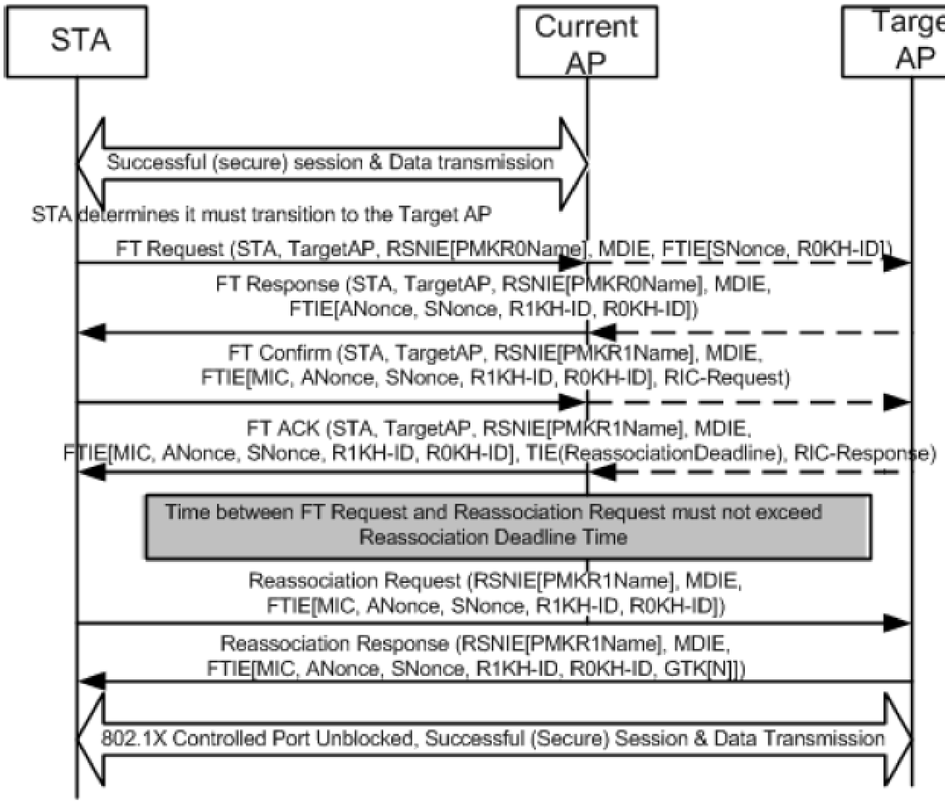

STA
Current AP
Target AP
Successful (secure) session & Data transmission
STA determines it must transition to the Target AP
FT Request (STA, TargetAP, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])
FT Response (STA, TargetAP, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])
Time between FT Request and Reassociation Request must not exceed Reassociation Deadline Time
Reassociation Request (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, Snonce, R1KH-ID, R0KH-ID], RIC-Request)
Reassociation Response(RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]], RIC-Response)
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

FIGURE 17 – Over-the-DS FT Protocol in an RSN (with Resource Request)

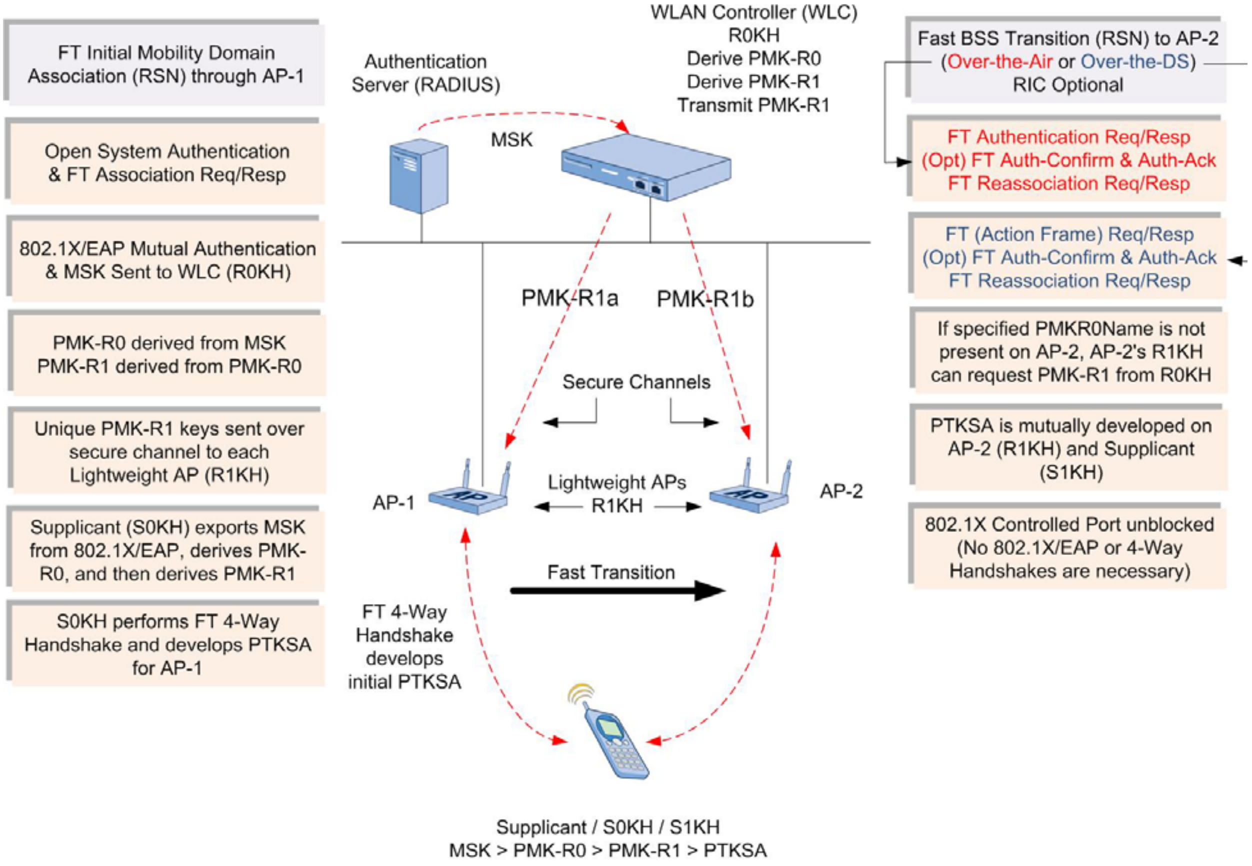

STA
Current AP
Target AP
Successful (secure) session & Data transmission
STA determines it must transition to the Target AP
FT Request (STA, TargetAP, RSNIE[PMKR0Name], MDIE, FTIE[SNonce, R0KH-ID])
FT Response (STA, TargetAP, RSNIE[PMKR0Name], MDIE, FTIE[ANonce, SNonce, R1KH-ID, R0KH-ID])
FT Confirm (STA, TargetAP, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], RIC-Request)
FT ACK (STA, TargetAP, RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID], TIE(ReassociationDeadline), RIC-Response)
Time between FT Request and Reassociation Request must not exceed Reassociation Deadline Time
Reassociation Request (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID])
Reassociation Response (RSNIE[PMKR1Name], MDIE, FTIE[MIC, ANonce, SNonce, R1KH-ID, R0KH-ID, GTK[N]])
802.1X Controlled Port Unblocked, Successful (Secure) Session & Data Transmission

IEEE 802.11r Fast BSS Transition Summary

Keep in mind that in an FT Initial Mobility Domain Association, an FT 4-Way Handshake is used to develop a PTKSA between a Supplicant and Authenticator, but thereafter FT Protocols (over-the-air or over-the-DS) are used in place of the FT 4-Way Handshake. The FT Protocol exchanges are used to swap information between the Supplicant and Authenticator for the purpose of developing a fresh PTKSA for each target AP. Frame exchanges vary slightly based on whether over-the-air or over-the-DS is used and whether Resource Requests are necessary, but the results are still the same: a fast, fresh PTK.

Figure 18 illustrates the step-by-step process summary (left-to-right, top-to-bottom) of an FT Initial Mobility Domain Association and a fast BSS transitional (over-the-air or over-the-DS).

FIGURE 18 (Summary) – 802.11r Fast BSS Transition Processes


FT Initial Mobility Domain Association (RSN) through AP-1
Open System Authentication & FT Association Req/Resp
802.1X/EAP Mutual Authentication & MSK Sent to WLC (R0KH)
PMK-R0 derived from MSK PMK-R1 derived from PMK-R0
Unique PMK-R1 keys sent over secure channel to each Lightweight AP (R1KH)
Supplicant (S0KH) exports MSK from 802.1X/EAP, derives PMK-R0, and then derives PMK-R1
S0KH performs FT 4-Way Handshake and develops PTKSA for AP-1
Authentication Server (RADIUS)
WLAN Controller (WLC)
R0KH
Derive PMK-R0
Derive PMK-R1
Transmit PMK-R1
MSK
PMK-R1a
PMK-R1b
Secure Channels
AP-1
Lightweight APs
R1KH
AP-2
Fast Transition
FT 4-Way Handshake develops initial PTKSA
Supplicant / S0KH / S1KH
MSK > PMK-R0 > PMK-R1 > PTKSA
Fast BSS Transition (RSN) to AP-2
(Over-the-Air or Over-the-DS)
RIC Optional
FT Authentication Req/Resp
(Opt) FT Auth-Confirm & Auth-Ack
FT Reassociation Req/Resp
FT (Action Frame) Req/Resp
(Opt) FT Auth-Confirm & Auth-Ack
FT Reassociation Req/Resp
If specified PMKR0Name is not present on AP-2, AP-2's R1KH can request PMK-R1 from R0KH
PTKSA is mutually developed on AP-2 (R1KH) and Supplicant (S1KH)
802.1X Controlled Port unblocked (No 802.1X/EAP or 4-Way Handshakes are necessary)

OKC / 802.11r Comparison

OKC is modestly-well adopted in the industry and reduces the burden on RADIUS servers. In fact, OKC and 802.11r reduce the burden on RADIUS in the same way to the same degree. 802.11r adds two additional "features" beyond the capabilities of OKC:

- QoS reservations at association – This is a two-frame savings.

- The ability to spread the key hierarchy across controllers – This is a minor benefit, because inter-controller protocols (when vendors create and use them) can make OKC work in exactly the same way. Remember, the R0KH generates both PMKs and sends PMK-R1 off to the R1KH securely. That's the same as the home authenticator sending the 802.11i PMK to the foreign controller.

Thus, 802.11r and OKC are nearly equivalent in functionality. 802.11r won't add any intelligence above what exists today. Instead, it's really going to be more of a "clean up" of the mechanisms rather than breaking new ground. One additional item to consider is that neither 802.11r nor OKC address **how** the handoff decision is made – only how quickly a handoff can occur. In the end, the station will make the roaming decision based on the vendor's roaming algorithm, which is typically based on RSSI values.

Non-traditional Roaming Mechanisms

Some vendors take a non-traditional approach to handling FT. Two such vendors are Meru Networks and Extricom (and of course any companies who are rebranding their solutions). By architecting the wireless network so that all APs are on the same channel using the same BSSID, wireless stations are "tricked" into thinking that there is only one AP in the environment – an AP with incredible coverage! This architecture is generically called the "Single Channel Architecture (SCA)." Ironically, Meru's and Extricom's solutions yield the same basic result when it comes to client roaming but are implemented in almost opposite fashion – Meru with highly-intelligent APs and Extricom with ultra-thin APs. Each vendor touts, "zero handoff." What they mean by "zero handoff" is that it's so close to zero that it shouldn't be a topic for discussion – and we feel they are correct in this assertion. It's typical to see sub 3 ms handoff times in this architecture. In contrast, the best BSS transition time seen to date in a traditional Multiple Channel Architecture (MCA) roaming model is approximately 20 ms. By removing the roaming process decision making from the station and placing it on the controller, roaming is optimized. Stations from various vendors each make roaming decisions differently. For consistency and predictability, it is our opinion that this decision making process is optimally accomplished at the controller.

Mobile stations have to scan across more than 30 channels (in the U.S.) in multiple bands just to identify which APs might serve them. If the AP chosen doesn't have the resources to handle the new association, the station is then forced to repeat the scanning, authentication, and association processes. This is like a travelling salesman going door-to-door hoping to find someone who will just let him in. In this case, the station may go a long time, hat-in-hand, without being served. 802.11r does nothing to address this problem. That's what makes 802.11r mostly into another security protocol - one that just tries to build more formality around key caching than OKC, but still solving the same problem. The real solution must be elsewhere.

Enter 802.11k

802.11k-2008, ratified June 12, 2008 detailing Radio Resource Measurement, was conceived as a potpourri of request-report exchanges, ones that APs and stations can send to each other to learn more about how the network is performing. Part of this communication is the Neighbor Report. The Neighbor Report mechanism allows a station to locate its "neighbors" - APs that are in the area - so that less scanning is required. By minimizing the scanning required to locate neighboring APs, the handoff decision should be faster and thus the handoff process en total should be much faster. As nice as this vision is, it is still missing a few things. Neighbor Reports don't effectively guide the stations, and 802.11k doesn't specify what a station is supposed to do with the information once it receives it.

The whole concept of 802.11k is predicated around the idea of a “neighbor.” So, what is a neighbor anyway?

- Is a neighbor an AP that is close to the station itself? If so, then how does the AP reliably know what other APs are in range of the station?

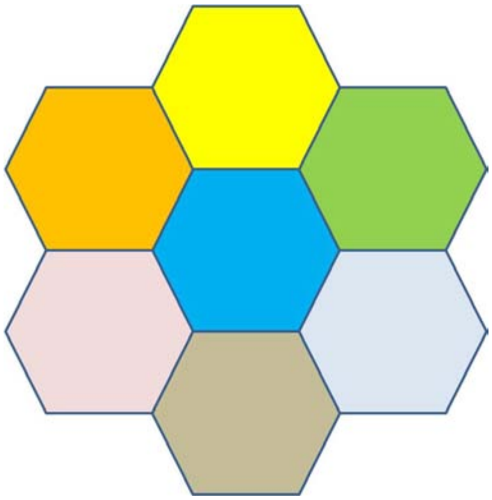
- Is a neighbor an AP that is near the AP from which the station requests the Neighbor Report? If so, then the AP is substituting its neighborhood for the station’s. That doesn’t help much because the AP could have a large set of APs as neighbors that are on the opposite side of the stations, and hence out of reach.

This is an intractable problem, and the best the network infrastructure can hope to do is to guess and hope that it gets close.

Unfortunately, it seems that everyone has a different understanding of what the information in a Neighbor Report should do; however, the standard is clear: a station isn’t required to do anything with the Neighbor Report. IEEE 802.11k-2008, Section 11.10.9 states, “*Since the information in the Neighbor Report may be stale, it should be considered advisory; information obtained by the report recipient through a scan or other sources may also be considered, possibly overriding information in the Neighbor Report.*”
Some people say that the Neighbor Report should supplant scanning; the station should be forced to choose from the list the AP returns. What if the AP returns nothing but out-of-range candidates? The station would have to ignore the Neighbor Report and move on, at a significant penalty of wasted time. Others say that the list should just supplement scanning, and the station should choose whether to believe its own scans or the AP’s Neighbor Report. But that doesn’t shrink the client’s list of candidate neighbors, but instead only adds to it.

Frankly, 802.11k is not designed to implement any form of “directed handoff” like you might expect from the cellular world. Interestingly, 802.11k is less effective as the number of channels in a deployment increase. As you add more channels, the situation can become very complicated. Think about how with 802.11n, designers want to use as many 5GHz Dynamic Frequency Selection (DFS) channels as they can. The primary goal of Multiple Channel Architecture (MCA) is to limit co-channel and adjacent channel interference by deploying neighboring APs on different channels (frequency sets). The more channels you have available, the more MCA designers use them to avoid co-channel and adjacent channel interference. Another way of saying this is that a properly designed MCA network will have only one, or a very small number, of APs (that are configured for a specific channel) to be available at any given physical location. Picture how that works for 802.11k. The odds are good that there will be only one AP in a Neighbor Report for any given channel. That means that coverage looks like the following:

FIGURE 19 – Channel Reuse in Multiple Channel Architectures



The Neighbor Report from the center AP will include the six neighboring APs. If these neighbors are in a band requiring DFS support (such as UNII-2 and UNII-2e), then the station is required to not trust the Neighbor Report's indication that an AP is on the channel. What that means is that the station can tune to the channel, but it is forced to wait for the next beacon before it can start association (which requires transmission) per FCC Part 15 rules. That can be a wait of up to 100 ms per channel since active probing isn't allowed. FCC Part 15.407(h)(2)(ii) states:

(ii) Channel Availability Check Time. A U-NII device shall check if there is a radar system already operating on the channel before it can initiate a transmission on a channel and when it has to move to a new channel. The U-NII device may start using the channel if no radar signal with a power level greater than the interference threshold values listed in paragraph (h)(2) of this part, is detected within 60 seconds.

Keep in mind that it is likely that only one of those six APs is going to be in range of the station because it moved away from the center AP towards the edge (of the diagram). An MCA deployment tries to have only one AP at a time covering any area. Therefore, the station will be forced to pick one of the six neighboring channels, and thus has only a 17% (1 in 6) chance of choosing a channel with an AP in range. In the worst case, it will have to dwell for up to 100 ms on each of the six channels before it finds its neighbor. This adds up to a possible 600 ms of passive scan time, and if the neighbor AP is full (no available resources), then there is no RF redundancy, and the station loses the connection. Therefore, 802.11k's hints become less useful just as their need becomes greater.

Ultimately, what we have is a faster security renegotiation standard (802.11r) and a series of hints to the station about what APs are out there and what services they are providing (802.11k, 802.11e). There is nothing to make the handoff predictable, and each station must figure out when and how to hand off on its own. Not surprisingly, this is the value of architectures where the handoff decisions are handled within the network infrastructure. As it turns out, 802.11k and 802.11r work best in environments such as these (where they are needed the least). That's because in layered SCA deployments, the 802.11k Neighbor Report is nearly always accurate no matter where the station is located - the report simply containing the list of layered channels. Following that, 802.11r can help, not with intra-channel handoff which is already solved, but with inter-channel load balancing and inter-controller handoff.

Summary

The IEEE 802.11-1999 standard did not provide guidance for how client stations or the wireless infrastructure is to handle BSS transition, and the amendments to the standard since 1999 haven't kept pace with advances in WLAN architecture. With the ratification of IEEE 802.11-2007, the standard still only offers PMK caching and preauthentication as standards-based fast BSS transition mechanisms.

With enterprise WLANs growing at an astounding pace and customers placing more demanding applications on the WLAN, vendors have had to improvise in a number of ways to keep the pace. Some of this improvisation has been the creation of proprietary protocols and some has been a complete change in architecture, as with the Single Channel Architecture. Fortunately, most of today's advancements are taking into consideration the Split-MAC (centralized) architecture rather than focusing only on the autonomous AP (distributed) architecture, but even still inter-controller handoff mechanisms are often slow and cause problems for real-time applications.

Opportunistic Key Caching (OKC) has never reached broad acceptance in the market, though it is currently the de-facto standard for fast BSS transition since it's really all we have had to date. 802.11r and 802.11k will only "clean up" OKC, but will not revolutionize how stations transition between BSSs beyond what OKC currently offers. Fortunately, almost every vendor has voiced support for 802.11r and 802.11k, which will broaden support for a standardized transition method, which is sorely needed.