Methoden für 802.11-WLAN und Fast-Secure Roaming auf dem CUWN ermitteln

Inhalt

Einleitung

In diesem Dokument werden die Wireless-Roaming-Typen und die schnellsicheren Roaming-Typen beschrieben, die für IEEE 802.11 Wireless LANs (WLANs) im Unified Wireless Network

(CUWN) verfügbar sind.

Voraussetzungen

Anforderungen

Cisco empfiehlt, dass Sie über Kenntnisse in folgenden Bereichen verfügen:

- IEEE 802.11 WLAN-Grundlagen
- IEEE 802.11: WLAN-Sicherheit
- IEEE 802.1x/EAP - Grundlagen

Verwendete Komponenten

Die in diesem Dokument enthaltenen Informationen basieren auf der Cisco WLAN Controller-Software Version 7.4.

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle verstehen.

Hintergrundinformationen

Die Informationen in diesem Dokument basieren auf der Cisco WLAN Controller Software Version 7.4. Die meisten der beschriebenen Debug-Ausgaben und -Verhaltensweisen können jedoch auf jede Softwareversion angewendet werden, die die beschriebenen Methoden unterstützt. Die Details aller hier beschriebenen Methoden bleiben auf späteren Cisco WLAN Controller-Codes gleich (bis zur Version 8.3 bis zum Zeitpunkt der Aktualisierung dieses Artikels).

In diesem Dokument werden die verschiedenen Wireless-Roaming- und Fast-Secure-Roaming-Methoden beschrieben, die für IEEE 802.11 Wireless LANs (WLANs) verfügbar sind, die vom Cisco Unified Wireless Network (CUWN) unterstützt werden.

Das Dokument enthält nicht alle Details zur Funktionsweise der einzelnen Methoden oder zu ihrer Konfiguration. Der Hauptzweck dieses Dokuments besteht darin, die Unterschiede zwischen den verschiedenen verfügbaren Techniken, ihre Vorteile und Einschränkungen sowie den Austausch von Frames für jede Methode zu beschreiben. Es werden Beispiele für WLAN Controller (WLC)-Fehlerbehebungen bereitgestellt, und Wireless-Paket-Images werden verwendet, um die Ereignisse zu analysieren und zu erläutern, die bei den einzelnen beschriebenen Roaming-Methoden auftreten.

Bevor eine Beschreibung der verschiedenen für WLANs verfügbaren schnellsicheren Roaming-Methoden gegeben wird, ist es wichtig zu verstehen, wie der WLAN-Zuordnungsprozess funktioniert und wie ein reguläres Roaming-Ereignis auftritt, wenn für den Service Set Identifier (SSID) keine Sicherheit konfiguriert ist.

Wenn ein 802.11-Wireless-Client eine Verbindung zu einem Access Point (AP) herstellt, muss er zunächst den grundlegenden 802.11-Authentifizierungsprozess des offenen Systems durchlaufen, bevor er Datenverkehr (Wireless-Datenframes) weiterleitet. Dann muss der Zuordnungsprozess

abgeschlossen sein. Der Open System-Authentifizierungsprozess ähnelt einer Kabelverbindung am AP, die der Client auswählt. Dies ist ein sehr wichtiger Punkt, da immer der Wireless-Client den bevorzugten Access Point auswählt und die Entscheidung auf unterschiedliche Faktoren stützt, die von Anbieter zu Anbieter variieren. Aus diesem Grund beginnt der Client diesen Prozess, indem er den Authentifizierungsrahmen an den ausgewählten Access Point sendet, wie weiter unten in diesem Dokument gezeigt. Der Access Point kann nicht verlangen, dass Sie eine Verbindung herstellen.

Sobald der Authentifizierungsprozess des offenen Systems erfolgreich mit einer Antwort des Access Points abgeschlossen wurde ("über Kabel verbunden"), schließt der Zuordnungsprozess im Wesentlichen die 802.11 Layer 2 (L2)-Aushandlung ab, die die Verbindung zwischen dem Client und dem Access Point herstellt. Der WAP weist dem Client bei erfolgreicher Verbindung eine Zuordnungs-ID zu und bereitet diese vor, um Datenverkehr weiterzuleiten oder eine übergeordnete Sicherheitsmethode auszuführen, falls diese für die SSID konfiguriert wurde. Der Authentifizierungsprozess des offenen Systems besteht aus zwei Management-Frames sowie dem Assoziierungsprozess. Authentifizierungs- und Zuordnungs-Frames sind Wireless-**Management**-Frames, keine Daten-Frames, die im Wesentlichen für den Verbindungsprozess mit dem Access Point verwendet werden.

Hier ist ein Bild der drahtlosen Frames über die Luft für diesen Prozess:

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462	Authentication, SN=2443, FN=0, Flags=....
2	0.000784	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462	Authentication, SN=2771, FN=0, Flags=....
3	0.002428	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462	Association Request, SN=2444, FN=0, Flags
4	0.007122	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462	Association Response, SN=2772, FN=0, Flag
5	0.995428	0.0.0.0	255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462	DHCP Discover - Transaction ID 0xba2bf0a4
6	2.996191	1.1.1.1	172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462	DHCP Offer - Transaction ID 0xba2bf0a4
7	2.998532	0.0.0.0	255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462	DHCP Request - Transaction ID 0xba2bf0a4
8	3.005016	1.1.1.1	172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462	DHCP ACK - Transaction ID 0xba2bf0a4

> **Hinweis:** Wenn Sie mehr über 802.11-Wireless-Sniffing und über die Filter/Farben erfahren möchten, die in Wireshark für die in diesem Dokument enthaltenen Bilder verwendet werden, besuchen Sie den Beitrag der Cisco Support Community mit dem Titel 802.11 Sniffer image Analysis (Sniffer-Bildanalyse).

Der Wireless-Client beginnt mit dem Authentifizierungsframe, und der WAP antwortet mit einem weiteren Authentifizierungsframe. Der Client sendet dann den Zuordnungsanforderungsrahmen, und der Zugangspunkt wird in einer Antwort mit dem Zuordnungsanantwortrahmen beendet. Wie aus den DHCP-Paketen ersichtlich, beginnt der Client nach der Übergabe der 802.11 Open System-Authentifizierungs- und -Zuordnungsprozesse, Daten-Frames zu übergeben. In diesem Fall ist keine Sicherheitsmethode auf der SSID konfiguriert, sodass der Client sofort beginnt, nicht verschlüsselte Datenframes (in diesem Fall DHCP) zu senden.

Wenn die Sicherheit auf der SSID aktiviert ist, werden, wie weiter unten in diesem Dokument gezeigt, für die spezifische Sicherheitsmethode Authentifizierungs- und Verschlüsselungs-Handshake-Frames auf höherer Ebene bereitgestellt, unmittelbar nach der Zuordnungsantwort und vor dem Senden von Client-Datenverkehr-Frames, wie DHCP, Address Resolution Protocol (ARP) und verschlüsselte Anwendungspakete. Daten-Frames können nur gesendet werden, bis der Client vollständig authentifiziert ist und die Verschlüsselungsschlüssel entsprechend der konfigurierten Sicherheitsmethode ausgehandelt wurden.

Basierend auf dem vorherigen Bild sind hier die Meldungen aufgeführt, die Sie in den Ausgaben des WLC-**Debug-Client**-Befehls sehen, wenn der Wireless-Client eine neue Verbindung zum WLAN beginnt:

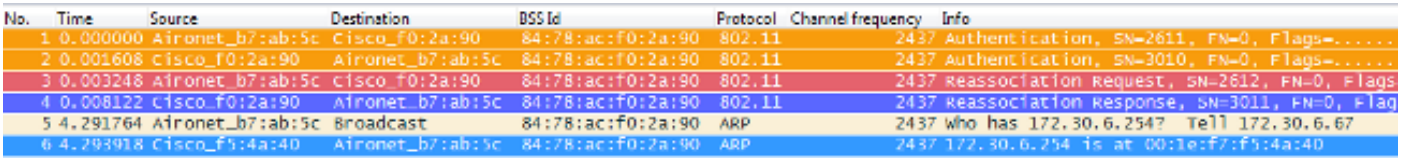
```
*apfMsConnTask_0: Jun 21 18:55:14.221: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d0
!--- This is the Association Request from the wireless client
     to the selected AP.

*apfMsConnTask_0: Jun 21 18:55:14.222: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d0
  (status 0) ApVapId 1 Slot 0
!--- This is the Association Response from the AP to the client.
```

> **Hinweis**: Das WLC-Debugging für die in diesem Dokument dargestellten Ausgaben ist der **Debug-Client**-Befehl, und die Beispiele zeigen nur einige relevante Meldungen, nicht die gesamte Ausgabe. Weitere Informationen zu diesem Debug-Befehl finden Sie im Dokument Understand the Debug Client on Wireless LAN Controllers (WLCs).

Diese Meldungen zeigen die Zuordnungsanforderung und Antwort-Frames an. Die anfänglichen Authentifizierungs-Frames werden nicht am WLC protokolliert, da dieser Handshake schnell auf AP-Ebene im CUWN stattfindet.

Welche Informationen werden beim Roaming des Clients angezeigt? Der Client tauscht beim Herstellen einer Verbindung zu einem AP immer vier Management-Frames aus, was entweder auf den Client-Aufbau einer Verbindung oder auf ein Roaming-Ereignis zurückzuführen ist. Der Client hat jeweils nur eine Verbindung mit nur einem WAP hergestellt. Der einzige Unterschied beim Frame-Austausch zwischen einer neuen Verbindung zur WLAN-Infrastruktur und einem Roaming-Ereignis besteht darin, dass die Zuordnungs-Frames eines Roaming-Ereignisses **Reassoziations**-Frames genannt werden, die anzeigen, dass der Client tatsächlich von einem anderen WAP Roaming durchführt, ohne dass versucht wird, eine neue Verbindung zum WLAN herzustellen. Diese Frames können verschiedene Elemente enthalten, die zum Aushandeln des Roaming-Ereignisses verwendet werden. Dies hängt von der Konfiguration ab, diese Details werden jedoch nicht in diesem Dokument behandelt.

Hier ein Beispiel für den Frame-Austausch:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:90	84:78:ac:f0:2a:90	802.11	2437	Authentication, SN=2611, FN=0, Flags=.....
2	0.001608	Cisco_f0:2a:90	Aironet_b7:ab:5c	84:78:ac:f0:2a:90	802.11	2437	Authentication, SN=3010, FN=0, Flags=.....
3	0.003248	Aironet_b7:ab:5c	Cisco_f0:2a:90	84:78:ac:f0:2a:90	802.11	2437	Reassociation Request, SN=2612, FN=0, Flags
4	0.008122	Cisco_f0:2a:90	Aironet_b7:ab:5c	84:78:ac:f0:2a:90	802.11	2437	Reassociation Response, SN=3011, FN=0, Flag
5	4.291764	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:90	ARP	2437	Who has 172.30.6.254? Tell 172.30.6.67
6	4.293918	Cisco_f5:4a:40	Aironet_b7:ab:5c	84:78:ac:f0:2a:90	ARP	2437	172.30.6.254 is at 00:1e:f7:f5:4a:40

Diese Meldungen werden in der Debug-Ausgabe angezeigt:

```
*apfMsConnTask_2: Jun 21 19:02:19.709: 00:40:96:b7:ab:5c
  Reassociation received from mobile on BSSID 84:78:ac:f0:2a:90
!--- This is the Reassociation Request from the wireless client
     to the selected AP.

*apfMsConnTask_2: Jun 21 19:02:19.710: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:90
  (status 0) ApVapId 1 Slot 0
!--- This is the Reassociation Response from the AP to the client.
```

Wie dargestellt, führt der Client erfolgreich ein Roaming-Ereignis aus, nachdem die

Neuzuordnungsanforderung an den neuen WAP gesendet wurde, und empfängt die Neuzuordnungsanwort vom WAP. Da der Client bereits über eine IP-Adresse verfügt, sind die ersten Daten-Frames für ARP-Pakete.

Wenn Sie ein Roaming-Ereignis erwarten, aber der Client eine Zuordnungsanforderung anstelle einer Neuzuordnungsanforderung sendet (die Sie anhand einiger Bilder und Debugs bestätigen können, die den zuvor in diesem Dokument beschriebenen ähneln), dann ist der Client kein wirkliches Roaming. Der Client beginnt eine neue Verbindung mit dem WLAN, als ob eine Trennung stattgefunden hätte, und versucht, die Verbindung von Grund auf wiederherzustellen. Dies kann aus mehreren Gründen geschehen, z. B. wenn ein Client sich von den Abdeckungsbereichen entfernt und dann einen Access Point mit ausreichender Signalqualität findet, um eine Verbindung herzustellen. Normalerweise weist dies jedoch auf ein Client-Problem hin, bei dem der Client aufgrund von Treibern, Firmware oder Softwareproblemen kein Roaming-Ereignis initiiert.

> **Hinweis**: Sie können sich an den Hersteller des Wireless-Clients wenden, um die Ursache des Problems zu ermitteln.

Roaming mit umfassender Sicherheit

Wenn die SSID zusätzlich zur grundlegenden 802.11-Authentifizierung des offenen Systems mit L2-Sicherheit konfiguriert wird, sind für die ursprüngliche Zuordnung und beim Roaming mehr Frames erforderlich. Die beiden gebräuchlichsten Sicherheitsmethoden, die für 802.11-WLANs standardisiert und implementiert werden, werden in diesem Dokument beschrieben:

- **WPA/WPA2-PSK (Pre-Shared Key)** - Authentifizierung von Clients mit einem Preshared Key.
- **WPA/WPA2-EAP (Extensible Authentication Protocol)** - Authentifizierung von Clients mit einer 802.1X/EAP-Methode, um sicherere Anmeldeinformationen durch die Verwendung eines Authentifizierungsservers wie Zertifikate, Benutzername und Kennwort und Token zu validieren.

Es ist wichtig zu wissen, dass diese beiden Methoden (PSK und EAP) die Clients zwar auf unterschiedliche Weise authentifizieren/validieren, aber beide im Wesentlichen die gleichen WPA/WPA2-Regeln für den Schlüsselverwaltungsprozess verwenden. Unabhängig davon, ob es sich um WPA/WPA2-PSK oder WPA/WPA2-EAP handelt, beginnt der als WPA/WPA2-4-Way-Handshake bekannte Prozess die Schlüsselverhandlung zwischen dem WLC/AP und dem Client mit einem Master Session Key (MSK) als ursprünglichem Schlüsselmaterial, sobald der Client mit der spezifischen verwendeten Authentifizierungsmethode validiert wurde.

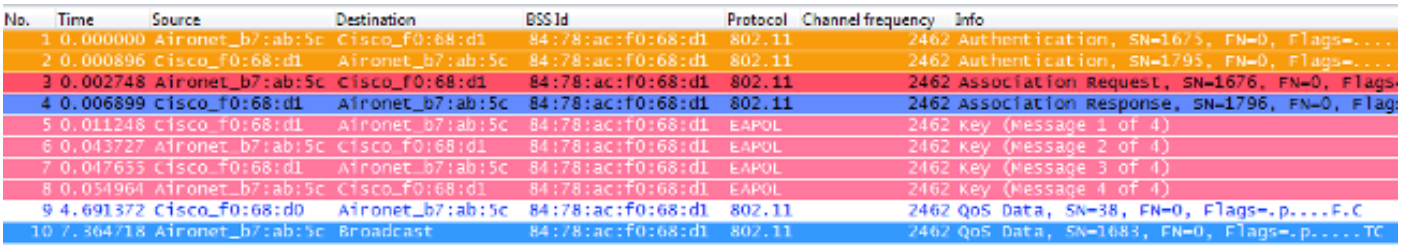
Im Folgenden finden Sie eine Zusammenfassung des Prozesses:

1. Ein MSK wird aus der EAP-Authentifizierungsphase abgeleitet, wenn 802.1X/EAP-Sicherheit verwendet wird, oder aus dem PSK, wenn WPA/WPA2-PSK als Sicherheitsmethode verwendet wird.
2. Von diesem MSK leiten der Client und WLC/AP den Pairwise Master Key (PMK) ab, und der WLC/AP generiert einen Group Master Key (GMK).
3. Sobald diese beiden Hauptschlüssel bereit sind, initiieren der Client und der WLC/AP den 4-Wege-Handshake von WPA/WPA2 (der später in diesem Dokument mit einigen Screenshots und Debugs veranschaulicht wird) mit den Hauptschlüsseln als Grundlage für die Aushandlung der eigentlichen Verschlüsselungsschlüssel.

4. Diese endgültigen Verschlüsselungsschlüssel werden als Pairwise Transient Key (PTK) und Group Transient Key (GTK) bezeichnet. Die PTK wird aus der PMK abgeleitet und zur Verschlüsselung von Unicast-Frames mit dem Client verwendet. Der Group Transient Key (GTK) wird aus dem GMK abgeleitet und zur Verschlüsselung von Multicast/Broadcast für diese spezifische SSID/diesen AP verwendet.

WPA/WPA2-PSK

Wenn WPA-PSK oder WPA2-PSK über TKIP (Temporal Key Integrity Protocol) oder AES (Advanced Encryption Standard) für die Verschlüsselung ausgeführt wird, muss der Client den als WPA 4-Way Handshake bekannten Prozess sowohl für die ursprüngliche Zuordnung als auch beim Roaming durchlaufen. Wie zuvor erläutert, ist dies im Wesentlichen der Schlüsselverwaltungsprozess, der verwendet wird, damit WPA/WPA2 die Verschlüsselungsschlüssel ableiten kann. Wenn jedoch PSK ausgeführt wird, wird es auch verwendet, um zu überprüfen, ob der Client über einen gültigen vorinstallierten Schlüssel verfügt, um dem WLAN beizutreten. Dieses Bild zeigt den anfänglichen Zuordnungsprozess, wenn WPA oder WPA2 mit PSK ausgeführt wird:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d1	84:78:ac:f0:68:d1	802.11	2462	Authentication, SN=1675, FN=0, Flags=....
2	0.000896	Cisco_f0:68:d1	Aironet_b7:ab:5c	84:78:ac:f0:68:d1	802.11	2462	Authentication, SN=1795, FN=0, Flags=....
3	0.002748	Aironet_b7:ab:5c	Cisco_f0:68:d1	84:78:ac:f0:68:d1	802.11	2462	Association Request, SN=1676, FN=0, Flags
4	0.006899	Cisco_f0:68:d1	Aironet_b7:ab:5c	84:78:ac:f0:68:d1	802.11	2462	Association Response, SN=1796, FN=0, Flag
5	0.011248	Cisco_f0:68:d1	Aironet_b7:ab:5c	84:78:ac:f0:68:d1	EAPOL	2462	Key (Message 1 of 4)
6	0.043727	Aironet_b7:ab:5c	Cisco_f0:68:d1	84:78:ac:f0:68:d1	EAPOL	2462	Key (Message 2 of 4)
7	0.047655	Cisco_f0:68:d1	Aironet_b7:ab:5c	84:78:ac:f0:68:d1	EAPOL	2462	Key (Message 3 of 4)
8	0.054964	Aironet_b7:ab:5c	Cisco_f0:68:d1	84:78:ac:f0:68:d1	EAPOL	2462	Key (Message 4 of 4)
9	4.691372	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d1	802.11	2462	QoS Data, SN=38, FN=0, Flags=.p....F.C
10	7.364718	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:68:d1	802.11	2462	QoS Data, SN=1683, FN=0, Flags=.p.....TC

Wie gezeigt, gibt es nach dem 802.11 Open System Authentifizierungs- und Zuordnungsprozess vier EAPOL-Frames vom 4-Wege-WPA-Handshake, die vom WAP mit **message-1** initiiert und vom Client mit **message-4** beendet werden. Nach einem erfolgreichen Handshake beginnt der Client, Datenframes (wie DHCP) zu übergeben, die in diesem Fall mit den vom 4-Wege-Handshake abgeleiteten Schlüsseln verschlüsselt werden (deshalb können Sie den tatsächlichen Inhalt und die Art des Verkehrs von den Wireless-Bildern nicht sehen).

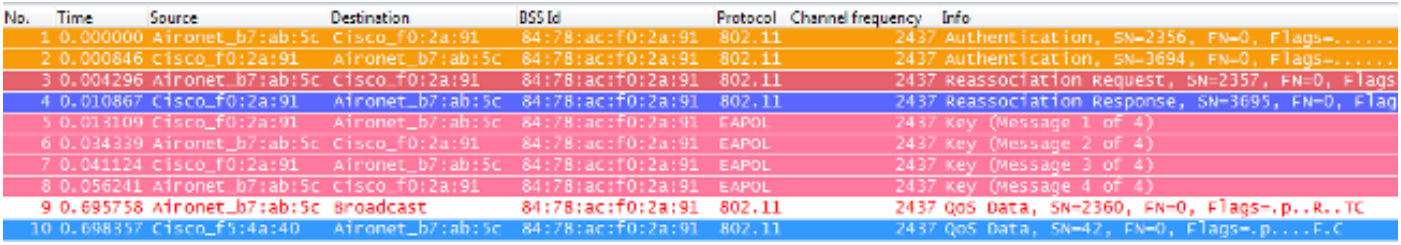
> **Hinweis:** EAPOL-Frames werden verwendet, um alle wichtigen Management-Frames und 802.1X/EAP-Authentifizierungsframes drahtlos zwischen dem AP und dem Client zu übertragen. Sie werden als Wireless-Datenframes übertragen.

Diese Meldungen werden in den Debug-Ausgaben angezeigt:

```
*apfMsConnTask_0: Jun 21 19:30:05.172: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d1
*apfMsConnTask_0: Jun 21 19:30:05.173: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d1
  (status 0) ApVapId 2 Slot 0
!--- The Association handshake is finished.

*dot1xMsgTask: Jun 21 19:30:05.178: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00
!--- Message-1 of the WPA/WPA2 4-Way handshake is sent
     from the WLC/AP to the client.
```

```
*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.289: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.289: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2)
  from mobile 00:40:96:b7:ab:5c
!--- Message-2 of the WPA/WPA2 4-Way handshake is successfully
     received from the client.

*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.290: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
!--- Message-3 of the WPA/WPA2 4-Way handshake is sent
     from the WLC/AP to the client.

*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.309: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.310: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile 00:40:96:b7:ab:5c
!--- Message-4 (final message) of the WPA/WPA2 4-Way handshake
     is successfully received from the client, which confirms
     the installation of the derived keys. They can now be used in
     order to encrypt data frames with current AP.
```

Beim Roaming verfolgt der Client im Grunde den gleichen Frame Exchange, bei dem der WPA 4-Wege Handshake erforderlich ist, um neue Verschlüsselungsschlüssel mit dem neuen AP abzuleiten. Dies liegt an den vom Standard festgelegten Sicherheitsgründen und daran, dass der neue Access Point die ursprünglichen Schlüssel nicht kennt. Der einzige Unterschied besteht darin, dass anstelle von Zuordnungsrahmen Neuzuordnungsrahmen vorhanden sind, wie in diesem Bild gezeigt:



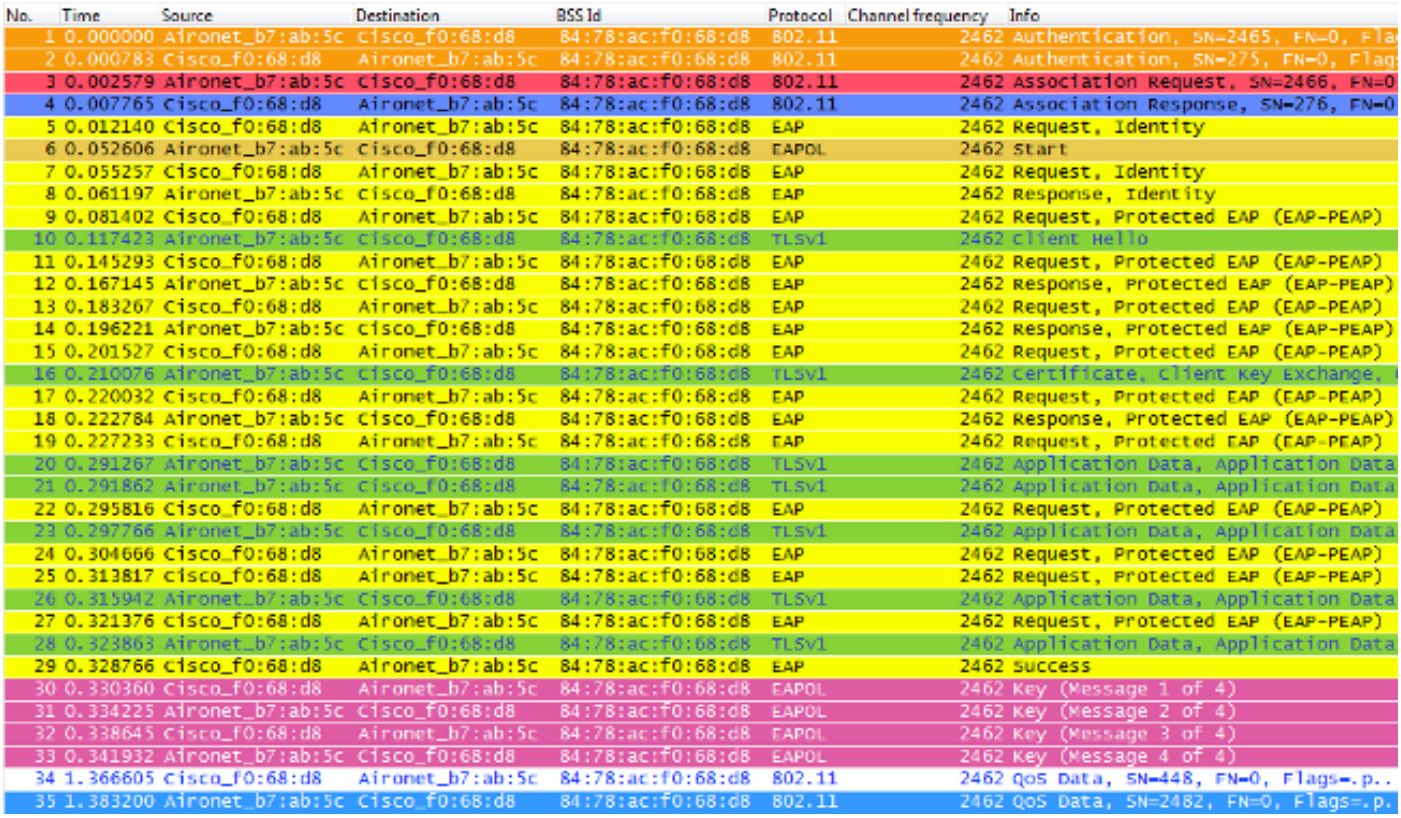
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3	0.004296	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Reassociation Request, SN=2357, FN=0, Flags
4	0.010867	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Reassociation Response, SN=3695, FN=0, Flag
5	0.013109	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	EAPOL	2437	Key (Message 1 of 4)
6	0.034339	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	EAPOL	2437	Key (Message 2 of 4)
7	0.041124	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	EAPOL	2437	Key (Message 3 of 4)
8	0.056241	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	EAPOL	2437	Key (Message 4 of 4)
9	0.695758	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:91	802.11	2437	QoS Data, SN=2360, FN=0, Flags=.p..R..TC
10	0.698357	Cisco_f5:4a:40	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	QoS Data, SN=42, FN=0, Flags=.p....F.C

Sie sehen die gleichen Meldungen in den Debug-Ausgaben, aber das erste Paket vom Client ist eine Neuzuordnung anstelle einer Zuordnung, wie zuvor gezeigt und erläutert.

WPA/WPA2-EAP

Wenn eine 802.1X/EAP-Methode verwendet wird, um die Clients auf einer sicheren SSID zu authentifizieren, sind noch mehr Frames erforderlich, bevor der Client beginnt, Datenverkehr zu übertragen. Diese zusätzlichen Frames werden zur Authentifizierung der Client-Anmeldeinformationen verwendet. Je nach EAP-Methode können zwischen vier und zwanzig Frames liegen. Diese kommen nach der Zuordnung/Neuzuordnung, aber vor dem 4-Wege-Handshake von WPA/WPA2, da die Authentifizierungsphase das MSK ableitet, das als Seed für die endgültige Verschlüsselungsschlüsselgenerierung im Schlüsselverwaltungsprozess verwendet wird (4-Wege-Handshake).

Dieses Bild zeigt ein Beispiel für die Frames, die bei der anfänglichen Zuordnung zwischen dem Access Point und dem Wireless-Client per Funk ausgetauscht werden, wenn WPA mit PEAPv0/EAP-MSCHAPv2 ausgeführt wird:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	Authentication, SN=2465, FN=0, Fla
2	0.000783	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	Authentication, SN=275, FN=0, Flag
3	0.002579	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	Association Request, SN=2466, FN=0
4	0.007765	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	Association Response, SN=276, FN=0
5	0.012140	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Identity
6	0.052606	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Start
7	0.055257	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Identity
8	0.061197	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Identity
9	0.081402	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
10	0.117423	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Client Hello
11	0.145293	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.167145	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.183267	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.196221	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
15	0.201527	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	0.210076	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Certificate, Client Key Exchange,
17	0.220032	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	0.222784	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
19	0.227233	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
20	0.291267	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
21	0.291862	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
22	0.295816	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	0.297766	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
24	0.304666	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	0.313817	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
26	0.315942	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
27	0.321376	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
28	0.323863	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
29	0.328766	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Success
30	0.330360	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 1 of 4)
31	0.334225	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 2 of 4)
32	0.338645	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 3 of 4)
33	0.341932	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 4 of 4)
34	1.366605	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	QoS Data, SN=448, FN=0, Flags=.p..
35	1.383200	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	QoS Data, SN=2482, FN=0, Flags=.p.

Manchmal zeigt dieser Austausch mehr oder weniger Frames, was von mehreren Faktoren abhängt, wie der EAP-Methode, Neuübertragungen aufgrund von Problemen, Client-Verhalten (wie die beiden Identitätsanforderungen in diesem Beispiel, weil der Client einen **EAPOL START** sendet, nachdem der WAP die erste Identitätsanforderung sendet), oder wenn der Client das Zertifikat bereits mit dem Server ausgetauscht hat. Wenn die SSID für eine 802.1X/EAP-Methode konfiguriert wird, gibt es mehr Frames (für die Authentifizierung), und daher dauert es länger, bis der Client beginnt, Datenframes zu senden.

Hier eine Zusammenfassung der Debug-Meldungen:

```
*apfMsConnTask_0: Jun 21 23:41:19.092: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d8
*apfMsConnTask_0: Jun 21 23:41:19.094: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d8
  (status 0) ApVapId 9 Slot 0
!--- The Association handshake is finished.

*dot1xMsgTask: Jun 21 23:41:19.098: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 1)
!--- The EAP Identity Request is sent to the client once it is
     associated in order to begin the higher-level authentication
     process. This informs the client that an identity to start
     this type of 802.1X/EAP authentication must be provided.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.226: 00:40:96:b7:ab:5c
  Received EAPOL START from mobile 00:40:96:b7:ab:5c
!--- The wireless client decides to start the EAP authentication
     process, and informs the AP with an EAPOL START data frame.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.227: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 2)
!--- WLC/AP sends another EAP Identity Request to the client.
```

```
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.235: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.235: 00:40:96:b7:ab:5c
  Received Identity Response (count=2) from mobile 00:40:96:b7:ab:5c
!--- The client responds with an EAP Identity Response on an EAPOL
     frame.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.301: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.301: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 3)
!--- Once the WLC/AP sends the client response to the Authentication
     Server on a RADIUS Access-Request packet, the server responds
     with a RADIUS Access-Challenge in order to officially start the
     EAP negotiation, handshake, and authentication with the client
     (sometimes with mutual authentication, dependent upon the EAP
     method). This response received by the WLC/AP is sent to the client.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.344: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.344: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 3, EAP Type 25)
!--- The client responds with an EAP Response on an EAPOL frame, which
     is sent to the Authentication Server on a RADIUS Access-Request
     packet. The server responds with another RADIUS Access-Challenge.
     This process continues, dependent upon the EAP method (the exchange
     of certificates when used, the building of TLS tunnels, validation
     of client credentials, client validation of server identity when
     applicable). Hence, the next few messages are basically the same on
     the WLC/AP side, as this acts as a "proxy" between the client and
     the Authentication Server exchanges.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.347: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.347: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 4)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.375: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.375: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 4, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.377: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.377: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 5)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.403: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.403: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 5, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.404: 00:40:96:b7:ab:5c
```

```
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.404: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 6)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.414: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.414: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 6, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.421: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.421: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 7)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.425: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.425: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 7, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.427: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.427: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 8)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.434: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.434: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 8, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.436: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.436: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 9)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.440: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.440: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 9, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.442: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.442: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 10)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.449: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c
```

```
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.449: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 10, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.452: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.452: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 11)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.457: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.457: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 11, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.459: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.459: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 13)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.469: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.469: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 13, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.472: 00:40:96:b7:ab:5c
  Processing Access-Accept for mobile 00:40:96:b7:ab:5c
!--- The authentication finishes and is successful for this client,
     so the RADIUS Server sends a RADIUS Access-Accept to the WLC/AP.
     This RADIUS Access-Accept comes with the special attributes
     that are assigned to this client (if any are configured on the
     Authentication Server for this client). This Access-Accept also
     comes with the MSK derived with the client in the EAP
     authentication process, so the WLC/AP installs it in order to
     initiate the WPA/WPA2 4-Way handshake with the wireless client.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.473: 00:40:96:b7:ab:5c
  Sending EAP-Success to mobile 00:40:96:b7:ab:5c
  (EAP Id 13)
!--- The accept/pass of the authentication is sent to the client as
     an EAP-Success message.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.473: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00
!--- Message-1 of the WPA/WPA2 4-Way handshake is sent from the
     WLC/AP to the client.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.481: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.481: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2)
  from mobile 00:40:96:b7:ab:5c
!--- Message-2 of the WPA/WPA-2 4-Way handshake is successfully
     received from the client.
```

```
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.481: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
!--- Message-3 of the WPA/WPA2 4-Way handshake is sent from the
     WLC/AP to the client.

*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.487: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.487: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile 00:40:96:b7:ab:5c
!--- Message-4 (final message) of the WPA/WPA2 4-Way handshake
     is successfully received from the client, which confirms the
     installation of the derived keys. They can now be used in
     order to encrypt data frames with the current AP.
```

Wenn der Wireless-Client hier ein reguläres Roaming durchführt (das normale Verhalten, ohne dass eine schnelle Roaming-Methode implementiert wird), muss der Client den gleichen Prozess durchlaufen und eine vollständige Authentifizierung gegenüber dem Authentifizierungsserver durchführen, wie in den Bildern gezeigt. Der einzige Unterschied besteht darin, dass der Client eine Neuzuordnungsanforderung verwendet, um den neuen WAP darüber zu informieren, dass er tatsächlich von einem anderen WAP aus Roaming nutzt. Der Client muss jedoch noch eine vollständige Validierung und die Generierung eines neuen Schlüssels durchlaufen:

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=2637, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=96, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2638, FN=0, Flags=...
4	0.008646	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Reassociation Response, SN=97, FN=0, Flags=....
5	0.014409	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
6	0.029712	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Start
7	0.035034	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
8	0.053240	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAP	2437	Response, Identity
9	0.062770	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Protected EAP (EAP-PEAP)
10	0.065313	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Client Hello
11	0.071282	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Server Hello, Change Cipher Spec, Encrypted Han
12	0.077740	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Change Cipher Spec, Encrypted Handshake Message
13	0.083816	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Application Data
14	0.092138	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Success
15	0.093699	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 1 of 4)
16	0.097014	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 2 of 4)
17	0.100739	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 3 of 4)
18	0.105180	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 4 of 4)
19	1.125063	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=76, FN=0, Flags=.p....F.C
20	4.383568	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=2647, FN=0, Flags=.p.....TC

Wie gezeigt, müssen auch bei weniger Frames als bei der anfänglichen Authentifizierung (die durch mehrere Faktoren verursacht wird, wie bereits erwähnt), wenn der Client zu einem neuen AP roamt, die EAP-Authentifizierung und die WPA-Schlüsselverwaltungsprozesse noch abgeschlossen sein, um weiterhin Datenframes weiterzuleiten (auch wenn der Datenverkehr vor dem Roaming aktiv gesendet wurde). Wenn der Client daher über eine aktive Anwendung verfügt, die empfindlich auf Verzögerungen reagiert (z. B. Anwendungen für Sprachdatenverkehr oder Anwendungen, die empfindlich auf Zeitüberschreitungen reagieren), kann der Benutzer Probleme beim Roaming erkennen, z. B. Audiolücken oder Anwendungsunterbrechungen. Dies hängt davon ab, wie lange der Prozess dauert, bis der Client weiterhin Datenframes sendet/empfängt. Diese Verzögerung kann länger sein und hängt von der Funkumgebung, der Anzahl der Clients, der Round-Trip-Zeit zwischen dem WLC und den LAPs sowie mit dem Authentifizierungsserver und anderen Gründen ab.

Im Folgenden finden Sie eine Zusammenfassung der Debug-Meldungen für dieses Roaming-Ereignis (im Wesentlichen identisch mit den vorherigen Meldungen, sodass diese nicht weiter beschrieben werden):

```
*apfMsConnTask_2: Jun 21 23:47:54.872: 00:40:96:b7:ab:5c
```

```
  Reassociation received from mobile on BSSID 84:78:ac:f0:2a:98

*apfMsConnTask_2: Jun 21 23:47:54.874: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:98
  (status 0) ApVapId 9 Slot 0

*dot1xMsgTask: Jun 21 23:47:54.879: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 1)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.895: 00:40:96:b7:ab:5c
  Received EAPOL START from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.895: 00:40:96:b7:ab:5c
  dot1x - moving mobile 00:40:96:b7:ab:5c into Connecting state

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.895: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 2)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.922: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.922: 00:40:96:b7:ab:5c
  Received Identity Response (count=2) from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.929: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.929: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 3)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.941: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.941: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 3, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.943: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.943: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 4)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.954: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.954: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 4, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.956: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.957: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 7)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.976: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c
```

```
*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.976: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 7, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.978: 00:40:96:b7:ab:5c
  Processing Access-Accept for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.978: 00:40:96:b7:ab:5c
  Sending EAP-Success to mobile 00:40:96:b7:ab:5c
  (EAP Id 7)

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.978: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.995: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.995: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2)
  from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:54.995: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_4: Jun 21 23:47:55.005: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 23:47:55.005: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile 00:40:96:b7:ab:5c
```

So funktionieren 802.1X/EAP und das WPA/WPA2-Sicherheits-Framework. Um die Auswirkungen von Anwendungen/Diensten auf Verzögerungen durch ein reguläres Roaming-Ereignis zu vermeiden, werden von der WiFi-Branche mehrere schnelle Roaming-Methoden entwickelt und implementiert, um den Roaming-Prozess zu beschleunigen, wenn auf dem WLAN/SSID Sicherheit verwendet wird. Die Clients müssen mit einer gewissen Latenz rechnen, wenn sie beim Roaming zwischen den APs weiterhin Datenverkehr weiterleiten, indem sie im WLAN ein hohes Maß an Sicherheit implementieren. Dies ist auf die EAP-Authentifizierung und den Schlüsselmanagement-Frame-Austausch zurückzuführen, die für die Sicherheitseinrichtung erforderlich sind, wie zuvor erläutert.

Es ist wichtig zu verstehen, dass schnelles sicheres Roaming nur der Begriff ist, der von der Branche für die Implementierung einer Methode/eines Schemas verwendet wird, die/das den Roaming-Prozess beschleunigt, wenn die Sicherheit im WLAN konfiguriert ist. Im nächsten Abschnitt werden die verschiedenen, für WLANs verfügbaren und vom CUWN unterstützten Methoden und Schemata für schnelles und sicheres Roaming erläutert.

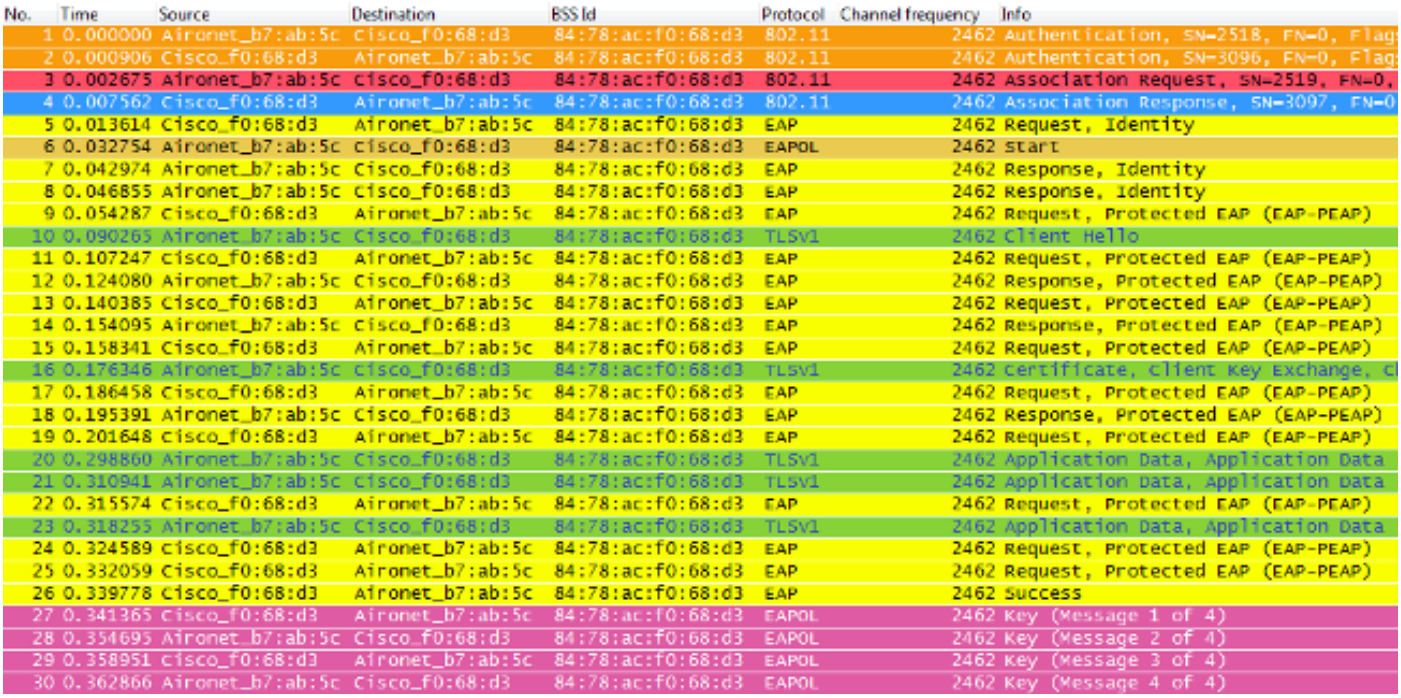
Schnelles und sicheres Roaming mit CCKM

Cisco Centralized Key Management (CCKM) ist die erste schnelle und sichere Roaming-Methode, die in Unternehmens-WLANs entwickelt und implementiert wurde. Sie wurde von Cisco als Lösung entwickelt, um die bislang erläuterten Verzögerungen zu reduzieren, wenn 802.1X/EAP-Sicherheit im WLAN verwendet wird. Da es sich um ein proprietäres Protokoll von Cisco handelt, wird es nur von Cisco WLAN-Infrastrukturgeräten und Wireless-Clients (von verschiedenen Anbietern) unterstützt, die mit Cisco Compatible Extension (CCX) kompatibel für CCKM sind.

CCKM kann mit allen verfügbaren Verschlüsselungsmethoden für WLANs implementiert werden, darunter WEP, TKIP und AES. Sie wird auch von den meisten für WLANs verwendeten 802.1X/EAP-Authentifizierungsmethoden unterstützt, je nach der von den Geräten unterstützten CCX-Version.

> **Hinweis:** Einen Überblick über die von den verschiedenen Versionen der CCX-Spezifikation unterstützten Funktionen (einschließlich unterstützter EAP-Methoden) finden Sie im Dokument "CCX-Versionen und -Funktionen" und überprüfen Sie die genaue CCX-Version, die von Ihren Wireless-Clients unterstützt wird (sofern diese CCX-kompatibel sind), damit Sie überprüfen können, ob die gewünschte Sicherheitsmethode mit CCKM implementiert werden kann.

Dieses Wireless-Image bietet ein Beispiel für die Frames, die bei der ersten Zuordnung ausgetauscht werden, wenn Sie CCKM mit TKIP als Verschlüsselung und PEAPv0/EAP-MSCHAPv2 als 802.1X/EAP-Methode durchführen. Dies ist im Wesentlichen der gleiche Austausch, als ob WPA/TKIP mit PEAPv0/EAP-MSCHAPv2 ausgeführt wird, aber diesmal wird CCKM zwischen dem Client und der Infrastruktur ausgehandelt, sodass sie unterschiedliche Schlüsselhierarchie- und Cache-Methoden verwenden, um Fast Secure Roaming auszuführen, wenn der Client Roaming ausführen muss:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	802.11	2462	Authentication, SN=2518, FN=0, Flag
2	0.000906	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	802.11	2462	Authentication, SN=3096, FN=0, Flag
3	0.002675	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	802.11	2462	Association Request, SN=2519, FN=0,
4	0.007562	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	802.11	2462	Association Response, SN=3097, FN=0
5	0.013614	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Identity
6	0.032754	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAPOL	2462	Start
7	0.042974	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAP	2462	Response, Identity
8	0.046855	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAP	2462	Response, Identity
9	0.054287	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
10	0.090265	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	TLSv1	2462	Client Hello
11	0.107247	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.124080	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.140385	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.154095	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAP	2462	Response, Protected EAP (EAP-PEAP)
15	0.158341	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	0.176346	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	TLSv1	2462	Certificate, Client Key Exchange, C
17	0.186458	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	0.195391	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAP	2462	Response, Protected EAP (EAP-PEAP)
19	0.201648	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
20	0.298860	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	TLSv1	2462	Application Data, Application Data
21	0.310941	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	TLSv1	2462	Application Data, Application Data
22	0.315574	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	0.318255	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	TLSv1	2462	Application Data, Application Data
24	0.324589	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	0.332059	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Request, Protected EAP (EAP-PEAP)
26	0.339778	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAP	2462	Success
27	0.341365	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAPOL	2462	Key (Message 1 of 4)
28	0.354695	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAPOL	2462	Key (Message 2 of 4)
29	0.358951	Cisco_f0:68:d3	Aironet_b7:ab:5c	84:78:ac:f0:68:d3	EAPOL	2462	Key (Message 3 of 4)
30	0.362866	Aironet_b7:ab:5c	Cisco_f0:68:d3	84:78:ac:f0:68:d3	EAPOL	2462	Key (Message 4 of 4)

Hier eine Zusammenfassung der Debug-Meldungen (mit einigen EAP-Austauschen entfernt, um die Ausgabe zu reduzieren):

```
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d3
!--- This is the Association Request from the client.

*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Processing WPA IE type 221, length 22 for mobile
```

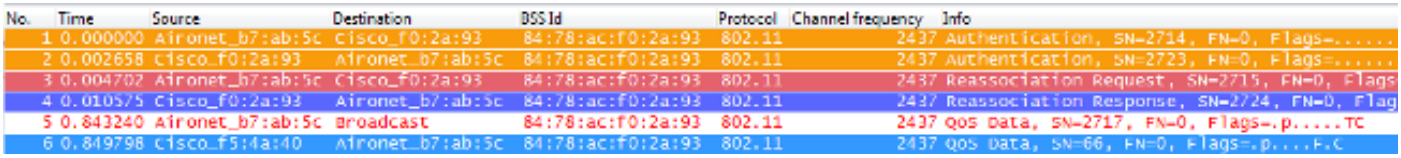
```
  00:40:96:b7:ab:5c
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  CCKM: Mobile is using CCKM
!--- The WLC/AP finds an Information Element that claims CCKM
     support on the Association request that is sent from the client.

*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
!--- This is the key cache index for this client, which is set temporally.

*apfMsConnTask_0: Jun 25 15:41:41.508: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d3
  (status 0) ApVapId 4 Slot 0
!--- The Association Response is sent to the client.

*dot1xMsgTask: Jun 25 15:41:41.513: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 1)
!--- An EAP Identity Request is sent to the client once it is
     associated in order to begin the higher-level authentication
     process. This informs the client that an identity to start
     this type of 802.1X/EAP authentication must be provided.
     Further EAP messages are not described, as they are basically
     the same as the ones previously-explained.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.536: 00:40:96:b7:ab:5c
  Received EAPOL START from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.536: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 2)

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.546: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.546: 00:40:96:b7:ab:5c
  Received EAP Response packet with mismatching id
  (currentid=2, eapid=1) from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.550: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.550: 00:40:96:b7:ab:5c
  Received Identity Response (count=2) from mobile
  00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.555: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.555: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 3)

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.594: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.594: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 3, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.840: 00:40:96:b7:ab:5c
  Processing Access-Accept for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
```

```
  Creating a PKC PMKID Cache entry for station 00:40:96:b7:ab:5c
  (RSN 0)<br/ >
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 0
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  CCKM: Create a global PMK cache entry
!--- WLC creates a global PMK cache entry for this client,
     which is for CCKM in this case.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Sending EAP-Success to mobile 00:40:96:b7:ab:5c
  (EAP Id 13)
!--- The client is informed of the successful EAP authentication.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c state
  INITPMK(message 1),replay counter 00.00.00.00.00.00.00.00
!--- Message-1 of the initial 4-Way handshake is sent from the
     WLC/AP to the client.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2) from mobile
  00:40:96:b7:ab:5c
!--- Message-2 of the initial 4-Way handshake is received
     successfully from the client.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: 00:40:96:b7:ab:5c
  CCKM: Sending cache add
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: CCKM: Sending CCKM PMK
  (Version_1) information to mobility group
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: CCKM: Sending CCKM PMK
  (Version_2) information to mobility group
!--- The CCKM PMK cache entry for this client is shared with
     the WLCs on the mobility group.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.858: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
!--- Message-3 of the initial 4-Way handshake is sent from the
     WLC/AP to the client.

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.866: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.866: 00:40:96:b7:ab:5c Received
  EAPOL-key in PTKINITNEGOTIATING state (message 4) from mobile
  00:40:96:b7:ab:5c
!--- Message-4 (final message) of this initial 4-Way handshake
     is received successfully from the client, which confirms the
     installation of the derived keys. They can now be used in order
     to encrypt data frames with the current AP.
```

Bei CCKM ähnelt die anfängliche Zuordnung zum WLAN der regulären WPA/WPA2, bei der ein MSK (auch als Network Session Key (NSK) bezeichnet) gemeinsam mit dem Client und dem RADIUS-Server abgeleitet wird. Dieser Primärschlüssel wird nach erfolgreicher Authentifizierung vom Server an den WLC gesendet und als Grundlage für die Ableitung aller nachfolgenden Schlüssel für die Lebensdauer der Client-Verknüpfung mit diesem WLAN zwischengespeichert. Von hier aus leiten der WLC und der Client die Seed-Informationen ab, die für ein schnelles und sicheres Roaming auf der Basis von CCKM verwendet werden. Dies erfolgt ähnlich wie bei

WPA/WPA2 über einen 4-Wege-Handshake, um die Unicast- (PTK) und Multicast-/Broadcast- (GTK) Verschlüsselungsschlüssel mit dem ersten AP abzuleiten.

Der große Unterschied wird beim Roaming bemerkt. In diesem Fall sendet der CCKM-Client einen einzelnen Frame für eine Neuzuordnungsanforderung an den AP/WLC (der ein MIC und eine sequenziell inkrementierende Zufallszahl enthält) und stellt genügend Informationen bereit (einschließlich der neuen AP-MAC-Adresse **-BSSID-**), um die neue PTK abzuleiten. Mit dieser Reassoziationsanforderung verfügen der WLC und der neue AP auch über genügend Informationen, um die neue PTK abzuleiten, sodass sie einfach mit einer Reassoziationsantwort antworten. Der Client kann nun weiterhin Datenverkehr weiterleiten, wie in diesem Bild gezeigt:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:93	84:78:ac:f0:2a:93	802.11	2437	Authentication, SN=2714, FN=0, Flags=.....
2	0.002658	Cisco_f0:2a:93	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	Authentication, SN=2723, FN=0, Flags=.....
3	0.004702	Aironet_b7:ab:5c	Cisco_f0:2a:93	84:78:ac:f0:2a:93	802.11	2437	Reassociation Request, SN=2715, FN=0, Flags
4	0.010575	Cisco_f0:2a:93	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	Reassociation Response, SN=2724, FN=0, Flag
5	0.843240	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:93	802.11	2437	QoS Data, SN=2717, FN=0, Flags=.p.....TC
6	0.849798	Cisco_f5:4a:40	Aironet_b7:ab:5c	84:78:ac:f0:2a:93	802.11	2437	QoS Data, SN=66, FN=0, Flags=.p....F.C

Nachfolgend finden Sie eine Zusammenfassung der WLC-Fehlerbehebungen für dieses Roaming-Ereignis:

```
*apfMsConnTask_2: Jun 25 15:43:33.749: 00:40:96:b7:ab:5c
  CCKM: Received REASSOC REQ IE
*apfMsConnTask_2: Jun 25 15:43:33.749: 00:40:96:b7:ab:5c
  Reassociation received from mobile on BSSID
  84:78:ac:f0:2a:93
*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  Processing WPA IE type 221, length 22 for mobile
  00:40:96:b7:ab:5c
*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  CCKM: Mobile is using CCKM
!--- The Reassociation Request is received from the client,
     which provides the CCKM information needed in order to
     derive the new keys with a fast-secure roam.

*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  Setting active key cache index 0 ---> 8

*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  CCKM: Processing REASSOC REQ IE

*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  CCKM: using HMAC MD5 to compute MIC
!--- WLC computes the MIC used for this CCKM fast-roaming
     exchange.

*apfMsConnTask_2: Jun 25 15:43:33.750: 00:40:96:b7:ab:5c
  CCKM: Received a valid REASSOC REQ IE

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  CCKM: Initializing PMK cache entry with a new PTK
!--- The new PTK is derived.

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
```

```
  Setting active key cache index 8 ---> 0

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Creating a PKC PMKID Cache entry for station
  00:40:96:b7:ab:5c (RSN 0) on BSSID 84:78:ac:f0:2a:93
!--- The new PMKID cache entry is created for this new
     AP-to-client association.

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  CCKM: using HMAC MD5 to compute MIC
*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Including CCKM Response IE (length 62) in Assoc Resp to mobile
*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:93
  (status 0) ApVapId 4 Slot 0
!--- The Reassociation Response is sent from the WLC/AP to
     the client, which includes the CCKM information required
     in order to confirm the new fast-roam and key derivation.

*dot1xMsgTask: Jun 25 15:43:33.757: 00:40:96:b7:ab:5c
  Skipping EAP-Success to mobile 00:40:96:b7:ab:5c
!--- EAP is skipped due to the fast roaming, and CCKM does not
     require further key handshakes. The client is now ready to
     pass encrypted data frames on the new AP.
```

Wie gezeigt, wird schnelles sicheres Roaming durchgeführt, während die EAP-Authentifizierungs-Frames vermieden werden und noch mehr 4-Wege-Handshakes, da die neuen Verschlüsselungsschlüssel immer noch abgeleitet werden, aber auf dem CCKM-Verhandlungsschema basieren. Dies wird durch die Roaming-Reassoziations-Frames und die zuvor vom Client und vom WLC zwischengespeicherten Informationen ergänzt.

FlexConnect mit CCKM

- Die zentrale Authentifizierung wird unterstützt. Dies umfasst das lokale und zentrale Daten-Switching. Die APs müssen Teil derselben FlexConnect-Gruppe sein.
- Flex Local Authentication wird unterstützt. Im verbundenen Modus kann der Cache vom Access Point zum Controller und dann zu den übrigen Access Points der FlexConnect-Gruppe verteilt werden.
- Der Standalone-Modus wird unterstützt. Wenn der Cache (aufgrund vorheriger Verteilung) bereits auf dem Access Point vorhanden ist, wird schnelles Roaming funktionieren. Die neue Authentifizierung im Standalone-Modus unterstützt kein schnelles sicheres Roaming.

Vorteile mit CCKM

- CCKM ist die schnellste und sicherste Roaming-Methode, die hauptsächlich in Unternehmens-WLANs eingesetzt wird. Clients müssen bei einem Wechsel zwischen den APs keinen Schlüsselmanagement-Handshake überspringen, um neue Schlüssel abzuleiten. Sie müssen während der Client-Lebensdauer in diesem WLAN nie wieder eine vollständige 802.1X/EAP-Authentifizierung mit neuen APs durchführen.
- CCKM unterstützt alle im 802.11-Standard verfügbaren Verschlüsselungsmethoden (WEP, TKIP und AES), zusätzlich zu einigen proprietären, von Cisco noch auf Legacy-Clients verwendeten Methoden.

Nachteile von CCKM

- CCKM ist eine proprietäre Methode von Cisco, die die Implementierung und den Support auf die Cisco WLAN-Infrastruktur und CCX Wireless Clients beschränkt.
- CCX-Version 5 wird nur in geringem Umfang eingesetzt, daher wird CCKM mit WPA2/AES von vielen CCX-Wireless-Clients nicht unterstützt (vor allem, weil die meisten von ihnen bereits CCKM mit WPA/TKIP unterstützen, was immer noch sehr sicher ist).

Schnelles und sicheres Roaming mit PMKID-Caching/Sticky Key-Caching

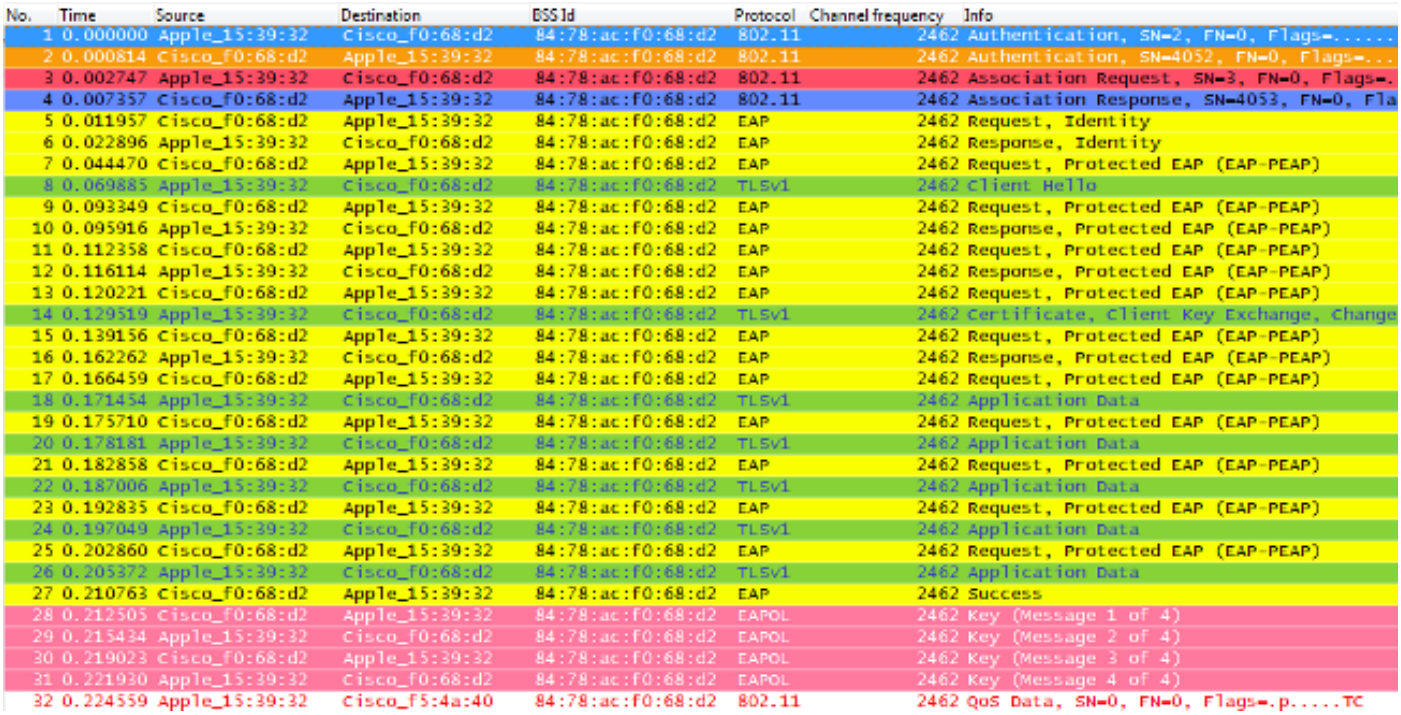
Pairwise betrachtet das PMKID-Caching (Key ID Caching) oder **Sticky Key Caching (SKC)** als die erste vom IEEE 802.11-Standard vorgeschlagene schnellsichere Roaming-Methode im Rahmen der 802.11i-Sicherheitsänderung, deren Hauptzweck die Standardisierung eines hohen Sicherheitsniveaus für WLANs ist. Diese Technik für schnelles und sicheres Roaming wurde als optionale Methode für WPA2-Geräte hinzugefügt, um das Roaming zu verbessern, als diese Sicherheit implementiert wurde.

Dies ist möglich, da Client und Authentifizierungsserver jedes Mal, wenn ein Client vollständig EAP-authentifiziert ist, ein MSK ableiten, das zur Ableitung des PMK verwendet wird. Dieser wird als Seed für den 4-Wege-WPA2-Handshake verwendet, um den endgültigen Unicast-Verschlüsselungsschlüssel (PTK) abzuleiten, der für die Sitzung verwendet wird (bis der Client zu einem anderen WAP wechselt oder die Sitzung abläuft). Daher verhindert diese Methode die EAP-Authentifizierungsphase beim Roaming, da sie den ursprünglichen PMK, der vom Client und dem WAP zwischengespeichert wurde, wiederverwendet. Der Client muss nur den 4-Wege-Handshake von WPA2 durchlaufen, um neue Verschlüsselungsschlüssel abzuleiten.

Diese Methode wird als die empfohlene schnelle sichere 802.11-Roaming-Standardmethode aus den folgenden Gründen nur selten eingesetzt:

- Diese Methode ist optional und wird nicht von allen WPA2-Geräten unterstützt, da der Zweck der 802.11i-Änderung nicht das schnelle sichere Roaming betrifft. Die IEEE hat bereits an einer anderen Änderung gearbeitet, um das schnelle sichere Roaming für WLANs zu standardisieren (802.11r, auf die später in diesem Dokument eingegangen wird).
- Diese Methode hat eine große Einschränkung bei ihrer Implementierung: Wireless-Clients können nur dann ein schnelles Roaming durchführen, wenn sie zu einem Access Point zurückkehren, bei dem sie sich zuvor authentifiziert/verbunden hatten.

Bei dieser Methode ist die anfängliche Zuordnung zu einem WAP wie eine reguläre Erstauthentifizierung mit dem WLAN, bei der die gesamte 802.1X/EAP-Authentifizierung gegenüber dem Authentifizierungsserver und der 4-Wege-Handshake zur Schlüsselgenerierung erfolgen muss, bevor der Client Datenframes senden kann, wie in diesem Screenimage gezeigt:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=2, FN=0, Flags=.....
2	0.000814	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=4052, FN=0, Flags=...
3	0.002747	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Association Request, SN=3, FN=0, Flags=.
4	0.007357	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Association Response, SN=4053, FN=0, Fla
5	0.011957	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Identity
6	0.022896	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Identity
7	0.044470	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
8	0.069885	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Client Hello
9	0.093349	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
10	0.095916	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
11	0.112358	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.116114	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.120221	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.129519	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Certificate, Client Key Exchange, Change
15	0.139156	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	0.162262	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
17	0.166459	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	0.171454	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data
19	0.175710	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
20	0.178181	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data
21	0.182858	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
22	0.187006	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data
23	0.192835	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
24	0.197049	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data
25	0.202860	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
26	0.205372	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data
27	0.210763	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAP	2462	Success
28	0.212505	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 1 of 4)
29	0.215434	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 2 of 4)
30	0.219023	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 3 of 4)
31	0.221930	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 4 of 4)
32	0.224559	Apple_15:39:32	Cisco_f5:4a:40	84:78:ac:f0:68:d2	802.11	2462	QoS Data, SN=0, FN=0, Flags=.p.....TC

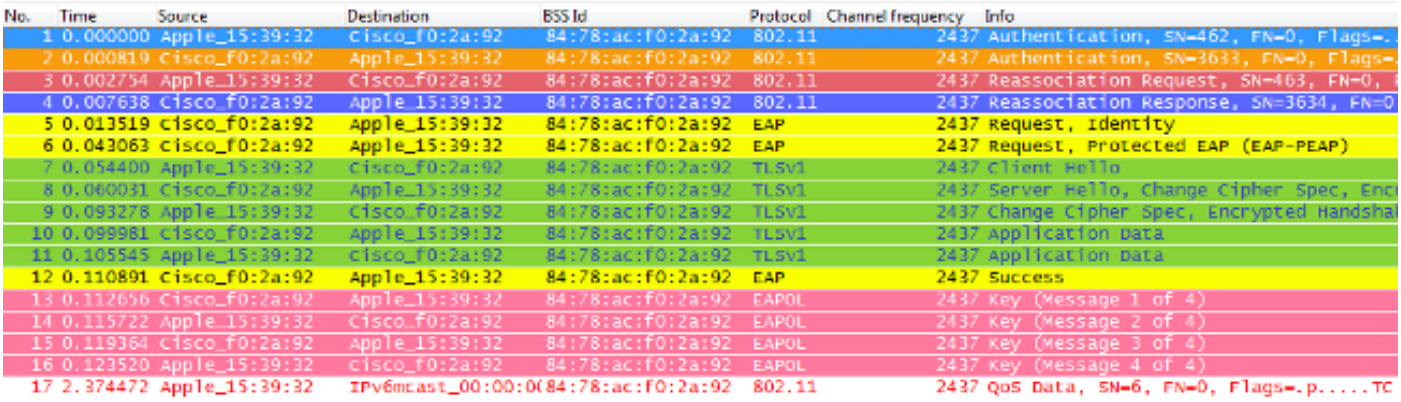
Die Debugs zeigen den gleichen EAP-Authentifizierungs-Frame-Austausch wie die anderen Methoden bei der Erstauthentifizierung im WLAN, wobei einige Ausgaben in Bezug auf die hier verwendeten Schlüssel-Caching-Techniken hinzugefügt wurden. Diese Debug-Ausgaben werden ausgeschnitten, um hauptsächlich die neuen Informationen anzuzeigen, nicht den gesamten EAP-Frame-Austausch, da im Grunde jedes Mal die gleichen Informationen zur Authentifizierung des Clients gegenüber dem Authentifizierungsserver ausgetauscht werden. Dies wird bisher gezeigt und korreliert mit den in den Paketbildern gezeigten EAP-Authentifizierungsframes, sodass die meisten EAP-Nachrichten aus Gründen der Einfachheit von den Debug-Ausgängen entfernt werden:

```
*apfMsConnTask_0: Jun 22 00:23:15.097: ec:85:2f:15:39:32
  Association received from mobile on BSSID 84:78:ac:f0:68:d2
!--- This is the Association Request from the client.

*apfMsConnTask_0: Jun 22 00:23:15.098: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 20 for mobile ec:85:2f:15:39:32
!--- The WLC/AP finds an Information Element that claims PMKID
     Caching support on the Association request that is sent
     from the client.

*apfMsConnTask_0: Jun 22 00:23:15.098: ec:85:2f:15:39:32
  Received RSN IE with 0 PMKIDs from mobile ec:85:2f:15:39:32
!--- Since this is an initial association, the Association
     Request comes without any PMKID.

*apfMsConnTask_0: Jun 22 00:23:15.098: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 8

*apfMsConnTask_0: Jun 22 00:23:15.099: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d2
  (status 0) ApVapId 3 Slot 0
!--- The Association Response is sent to the client.

*dot1xMsgTask: Jun 22 00:23:15.103: ec:85:2f:15:39:32
  Sending EAP-Request/Identity to mobile ec:85:2f:15:39:32
```

```
  (EAP Id 1)

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.118: ec:85:2f:15:39:32
  Received EAPOL EAPPKT from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.118: ec:85:2f:15:39:32
  Received Identity Response (count=1) from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.126: ec:85:2f:15:39:32
  Processing Access-Challenge for mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.126: ec:85:2f:15:39:32
  Sending EAP Request from AAA to mobile ec:85:2f:15:39:32
  (EAP Id 2)

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.146: ec:85:2f:15:39:32
  Received EAPOL EAPPKT from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.146: ec:85:2f:15:39:32
  Received EAP Response from mobile ec:85:2f:15:39:32
  (EAP Id 2, EAP Type 25)

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Processing Access-Accept for mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Creating a PKC PMKID Cache entry for station ec:85:2f:15:39:32
  (RSN 2)
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 8
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 0
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Adding BSSID 84:78:ac:f0:68:d2 to PMKID cache at index 0
  for station ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274:
  New PMKID: (16)
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274:
  [0000] c9 4d 0d 97 03 aa a9 0f 1b c8 33 73 01 f1 18 f5
!--- WLC creates a PMK cache entry for this client, which is
     used for SKC in this case, so the PMKID is computed with
     the AP MAC address (BSSID 84:78:ac:f0:68:d2).

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.274: ec:85:2f:15:39:32
  Sending EAP-Success to mobile ec:85:2f:15:39:32
  (EAP Id 12)

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.275:
  Including PMKID in M1  (16)
!--- The hashed PMKID is included on the Message-1 of the
     WPA/WPA2 4-Way handshake.

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.275:
  [0000] c9 4d 0d 97 03 aa a9 0f 1b c8 33 73 01 f1 18 f5
!--- This is the hashed PMKID.

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.275: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00
!--- Message-1 of the WPA/WPA2 4-Way handshake is sent from
     the WLC/AP to the client.

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
```

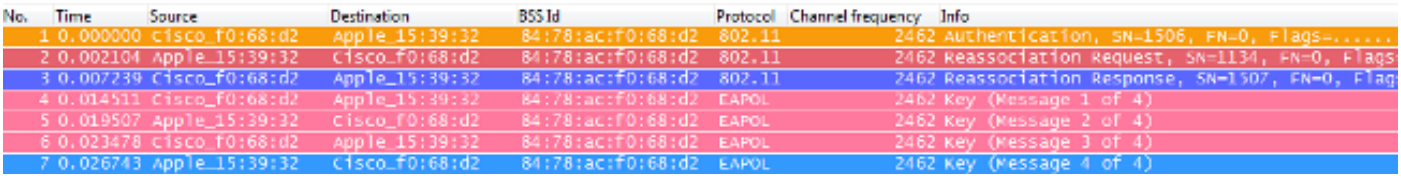
```
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from mobile
  ec:85:2f:15:39:32
!--- Message-2 of the WPA/WPA-2 4-Way handshake is successfully
     received from the client.

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  PMK: Sending cache add

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.285: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
!--- Message-3 of the WPA/WPA2 4-Way handshake is sent from
     the WLC/AP to the client.

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.291: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.291: ec:85:2f:15:39:32
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile ec:85:2f:15:39:32
!--- Message-4 (final message) of this initial WPA/WPA2 4-Way
     handshake is successfully received from the client, which
     confirms the installation of the derived keys. They can
     now be used in order to encrypt data frames with the current AP.
```

Bei diesem Verfahren zwischenspeichern der WAP und der WLAN-Client die PMKs der bereits aufgebauten sicheren Verknüpfungen. Wenn der Wireless-Client also zu einem neuen Access Point wechselt, mit dem er noch nie verbunden war, muss der Client erneut eine vollständige EAP-Authentifizierung durchführen, wie in diesem Bild dargestellt. Dabei wechselt der Client zu einem neuen Access Point:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	802.11	2437	Authentication, SN=462, FN=0, Flags=.
2	0.000819	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	802.11	2437	Authentication, SN=3633, FN=0, Flags=
3	0.002754	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	802.11	2437	Reassociation Request, SN=463, FN=0,
4	0.007638	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	802.11	2437	Reassociation Response, SN=3634, FN=0
5	0.013519	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	EAP	2437	Request, Identity
6	0.043063	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	EAP	2437	Request, Protected EAP (EAP-PEAP)
7	0.054400	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	TLSv1	2437	Client Hello
8	0.060031	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	TLSv1	2437	Server Hello, Change Cipher Spec, Enc
9	0.093278	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	TLSv1	2437	Change Cipher Spec, Encrypted Handsha
10	0.099981	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	TLSv1	2437	Application Data
11	0.105545	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	TLSv1	2437	Application Data
12	0.110891	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	EAP	2437	Success
13	0.112656	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 1 of 4)
14	0.115722	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 2 of 4)
15	0.119364	Cisco_f0:2a:92	Apple_15:39:32	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 3 of 4)
16	0.123520	Apple_15:39:32	Cisco_f0:2a:92	84:78:ac:f0:2a:92	EAPOL	2437	Key (Message 4 of 4)
17	2.374472	Apple_15:39:32	IPv6mcast_00:00:0(	84:78:ac:f0:2a:92	802.11	2437	QoS Data, SN=6, FN=0, Flags=.p.....TC

Wenn der Wireless-Client jedoch zu einem Access Point zurückkehrt, an dem zuvor eine Zuordnung bzw. Authentifizierung stattgefunden hat, sendet der Client einen Frame für eine Neuzuordnungsanforderung, in dem mehrere PMKIDs aufgelistet sind. Dieser informiert den Access Point über die PMKs, die von allen Access Points, an denen der Client zuvor authentifiziert wurde, zwischengespeichert wurden. Da der Client also zu einem WAP zurückkehrt, der auch über einen PMK verfügt, der für diesen Client zwischengespeichert ist, muss der Client sich nicht erneut über EAP authentifizieren, um einen neuen PMK abzuleiten. Der Client leitet die neuen transienten Verschlüsselungsschlüssel einfach über den 4-Wege-WPA2-Handshake ab:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=1506, FN=0, Flags=.....
2	0.002104	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Reassociation Request, SN=1134, FN=0, Flags
3	0.007239	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Reassociation Response, SN=1507, FN=0, Flag
4	0.014511	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 1 of 4)
5	0.019507	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 2 of 4)
6	0.023478	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 3 of 4)
7	0.026743	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 4 of 4)

Hinweis: Dieses Bild zeigt nicht den ersten 802.11 Open System Authentifizierungsframe vom Client, aber dies liegt nicht an der implementierten Methode, da dieser Frame immer erforderlich ist. Der Grund dafür ist, dass dieser spezielle Frame nicht von dem Adapter oder der Wireless-Paket-Image-Software abgebildet wird, die verwendet wird, um die Over-the-Air-Frames für dieses Beispiel zu erschnüffeln, sondern dass er für Unterrichtszwecke so auf dem Beispiel belassen wird. Beachten Sie, dass dies bei der Übertragung von Paket-Images möglich ist. Manche Frames können vom Image verpasst werden, werden jedoch zwischen dem Client und dem Access Point ausgetauscht. Andernfalls wird das Roaming in diesem Beispiel nie gestartet.

Im Folgenden finden Sie eine Zusammenfassung der WLC-Fehlerbehebungen für diese hochsichere Roaming-Methode:

```
*apfMsConnTask_0: Jun 22 00:26:40.787: ec:85:2f:15:39:32
  Reassociation received from mobile on BSSID
  84:78:ac:f0:68:d2
!--- This is the Reassociation Request from the client.

*apfMsConnTask_0: Jun 22 00:26:40.787: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 38 for mobile
  ec:85:2f:15:39:32
!--- The WLC/AP finds an Information Element that claims PMKID
     Caching support on the Association request that is sent
     from the client.

*apfMsConnTask_0: Jun 22 00:26:40.787: ec:85:2f:15:39:32
  Received RSN IE with 1 PMKIDs from mobile
  ec:85:2f:15:39:32
!--- The Reassociation Request from the client comes with
     one PMKID.

*apfMsConnTask_0: Jun 22 00:26:40.787:
  Received PMKID:  (16)
*apfMsConnTask_0: Jun 22 00:26:40.788:
  [0000] c9 4d 0d 97 03 aa a9 0f 1b c8 33 73 01 f1 18 f5
!--- This is the PMKID that is received.

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Searching for PMKID in MSCB PMKID cache for mobile
  ec:85:2f:15:39:32
!--- WLC searches for a matching PMKID on the database.

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Found an cache entry for BSSID 84:78:ac:f0:68:d2 in
  PMKID cache at index 0 of station ec:85:2f:15:39:32

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Found a valid PMKID in the MSCB PMKID cache for mobile
  ec:85:2f:15:39:32
!--- The WLC validates the PMKID provided by the client,
     and confirms that it has a valid PMK cache for this
     client-and-AP pair.

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
```

```
  Setting active key cache index 1 ---> 0

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:68:d2(status 0) ApVapId 3 Slot 0
!--- The Reassociation Response is sent to the client, which
     validates the fast-roam with SKC.

*dot1xMsgTask: Jun 22 00:26:40.795: ec:85:2f:15:39:32
  Initiating RSN with existing PMK to mobile
  ec:85:2f:15:39:32
!--- WLC initiates a Robust Secure Network association with
     this client-and-AP pair based on the cached PMK found.
Hence, EAP is avoided as per the next message.

*dot1xMsgTask: Jun 22 00:26:40.795: ec:85:2f:15:39:32
  Skipping EAP-Success to mobile ec:85:2f:15:39:32

*dot1xMsgTask: Jun 22 00:26:40.795: ec:85:2f:15:39:32
  Found an cache entry for BSSID 84:78:ac:f0:68:d2 in
  PMKID cache at index 0 of station ec:85:2f:15:39:32

*dot1xMsgTask: Jun 22 00:26:40.795: Including PMKID in M1(16)
!--- The hashed PMKID is included on the Message-1 of the
     WPA/WPA2 4-Way handshake.

*dot1xMsgTask: Jun 22 00:26:40.795:
  [0000] c9 4d 0d 97 03 aa a9 0f 1b c8 33 73 01 f1 18 f5
!--- The PMKID is hashed. The next messages are the same
     WPA/WPA2 4-Way handshake messages described thus far
     that are used in order to finish the encryption keys
     generation/installation.

*dot1xMsgTask: Jun 22 00:26:40.795: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  INITPMK (message 1), replay counter 00.00.00.00.00.00.00.00

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.811: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.812: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from mobile
  ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.812: ec:85:2f:15:39:32
  PMK: Sending cache add

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.812: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.820: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 22 00:26:40.820: ec:85:2f:15:39:32
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile ec:85:2f:15:39:32
```

FlexConnect mit PMKID-Caching/Sticky Key-Caching

- Wenn Sie diese Methode in einer FlexConnect-Konfiguration verwenden, könnte sie funktionieren, und das Verhalten kann ähnlich wie zuvor beschrieben aussehen, wenn Sie die

zentrale Authentifizierung für den WLC verwenden (entweder mit zentralem oder lokalem Switching). Diese SKC-Methode wird jedoch auf FlexConnect nicht unterstützt.
- Diese Methode wird nur offiziell auf CUWN mit APs im lokalen Modus unterstützt, nicht auf FlexConnect oder anderen Modi.

Profis mit PMKID Caching / Sticky Key Caching

Diese Methode kann lokal durch autonome unabhängige Zugangspunkte implementiert werden, ohne dass ein zentralisiertes Gerät zur Verwaltung der zwischengespeicherten Schlüssel erforderlich ist.

Nachteile mit PMKID-Zwischenspeicherung/Zwischenspeicherung von Kurztasten

- Wie bereits in diesem Dokument erwähnt, besteht die wichtigste Einschränkung dieses Verfahrens darin, dass der Client beim Roaming zu einem Access Point, dem er zuvor zugeordnet/authentifiziert wurde, nur ein schnelles Roaming durchführen kann. Wenn das Roaming zu einem neuen Access Point erfolgt, muss der Client die vollständige EAP-Authentifizierung erneut durchführen.
- Der Wireless-Client und die APs müssen sich alle PMKs merken, die bei jeder neuen Authentifizierung abgeleitet werden. Daher ist diese Funktion normalerweise auf eine bestimmte Anzahl von PMKs beschränkt, die zwischengespeichert werden. Da dieser Grenzwert im Standard nicht klar definiert ist, können die Anbieter verschiedene Grenzwerte für ihre SKC-Implementierungen definieren. Beispielsweise können die Cisco WLAN Controller derzeit die PMKs eines Clients für bis zu acht APs zwischenspeichern. Wenn ein Client zu mehr als acht APs pro Sitzung wechselt, werden die ältesten APs aus der Cache-Liste entfernt, um die neu zwischengespeicherten Einträge zu speichern.
- Diese Methode ist optional und wird von vielen WPA2-Geräten immer noch nicht unterstützt. Daher wird sie nur selten verwendet und bereitgestellt.
- SKC wird nicht unterstützt, wenn Sie das Roaming zwischen Controllern durchführen, d. h. wenn Sie zwischen APs wechseln, die von verschiedenen WLCs verwaltet werden, selbst wenn diese derselben Mobilitätsgruppe angehören.

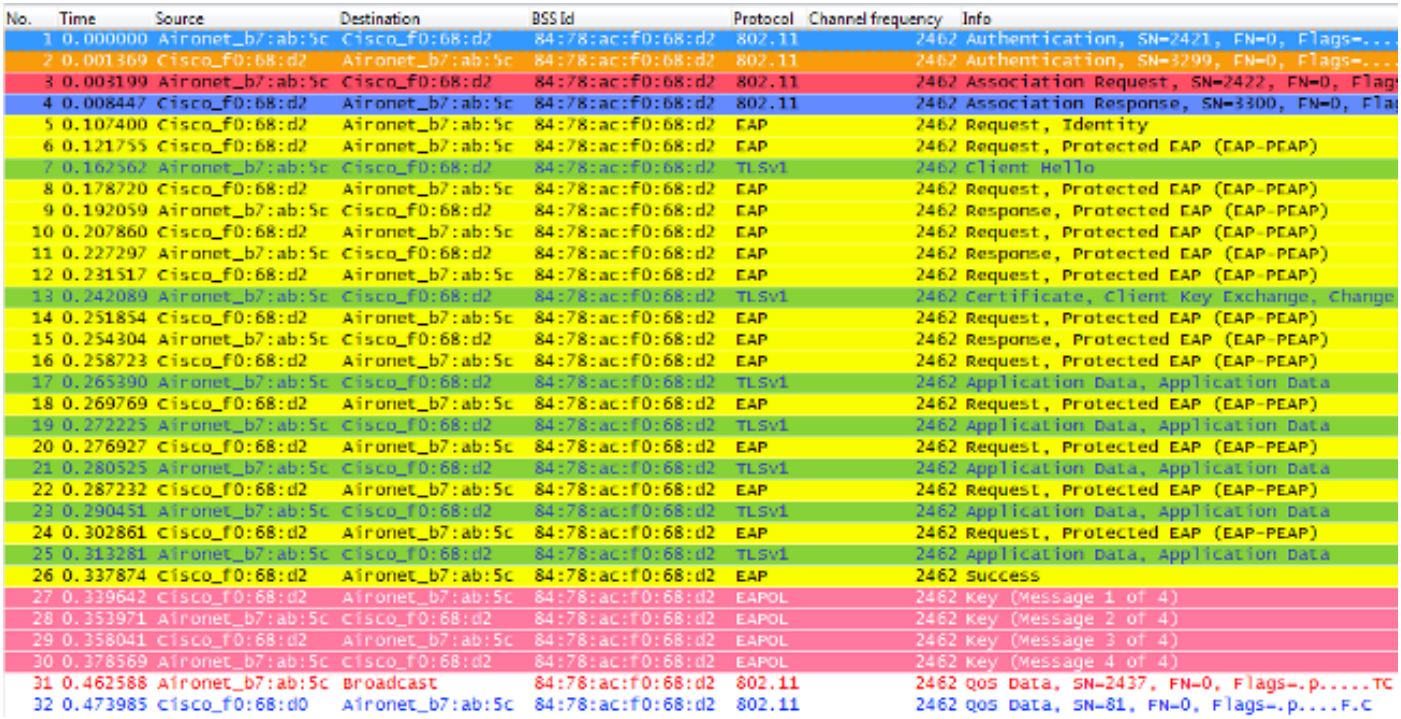
Schnelles und sicheres Roaming mit opportunistischem Schlüssel-Caching

Opportunistic Key Caching (OKC), auch bekannt als Proactive Key Caching (PKC) (dieser Begriff wird in einem nächsten Hinweis näher erläutert), ist grundsätzlich eine Erweiterung der zuvor beschriebenen WPA2 PMKID Caching-Methode, weshalb sie auch als Proactive/Opportunistic PMKID Caching bezeichnet wird. Es ist daher wichtig zu beachten, dass es sich hierbei nicht um eine durch den 802.11-Standard definierte schnelle Roaming-Methode handelt, die von vielen Geräten nicht unterstützt wird. Wie beim PMKID-Caching funktioniert sie jedoch mit WPA2-EAP.

Diese Technik ermöglicht es dem WLAN-Client und der WLAN-Infrastruktur, nur einen PMK für die Lebensdauer der Client-Verbindung mit diesem WLAN (abgeleitet vom MSK nach der anfänglichen 802.1X/EAP-Authentifizierung mit dem Authentifizierungsserver) zwischenzuspeichern, selbst wenn das Roaming zwischen mehreren APs erfolgt, da alle den ursprünglichen PMK teilen, der als Seed für alle 4-Wege-WPA2-Handshakes verwendet wird. Dies

ist weiterhin erforderlich, genau wie in SKC, um bei jeder Neuzuordnung des Clients zu den APs neue Verschlüsselungsschlüssel zu generieren. Damit die APs diesen ursprünglichen PMK aus der Client-Sitzung gemeinsam nutzen können, müssen sie alle unter einer administrativen Kontrolle stehen und über ein zentrales Gerät verfügen, das den ursprünglichen PMK zwischenspeichert und für alle APs verteilt. Dies ähnelt dem CUWN, bei dem der WLC diese Aufgabe für alle ihm unterstehenden LAPs ausführt und die Mobilitätsgruppen verwendet, um diese PMK zwischen mehreren WLCs zu verwalten. Dies stellt daher eine Beschränkung für autonome AP-Umgebungen dar.

Wie beim PMKID-Caching (SKC) ist bei dieser Methode die Erstzuordnung zu einem AP eine reguläre Erstauthentifizierung zum WLAN, bei der Sie die gesamte 802.1X/EAP-Authentifizierung gegen den Authentifizierungsserver und den 4-Wege-Handshake zur Schlüsselgenerierung abschließen müssen, bevor Sie Datenframes senden können. Das folgende Screenbild veranschaulicht dies:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=2421, FN=0, Flags=...
2	0.001369	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=3299, FN=0, Flags=...
3	0.003199	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Association Request, SN=2422, FN=0, Flags
4	0.008447	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	802.11	2462	Association Response, SN=3300, FN=0, Flags
5	0.107400	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Identity
6	0.121755	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
7	0.162562	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Client Hello
8	0.178720	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
9	0.192059	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
10	0.207860	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
11	0.227297	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
12	0.231517	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
13	0.242089	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Certificate, Client Key Exchange, Change
14	0.251854	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
15	0.254304	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAP	2462	Response, Protected EAP (EAP-PEAP)
16	0.258723	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
17	0.265390	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data, Application Data
18	0.269769	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
19	0.272225	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data, Application Data
20	0.276927	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
21	0.280525	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data, Application Data
22	0.287232	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	0.290451	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data, Application Data
24	0.302861	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	0.313281	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	TLSv1	2462	Application Data, Application Data
26	0.337874	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAP	2462	Success
27	0.339642	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 1 of 4)
28	0.353971	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 2 of 4)
29	0.358041	Cisco_f0:68:d2	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 3 of 4)
30	0.378569	Aironet_b7:ab:5c	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 4 of 4)
31	0.462588	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:68:d2	802.11	2462	QoS Data, SN=2437, FN=0, Flags=.p.....TC
32	0.473985	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d2	802.11	2462	QoS Data, SN=81, FN=0, Flags=.p....F.C

Die Debug-Ausgaben zeigen im Wesentlichen den gleichen EAP-Authentifizierungs-Frame-Austausch wie die anderen in diesem Dokument beschriebenen Methoden bei der Erstauthentifizierung im WLAN (wie in den Bildern gezeigt), zusammen mit der Hinzufügung einiger Ausgaben, die die Schlüssel-Caching-Techniken betreffen, die vom WLC hier verwendet werden. Diese Debug-Ausgabe wird ebenfalls ausgeschnitten, um nur die relevanten Informationen anzuzeigen:

```
*apfMsConnTask_0: Jun 21 21:46:06.515: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID
  84:78:ac:f0:68:d2
!--- This is the Association Request from the client.

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Processing RSN IE type 48, length 20 for mobile
  00:40:96:b7:ab:5c
!--- The WLC/AP finds an Information Element that claims
     PMKID Caching support on the Association request that
     is sent from the client.

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
```

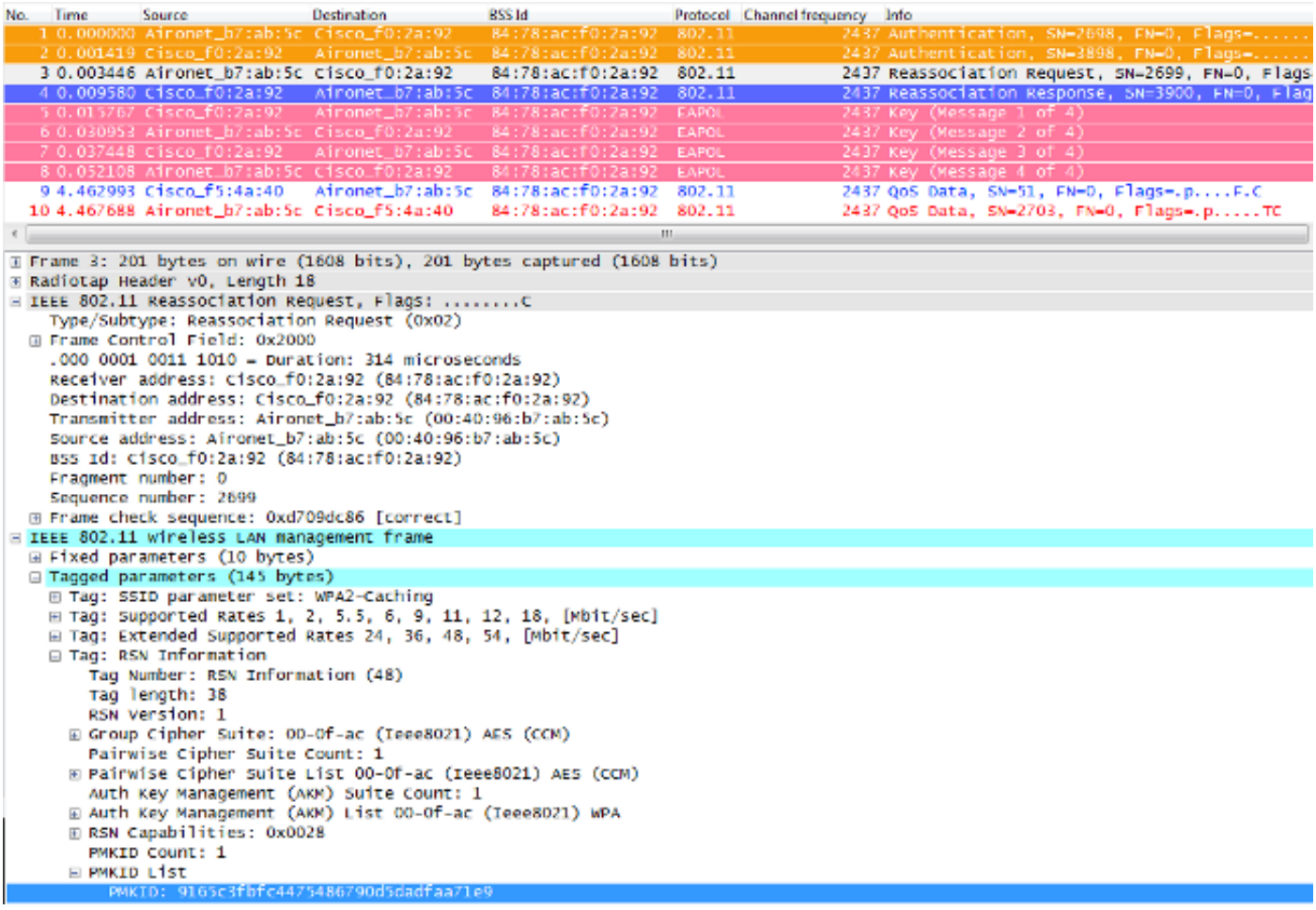
```
  Received RSN IE with 0 PMKIDs from mobile
  00:40:96:b7:ab:5c
!--- Since this is an initial association, the Association
     Request comes without any PMKID.

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Setting active key cache index 0 ---> 8

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:68:d2 (status 0) ApVapId 3 Slot
!--- The Association Response is sent to the client.

*dot1xMsgTask: Jun 21 21:46:06.522: 00:40:96:b7:ab:5
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 1)

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.614: 00:40:96:b7:ab:5c
  Received EAPOL START from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.614: 00:40:96:b7:ab:5c
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 2)

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.623: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.623: 00:40:96:b7:ab:5c
  Received Identity Response (count=2) from mobile
  00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.630: 00:40:96:b7:ab:5c
  Processing Access-Challenge for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.630: 00:40:96:b7:ab:5c
  Sending EAP Request from AAA to mobile 00:40:96:b7:ab:5c
  (EAP Id 3)

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.673: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.673: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 3, EAP Type 25)

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.843: 00:40:96:b7:ab:5c
  Processing Access-Accept for mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Creating a PKC PMKID Cache entry for station
  00:40:96:b7:ab:5c (RSN 2)
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 0
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Adding BSSID 84:78:ac:f0:68:d2 to PMKID cache at index 0
  for station 00:40:96:b7:ab:5
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: New PMKID: (16)
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844:
  [0000] 4e a1 7f 5a 75 48 9c f9 96 e3 a8 71 25 6f 11 d0
!--- WLC creates a PMK cache entry for this client, which is
     used for OKC in this case, so the PMKID is computed
     with the AP MAC address (BSSID 84:78:ac:f0:68:d2).
```

```
*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  PMK sent to mobility group
!--- The PMK cache entry for this client is shared with the
     WLCs on the mobility group.

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Sending EAP-Success to mobile 00:40:96:b7:ab:5c (EAP Id 13)

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Found an cache entry for BSSID 84:78:ac:f0:68:d2 in PMKID
  cache at index 0 of station 00:40:96:b7:ab:5

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: Including PMKID
  in M1  (16)
!--- The hashed PMKID is included on the Message-1 of the
     WPA/WPA2 4-Way handshake.

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844:
  [0000] 4e a1 7f 5a 75 48 9c f9 96 e3 a8 71 25 6f 11 d0
!--- This is the hashed PMKID. The next messages are the same
     WPA/WPA2 4-Way handshake messages described thus far that
     are used in order to finish the encryption keys
     generation/installation.

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c state
  INITPMK (message 1), replay counter 00.00.00.00.00.00.00.00

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.865: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.865: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2)
  from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.865: 00:40:96:b7:ab:5c
  PMK: Sending cache add

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.865: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.889: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.890: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile 00:40:96:b7:ab:5c
```

Bei dieser Methode zwischenspeichern der Wireless-Client und der WLC (für alle verwalteten APs) den ursprünglichen PMK der ursprünglich eingerichteten sicheren Verbindung. Grundsätzlich wird bei jeder Verbindung des Wireless-Clients mit einem bestimmten WAP eine PMKID gehasht, die auf der Client-MAC-Adresse, der WAP-MAC-Adresse (BSSID des WLAN) und der mit diesem WAP abgeleiteten PMK basiert. Da OKC daher für alle Zugangspunkte und den jeweiligen Client denselben ursprünglichen PMK zwischenspeichert, ist bei einer (erneuten) Zuordnung dieses Clients zu einem anderen Zugangspunkt der einzige Wert, der sich ändert, um die neue PMKID zu hash, die neue AP-MAC-Adresse.

Wenn der Client das Roaming zu einem neuen WAP initiiert und den Frame der Neuzuordnungsanforderung sendet, fügt er die PMKID zum WPA2 RSN-Informationselement

hinzu, wenn er den WAP darüber informieren möchte, dass ein zwischengespeicherter PMK für das schnelle sichere Roaming verwendet wird. Er kennt bereits die MAC-Adresse des BSSID (AP), wohin er roamt, dann hasht der Client einfach die neue PMKID, die für diese Neuzuordnungsanforderung verwendet wird. Wenn der WAP diese Anforderung vom Client empfängt, hasht er auch die PMKID mit den Werten, über die er bereits verfügt (die zwischengespeicherte PMK, die Client-MAC-Adresse und seine eigene WAP-MAC-Adresse), und antwortet mit der erfolgreichen Antwort zur erneuten Zuordnung, die bestätigt, dass die PMKIDs zugeordnet wurden. Der zwischengespeicherte PMK kann als Seed verwendet werden, der einen 4-Wege-WPA2-Handshake startet, um die neuen Verschlüsselungsschlüssel abzuleiten (und EAP zu überspringen):



In diesem Bild wird der Frame für die Neuzuordnungsanforderung vom Client ausgewählt und erweitert, sodass Sie weitere Details des Frames sehen können. Die MAC-Adressinformationen sowie das Robust Security Network (RSN, gemäß 802.11i - WPA2) Information Element, in dem Informationen über die WPA2-Einstellungen angezeigt werden, die für diese Zuordnung verwendet werden (hervorgehoben ist die PMKID, die aus der Hashformel erhalten wird).

Nachfolgend finden Sie eine Zusammenfassung der WLC-Debugs für diese hochsichere Roaming-Methode mit OKC:

```
*apfMsConnTask_2: Jun 21 21:48:50.562: 00:40:96:b7:ab:5c
  Reassociation received from mobile on BSSID
  84:78:ac:f0:2a:92
!--- This is the Reassociation Request from the client.

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Processing RSN IE type 48, length 38 for mobile
```

```
  00:40:96:b7:ab:5c
!--- The WLC/AP finds and Information Element that claims
     PMKID Caching support on the Association request that
     is sent from the client.

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Received RSN IE with 1 PMKIDs from mobile
  00:40:96:b7:ab:5c
!--- The Reassociation Request from the client comes with
     one PMKID.

*apfMsConnTask_2: Jun 21 21:48:50.563:
  Received PMKID:  (16)

*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Searching for PMKID in MSCB PMKID cache for mobile
  00:40:96:b7:ab:5c
*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  No valid PMKID found in the MSCB PMKID cache for mobile
  00:40:96:b7:ab:5
!--- As the client has never authenticated with this new AP,
     the WLC cannot find a valid PMKID to match the one provided
     by the client. However, since the client performs OKC
     and not SKC (as per the following messages), the WLC computes
     a new PMKID based on the information gathered (the cached PMK,
     the client MAC address, and the new AP MAC address).

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Trying to compute a PMKID from MSCB PMK cache for mobile
  00:40:96:b7:ab:5c

*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: BSSID =  (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 90
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: realAA =  (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 92
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: PMKID =  (16)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: AA (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 92
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: SPA (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 00 40 96 b7 ab 5c
*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Adding BSSID 84:78:ac:f0:2a:92 to PMKID cache at
  index 0 for station 00:40:96:b7:ab:5c
*apfMsConnTask_2: Jun 21 21:48:50.563:
  New PMKID: (16)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Computed a valid PMKID from MSCB PMK cache for mobile
  00:40:96:b7:ab:5c
```

!--- The new PMKID is computed and validated to match the one provided by the client, which is also computed with the same information. Hence, the fast-secure roam is possible.

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Setting active key cache index 0 ---> 0

*apfMsConnTask_2: Jun 21 21:48:50.564: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:92
  (status 0) ApVapId 3 Slot
!--- The Reassociation response is sent to the client, which validates the fast-roam with OKC.

*dot1xMsgTask: Jun 21 21:48:50.570: 00:40:96:b7:ab:5c
  Initiating RSN with existing PMK to mobile
  00:40:96:b7:ab:5c
!--- WLC initiates a Robust Secure Network association with this client-and AP pair with the cached PMK found. Hence, EAP is avoided, as per the the next message.

*dot1xMsgTask: Jun 21 21:48:50.570: 00:40:96:b7:ab:5c
  Skipping EAP-Success to mobile 00:40:96:b7:ab:5c

*dot1xMsgTask: Jun 21 21:48:50.570: 00:40:96:b7:ab:5c
  Found an cache entry for BSSID 84:78:ac:f0:2a:92 in
  PMKID cache at index 0 of station 00:40:96:b7:ab:5c

*dot1xMsgTask: Jun 21 21:48:50.570:
  Including PMKID in M1 (16)
!--- The hashed PMKID is included on the Message-1 of the WPA/WPA2 4-Way handshake.

*dot1xMsgTask: Jun 21 21:48:50.570:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
!--- The PMKID is hashed. The next messages are the same WPA/WPA2 4-Way handshake messages described thus far, which are used in order to finish the encryption keys generation/installation.

*dot1xMsgTask: Jun 21 21:48:50.570: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c state
  INITPMK (message 1), replay counter 00.00.00.00.00.00.00.00

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.589: 00:40:96:b7:ab:5
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.589: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTK_START state (message 2) from mobile
  00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.589: 00:40:96:b7:ab:5c
  PMK: Sending cache add

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.590: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.610: 00:40:96:b7:ab:5c
  Received EAPOL-Key from mobile 00:40:96:b7:ab:5c

*Dot1x_NW_MsgTask_4: Jun 21 21:48:50.610: 00:40:96:b7:ab:5c
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)

```
from mobile 00:40:96:b7:ab:5c
```

Wie zu Beginn der Debugs gezeigt, muss die PMKID berechnet werden, nachdem die Neuzuordnungsanforderung vom Client empfangen wurde. Dies ist erforderlich, um die PMKID zu validieren und zu bestätigen, dass der zwischengespeicherte PMK mit dem 4-Wege-WPA2-Handshake verwendet wird, um die Verschlüsselungsschlüssel abzuleiten und das schnelle sichere Roaming zu beenden. Verwechseln Sie nicht die CCKM-Einträge auf den Debugs; dies wird nicht verwendet, um CCKM auszuführen, sondern OKC, wie zuvor erläutert. CCKM ist hier einfach ein Name, der vom WLC für diese Ausgaben verwendet wird, z. B. der Name einer Funktion, die die Werte verarbeitet, um die PMKID zu berechnen.

FlexConnect mit opportunistischem Key-Caching

- Die zentrale Authentifizierung wird unterstützt. Dies umfasst das lokale und zentrale Daten-Switching. Wenn der AP zur gleichen FlexConnect-Gruppe gehört, erfolgt das schnelle sichere Roaming durch den AP, ansonsten erfolgt das schnelle sichere Roaming durch den Controller.
  Hinweis: Diese Konfiguration funktioniert, wenn die Access Points nicht derselben FlexConnect-Gruppe angehören. Dies wird jedoch nicht empfohlen oder unterstützt.
- Flex Local Authentication wird unterstützt. Im verbundenen Modus kann der Cache vom Access Point zum Controller und dann zu den übrigen Access Points der FlexConnect-Gruppe verteilt werden.
- Der Standalone-Modus wird unterstützt. Wenn der Cache (aufgrund früherer Verteilung) bereits auf dem Access Point vorhanden ist, muss das schnelle Roaming funktionieren. Die neue Authentifizierung im Standalone-Modus unterstützt kein schnelles sicheres Roaming.

Vorteile mit opportunistischem Schlüssel-Caching

- Der WLAN-Client und die WLAN-Infrastruktur müssen sich nicht mehrere PMKIDs merken, sondern lediglich die ursprüngliche PMK aus der ursprünglichen Authentifizierung in das WLAN zwischenspeichern. Anschließend müssen Sie die richtige PMKID (für die Anforderung zur erneuten Zuordnung) neu erstellen, die für jede sichere AP-Zuordnung erforderlich ist, um das schnelle Roaming zu validieren.
- Hier führt der Wireless-Client ein schnelles und sicheres Roaming zu einem neuen WAP im selben WLAN/SSID durch, selbst wenn er nie mit diesem WAP verbunden war (nicht der Fall bei SKC). Solange der Client die anfängliche 802.1X/EAP-Authentifizierung mit einem von der zentralisierten Bereitstellung verwalteten WAP durchführt, der den PMK-Cache für alle WAPs verwaltet, für die der Client Roaming durchführt, sind für die restliche Lebensdauer des Clients in diesem WLAN keine vollständigen Authentifizierungen mehr erforderlich.

Nachteile: Opportunistisches Schlüssel-Caching

- Diese Methode wird nur in einer zentralisierten Umgebung bereitgestellt, in der alle Access Points einer Art administrativen Kontrolle (z. B. einem WLAN-Controller) unterliegen, die für das Caching und die gemeinsame Nutzung des ursprünglichen PMK aus der Client-Sitzung zuständig ist. Dies stellt daher eine Beschränkung für autonome AP-Umgebungen dar.
- Die Techniken, die bei diesem Verfahren angewendet werden, werden nicht auf dem 802.11-Standard vorgeschlagen oder beschrieben, sodass die Unterstützung von Gerät zu Gerät stark variiert. Dennoch ist dies immer noch die Methode, die mehr während der Wartezeit auf

802.11r angenommen wurde.

Hinweis zum Begriff "proaktives Schlüssel-Caching"

Proaktives Schlüssel-Caching (oder PKC) wurde als OKC (Opportunistisches Schlüssel-Caching) bezeichnet. Beide Begriffe werden synonym verwendet, wenn sie dieselbe, hier erläuterte Methode beschreiben. Dies war jedoch nur ein Begriff, der 2001 von Airspace für eine alte Schlüssel-Caching-Methode verwendet wurde, die dann vom 802.11i-Standard als Grundlage für die "Vorauthentifizierung" verwendet wurde (eine weitere Fast Secure Roaming-Methode, die im Folgenden kurz erläutert wird). PKC ist nicht Preauthentication oder OKC (Opportunistic Key Caching), aber wenn Sie PKC hören oder lesen, wird im Grunde auf OKC verwiesen, und nicht auf Preauthentication.

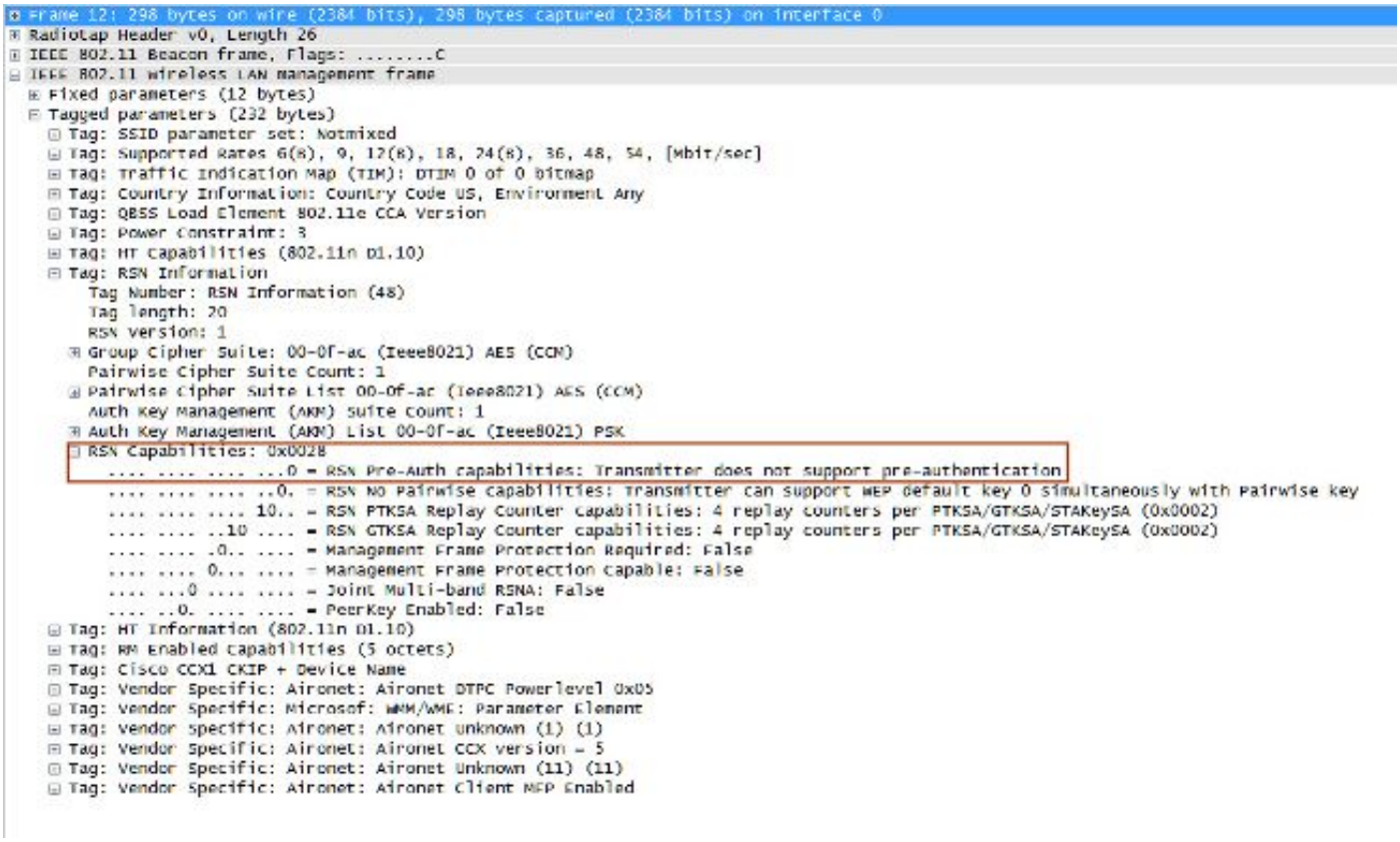
Schnelles und sicheres Roaming mit Vorauthentifizierung

Diese Methode wird auch vom IEEE 802.11-Standard im Rahmen der 802.11i-Sicherheitsänderung empfohlen. Sie funktioniert also auch mit WPA2, ist jedoch die einzige Fast Secure Roaming-Methode, die von der Cisco WLAN-Infrastruktur nicht unterstützt wird. Aus diesem Grund wird sie hier nur kurz und ohne Ausgänge erläutert.

Mit der Vorauthentifizierung können sich die Wireless-Clients mit mehreren WAPs gleichzeitig authentifizieren, während sie mit dem aktuellen WAP verbunden sind. In diesem Fall sendet der Client die EAP-Authentifizierungs-Frames an den aktuellen WAP, mit dem er verbunden ist. Er ist jedoch für die anderen WAPs bestimmt, bei denen der Client die Vorauthentifizierung durchführen möchte (benachbarte WAPs, die möglicherweise für Roaming infrage kommen). Der aktuelle WAP sendet diese Frames über das Verteilungssystem an die Ziel-WAPs. Der neue WAP führt für diesen Client eine vollständige Authentifizierung gegenüber dem RADIUS-Server durch, sodass ein kompletter neuer EAP-Authentifizierungshandshake abgeschlossen ist und dieser neue WAP als Authentifizierer fungiert.

Die Idee besteht darin, eine Authentifizierung durchzuführen und PMK mit den benachbarten APs abzurufen, bevor der Client tatsächlich zu ihnen roamt. Wenn es also an der Zeit ist, zu roamen, ist der Client bereits authentifiziert und mit einem PMK, der bereits für diese neue sichere AP-to-Client-Verbindung zwischengespeichert ist, sodass sie nur den 4-Wege-Handshake durchführen müssen und einen schnellen Roam erleben, nachdem der Client seine anfängliche Neuzuordnungsanforderung gesendet hat.

Das folgende Bild zeigt ein AP-Beacon mit dem RSN IE-Feld, das die Unterstützung für die Vorauthentifizierung ankündigt (diese ist von einem Cisco AP, bei dem bestätigt wird, dass die Vorauthentifizierung nicht unterstützt wird):



Vorteile mit Vorauthentifizierung

Für jede sichere Zuordnung zwischen APs und Clients gibt es einen PMK. Dies könnte als Sicherheitsvorteil angesehen werden, wenn ein AP kompromittiert wird und die Schlüssel gestohlen werden (kann nicht mit anderen APs verwendet werden). Dieser Sicherheitsvorteil wird jedoch von der WLAN-Infrastruktur auf verschiedene Weise und mit anderen Methoden genutzt.

Nachteile: Vorauthentifizierung

- Da es einen PMK pro AP gibt, können die Clients nur eine begrenzte Anzahl von APs vorauthentifizieren.
- Jedes Mal, wenn ein Client eine Vorauthentifizierung mit einem neuen WAP durchführt, findet ein vollständiger EAP-Authentifizierungsaustausch statt, was eine höhere Auslastung des Netzwerks und des Authentifizierungsservers bedeutet.
- Die meisten Wireless-Clients unterstützen diese Methode nicht, da sie nie sehr häufig eingesetzt wurde (OKC wurde eher eingesetzt).

Schnelles und sicheres Roaming mit 802.11r

Die auf dem 802.11r-Zusatz (offiziell **Fast BSS Transition** durch den 802.11-Standard und **FT)** basierende Fast-Secure-Roaming-Technik ist die erste offiziell (2008) vom IEEE für den 802.11-Standard als Lösung zur Durchführung schneller Übergänge zwischen APs ratifizierte Methode (Basic Service Sets oder BSSs), die die Schlüsselhierarchie klar definiert, die beim Verarbeiten und Zwischenspeichern von Schlüsseln in einem WLAN verwendet wird. Die Einführung erfolgte jedoch nur langsam, was vor allem auf die anderen Lösungen zurückzuführen ist, die bereits zur Verfügung standen, wenn tatsächlich schnelle Übergänge erforderlich waren, z. B. bei VoWLAN-Implementierungen, wenn sie mit einer der zuvor in diesem Dokument beschriebenen Methoden verwendet wurden. Derzeit unterstützen nur wenige Geräte einige der FT-Optionen (bis 2013).

Diese Methode ist komplexer zu erklären als die anderen Methoden, da sie neue Konzepte und mehrere Ebenen von PMKs einführt, die auf verschiedenen Geräten zwischengespeichert werden (jedes Gerät mit einer anderen Rolle), und noch mehr Optionen für schnelles und sicheres Roaming bietet. Daher wird eine kurze Zusammenfassung über diese Methode und die Art ihrer Implementierung mit jeder verfügbaren Option bereitgestellt.

802.11r unterscheidet sich von SKC und OKC vor allem aus den folgenden Gründen:

- Handshake-Messaging (z. B. PMKID-, ANonce- und SNonce-Austausch) erfolgt in Authentifizierungs-Frames des Standards 802.11 oder in Action-Frames anstelle von Reassociation-Frames. Im Gegensatz zu PMKID-Caching-Verfahren wird die separate 4-Wege-Handshake-Phase, die nach dem (Re)Zuordnungs-Nachrichtenaustausch durchgeführt wird, vermieden. Der Schlüsselhandshake mit dem neuen Access Point beginnt, bevor der Client das Roaming bzw. die Neuzuordnung mit diesem neuen Access Point vollständig durchführt.
- Sie bietet zwei Methoden für den schnellen Roaming-Handshake: über den AIR und über das Distribution System (DS).
- 802.11r verfügt über mehr Ebenen der Schlüsselhierarchie.
- Da dieses Protokoll den 4-Wege-Handshake für die Schlüsselverwaltung beim Roaming eines Clients vermeidet (erzeugt neue Verschlüsselungsschlüssel - PTK und GTK - ohne diesen Handshake), kann es auch für WPA2-Einrichtungen mit einem PSK angewendet werden, und nicht nur, wenn 802.1X/EAP für die Authentifizierung verwendet wird. Dies beschleunigt das Roaming noch mehr für diese Konfigurationen, bei denen kein EAP- oder 4-Wege-Handshake-Austausch stattfindet.

Bei diesem Verfahren führt der WLAN-Client beim Herstellen einer Verbindung mit dem ersten WAP nur eine erste Authentifizierung gegenüber der WLAN-Infrastruktur durch und führt beim Roaming zwischen WAPs derselben FT-Mobilitätsdomäne ein schnelles und sicheres Roaming durch.

Dies ist eines der neuen Konzepte, das sich im Wesentlichen auf die APs bezieht, die die gleiche SSID (Extended Service Set oder ESS) verwenden und die gleichen FT-Schlüssel verarbeiten. Dies ist vergleichbar mit den anderen bisher erläuterten Verfahren. Die Art und Weise, wie die APs die FT-Mobilitätsdomänenschlüssel handhaben, basiert normalerweise auf einer zentralisierten Einrichtung, wie dem WLC oder Mobilitätsgruppen; diese Methode kann jedoch auch in autonomen AP-Umgebungen implementiert werden.

Hier eine Zusammenfassung der Haupthierarchie:

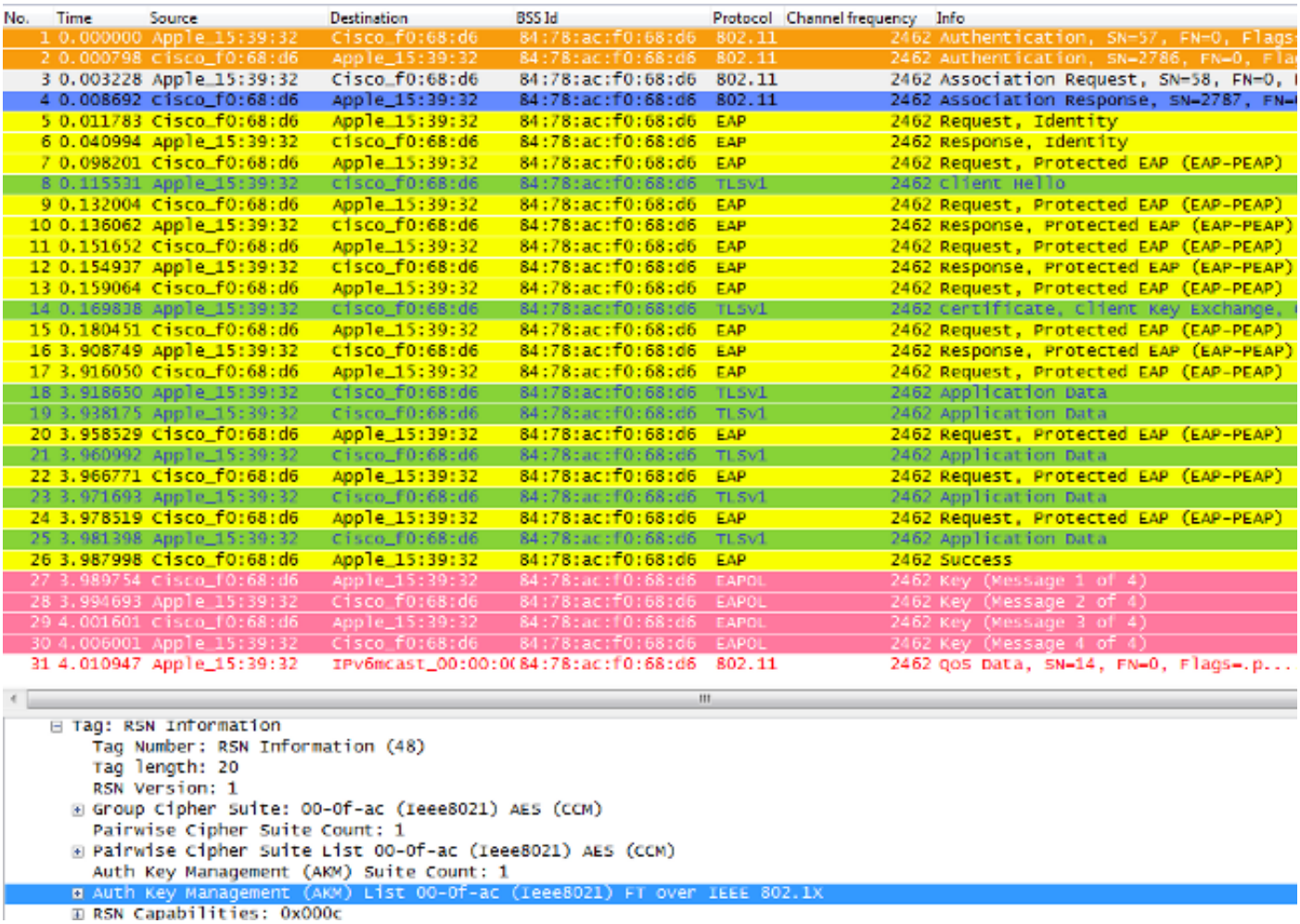
- Ein MSK wird auf der Client-Komponente und dem Authentifizierungsserver weiterhin von der anfänglichen 802.1X/EAP-Authentifizierungsphase abgeleitet (vom Authentifizierungsserver an den Authentifizierer (WLC) übertragen, sobald die Authentifizierung erfolgreich ist). Diese MSK wird wie bei den anderen Methoden als Ausgangspunkt für die FT-Schlüsselhierarchie verwendet. Wenn Sie WPA2-PSK anstelle einer EAP-Authentifizierungsmethode verwenden, ist der PSK im Grunde dieser MSK.
- Ein Pairwise Master Key R0 (PMK-R0) wird aus dem MSK abgeleitet, dem First-Level-Key der FT-Schlüsselhierarchie. Die Schlüsselhalter für diese PMK-R0 sind der WLC und der Client.
- Aus dem PMK-R0 wird ein Schlüssel der zweiten Ebene abgeleitet, der als Pairwise Master Key R1 (PMK-R1) bezeichnet wird. Die Schlüsselhalter sind der Client und die APs, die vom WLC verwaltet werden, der den PMK-R0 enthält.
- Der dritte und letzte Schlüssel der FT-Schlüsselhierarchie ist der PTK. Dies ist der letzte

Schlüssel, der zur Verschlüsselung der 802.11-Unicast-Datenframes verwendet wird (ähnlich den anderen Methoden, die WPA/TKIP oder WPA2/AES verwenden). Diese PTK wird auf FT von der PMK-R1 abgeleitet, und die Schlüsselhalter sind der Client und die APs, die vom WLC verwaltet werden.

Hinweis: Je nach WLAN-Anbieter und den Implementierungseinrichtungen (z. B. autonome APs, FlexConnect oder Mesh) kann die WLAN-Infrastruktur die Schlüssel auf andere Weise übertragen und verarbeiten. Es kann sogar die Rollen der Schlüsselhalter ändern, da dies jedoch nicht im Rahmen dieses Dokuments enthalten ist, stehen die Beispiele, die auf der zuvor angegebenen Zusammenfassung der Schlüsselhierarchie basieren, im Mittelpunkt. Die Unterschiede sind für das Verständnis des Prozesses nicht relevant, es sei denn, Sie müssen die Infrastrukturgeräte (und ihren Code) eingehend analysieren, um ein Softwareproblem zu erkennen.

Schnelle drahtlose BSS-Umstellung

Bei dieser Methode ist die erste Zuordnung zu einem WAP eine reguläre Erstauthentifizierung im WLAN, wobei die gesamte 802.1X/EAP-Authentifizierung gegenüber dem Authentifizierungsserver und der 4-Wege-Handshake zur Schlüsselgenerierung erfolgen muss, bevor Datenframes gesendet werden, wie in diesem Screenimage gezeigt:



Die Hauptunterschiede sind:

- Die Aushandlung für das Authentifizierungsschlüsselmanagement unterscheidet sich geringfügig von der Aushandlung für das reguläre WPA/WPA2, sodass zusätzliche

Informationen verwendet werden, um diese Aushandlung durchzuführen, wenn die Zuordnung zu einer WLAN-Infrastruktur erfolgt, die FT unterstützt. Wie im Bild gezeigt, wird der Zuordnungsanforderungsrahmen vom Client ausgewählt, und das AKM-Feld des RNS-Informationselements wird hervorgehoben, um zu zeigen, dass dieser Client FT über 802.1X/EAP ausführen möchte.

- Ebenfalls abgebildet ist das Mobility Domain Information Element (Teil von FT), bei dem das Feld "**FT Capability and Policy**" angibt, ob der schnelle BSS-Übergang beim schnellen Roaming Over-the-Air oder Over-the-DS abgeschlossen ist (in diesem Bild "Over-the-Air").
- Ein weiteres Informationselement (Fast BSS Transition oder FT IE, das später in diesem Dokument beschrieben wird) wird mit Informationen ergänzt, die erforderlich sind, um die FT-Authentifizierungssequenz beim FT-Roaming durchzuführen.
- Die Schlüsselgenerierung unterscheidet sich aufgrund der Schlüsselhierarchie. Obwohl der FT 4-Way-Handshake ähnlich aussieht wie der WPA/WPA2 4-Way-Handshake, ist er inhaltlich tatsächlich etwas anders.

Die Debugs zeigen im Wesentlichen den gleichen EAP-Authentifizierungs-Frame-Austausch wie die übrigen Methoden bei der Erstauthentifizierung im WLAN (wie aus den Bildern hervorgeht), aber einige Ausgaben, die die Schlüsselcaching-Techniken betreffen, die vom WLC verwendet werden, werden hinzugefügt; daher wird diese Debug-Ausgabe ausgeschnitten, um nur die relevanten Informationen anzuzeigen:

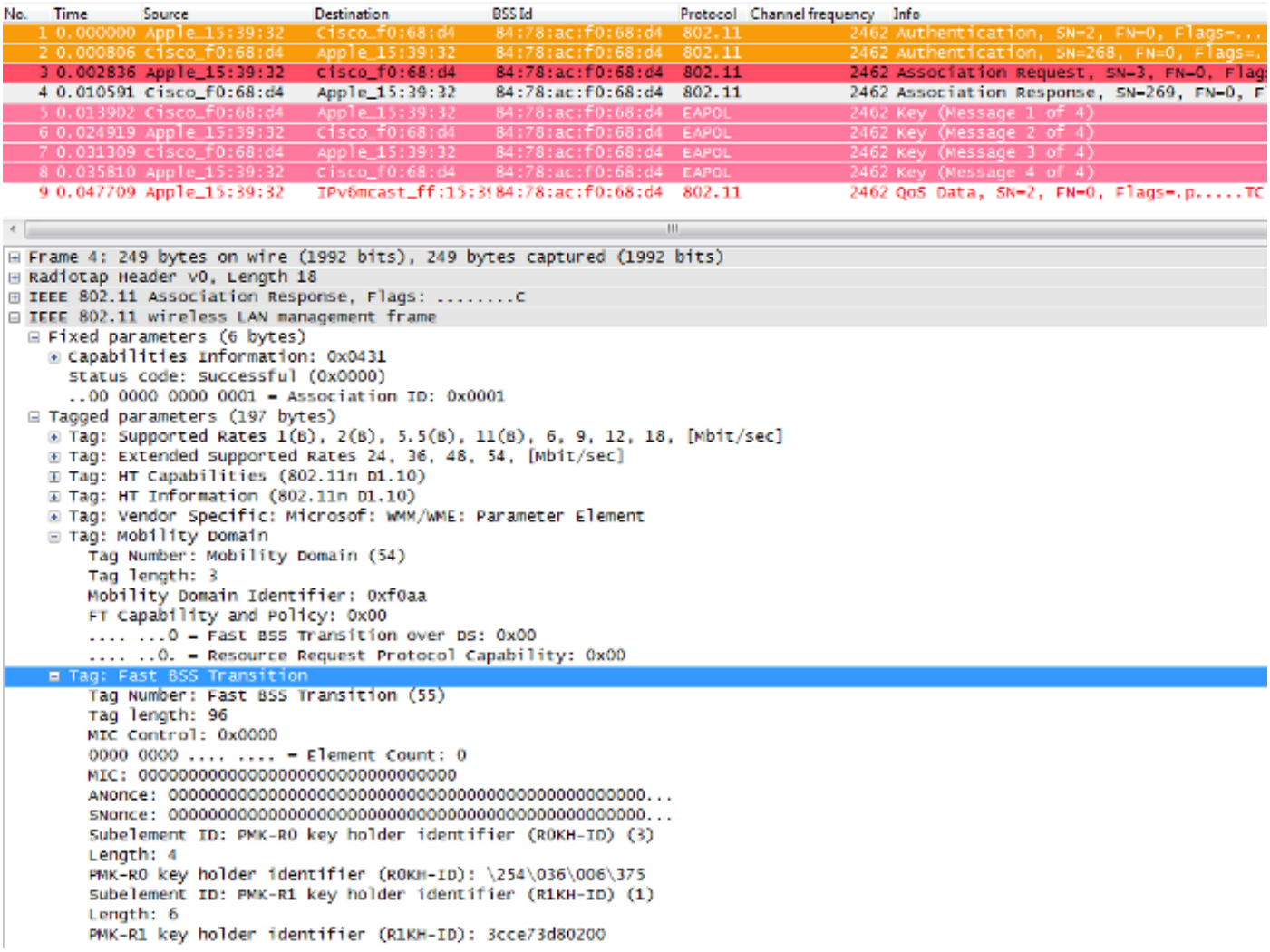
```
*apfMsConnTask_0: Jun 27 19:25:23.426: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d6
!--- This is the Association request from the client.

*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.
!--- WLC recognizes that the client is 802.11r-capable.

*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 20 for mobile
  ec:85:2f:15:39:32
!--- The WLC/AP finds an Information Element that claims FT
     support on the Association request that is sent from the client.

*apfMsConnTask_0: Jun 27 19:25:23.427:
  Sending assoc-resp station:ec:85:2f:15:39:32
  AP:84:78:ac:f0:68:d0-00 thread:144be808
*apfMsConnTask_0: Jun 27 19:25:23.427:
  Adding MDIE, ID is:0xaaf0
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in Initial
  assoc Resp to mobile
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Sending R0KH-ID as:-84.30.6.-3
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Sending R1KH-ID as 3c:ce:73:d8:02:00
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Including FT IE (length 98) in Initial Assoc Resp to mobile
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d6
  (status 0) ApVapId 7 Slot 0
!--- The Association Response is sent to the client once the
     FT information is computed (as per the previous messages),
     so this is included in the response.
```

```
*dot1xMsgTask: Jun 27 19:25:23.432: ec:85:2f:15:39:32
  Sending EAP-Request/Identity to mobile ec:85:2f:15:39:32
  (EAP Id 1)
!--- EAP begins, and follows the same exchange explained so far.

*apfMsConnTask_0: Jun 27 19:25:23.436: ec:85:2f:15:39:32
  Got action frame from this client.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.449: ec:85:2f:15:39:32
  Received EAPOL EAPPKT from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.449: ec:85:2f:15:39:32
  Received Identity Response (count=1) from mobile
  ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.456: ec:85:2f:15:39:32
  Processing Access-Challenge for mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.456: ec:85:2f:15:39:32
  Sending EAP Request from AAA to mobile ec:85:2f:15:39:32
  (EAP Id 2)

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.479: ec:85:2f:15:39:32
  Received EAPOL EAPPKT from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.479: ec:85:2f:15:39:32
  Received EAP Response from mobile ec:85:2f:15:39:32
  (EAP Id 2, EAP Type 25)

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.627: ec:85:2f:15:39:32
  Processing Access-Accept for mobile ec:85:2f:15:39:32
!--- The client is validated/authenticated by the RADIUS Server.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.627: ec:85:2f:15:39:32
  Creating a PKC PMKID Cache entry for station
  ec:85:2f:15:39:32 (RSN 2)
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.627: ec:85:2f:15:39:32
  Resetting MSCB PMK Cache Entry 0 for station
  ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.627: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 8
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.628: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 0
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.628: ec:85:2f:15:39:32
  Adding BSSID 84:78:ac:f0:68:d6 to PMKID cache at index 0
  for station ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.628: New PMKID: (16)
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.628:
  [0000] 52 b8 8f cf 50 a7 90 98 2b ba d6 20 79 e4 cd f9
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.629: ec:85:2f:15:39:32
  Created PMK Cache Entry for TGr AKM:802.1x ec:85:2f:15:39:32
!--- WLC creates a PMK cache entry for this client, which is
     used for FT with 802.1X in this case, so the PMKID is
     computed with the AP MAC address (BSSID 84:78:ac:f0:68:d6).

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.629:
  ec:85:2f:15:39:32   R0KH-ID:172.30.6.253
  R1KH-ID:3c:ce:73:d8:02:00  MSK Len:48 pmkValidTime:1807
!--- The R0KH-ID and R1KH-ID are defined, as well as the PMK
     cache validity period.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: ec:85:2f:15:39:32
  PMK sent to mobility group
!--- The FT PMK cache entry for this client is shared with the
```

```
     WLCs on the mobility group.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: ec:85:2f:15:39:32
  Sending EAP-Success to mobile ec:85:2f:15:39:32 (EAP Id 12)

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: ec:85:2f:15:39:32
  Found an cache entry for BSSID 84:78:ac:f0:68:d6 in PMKID
  cache at index 0 of station ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: Including PMKID in
  M1  (16)
!--- The hashed PMKID is included on the Message-1 of the
     initial FT 4-Way handshake.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630:
  [0000] 52 b8 8f cf 50 a7 90 98 2b ba d6 20 79 e4 cd f9

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  INITPMK (message 1), replay counter 00.00.00.00.00.00.00.0
!--- Message-1 of the FT 4-Way handshake is sent from the
     WLC/AP to the client.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from
  mobile ec:85:2f:15:39:32
!--- Message-2 of the FT 4-Way handshake is received
     successfully from the client.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Calculating PMKR0Name
!--- The PMKR0Name is calculated.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  DOT11R: Sending cache add

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: Adding MDIE,
  ID is:0xaaf0
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Adding TIE for reassociation deadtime:20000 milliseconds
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Adding TIE for R0Key-Data valid time :1807
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.640: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
!--- After the MDIE, TIE for reassociation deadtime, and TIE
     for R0Key-Data valid time are calculated, the Message-3
     of this FT 4-Way handshake is sent from the WLC/AP to the
     client with this information.

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.651: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.651: ec:85:2f:15:39:32
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile ec:85:2f:15:39:32
!--- Message-4 (final message) of this initial FT 4-Way handshake
     is received successfully from the client, which confirms the
     installation of the derived keys. They can now be used in order
     to encrypt data frames with the current AP.
```

Hinweis: Um diese Methode zu debuggen und die hier gezeigten zusätzlichen 802.11r/FT-Ausgaben zu erreichen, wird zusammen mit dem **Debug-Client** ein zusätzliches Debug aktiviert, das **Aktivieren von Debug-FT-Ereignissen**.

Hier sehen Sie die Bilder und Fehlerbehebungen einer anfänglichen Verknüpfung mit dem WLAN, wenn Sie FT mit WPA2-PSK (anstelle einer 802.1X/EAP-Methode) durchführen. Dabei wird der Zuordnungsantwortrahmen vom WAP ausgewählt, um das Fast BSS Transition Information Element (hervorgehoben) anzuzeigen. Einige der wichtigsten Informationen, die für den FT 4-Way-Handshake erforderlich sind, werden ebenfalls angezeigt:




```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 20 for mobile
  ec:85:2f:15:39:32
```

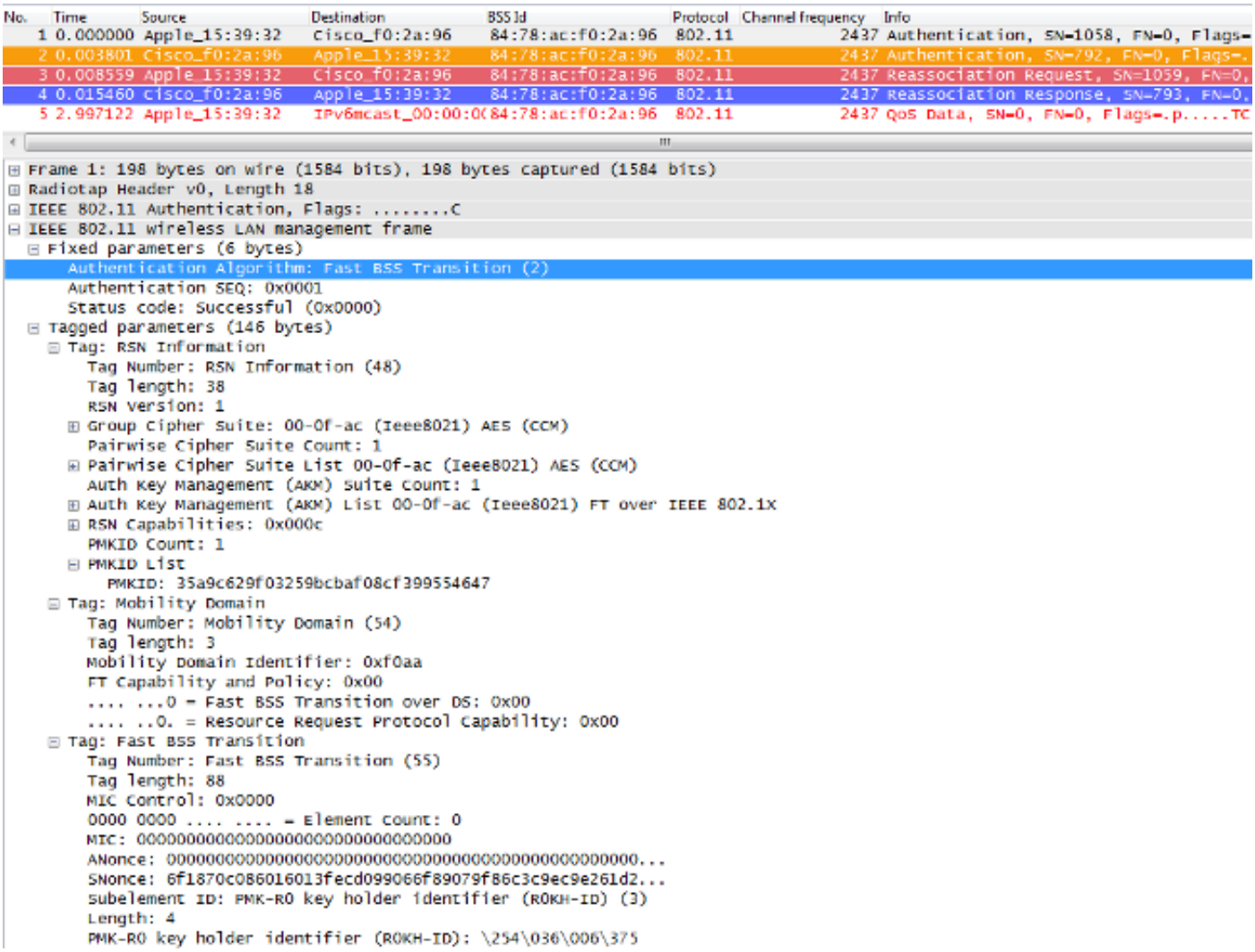
```
*apfMsConnTask_0: Jun 27 19:29:09.137: Sending assoc-resp
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:68:d0-00
  thread:144be808

*apfMsConnTask_0: Jun 27 19:29:09.137: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in Initial
  assoc Resp to mobile

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Sending R0KH-ID as:-84.30.6.-3

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Sending R1KH-ID as 3c:ce:73:d8:02:00

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Including FT IE (length 98) in Initial Assoc Resp to mobile

*apfMsConnTask_0: Jun 27 19:29:09.138: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d4
  (status 0) ApVapId 5 Slot 0

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Creating a PKC PMKID Cache entry for station
  ec:85:2f:15:39:32 (RSN 2)

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Resetting MSCB PMK Cache Entry 0 for station
  ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 8

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 0

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Adding BSSID 84:78:ac:f0:68:d4 to PMKID cache at
  index 0 for station ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: New PMKID: (16)

*dot1xMsgTask: Jun 27 19:29:09.142:
  [0000] 17 4b 17 5c ed 5f c7 1d 66 39 e9 5d 3a 63 69 e7

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Creating global PMK cache for this TGr client

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Created PMK Cache Entry for TGr AKM:PSK
  ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  R0KH-ID:172.30.6.253   R1KH-ID:3c:ce:73:d8:02:00
  MSK Len:48 pmkValidTime:1813

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Initiating RSN PSK to mobile ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Found an cache entry for BSSID 84:78:ac:f0:68:d4 in
  PMKID cache at index 0 of station ec:85:2f:15:39:32
```

```
*dot1xMsgTask: Jun 27 19:29:09.142: Including PMKID
  in M1  (16)

*dot1xMsgTask: Jun 27 19:29:09.142:
  [0000] 17 4b 17 5c ed 5f c7 1d 66 39 e9 5d 3a 63 69 e7

*dot1xMsgTask: Jun 27 19:29:09.143: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00

*apfMsConnTask_0: Jun 27 19:29:09.144: ec:85:2f:15:39:32
  Got action frame from this client.

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.152: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from
  mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Calculating PMKR0Name

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: Adding MDIE,
  ID is:0xaaf0

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Adding TIE for reassociation deadtime:20000 milliseconds

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Adding TIE for R0Key-Data valid time :1813

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.154: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.163: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.163: ec:85:2f:15:39:32
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile ec:85:2f:15:39:32
```

Bei 802.11r ist die anfängliche Zuordnung zum WLAN die Grundlage, auf der die bei dieser Technik verwendeten Basisschlüssel abgeleitet werden, genau wie bei den anderen schnellsicheren Roaming-Methoden. Die Hauptunterschiede entstehen, wenn der Client Roaming beginnt; FT vermeidet nicht nur 802.1X/EAP, wenn dies verwendet wird, sondern es führt tatsächlich eine effizientere Roaming-Methode durch, die die ursprünglichen 802.11 Open System Authentication und Reassociation Frames (die immer verwendet werden und erforderlich sind, wenn Roaming zwischen APs) kombiniert, um FT-Informationen auszutauschen und neue dynamische Verschlüsselungsschlüssel anstelle des 4-Way-Handshake abzuleiten 1.

Das nächste Bild zeigt die Frames, die bei einem schnellen BSS-Übergang über das Internet mit 802.1X/EAP-Sicherheit ausgetauscht werden. Der Frame der offenen Systemauthentifizierung vom Client zum Access Point wird ausgewählt, damit die FT-Protokoll-Informationselemente angezeigt werden, die zum Starten der FT-Schlüsselaushandlung erforderlich sind. Diese wird verwendet, um die neue PTK mit dem neuen AP abzuleiten (basierend auf der PMK-R1). Das Feld, das den Authentifizierungsalgorithmus anzeigt, ist hervorgehoben, um anzuzeigen, dass

dieser Client keine einfache Open System Authentication, sondern einen Fast BSS Transition durchführt:

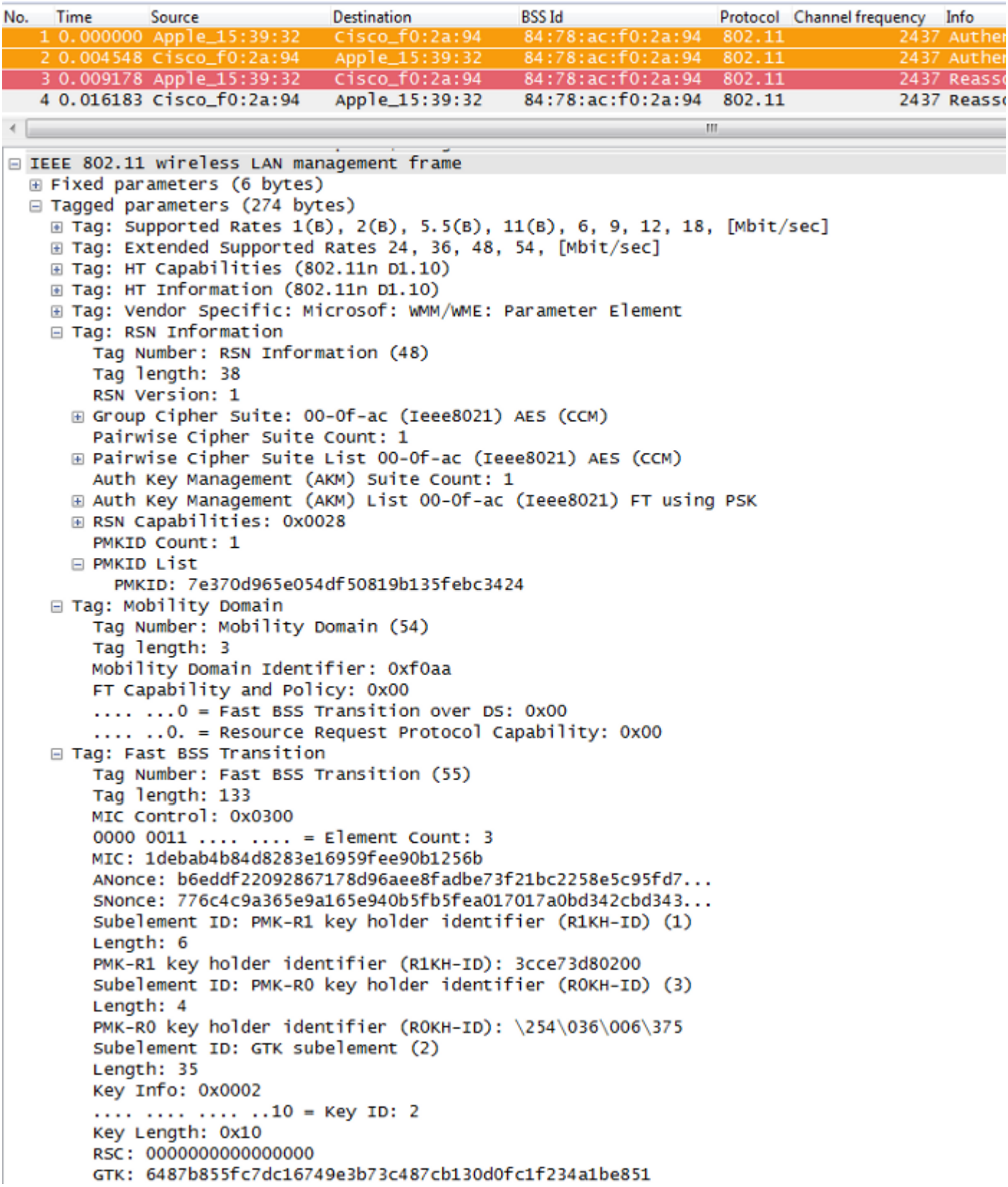


Nachfolgend sind die Debug-Ausgaben des WLC aufgeführt, wenn dieses FT-Roaming-Ereignis bei 802.1X/EAP auftritt:

```
*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Doing preauth for this client over the Air
!--- WLC begins FT fast-secure roaming over-the-Air with
      this client and performs a type of preauthentication,
      because the client asks for this with FT on the Authentication
      frame that is sent to the new AP over-the-Air
      (before the Reassociation Request).

*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Doing local roaming for destination address
  84:78:ac:f0:2a:96
!--- WLC performs the local roaming event with the new AP to
      which the client roams.

*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Got 1 AKMs in RSNIE
*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  RSNIE AKM matches with PMK cache entry :0x3
!--- WLC receives one PMK from this client (known as AKM here),
      which matches the PMK cache entry hold for this client.
```

```
*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Created a new preauth entry for AP:84:78:ac:f0:2a:96
*apfMsConnTask_2: Jun 27 19:25:48.751: Adding MDIE,
  ID is:0xaaf0
!--- WLC creates a new preauth entry for this AP-and-Client pair,
     and adds the MDIE information.

*apfMsConnTask_2: Jun 27 19:25:48.763: Processing assoc-req
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38
*apfMsConnTask_2: Jun 27 19:25:48.763: ec:85:2f:15:39:32
  Reassociation received from mobile on BSSID
  84:78:ac:f0:2a:96
!--- Once the client receives the Authentication frame reply from the
     WLC/AP, the Reassociation request is sent, which is received at
     the new AP to which the client roams.

*apfMsConnTask_2: Jun 27 19:25:48.764: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.

*apfMsConnTask_2: Jun 27 19:25:48.764: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 38 for mobile
  ec:85:2f:15:39:32

*apfMsConnTask_2: Jun 27 19:25:48.765: ec:85:2f:15:39:32
  Roaming succeed for this client.
!--- WLC confirms that the FT fast-secure roaming is successful
     for this client.

*apfMsConnTask_2: Jun 27 19:25:48.765: Sending assoc-resp
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38
*apfMsConnTask_2: Jun 27 19:25:48.766: Adding MDIE,
  ID is:0xaaf0
*apfMsConnTask_2: Jun 27 19:25:48.766: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in
  reassociation assoc Resp to mobile
*apfMsConnTask_2: Jun 27 19:25:48.766: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:96
  (status 0) ApVapId 7 Slot 0
!--- The Reassociation response is sent to the client, which
     includes the FT Mobility Domain IE.

*dot1xMsgTask: Jun 27 19:25:48.769: ec:85:2f:15:39:32
  Finishing FT roaming for mobile ec:85:2f:15:39:32
!--- FT roaming finishes and EAP is skipped (as well as any
     other key management handshake), so the client is ready
     to pass encrypted data frames with the current AP.

*dot1xMsgTask: Jun 27 19:25:48.769: ec:85:2f:15:39:32
  Skipping EAP-Success to mobile ec:85:2f:15:39:32
```

Das folgende Bild zeigt einen schnellen drahtlosen BSS-Übergang mit WPA2-PSK-Sicherheit, bei dem der endgültige Frame für die Antwort auf die Neuzuordnung vom Access Point zum Client ausgewählt wird, um weitere Details zu diesem FT-Austausch anzuzeigen:



Hier sind die Debug-Ausgaben, wenn dieses FT-Roaming-Ereignis mit PSK auftritt, die den Ausgaben bei Verwendung von 802.1X/EAP ähneln:

```
*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Doing preauth for this client over the Air

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Doing local roaming for destination address
```

```
  84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Got 1 AKMs in RSNIE

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  RSNIE AKM matches with PMK cache entry :0x4

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Created a new preauth entry for AP:84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.854: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_2: Jun 27 19:29:29.867: Processing assoc-req
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38

*apfMsConnTask_2: Jun 27 19:29:29.867: ec:85:2f:15:39:32
  Reassociation received from mobile on BSSID
  84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.868: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.

*apfMsConnTask_2: Jun 27 19:29:29.868: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 38 for mobile
  ec:85:2f:15:39:32

*apfMsConnTask_2: Jun 27 19:29:29.869: ec:85:2f:15:39:32
  Roaming succeed for this client.

*apfMsConnTask_2: Jun 27 19:29:29.869: Sending assoc-resp
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38

*apfMsConnTask_2: Jun 27 19:29:29.869: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_2: Jun 27 19:29:29.869: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in
  reassociation assoc Resp to mobile

*apfMsConnTask_2: Jun 27 19:29:29.870: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:2a:94 (status 0) ApVapId 5 Slot 0

*dot1xMsgTask: Jun 27 19:29:29.874: ec:85:2f:15:39:32
  Finishing FT roaming for mobile ec:85:2f:15:39:32
```

Wie in der Abbildung dargestellt, werden nach der Aushandlung des Fast BSS Transition bei der ersten Zuordnung zum WLAN die vier Frames, die für das Roaming verwendet werden und benötigt werden (Open System Authentication vom Client, Open System Authentication vom AP, Reassociation Request und Reassociation Response) im Wesentlichen als FT 4-Way Handshake verwendet, um den neuen PTK (Unicast Encryption Key) und GTK (Multicast/Broadcast Encryption Key) abzuleiten.

Dies ersetzt den 4-Wege-Handshake, der normalerweise auftritt, nachdem diese Frames ausgetauscht wurden, und die FT-Inhalte und Schlüsselaushandlung bei diesen Frames ist grundsätzlich gleich, ob Sie 802.1X/EAP oder PSK als Sicherheitsmethode verwenden. Wie im Bild gezeigt, ist das AKM-Feld der Hauptunterschied, der bestätigt, ob der Client FT mit PSK oder 802.1X durchführt. Daher ist es wichtig zu beachten, dass diese vier Frames normalerweise nicht

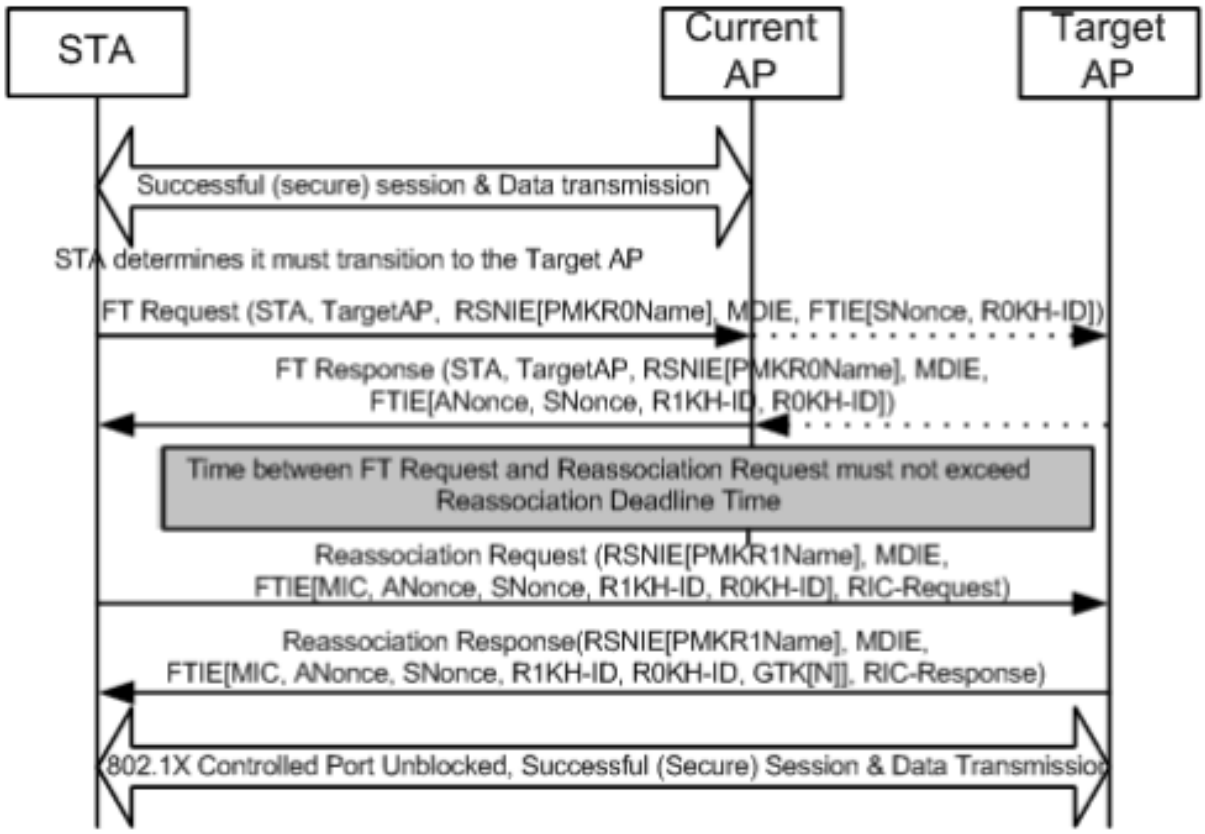
über diese Art von Sicherheitsinformationen für die Schlüsselaushandlung verfügen, sondern nur, wenn der Client-FT Roaming durchführt, wenn 802.11r implementiert ist und zwischen dem Client und der WLAN-Infrastruktur bei der ersten Zuordnung ausgehandelt wird.

Schneller BSS-Übergang über den DS

802.11r ermöglicht eine weitere Implementierung von Fast BSS Transition, bei der das FT-Roaming vom Client mit dem neuen AP initiiert wird, für den der Client über den DS (Distribution System) und nicht über das Funk roamt. In diesem Fall werden FT Action-Frames verwendet, um die Schlüsselaushandlung anstelle der Open System Authentication-Frames zu initiieren.

Sobald der Client beschließt, zu einem besseren Access Point wechseln zu können, sendet er einen FT Action Request Frame an den ursprünglichen Access Point, mit dem er derzeit verbunden ist, bevor er zum Roaming wechselt. Der Client gibt die BSSID (MAC-Adresse) des Ziel-AP an, in dem er Roaming betreiben möchte. Der ursprüngliche WAP leitet diesen FT Action Request Frame über das Verteilersystem (in der Regel die verkabelte Infrastruktur) an den Ziel-WAP weiter, und der Ziel-WAP antwortet dem Client mit einem FT Action Response Frame (auch über den DS, sodass er ihn schließlich drahtlos an den Client senden kann). Sobald dieser FT Action-Frame-Austausch erfolgreich war, beendet der Client das FT-Roaming; der Client sendet die Neuzuordnungsanforderung an den Ziel-AP (diesmal drahtlos) und erhält eine Neuzuordnungsanwort vom neuen AP, um die Ableitung des Roaming- und des endgültigen Schlüssels zu bestätigen.

Zusammenfassend gibt es vier Frames, die den schnellen BSS-Übergang aushandeln und neue Verschlüsselungsschlüssel ableiten, aber hier werden die Open System Authentication Frames durch die FT Action Request/Response Frames ersetzt, die mit dem Ziel-AP über das Distribution System mit dem aktuellen AP ausgetauscht werden. Dieses Verfahren gilt auch für die beiden Sicherheitsmethoden 802.1X/EAP und PSK, die alle von den Cisco Wireless LAN Controllern unterstützt werden. Da dieser Over-the-DS Übergang jedoch von den meisten Wireless Clients in der WiFi-Branche nicht unterstützt und implementiert wird (und da der Frame-Austausch und die Debug-Ausgaben im Wesentlichen gleich sind), werden in diesem Dokument keine Beispiele angegeben. Stattdessen wird dieses Bild verwendet, um den schnellen BSS-Übergang über den DS zu veranschaulichen:



FlexConnect mit 802.11r

- Die zentrale Authentifizierung wird unterstützt. Dies umfasst das lokale und zentrale Daten-Switching. Die APs müssen Teil derselben FlexConnect-Gruppe sein.
- Die lokale Authentifizierung wird nicht unterstützt.
- Der Standalone-Modus wird nicht unterstützt.

Vorteile mit 802.11r

- Diese Methode ist die erste, die eine Schlüsselhierarchie verwendet, die vom IEEE auf dem 802.11-Standard als Ergänzung (802.11r) klar definiert ist, sodass die Implementierung dieser FT-Techniken zwischen den Anbietern kompatibler ist und keine unterschiedlichen Interpretationen aufweist.
- 802.11r ermöglicht verschiedene Techniken, die je nach Ihren Anforderungen hilfreich sind (Over-the-Air und Over-the-DS, für 802.1x/EAP-Sicherheit und für PSK-Sicherheit).
- Der Wireless-Client führt ein schnelles und sicheres Roaming zu einem neuen WAP im selben WLAN/SSID durch, selbst wenn er nie mit diesem WAP verbunden war und ohne mehrere PMKIDs speichern zu müssen.
- Dies ist die erste schnellsichere Roaming-Methode, die auch bei PSK-Sicherheit ein schnelleres Roaming ermöglicht und den beim Roaming zwischen APs mit WPA/WPA2-PSK erforderlichen 4-Wege-Handshake vermeidet. Der Hauptzweck der Fast-Secure-Roaming-Methoden besteht darin, den 802.1X/EAP-Handshake bei der Implementierung dieser Sicherheitsmethode zu vermeiden; für die PSK-Sicherheit wird das Roaming-Ereignis jedoch mit 802.11r noch weiter beschleunigt, wenn der 4-Wege-Handshake vermieden wird.

Nachteile von 802.11r

- Es gibt einige Wireless-Client-Geräte, die Fast BSS Transitions unterstützen, und in den meisten Fällen unterstützen sie nicht alle auf 802.11r verfügbaren Techniken.
- Da diese Implementierungen noch sehr jung sind, gibt es weder genügend Testergebnisse aus realen Produktionsumgebungen noch genügend Debugging-Ergebnisse, um mögliche Probleme zu beheben.
- Wenn Sie eine WLAN/SSID konfigurieren, um eine der FT-Methoden zu verwenden, können nur Wireless-Clients, die 802.11r unterstützen, eine Verbindung zu dieser WLAN/SSID herstellen. Die FT-Einstellungen sind für die Clients nicht optional. Daher müssen sich Wireless-Clients, die 802.11r nicht unterstützen, mit einer separaten WLAN/SSID verbinden, wenn FT überhaupt nicht konfiguriert ist.

Adaptives 802.11r

- Einige ältere Clients können keine Verbindung zu einer WLAN/SSID herstellen, bei der 802.11r aktiviert ist, selbst im "gemischten Modus" (den Sie hoffentlich auf denselben SSID-Clients haben können, die 802.11r unterstützen und nicht unterstützen). In diesem Fall ist der Treiber der Client-Komponente, die für das Parsen des Robust Security Network Information Element (RSN IE) zuständig ist, alt und kennt die zusätzlichen AKM-Suites im IE nicht. Aufgrund dieser Einschränkung können Clients keine Zuordnungsanforderungen an WLANs senden, die 802.11r-Unterstützung ankündigen. Daher müssen Sie eine WLAN/SSID für 802.11r-Clients und eine separate WLAN/SSID für Clients, die 802.11r nicht unterstützen, konfigurieren.
- Um dieses Problem zu lösen, hat die Cisco Wireless LAN-Infrastruktur die Funktion Adaptive 802.11r eingeführt. Wenn der FT-Modus auf WLAN-Ebene auf "Adaptive" festgelegt ist, meldet das WLAN die 802.11r-Mobilitätsdomänen-ID in einem 802.11i-fähigen WLAN. Einige Apple iOS10-Client-Geräte identifizieren das Vorhandensein von MDIE in einem 80211i/WPA2-WLAN und führen einen proprietären Handshake aus, um eine 802.11r-Zuordnung herzustellen. Sobald der Client die erfolgreiche 802.11r-Zuordnung abgeschlossen hat, kann er das FT-Roaming wie in einem normalen 802.11r-fähigen WLAN ausführen. Der FT Adaptive Service gilt nur für ausgewählte Apple iOS10-Geräte (oder höher). Alle anderen Clients können weiterhin eine 802.11i/WPA2-Zuordnung im WLAN aufweisen und die entsprechende FSR-Methode wie unterstützt ausführen.
- Weitere Dokumentation zu dieser neuen Funktion, die für iOS10-Geräte eingeführt wurde, um 802.11r in einem WLAN/SSID auszuführen, in dem 802.11r nicht wirklich aktiviert ist (sodass andere Clients, die nicht 802.11r sind, sich erfolgreich verbinden können), finden Sie unter Best Practices für Cisco IOS-Geräte im Cisco Wireless LAN.

Schlussfolgerungen

- Beachten Sie, dass der Client immer derjenige ist, der sich für das Roaming zu einem bestimmten AP entscheidet, und der WLC/AP kann dies für den Client nicht entscheiden. Das Roaming-Ereignis wird vom Wireless-Client initiiert, sobald er es als Roaming betrachtet.
- Der WLC unterstützt eine Kombination der meisten oder aller FSR-Methoden (Fast-Secure Roaming) zusammen auf demselben WLAN/SSID. Beachten Sie jedoch, dass dies normalerweise nicht funktioniert, da es stark vom Clientverhalten abhängt (sehr

unterschiedlich bei verschiedenen Mobilgeräten), um das zu unterstützen oder sogar zu verstehen, was der WLC als unterstützt ankündigen möchte. Anstatt die Interoperabilität mit nur einer SSID zu erreichen, gibt es normalerweise mehr Probleme als die, die voraussichtlich behoben werden. Daher wird davon abgeraten. Es sind intensive Tests mit allen möglichen Clients, die in diesem WLAN verwendet werden können, durchzuführen, wenn dies wirklich erforderlich ist.

- Es ist sehr wichtig zu verstehen, dass schnelle sichere Roaming-Methoden entwickelt werden, um den WLAN-Roaming-Prozess zu beschleunigen, wenn Sie zwischen APs wechseln, wenn die Sicherheit für WLAN/SSID aktiviert ist. Wenn keine Sicherheit vorhanden ist, gibt es nichts zu beschleunigen, da der Client-AP lediglich die Wireless-Management-Frames austauscht, die beim Roaming zwischen APs vor dem Senden von Daten-Frames immer erforderlich sind (Open System Authentication vom Client, Open System Authentication vom AP, Reassoziationsanforderung und Reassoziationsantwort). Daher kann dies nicht schneller gehen. Wenn Roaming-Probleme ohne Sicherheit auftreten, gibt es keine schnellen Roaming-Methoden zur Verbesserung des Roaming, sondern nur Methoden, um zu bestätigen, ob die Einrichtung und das Design von WLAN/SSID für die drahtlosen Client-Stationen geeignet sind, entsprechend zwischen den WAP-Abdeckungszellen zu roamen.
- 802.11r/FT wird mit WPA2-PSK implementiert, um Roaming-Ereignisse mit dieser Sicherheit zu beschleunigen und den 4-Wege-Handshake zu vermeiden, wie im Abschnitt 802.11r erläutert.
- Alle Methoden haben ihre Vor- und Nachteile, aber am Ende müssen Sie immer überprüfen, ob die Wireless-Client-Stationen die gewünschte Methode unterstützen und ob die Cisco WLAN-Infrastruktur alle verfügbaren Methoden unterstützt. Daher müssen Sie die Methode auswählen, die von den Wireless-Clients, die eine Verbindung mit dem jeweiligen WLAN/SSID herstellen, am besten unterstützt wird. In einigen Bereitstellungen können Sie z. B. eine WLAN/SSID mit CCKM für Cisco Wireless IP-Telefone erstellen (die WPA2/AES mit CCKM, jedoch nicht 802.11r unterstützen), und dann eine weitere WLAN/SSID mit WPA2/AES über 802.11r/FT für Wireless-Clients, die diese Fast Secure Roaming-Methode unterstützen (oder OKC , wenn dies unterstützt wird).
- Wenn die Wireless-Clients keine der verfügbaren hochsicheren Roaming-Methoden unterstützen, müssen Sie akzeptieren, dass diese Clients beim Roaming zwischen APs in einem WLAN/SSID mit 802.1X/EAP-Sicherheit (die zu Unterbrechungen der Client-Anwendungen/Dienste führen kann) immer die in diesem Dokument beschriebenen Verzögerungen ausprobieren können.
- Alle Methoden mit Ausnahme von SKC (WPA2 PMKID-Caching) werden für schnelles Roaming zwischen APs unterstützt, die von verschiedenen WLCs verwaltet werden (Intercontroller-Roaming), sofern sie derselben Mobilitätsgruppe angehören.
- CUWN unterstützt alle in diesem Artikel behandelten Fast-Secure Roaming-Methoden, wenn die 802.1X/EAP-Authentifizierung für WPA/WPA2 verwendet wird. Bei Methoden, die mit WPA2-RSN (CCKM, PMKID Caching/SKC, OKC/PKC) arbeiten, wenn PSK (WPA2-Personal) verwendet wird, wobei Fast-Roaming-Methoden meist nicht benötigt werden, bietet CUWN keine Unterstützung für schnelles Roaming. CUWN unterstützt jedoch Fast-Secure Roaming bei WPA2-FT (802.11r) mit PSK, wie auch in diesem Artikel erläutert.

Zugehörige Informationen

- Implementierungsleitfaden für die schnelle Umrüstung von 802.11r BSS

- Technischer Support und Downloads von Cisco

Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.