确定802.11 WLAN和CUWN快速安全漫游的方法

目录

简介

本文档介绍适用于统一无线网络(CUWN)上的IEEE 802.11无线LAN(WLAN)的无线和快速安全漫游类型。

先决条件

要求

Cisco 建议您了解以下主题：

- IEEE 802.11 WLAN基础
- IEEE 802.11 WLAN安全
- IEEE 802.1X/EAP基础知识

使用的组件

本文档中的信息基于Cisco WLAN控制器软件版本7.4。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

本文档中的信息基于Cisco WLAN控制器软件版本7.4，但描述的大多数调试输出和行为可适用于支持所讨论方法的任何软件版本。在后面的Cisco WLAN控制器代码中，此处介绍的所有方法的具体内容保持不变（截至本文更新时版本8.3）。

本文档介绍适用于思科统一无线网络(CUWN)支持的IEEE 802.11无线LAN(WLAN)的不同类型的无线漫游和快速安全漫游方法。

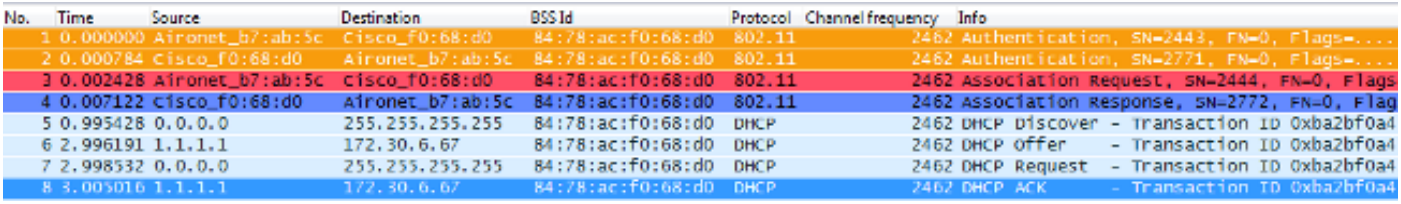
本文档并未提供每种方法的工作原理或配置方式的所有详细信息。本文档的主要目的是描述各种可用技术之间的差异、其优点和局限性，以及每种方法的帧交换。提供WLAN控制器(WLC)调试的示例，并使用无线数据包图像来分析和解释每种所述漫游方法所发生的事件。

在说明可用于WLAN的不同快速安全漫游方法之前，必须了解WLAN关联过程如何工作，以及在服务集标识符(SSID)上未配置安全时如何发生常规漫游事件。

当802.11无线客户端连接到接入点(AP)时，在它开始传递流量（无线数据帧）之前，它必须先通过基本的802.11开放系统身份验证过程。然后，必须完成关联过程。开放式系统身份验证过程类似于客户端选择的AP上的电缆连接。这一点非常重要，因为选择首选哪个AP的总是无线客户端，而且决策取决于供应商之间不同的多个因素。这就是客户端通过向选定AP发送身份验证帧来开始此过程的原因，如本文档后面部分所示。AP无法请求您建立连接。

使用来自AP（“电缆连接”）的响应成功完成开放系统身份验证过程后，关联过程基本上会完成802.11第2层(L2)协商，建立客户端和AP之间的链路。如果连接成功，则AP会向客户端分配关联ID，并对其进行准备，以便传递流量或执行更高级别的安全方法（如果在SSID上配置）。开放式系统身份验证过程包括两个管理帧以及关联过程。身份验证和关联帧是无线**管理**帧，而不是数据帧，数据帧基本上用于与AP的连接过程。

以下是此过程的无线帧图像：



注意：如果您想了解802.11无线嗅探技术，以及Wireshark上用于本文档中显示的图像的滤镜/颜色，请访问思科支持社区文章802.11嗅探器图像分析。

无线客户端以身份验证帧开始，AP以另一个身份验证帧作为回复。然后，客户端发送关联请求帧，然后AP使用关联响应帧完成回复。如DHCP数据包所示，一旦通过802.11开放系统身份验证和关联过程，客户端就会开始传送数据帧。在这种情况下，SSID上没有配置安全方法，因此客户端会立即开始发送未加密的数据帧（在本例中为DHCP）。

如本文档后面所示，如果SSID上启用了安全性，则特定安全方法会在关联响应之后和发送任何客户端流量数据帧之前(例如DHCP、地址解析协议(ARP)和应用数据包)出现更高级别的身份验证和加密握手帧，这些数据包经过加密。只有在客户端完全通过身份验证之后，才能发送数据帧，并且根据配置的安全方法协商加密密钥。

根据上一个映像，以下是当无线客户端开始与WLAN建立新关联时，您在WLC **debug client**命令的输出中看到的消息：

```
*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.
```

注：用于本文档中所示输出的WLC调试是**debug client**命令，示例仅显示一些相关消息，而不是整个输出。有关此debug命令的详细信息，请参阅名为了解无线LAN控制器(WLC)上的调试客户端的文档。

这些消息显示关联请求和响应帧；初始身份验证帧不会记录在WLC中，因为此握手在CUWN上的AP级别上快速发生。

客户端漫游时显示什么信息？客户端在与AP建立连接时始终交换四个管理帧，这归因于客户端建立关联或漫游事件。客户端每次仅建立到一个AP的一个连接。与WLAN基础设施的新连接和漫游事件之间的帧交换的唯一区别是，漫游事件的关联帧称为**Reassociation**帧，表示客户端实际上是从另一个AP漫游，没有尝试与WLAN建立新关联。这些帧可能包含用于协商漫游事件的不同元素；这取决于设置，但这些详细信息不在本文档的讨论范围之内。

以下是帧交换的示例：

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

调试输出中显示以下消息：

```
*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.
```

如图所示，客户端在向新AP发送重新关联请求后成功执行漫游事件，并从AP接收重新关联响应。由于客户端已经具有IP地址，因此第一个数据帧用于ARP数据包。

如果您期望漫游事件，但客户端发送关联请求而不是重新关联请求（您可以根据某些映像和调试确认该请求，类似于本文档前面所介绍的），则客户端实际上并不漫游。客户端开始与WLAN的新关联，就像断开连接一样，并尝试从头重新连接。发生这种情况的原因可能有多种，例如，客户端离开覆盖区域后发现具有足够信号质量的AP以开始关联，但通常表示客户端因驱动程序、固件或软件问题而无法发起漫游事件的客户端问题。

注：您可以咨询无线客户端供应商以确定问题的原因。

具有更高级别安全性的漫游

在基本802.11开放系统身份验证的基础上为SSID配置L2更高级别的安全时，初始关联和漫游时需要更多帧。本文档介绍两种最常用的802.11 WLAN标准化和实施的安全方法：

- **WPA/WPA2-PSK（预共享密钥）— 使用**预共享密钥对客户端进行身份验证。
- **WPA/WPA2-EAP（可扩展身份验证协议）— 使用**802.1X/EAP方法对客户端进行身份验证，以便通过使用身份验证服务器验证更安全的凭证，例如证书、用户名和密码以及令牌。

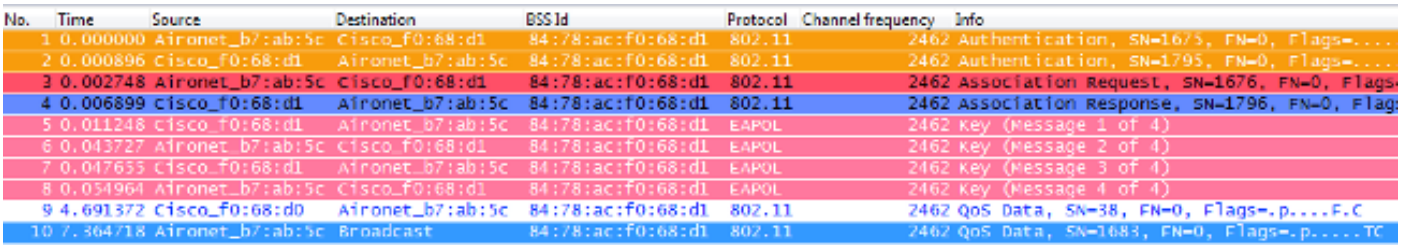
需要注意的是，尽管这两种方法（PSK和EAP）以不同方式验证/验证客户端，但两者都使用基本相同的WPA/WPA2规则进行密钥管理过程。无论安全是WPA/WPA2-PSK还是WPA/WPA2-EAP，称为WPA/WPA2 4次握手的过程都会在客户端与主会话密钥(MSK)作为原始密钥材料的WLC/AP之间开始密钥协商，一旦客户端使用所用的特定身份验证方法进行了验证。

以下是流程总结：

1. 当使用802.1X/EAP安全时，从EAP身份验证阶段或当使用WPA/WPA2-PSK作为安全方法时，从PSK派生MSK。
2. 从此MSK，客户端和WLC/AP派生成对主密钥(PMK),WLC/AP生成组主密钥(GMK)。
3. 一旦这两个主密钥准备就绪，客户端和WLC/AP将启动WPA/WPA2 4次握手（本文档后面部分将介绍一些屏幕图像和调试），主密钥将作为实际加密密钥协商的种子。
4. 这些最终加密密钥称为成对瞬时密钥(PTK)和组瞬时密钥(GTK)。PTK从PMK派生，用于加密与客户端的单播帧。组临时密钥(GTK)源自GMK，用于加密此特定SSID/AP上的组播/广播。

WPA/WPA2-PSK

当通过临时密钥完整性协议(TKIP)或高级加密标准(AES)执行用于加密的WPA-PSK或WPA2-PSK时，客户端必须在初始关联和漫游时执行称为WPA 4次握手的流程。如前所述，这基本上是用于WPA/WPA2派生加密密钥的密钥管理过程。但是，当执行PSK时，也使用它来验证客户端是否有有效的预共享密钥来加入WLAN。此图显示了执行WPA或WPA2与PSK的初始关联过程：



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

如图所示，在802.11开放系统身份验证和关联过程之后，WPA 4向握手中有四个EAPOL帧，它们由AP使用**message-1**发起，由客户端使用**message-4结束**。握手成功后，客户端开始传递数据帧（如DHCP），在这种情况下，数据帧使用源自四次握手的密钥进行加密（这就是为什么您无法看到来自无线映像的实际内容和流量类型）。

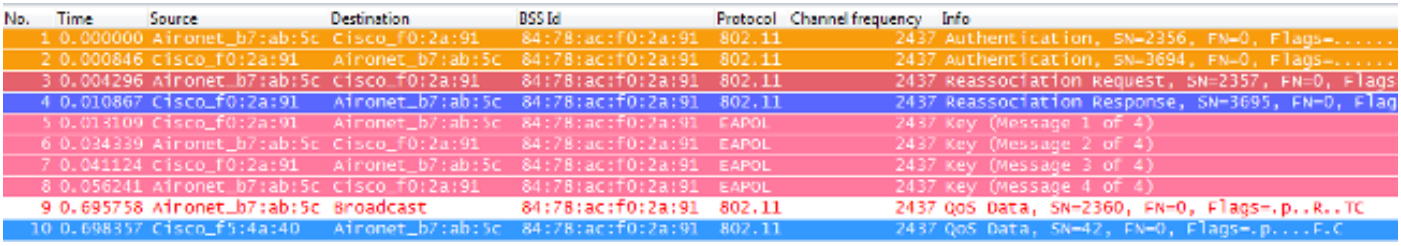
> **注**:EAPOL帧用于在AP和客户端之间通过空中传输所有密钥管理帧和802.1X/EAP身份验证帧；它们作为无线数据帧传输。

调试输出中显示以下消息：

```
*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.
```

漫游时，客户端基本上跟踪相同的帧交换，其中需要使用新的AP获取新的加密密钥，需要WPA 4次握手。这是因为标准确立的安全原因以及新AP不知道原始密钥的事实。唯一的区别是存在重新关联

帧而不是关联帧，如下图所示：



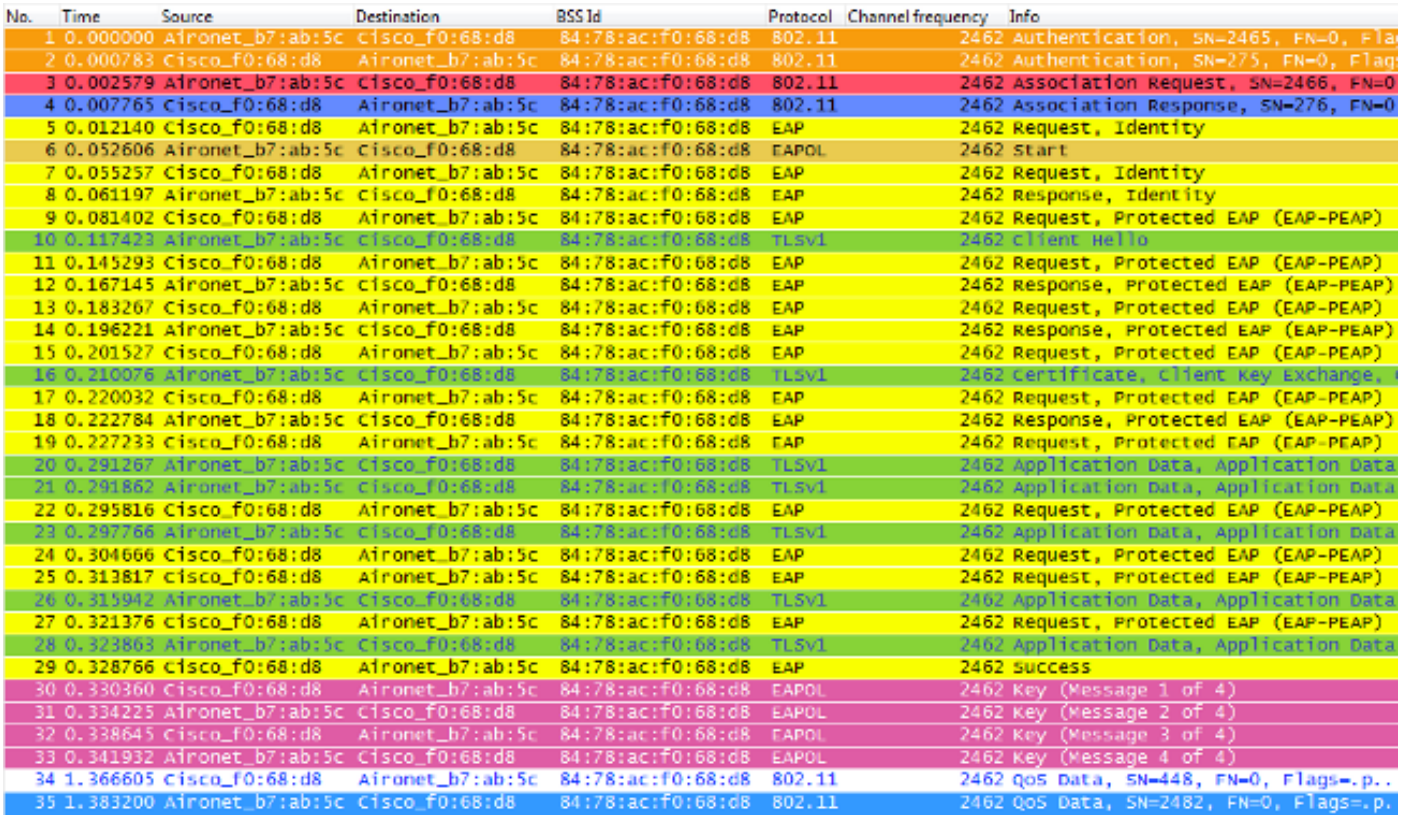
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3	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

您会在调试输出中看到相同的消息，但来自客户端的第一个数据包是重新关联而不是关联，如前文所示和解释。

WPA/WPA2-EAP

当使用802.1X/EAP方法对安全SSID上的客户端进行身份验证时，在客户端开始传递流量之前，需要更多帧。这些额外帧用于验证客户端凭证，根据EAP方法，可能有4到20个帧。这些密钥在关联/重新关联之后，但在WPA/WPA2 4次握手之前，因为身份验证阶段派生的MSK用作密钥管理过程（4次握手）中最终加密密钥生成的种子。

下图显示当执行具有PEAPv0/EAP-MSCHAPv2的WPA时，AP与无线客户端在初始关联时通过空中交换的帧的示例：



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.

有时，此交换显示或多或少帧，这取决于多种因素，例如EAP方法、由于问题导致的重新传输、客户端行为(例如本示例中的两个身份请求，因为客户端在AP发送第一个身份请求后发送**EAPOL START)，或者客户端已与服务器交换证书。**每当SSID配置为802.1X/EAP方法时，都会有更多帧（用于身份验证），因此，客户端开始发送数据帧前需要更多时间。

以下是调试消息的摘要：

```
*apfMsConnTask_0: Jun 21 23:41:19.092: 00:40:96:b7:ab:5c
```

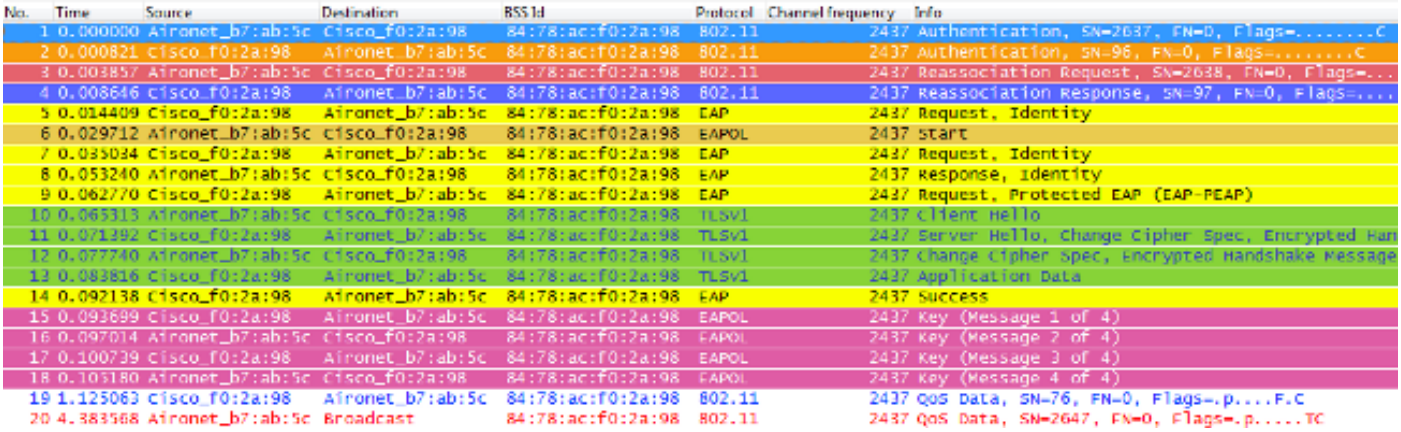
```
  Association received from mobile on BSSID 84:78:ac:f0:68:d8
*apfMsConnTask_0: Jun 21 23:41:19.094: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d8
  (status 0) ApVapId 9 Slot 0
!--- The Association handshake is finished.

*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.
```

当无线客户端在此处执行常规漫游（正常行为，未实施快速安全漫游方法）时，客户端必须经过完全相同的过程，并对身份验证服务器执行完全身份验证，如图所示。唯一的区别是客户端使用重新关联请求来通知新AP它实际上正在从另一个AP漫游，但客户端仍然需要进行完全验证并生成新密钥：



如图所示，即使帧少于初始身份验证（由前面提到的多个因素导致），当客户端漫游到新的AP时，也必须完成EAP身份验证和WPA密钥管理过程才能继续传递数据帧（即使流量在漫游前已主动发送）。因此，如果客户端有对延迟敏感的活动应用（如语音流量应用或对超时敏感的应用），则用

户会在漫游时察觉问题，如音频间隙或应用断开。这取决于该过程需要多长时间才能使客户端继续发送/接收数据帧。此延迟可能会更长，具体取决于：RF环境、客户端数量、WLC和LAP之间以及与身份验证服务器的往返时间以及其他原因。

以下是此漫游事件的调试消息的摘要（基本与前面的消息相同，因此这些消息不会进行进一步说明）：

```
*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
```

这是802.1X/EAP和WPA/WPA2安全框架的工作方式。为了防止应用/服务对常规漫游事件延迟的影响，WiFi行业开发和实施了多种快速安全漫游方法，以在WLAN/SSID上使用安全性时加快漫游过程。当客户端通过在WLAN上部署高级安全功能在AP之间漫游时继续传递流量时，会面临一些延迟。这是因为安全设置需要进行EAP身份验证和密钥管理帧交换，如前所述。

了解快速安全漫游是行业使用的术语，是指在WLAN上配置安全时实施加速漫游过程的方法/方案，这一点非常重要。 下一节将介绍可用于WLAN并受CUWN支持的各种快速安全漫游方法/方案。

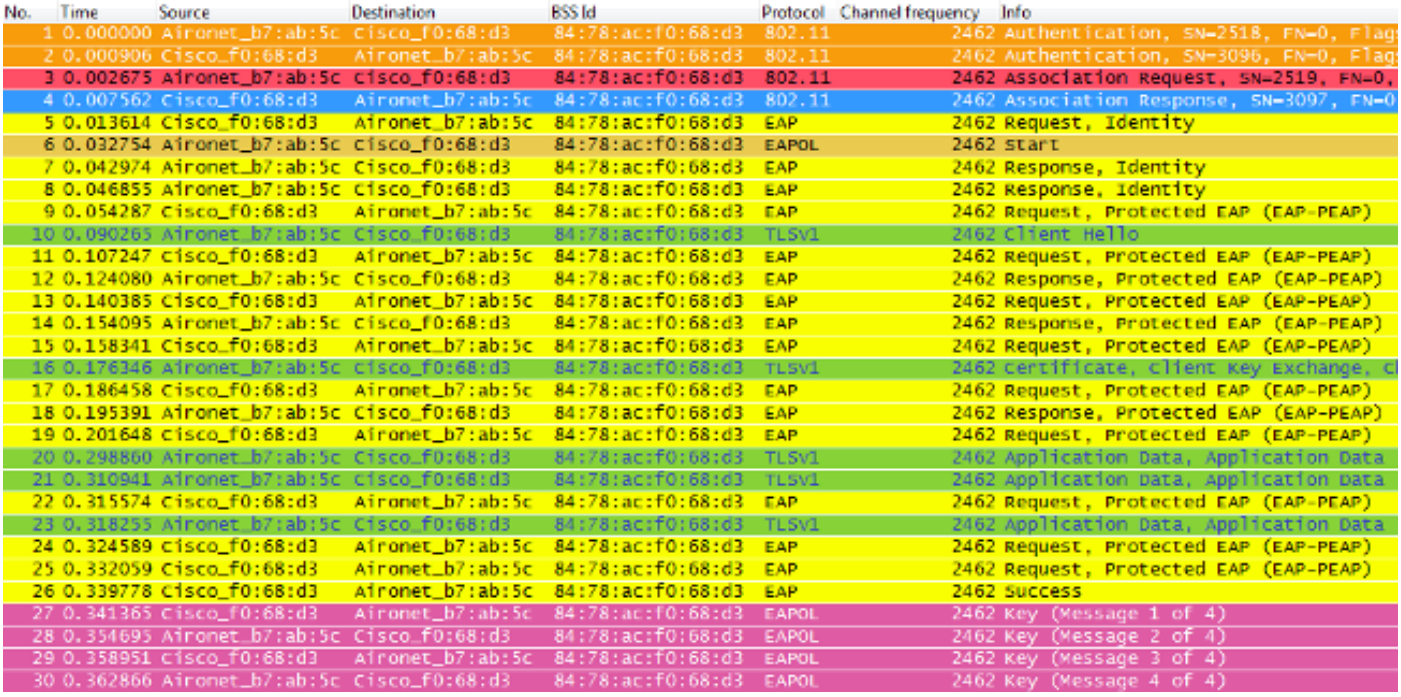
通过CCKM实现快速安全漫游

思科集中密钥管理(CCKM)是在企业WLAN上开发和实施的第一个快速安全漫游方法，由思科创建，作为在WLAN上使用802.1X/EAP安全时用于缓解迄今为止所解释的延迟的解决方案。由于这是思科专有协议，因此只有思科WLAN基础设施设备和无线客户端（来自多个供应商）支持CCKM的思科兼容扩展(CCX)。

CCKM可通过可用于WLAN的所有不同加密方法实施，包括：WEP、TKIP和AES。大多数用于WLAN的802.1X/EAP身份验证方法也支持此功能，具体取决于设备支持的CCX版本。

> **注意**：有关不同版本的CCX规范（包括支持的EAP方法）支持的功能内容的概述，请参阅CCX版本和功能文档，并验证无线客户端所支持的精确CCX版本（如果它们与CCX兼容），以便您可以确认是否可实施您希望用于CCKM的安全方法。

此无线映像提供初次关联时交换的帧的示例，当您执行CCKM时，使用TKIP作为加密，使用PEAPv0/EAP-MSCHAPv2作为802.1X/EAP方法。这基本上与执行PEAPv0/EAP-MSCHAPv2的WPA/TKIP交换相同，但这次协商客户端和基础设施之间的CCKM，以便客户端在必须漫游时使用不同的密钥层次结构和缓存方法来执行快速安全漫游：



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)

以下是调试消息的摘要（删除了某些EAP交换以减少输出）：

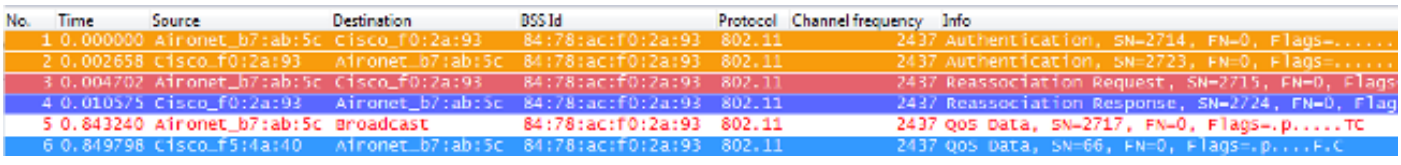
```
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d3
!--- This is the Association Request from the client.

*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.
```

使用CCKM时，与WLAN的初始关联类似于常规WPA/WPA2，其中MSK(在此也称为网络会话密钥(NSK))与客户端和RADIUS服务器相互派生。在身份验证成功后，此主密钥从服务器发送到WLC，并缓存为派生所有后续密钥的基础，以便在客户端与此WLAN关联的生存期内使用。从这里，WLC和客户端派生用于基于CCKM的快速安全漫游的种子信息，这将通过类似于WPA/WPA2的4次握手，以便派生带有第一个AP的单播(PTK)和组播/广播(GTK)加密密钥。

在漫游时，我们注意到最大的差异。在这种情况下，CCKM客户端向AP/WLC发送一个重新关联请求帧（包括MIC和按顺序递增的随机数），并提供足够的信息(包括新的AP MAC地址 — **BSSID-**)以派生新的PTK。有了此重新关联请求，WLC和新AP也有足够的信息来派生新的PTK，因此它们仅使用重新关联响应进行响应。客户端现在可以继续传递流量，如下图所示：



以下是此漫游事件的WLC调试摘要：

```
*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.
```

如图所示，由于新加密密钥仍然派生，但基于CCKM协商方案，因此可以在避免EAP身份验证帧和更多四向握手的同时执行快速安全漫游。这通过漫游重新关联帧以及客户端和WLC以前缓存的信息完成。

采用CCKM的FlexConnect

- 支持集中身份验证。这包括本地和中央数据交换。AP必须属于同一FlexConnect组。
- 支持Flex Local身份验证。在连接模式下，缓存可以从AP分配到控制器，然后分配到FlexConnect组中的其他AP。
- 支持独立模式。如果AP上已经存在缓存（由于以前的分发），则快速漫游将起作用。独立模式下的新身份验证不支持快速安全漫游。

使用CCKM的专业人士

- CCKM是最快的快速安全漫游方法，主要部署在企业WLAN上。 当在AP之间移动时，客户端不需要通过密钥管理握手来获取新密钥，并且在此WLAN的客户端生存期内无需再次对新AP执行完整的802.1X/EAP身份验证。
- CCKM支持802.11标准中可用的所有加密方法（WEP、TKIP和AES），以及一些仍在旧版客户端上使用的旧版思科专有方法。

CCKM的缺点

- CCKM是思科专有方法，它限制了对思科WLAN基础设施和CCX无线客户端的实施和支持。
- CCX第5版并未广泛采用，因此许多CCX无线客户端都不支持采用WPA2/AES的CCKM（主要是因为大多数无线客户端已经支持采用WPA/TKIP的CCKM，这仍然非常安全）。

通过PMKID缓存/粘滞密钥缓存实现快速安全漫游

成对思考密钥ID(PMKID)缓存或粘滞密钥缓存(SKC)是IEEE 802.11标准在802.11i安全修订版中建议的第一个快速安全漫游方法，其主要目的是为WLAN标准化高水平的安全性。此快速安全漫游技术已添加为WPA2设备的可选方法，以便在实施此安全性时改进漫游。

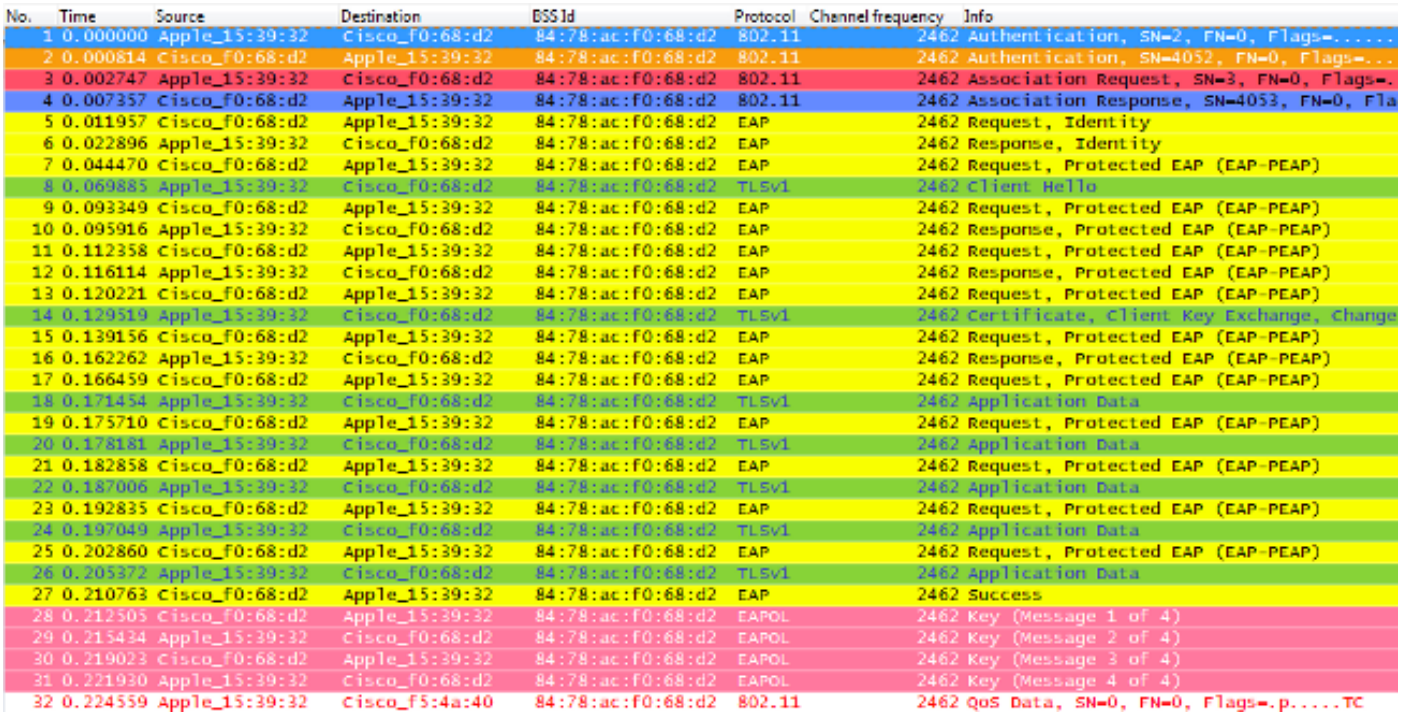
这是可能的，因为每次客户端完全通过EAP身份验证时，客户端和身份验证服务器都会派生MSK，MSK用于派生PMK。此方法用作WPA2 4次握手的种子，以生成用于会话的最终单播加密密钥(PTK)（直到客户端漫游到另一个AP或会话过期）；因此，此方法可防止漫游时的EAP身份验证阶段，因为它会重新利用客户端和AP缓存的原始PMK。客户端只需通过WPA2 4次握手即可获取新的加密密钥。

此方法没有广泛部署为推荐的802.11标准快速安全漫游方法，主要是由于以下原因：

- 此方法是可选的，并非所有WPA2设备都支持此方法，因为802.11i修正案的目的不涉及快速安全漫游，而IEEE已制定另一项修正案来标准化WLAN的快速安全漫游（本文档稍后将介绍802.11r）。
- 这种方法在实施方面存在很大的限制：无线客户端只有在漫游回他们之前已验证/连接的AP时才能执行快速安全漫游。

使用此方法，与任何AP的初始关联就像对WLAN的常规首次身份验证，其中针对身份验证服务器的

整个802.1X/EAP身份验证和密钥生成的4次握手必须在客户端能够发送数据帧之前进行，如下屏幕图像所示：



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

调试显示与对WLAN进行初始身份验证时其他方法相同的EAP身份验证帧交换，并添加了一些关于此处使用的密钥缓存技术的输出。这些debug输出被剪切，以便主要显示新信息，而不是整个EAP帧交换，因为每次交换基本相同的信息用于客户端对身份验证服务器的身份验证。这一点目前已经过演示，并与数据包图像中显示的EAP身份验证帧相关联，因此为了简单起见，大多数EAP消息已从调试输出中删除：

```
*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.
```

使用此方法，AP和无线客户端会缓存已建立的安全关联的PMK。因此，如果无线客户端漫游到从未关联的新AP，客户端必须再次执行完整的EAP身份验证，如以下图所示，客户端漫游到新AP:

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

但是，如果无线客户端漫游回发生先前关联/身份验证的AP，则客户端会发送列出多个PMKID的重新关联请求帧，该帧会向AP通知从客户端之前已进行身份验证的所有AP缓存的PMK。因此，由于客户端正在漫游回也为此客户端缓存了PMK的AP，因此客户端不需要通过EAP重新身份验证以派生新的PMK。客户端只需通过WPA2 4次握手即可获取新的临时加密密钥：

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)

注：此图像不显示来自客户端的第一个802.11开放系统身份验证帧，但是这不是由于实施的方法所致，因为始终需要此帧。原因在于，此特定帧未由用于嗅探本示例无线帧的适配器或无线数据包映像软件进行成像，但出于教育目的，本示例中保留此帧。请注意，执行无线数据包图像时可能会发生这种情况；图像可能会错过某些帧，但实际上这些帧是在客户端和AP之间交换的。否则，漫游不会在此示例中启动。

以下是此快速安全漫游方法的WLC调试摘要：

```
*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与PMKID缓存/粘滞密钥缓存

- 当您在FlexConnect设置上使用此方法时，它可能会起作用，并且行为类似于先前介绍的将中央身份验证用回WLC（使用中央或本地交换）；但是，FlexConnect不支持此SKC方法。
- 此方法仅在具有本地模式AP的CUWN上得到正式支持，但在FlexConnect或其他模式上不受支持。

使用PMKID缓存/粘滞密钥缓存的优势

此方法可由自主独立AP在本地实施，无需集中设备来管理缓存的密钥。

PMKID缓存/粘滞密钥缓存的缺点

- 如本文档前面所述，此方法的主要限制是，客户端在漫游回之前关联/身份验证的AP时只能执行快速安全漫游。如果漫游到新的AP，客户端必须再次完成完整的EAP身份验证。
- 无线客户端和AP必须记住每次新身份验证时派生的所有PMK，因此此功能通常限制为缓存一定

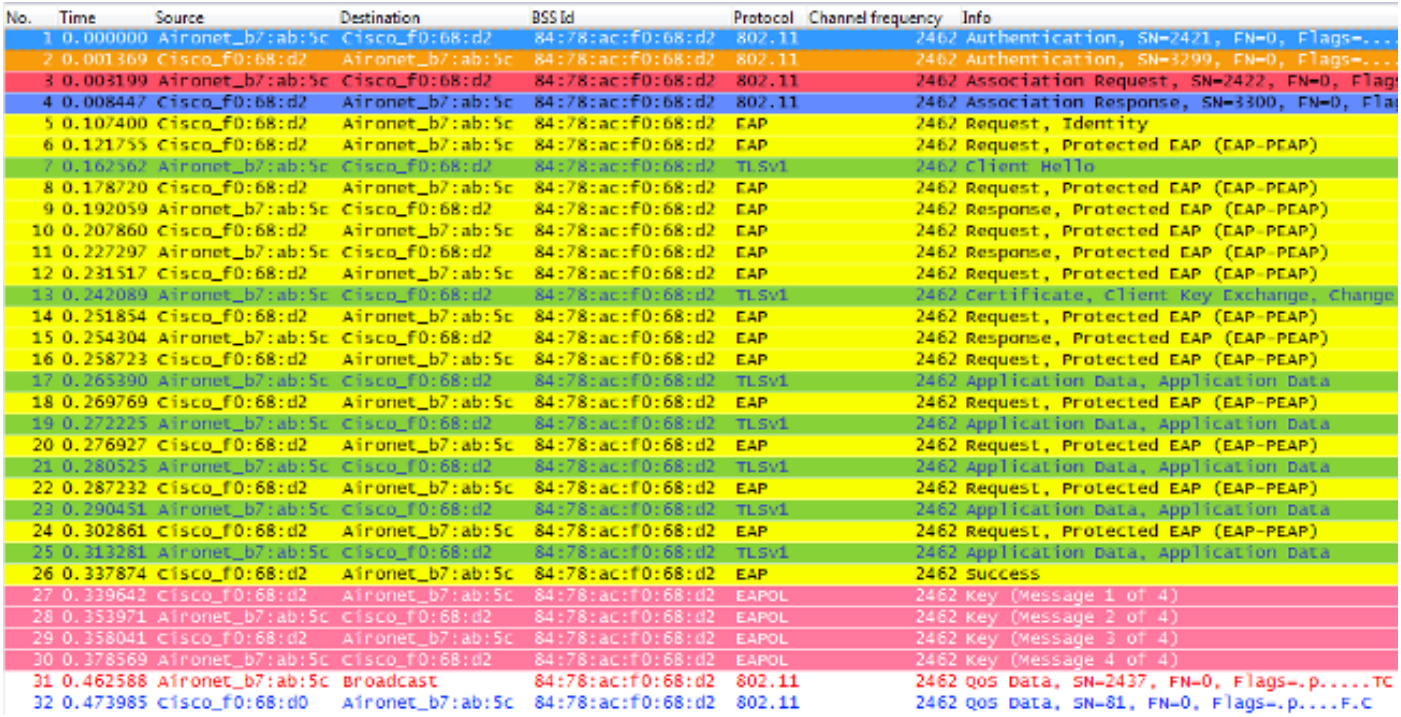
数量的PMK。由于此限制未在该标准中明确定义，因此供应商可对其实施SKC定义不同的限制。例如，思科WLAN控制器当前可以从客户端缓存最多8个AP的PMK。如果客户端漫游到每个会话中超过8个AP，最早的AP将从缓存列表中删除，以存储新缓存的条目。
- 此方法是可选的，许多WPA2设备仍不支持此方法；因此，此方法未被广泛采用和部署。
- 当您执行控制器间漫游时，不支持SKC，当您在不同的WLC管理的AP之间移动时，即使这些AP位于同一个移动组上，也会发生这种情况。

通过机会密钥缓存实现快速安全漫游

机会性密钥缓存(OKC)，也称为主动式密钥缓存(PKC)（这一术语将在接下来的一篇注释中详细介绍）基本上是对前面所述的WPA2 PMKID缓存方法的增强，这也是它也被称为主动/机会性PMKID缓存的原因。因此，请注意，这不是由802.11标准定义的快速安全漫游方法，并且许多设备不支持此方法，但与PMKID缓存一样，它与WPA2-EAP配合使用。

此技术允许无线客户端和WLAN基础设施在客户端与此WLAN关联的生存期内仅缓存一个PMK（从与身份验证服务器进行初始802.1X/EAP身份验证之后的MSK派生），即使在多个AP之间漫游时也是如此，因为它们都共享原始PMK，该PMK用作所有WPA2 4次握手中的种子。就像在SKC中一样，这仍然是必需的，以便在每次客户端重新与AP关联时生成新的加密密钥。要使AP共享来自客户端会话的这个原始PMK，它们必须都处于某种管理控制之下，并配备一台集中式设备，该设备会缓存原始PMK并将其分配给所有AP。这类似于CUWN，其中WLC对其控制下的所有LAP执行此作业，并使用移动组在多个WLC之间处理此PMK；因此，这是对自治AP环境的限制。

使用此方法，就像在PMKID缓存(SKC)中一样，与任何AP的初始关联都是对WLAN的常规首次身份验证，在此过程中，您必须根据身份验证服务器完成整个802.1X/EAP身份验证，并完成密钥生成的4次握手，然后才能发送数据帧。下面是一个屏幕图像，说明了这一点：



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, Flag
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, Flag
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

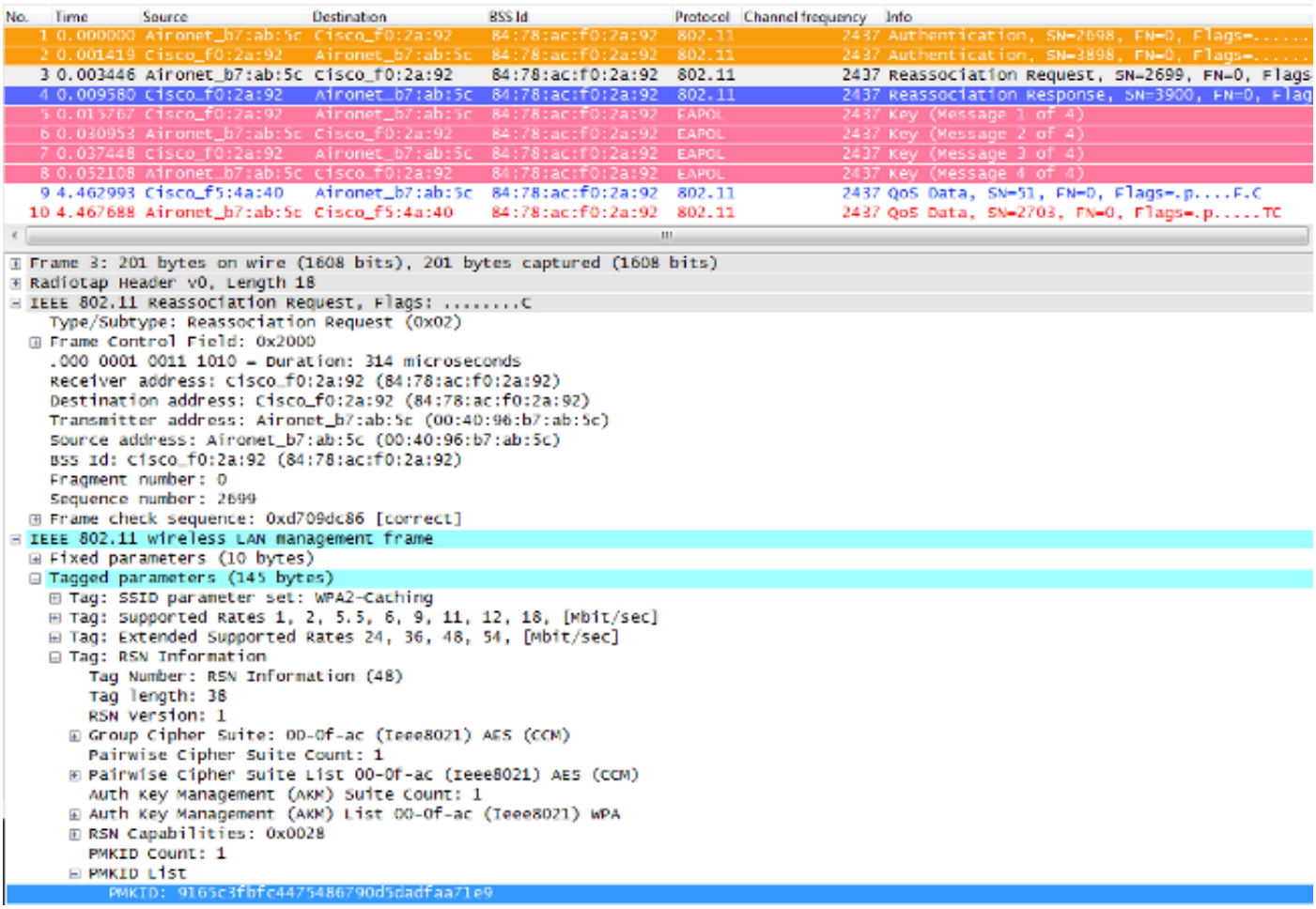
debug输出显示基本与WLAN初始身份验证时本文档中介绍的其余方法相同的EAP身份验证帧交换（如图所示），以及一些与WLC在此处使用的关键缓存技术相关的输出。此调试输出也将被剪切，以便仅显示相关信息：

```
*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
```

通过此方法，无线客户端和WLC（对于所有托管AP）缓存最初建立的安全关联的一个原始PMK。基本上，每次无线客户端连接到特定AP时，都会根据以下内容对PMKID进行哈希处理：客户端MAC地址、AP MAC地址（WLAN的BSSID）以及派生自该AP的PMK。因此，由于OKC为所有AP和特定客户端缓存相同的原始PMK，因此当此客户端（重新）关联到另一个AP时，为散列新PMKID而更改的唯一值是新的AP MAC地址。

当客户端向新AP发起漫游并发送重新关联请求帧时，如果它要通知AP缓存PMK用于快速安全漫游，则会在WPA2 RSN信息元素上添加PMKID。它已知道漫游位置的BSSID(AP)的MAC地址，则客户端只需散列此重新关联请求上使用的新PMKID。当AP收到来自客户端的此请求时，它还会使用已经有的值（缓存的PMK、客户端MAC地址及其自己的AP MAC地址）对PMKID进行哈希处理，并响应确认匹配的PMKID的重新关联响应成功。缓存PMK可用作启动WPA2 4次握手的种子，以生成新的加密密钥（并跳过EAP）：



在此图像中，来自客户端的“重新关联请求”帧被选中并展开，以便您可以看到该帧的更多详细信息。根据802.11i - WPA2)信息元素，MAC地址信息以及稳健安全网络(RSN)，其中显示了用于此关联的WPA2设置的相关信息（突出显示的是从散列公式获取的PMKID）。

以下是此带OKC的快速安全漫游方法的WLC调试摘要：

```
*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
```

如调试开始时所示，必须在收到来自客户端的重新关联请求后计算PMKID。验证PMKID并确认缓存的PMK与WPA2 4次握手配合使用以生成加密密钥并完成快速安全漫游，需要此步骤。请勿混淆调试上的CCKM条目；如前所述，这不是用于执行CCKM，而是OKC。这里的CCKM只是WLC用于这些输出的名称，例如处理值以计算PMKID的函数的名称。

带有机会性密钥缓存的FlexConnect

- 支持集中身份验证。这包括本地和中央数据交换。如果AP属于同一FlexConnect组，则AP将执行快速安全漫游，否则控制器将执行快速安全漫游。
  注：如果AP不在同一FlexConnect组中，则此设置可以运行，但这不是推荐的或支持的设置。
- 支持Flex Local身份验证。在连接模式下，缓存可以从AP分配到控制器，然后分配到FlexConnect组中的其他AP。
- 支持独立模式。 如果AP上已存在缓存（由于先前的分发），快速安全漫游将起作用。独立模式下的新身份验证不支持快速安全漫游。

随机密钥缓存的优势

- 无线客户端和WLAN基础设施不需要记住多个PMKID，只需将初始身份验证中的一个原始PMK缓存到WLAN即可。然后，您必须对每个AP安全关联所需的正确PMKID（用于重新关联请求）进行重新散列，以验证快速安全漫游。
- 在这里，无线客户端对同一WLAN/SSID上的新AP执行快速安全漫游，即使它从未与该AP关联（SKC的情况并非如此）。只要客户端使用一个由集中部署管理的AP执行初始802.1X/EAP身份验证，该集中部署为客户端漫游的所有AP处理PMK缓存，则此WLAN上的剩余客户端生命周期不需要更多完全身份验证。

使用机会性密钥缓存时的缺点

- 此方法仅部署在集中式环境中，其中所有AP都处于某种管理控制下（例如WLAN控制器），该管理控制负责缓存和共享来自客户端会话的一个原始PMK。因此，这是对自治AP环境的限制。
- 802.11标准未建议或描述这种方法中所使用的技术，因此不同设备之间的支持差异很大。然而，这种方法在等待802.11r时仍被更多地采用。

关于术语“主动密钥缓存”的说明

主动密钥缓存（Proactive Key Caching，简称PKC）也称为OKC（Opportunistic Key Caching，简称OKC），这两个术语在描述此处介绍的相同方法时可以互换使用。但是，这只是Airspace在2001年用于旧密钥缓存方法的术语，802.11i标准随后将其用作“预身份验证”（另一个快速安全漫游方法将在下面简要说明）的基础。PKC不是Preauthentication或OKC(Opportitional Key Caching)，但当您听到或读到PKC时，参考基本上是OKC，而不是预身份验证。

使用预身份验证的快速安全漫游

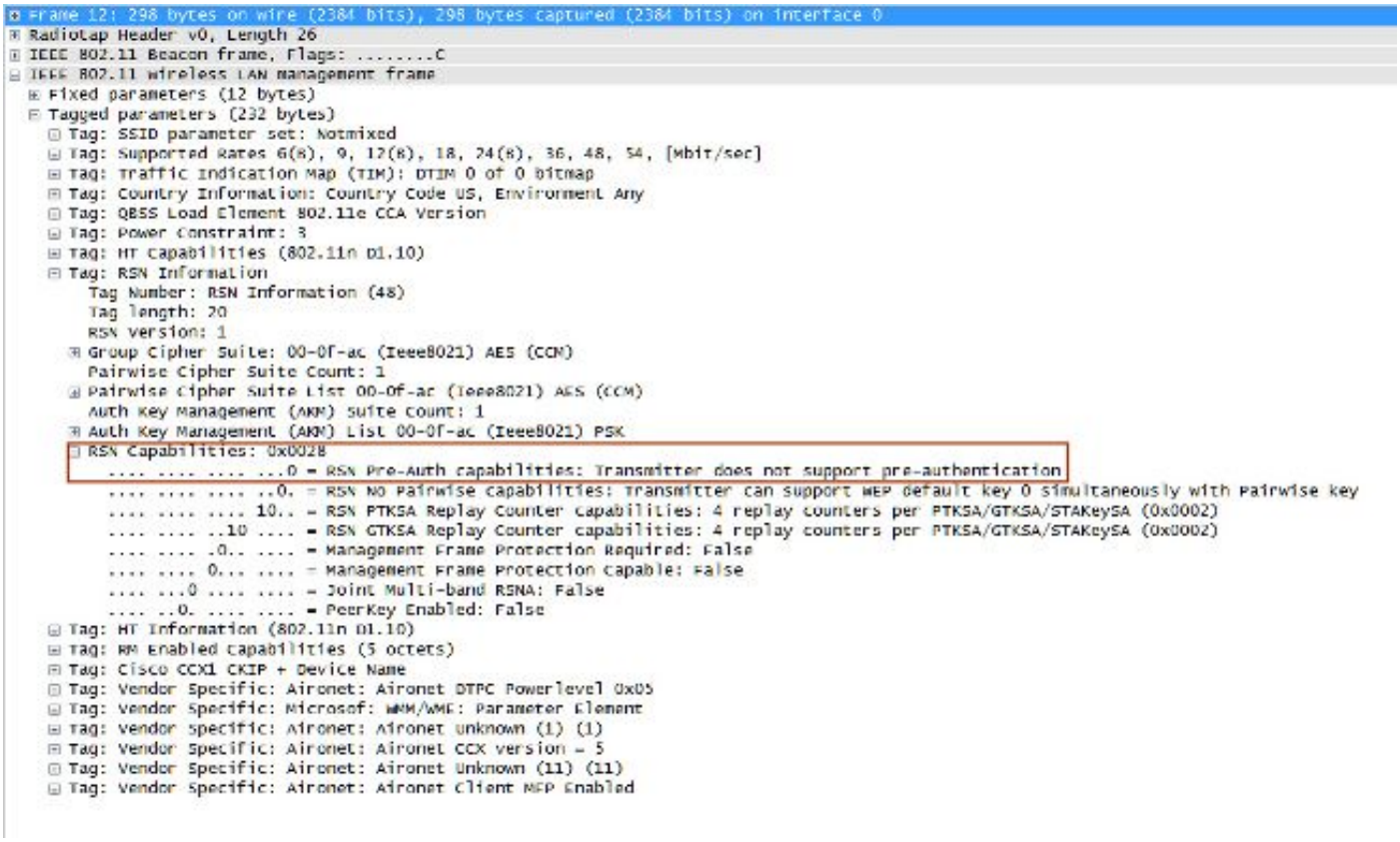
802.11i安全修订版中的IEEE 802.11标准也建议使用此方法，因此此方法也适用于WPA2，但它是思科WLAN基础设施不支持的唯一快速安全漫游方法。因此，这里仅作简要解释，不作输出。

使用预身份验证，无线客户端可以在与当前AP关联的同时一次对多个AP进行身份验证。发生这种情况时，客户端会将EAP身份验证帧发送到其连接的当前AP，但将其发往客户端希望执行预身份验证的其他AP（邻居AP可能是漫游的候选者）。当前AP通过分布系统将这些帧发送到目标AP。新

AP针对此客户端的RADIUS服务器执行完全身份验证，因此整个新EAP身份验证握手已完成，并且此新AP充当身份验证器。

其思路是在客户端实际漫游到相邻AP之前执行身份验证并派生PMK，因此当漫游时，客户端已经通过身份验证，并且已缓存一个PMK以用于新的AP到客户端安全关联，因此客户端只需执行4次握手，并在客户端发送其初始重新关联请求后体验快速漫游。

以下是来自AP信标的图像，其中显示通告支持预身份验证的RSN IE字段（此图像来自Cisco AP，其中确认不支持预身份验证）：



具有预身份验证的优点

每个AP到客户端安全关联都有一个PMK，当AP受到威胁且密钥被窃取（无法与其他AP一起使用）时，可以将其视为安全优势。但是，WLAN基础架构在其他方法上以不同方式处理此安全优势。

带有预身份验证的缺点

- 因为每个AP有一个PMK，所以客户端对可以预先进行身份验证的AP数量有限制。
- 每次客户端使用新的AP执行预身份验证时，都会有完整的EAP身份验证交换，这意味着网络和身份验证服务器上的负载会增加。
- 大多数无线客户端不支持此方法，因为它从未被高度采用（OKC更多地被采用）。

802.11r快速安全漫游

基于802.11r修正案的快速安全漫游技术(由802.11标准正式命名为**Fast BSS Transition**，称为**FT**)是IEEE正式批准（2008年）的第一种802.11标准方法，它作为在AP（基本服务集或BSS）之间执行快速转换的解决方案，它明确定义了在WLAN上处理密钥和缓存密钥时使用的密钥层次结构。但是，它的采用一直很慢，这主要是因为在确实需要快速转换时已经提供了其他解决方案，例如与本文

档前面介绍的方法之一一起使用时的VoWLAN实施。目前只有少数设备支持某些FT选项（到2013年）。

与其他方法相比，此技术解释起来更为复杂，因为它引入了缓存到不同设备（每个设备具有不同的角色）上的新概念和多层PMK，并且提供了更多快速安全漫游选项。因此，我们将简要介绍此方法以及每种可用选项的实施方式。

802.11r与SKC和OKC不同，主要是因为以下原因：

- 握手消息传送（例如PMKID、ANonce和SNonce交换）在802.11身份验证帧或操作帧中发生，而不是重新关联帧。不同于PMKID缓存方法，避免在（重新）关联消息交换之后进行单独的4次握手阶段。与新AP的密钥握手在客户端完全漫游/重新关联此新AP之前开始。
- 它为快速漫游握手提供了两种方法：通过AIR和分布式系统(DS)。
- 802.11r具有更多层密钥层次结构。
- 由于此协议避免了在客户端漫游时进行密钥管理的4次握手（无需此握手即可生成新的加密密钥 — PTK和GTK），因此它还可以应用于具有PSK的WPA2设置，而不仅仅是在使用802.1X/EAP进行身份验证时。对于不发生EAP或4路握手交换的设置，这会进一步加速漫游。

通过这种方法，当与第一个AP建立连接时，无线客户端仅对WLAN基础设施执行一次初始身份验证，并在同一FT移动域的AP之间漫游时执行快速安全漫游。

这是新概念之一，它基本上是指使用相同SSID（称为扩展服务集或ESS）并处理相同FT密钥的AP。这与迄今为止介绍的其他方法类似。AP处理FT移动域密钥的方式通常基于集中设置，例如WLC或移动组；但是，此方法也可以在自治AP环境中实施。

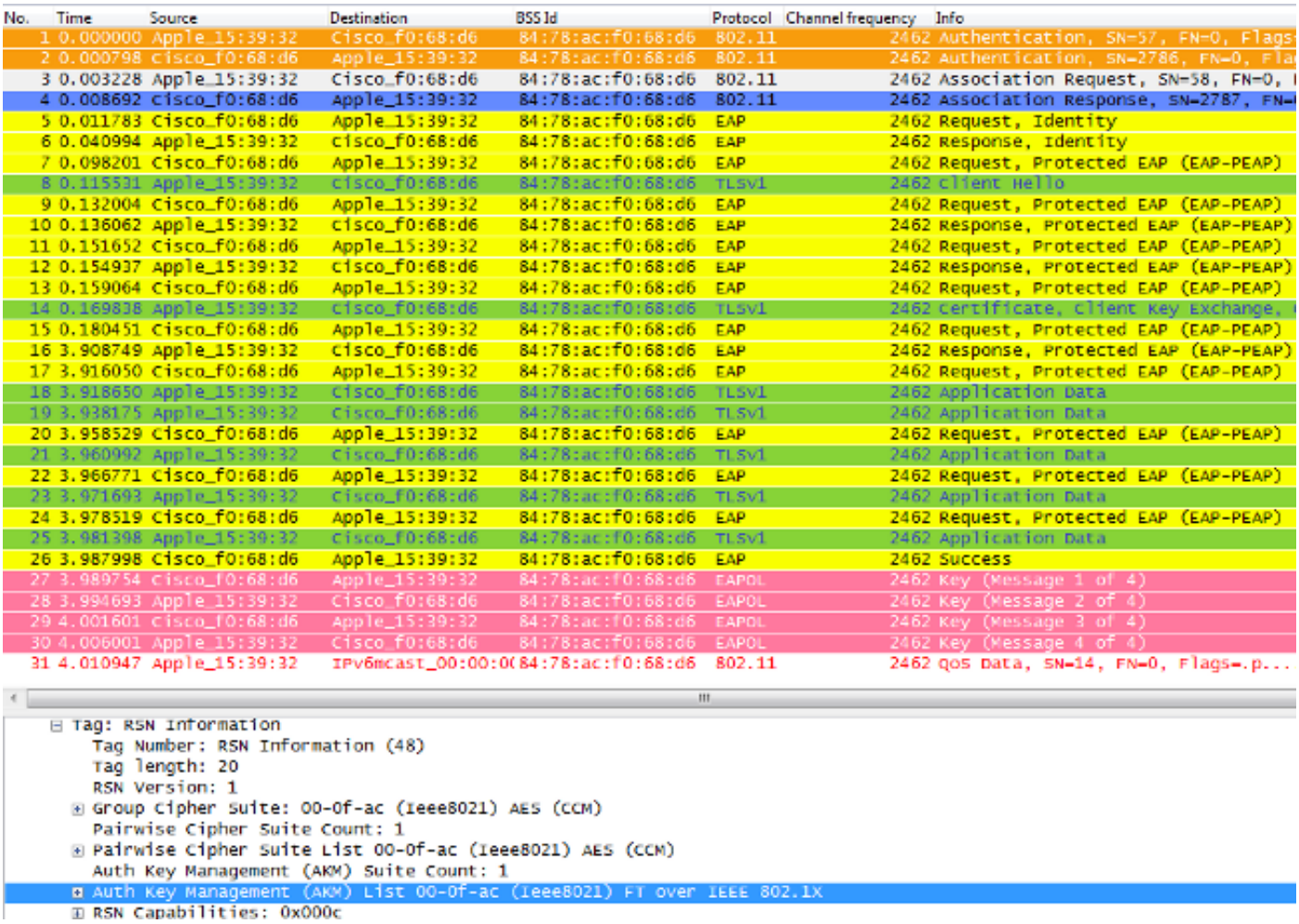
以下是关键层次结构的摘要：

- MSK仍然从初始802.1X/EAP身份验证阶段(身份验证成功后，从身份验证服务器传输到身份验证器(WLC))派生到客户端请求方和身份验证服务器。与其它方法一样，此MSK用作FT密钥层次结构的种子。当您使用WPA2-PSK而非EAP身份验证方法时，PSK基本上是此MSK。
- 成对主密钥R0(PMK-R0)从MSK派生，MSK是FT密钥层次结构的第一级密钥。此PMK-R0的密钥持有者是WLC和客户端。
- 第二级密钥称为成对主密钥R1(PMK-R1)，从PMK-R0派生，密钥持有者是客户端和由WLC管理、持有PMK-R0的AP。
- FT密钥层次结构的第三级也是最终级密钥是PTK，它是用于加密802.11单播数据帧的最终密钥（类似于使用WPA/TKIP或WPA2/AES的其他方法）。此PTK在FT上衍生自PMK-R1，密钥持有者为客户端和WLC管理的AP。

  注：根据WLAN供应商和实施设置（例如自治AP、FlexConnect或网状网），WLAN基础设施可以采用不同的方式传输和处理密钥。它甚至可以更改密钥持有者的角色，但由于这不在本文档的讨论范围之内，因此下面重点介绍基于前面给出的密钥层次结构汇总的示例。这些差异实际上与了解流程无关，除非您实际上需要深入分析基础设施设备（及其代码）以发现软件问题。

空中BSS快速过渡

使用此方法时，与任何AP的第一个关联是对WLAN的常规首次身份验证，其中针对身份验证服务器的整个802.1X/EAP身份验证和密钥生成的4次握手必须在发送数据帧之前进行，如下屏幕图像所示：



主要区别如下：

- 身份验证密钥管理协商与常规WPA/WPA2略有不同，因此当与支持FT的WLAN基础设施发生关联时，会使用一些额外信息来执行此协商。如图所示，选择来自客户端的关联请求帧，并突出显示RNS信息元素的AKM字段，以表明此客户端希望执行802.1X/EAP上的FT。
- 还显示了移动域信息元素（FT的一部分），其中**FT Capability and Policy**字段指示在快速漫游时快速BSS转换是通过Over-the-Air还是Over-the-DS完成（在此映像中指示Over-the-Air）。
- 还会添加另一个信息元素（快速BSS过渡或FT IE，本文档稍后将进行说明）以在FT漫游时执行FT身份验证序列所需的信息。
- 密钥生成因密钥层次结构而异，因此，即使FT四向握手看起来类似于WPA/WPA2四向握手，其内容实际上略有不同。

调试显示基本与WLAN初始身份验证时其他方法相同的EAP身份验证帧交换（如图所示），但添加了与WLC使用的密钥缓存技术相关的一些输出；因此，剪切此调试输出以仅显示相关信息：

```
*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
```

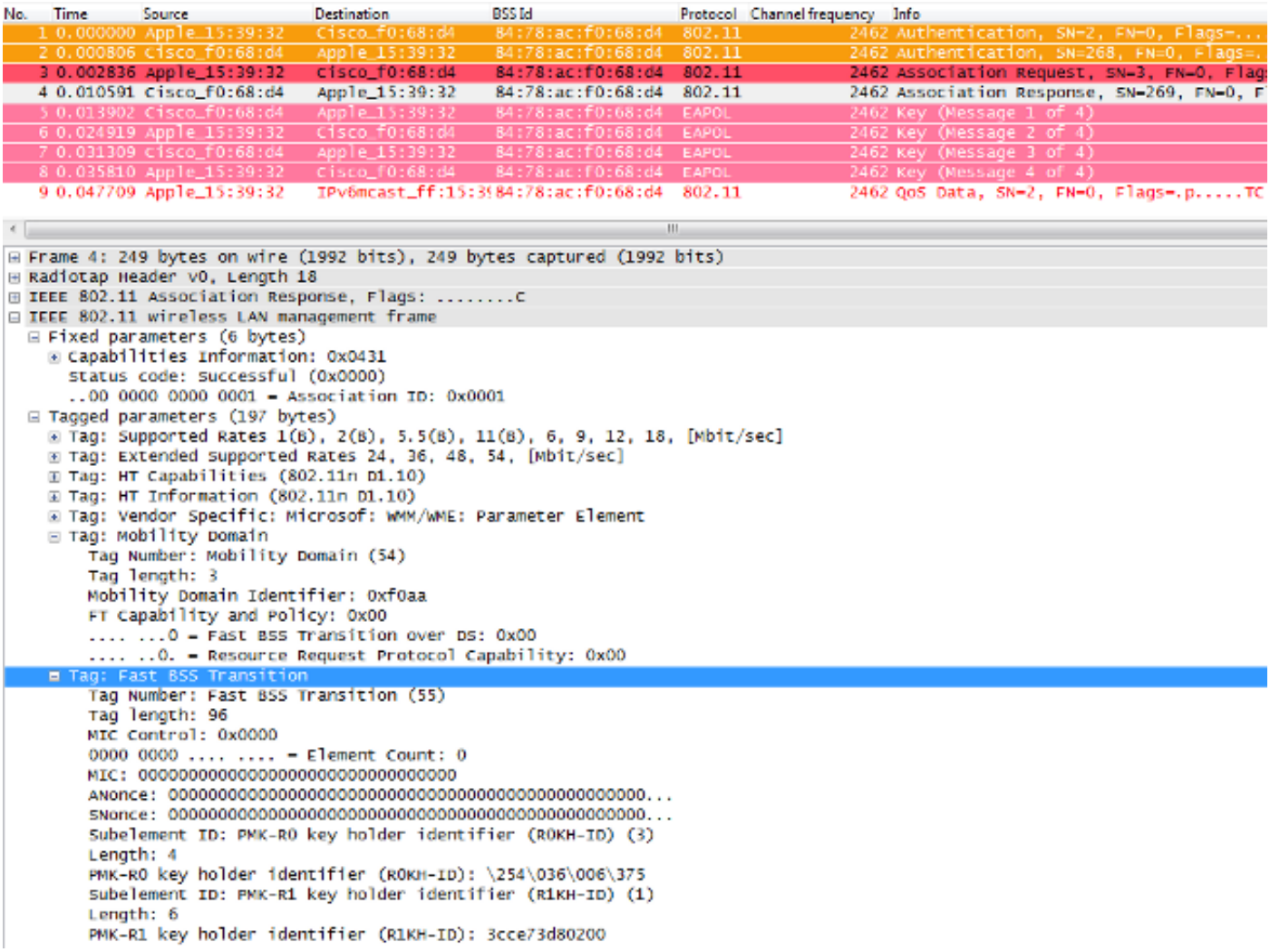
```
!--- The WLC/AP finds an Information Element that claims FT
      support on the Association request that is sent from the client.

*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.
```

注：为了调试此方法并达到此处所示的额外802.11r/FT输出，将启用附加调试以及**debug client**，即**debug ft events enable**。

以下是当您使用WPA2-PSK（而不是802.1X/EAP方法）执行FT时，与WLAN的初始关联的映像和调试，其中会选择AP的关联响应帧以显示快速BSS转换信息元素（突出显示）。执行FT 4次握手所需的一些关键信息也显示如下：



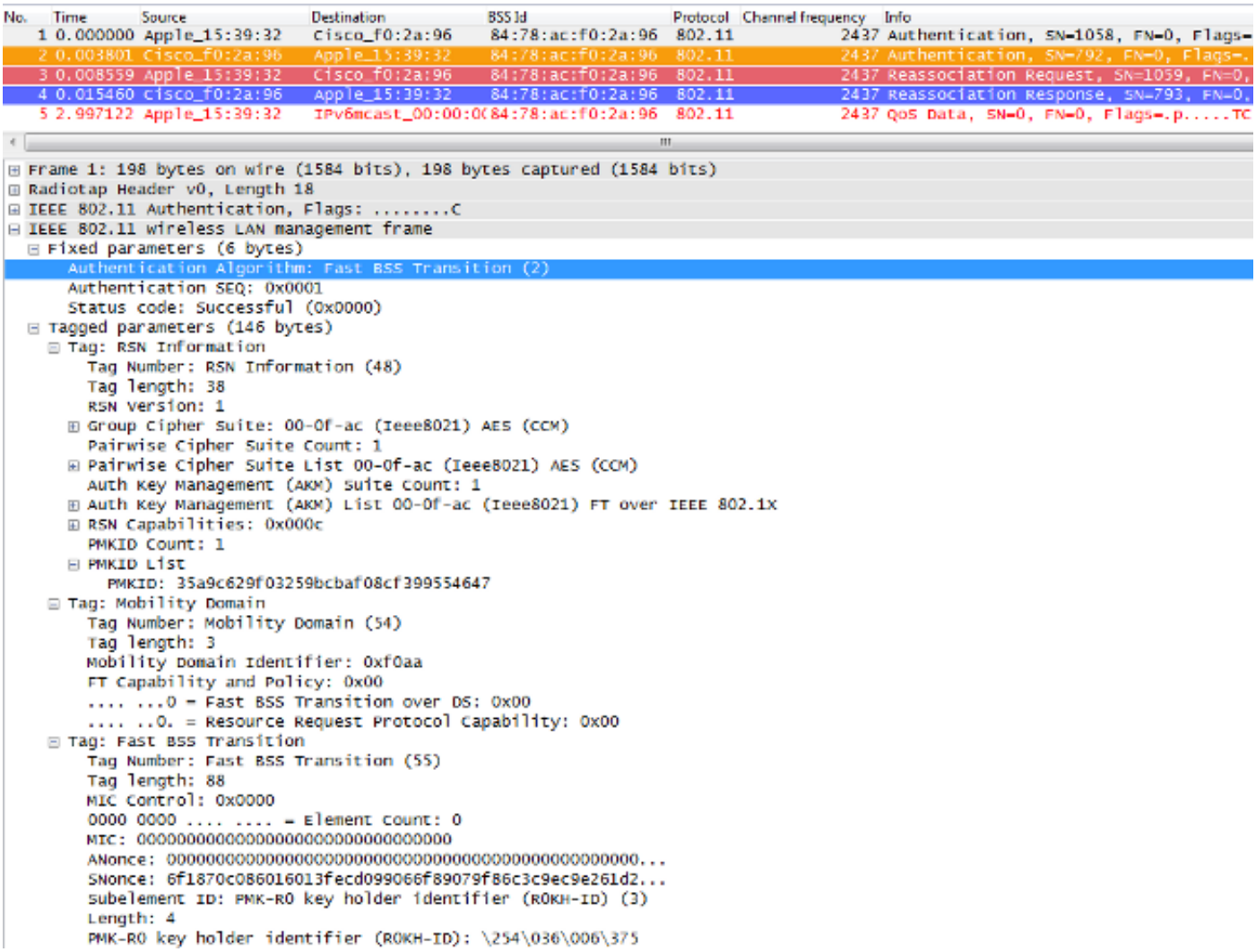

```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*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
```

与802.11r一样，与WLAN的初始关联是获取此技术使用的基本密钥的基础，其他快速安全漫游方法也是如此。主要区别在于客户端开始漫游时；FT不仅在使用时会避免802.1X/EAP，而且实际上会执行更高效的漫游方法，将初始802.11开放系统身份验证和重关联帧（在AP之间漫游时始终使用并且需要这些帧）合并起来，以便交换FT信息并衍生新的动态加密密钥来代替4次握手。

下图显示当执行具有802.1X/EAP安全的快速BSS空中转换时交换的帧。选择从客户端到AP的开放式系统身份验证帧，以便查看开始FT密钥协商所需的FT协议信息元素。此命令用于使用新AP（基于PMK-R1）生成新PTK。突出显示显示身份验证算法的字段，以显示此客户端不执行简单的开放系统身份验证，而是执行快速BSS转换：

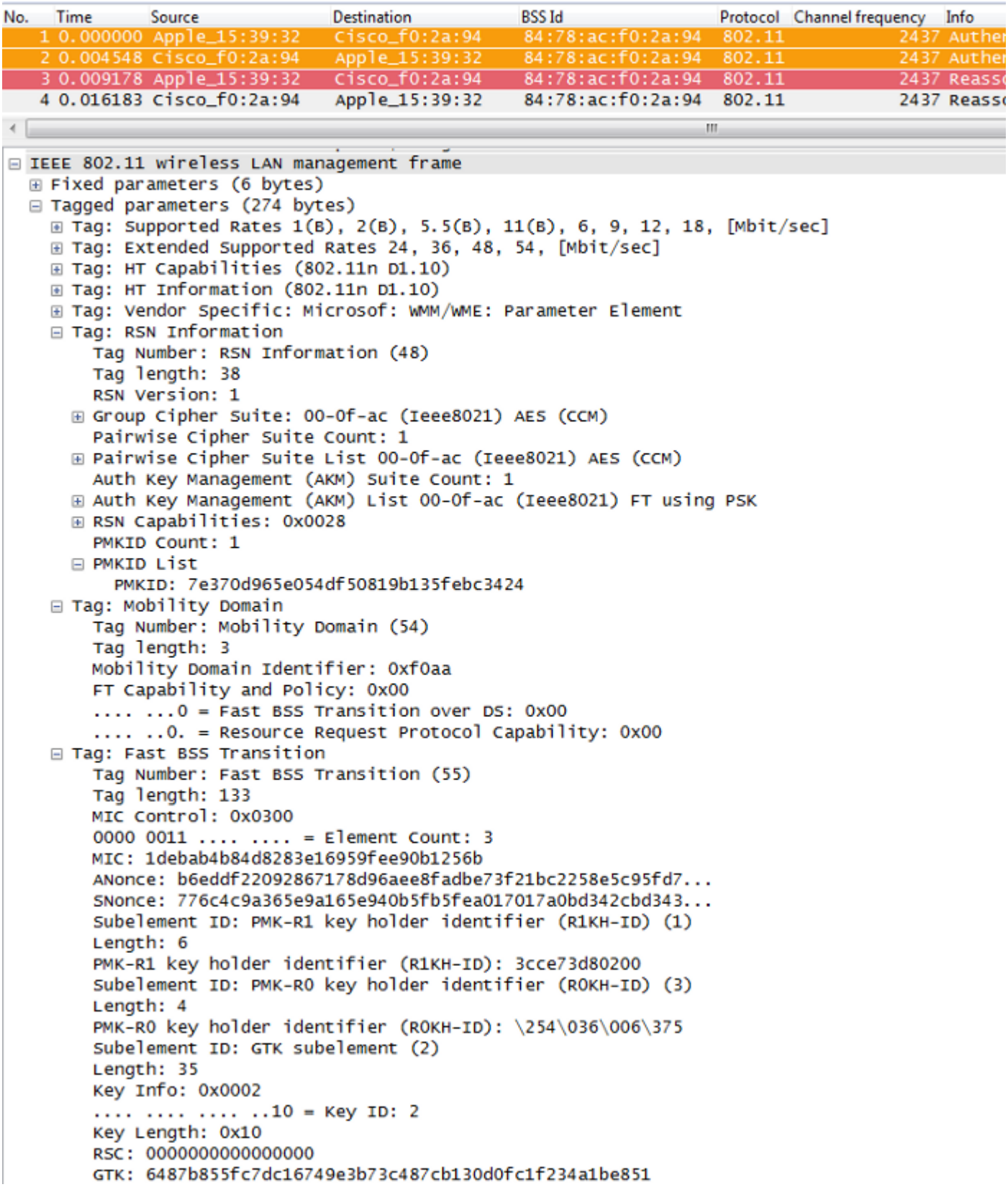


以下是当802.1X/EAP发生此FT漫游事件时WLC的调试输出：

```
*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
```

下图显示了具有WPA2-PSK安全性的Fast BSS Transition Over-the-Air，其中选择了从AP到客户端的最终Reassociation Response帧，以显示此FT交换的更多详细信息：



以下是使用PSK发生此FT漫游事件时的调试输出，类似于使用802.1X/EAP时的调试输出：

```
*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
```

如图所示，在与WLAN初始关联时协商快速BSS转换后，漫游时使用和需要的四个帧（来自客户端的开放系统身份验证、来自AP的开放系统身份验证、重新关联请求和重新关联响应）基本上被用作FT四向握手，以生成新的PTK（单播加密密钥）和GTK（组播/广播加密密钥）。

这代替了交换这些帧后通常发生的4次握手，并且无论使用802.1X/EAP还是PSK作为安全方法，这些帧上的FT内容和密钥协商基本相同。如图所示，AKM字段是主要区别，用于确认客户端是否使用PSK或802.1X执行FT。因此，必须注意的是，这四个帧通常没有此类用于密钥协商的安全信息，但仅当客户端FT漫游时，才实施了802.11r并在客户端和WLAN基础设施之间进行初始关联协商。

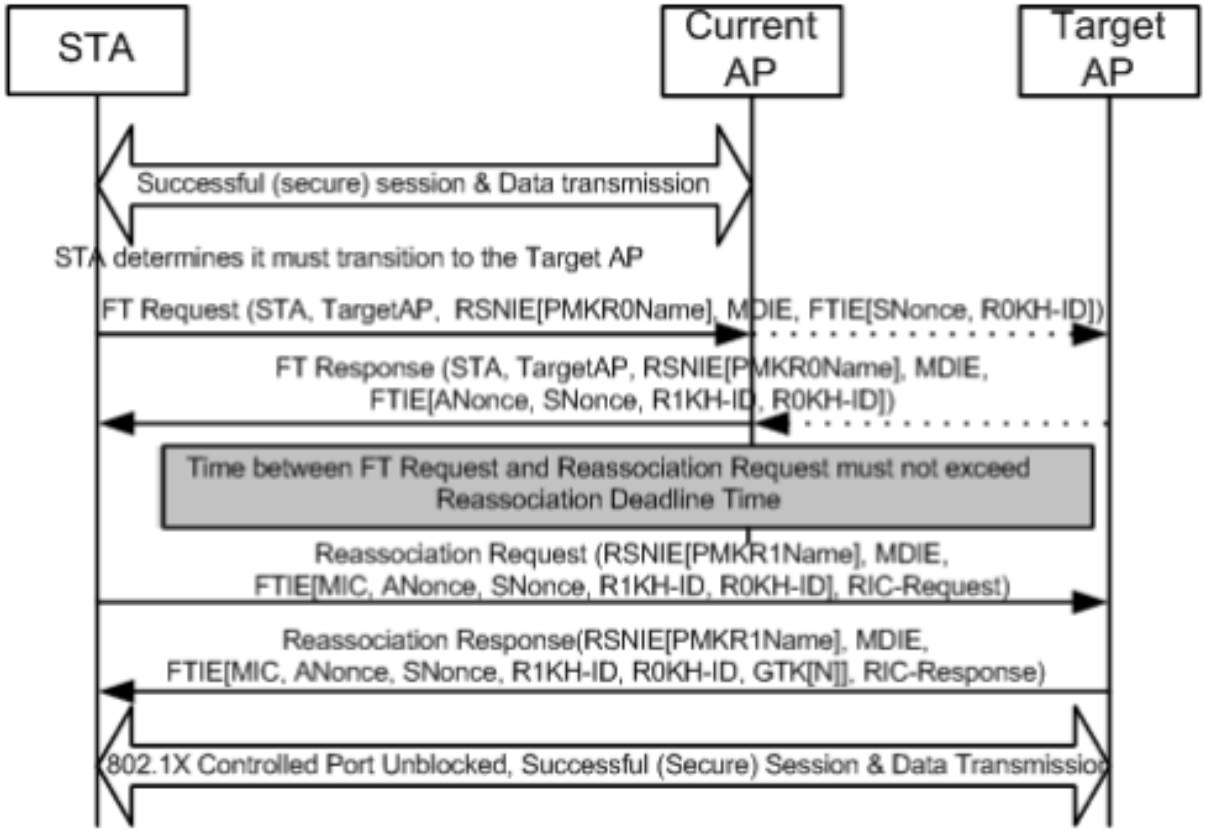
通过DS快速BSS过渡

802.11r允许实施另一种快速BSS过渡，即由客户端通过新的AP发起FT漫游，客户端通过DS（分布

系统）而非无线进行漫游。在这种情况下，使用FT Action帧来启动密钥协商，而不是开放式系统身份验证帧。

基本上，一旦客户端决定可以漫游到更好的AP，客户端就会向它当前连接的原始AP发送FT操作请求帧，然后进行漫游。客户端指示它要FT漫游的目标AP的BSSID（MAC地址）。原始AP通过分布系统（通常是有线基础设施）将此FT操作请求帧转发到目标AP，目标AP使用FT操作响应帧响应客户端（也通过DS，因此它最终可以无线发送到客户端）。此FT操作帧交换成功后，客户端完成FT漫游；客户端向目标AP发送重新关联请求（这次通过空中），并接收来自新AP的重新关联响应以确认漫游和最终密钥派生。

总之，有四个帧用于协商快速BSS转换和生成新的加密密钥，但此处开放系统身份验证帧替换为FT操作请求/响应帧，这些帧通过分布系统与当前AP交换到目标AP。此方法对802.1X/EAP和PSK安全方法也有效，所有这些方法都受Cisco无线LAN控制器支持；但是，由于WiFi行业中的大多数无线客户端不支持和实施此Over-the-DS转换（并且由于帧交换和调试输出基本相同），因此本文档不提供示例。相反，此图像用于显示Fast BSS Transition Over-the-DS:



采用802.11r的FlexConnect

- 支持集中身份验证。这包括本地和中央数据交换。AP必须属于同一FlexConnect组。
- 不支持本地身份验证。
- 不支持独立模式。

802.11r的优势

- 此方法首先使用IEEE在802.11标准上明确定义的密钥层次结构作为修正案(802.11r)，因此这些FT技术的实施在供应商之间更兼容，且没有不同的解释。
- 802.11r根据您的需求，允许使用多种有用的技术（空中和DS上，适用于802.1x/EAP安全性和

PSK安全性）。
- 无线客户端在同一WLAN/SSID上向新AP执行快速安全漫游，即使它从未与该AP关联，也不需要保存多个PMKID。
- 这是第一种快速安全漫游方法，即使使用PSK安全也允许更快的漫游，并避免在使用WPA/WPA2 PSK的AP之间漫游时所需的4次握手。快速安全漫游方法的主要目的是在实施此安全方法时避免802.1X/EAP握手；但是，对于PSK安全，在避免4次握手时，使用802.11r会加速漫游事件。

802.11r的缺点

- 有一些无线客户端设备实际上支持快速BSS过渡，而且在大多数情况下，它们并不支持802.11r上提供的所有技术。
- 由于这些实施非常年轻，因此没有来自实际生产环境的足够测试结果或足够的调试结果来解决可能出现的警告。
- 当您配置WLAN/SSID以使用任何FT方法时，只有支持802.11r的无线客户端能够连接到此WLAN/SSID。FT设置对于客户端不是可选的，因此不支持802.11r的无线客户端必须连接到没有配置FT的独立WLAN/SSID。

自适应802.11r

- 某些旧版客户端无法与已启用802.11r的WLAN/SSID关联，即使对“混合模式”也是如此（您希望可以在支持和不支持802.11r的相同SSID客户端上拥有该功能）。这表示负责解析强大的安全网络信息元素(RSN IE)的客户端请求方的驱动程序较旧，不了解IE中的其他AKM套件。由于此限制，客户端无法向通告802.11r支持的WLAN发送关联请求，因此，您需要为802.11r客户端配置一个WLAN/SSID，为不支持802.11r的客户端配置一个单独的WLAN/SSID。
- 为了克服这一点，思科无线局域网基础设施引入了自适应802.11r功能。当FT模式在WLAN级别设置为“自适应”时，WLAN将在启用802.11i的WLAN上通告802.11r移动域ID。某些Apple iOS10客户端设备识别80211i/WPA2 WLAN上存在MDIE，并执行专有握手以建立802.11r关联。一旦客户端成功完成802.11r关联，它就可以像在正常启用802.11r的WLAN中一样进行FT漫游。FT自适应仅适用于所选的Apple iOS10（及更高版本）设备。所有其他客户端都可以继续在WLAN上进行802.11i/WPA2关联，并执行所支持的适用FSR方法。
- 有关为iOS10设备在WLAN/SSID上执行802.11r（其中802.11r未真正启用，因此其他非802.11r客户端可以成功连接）引入的这一新功能的更多文档，请参阅Cisco无线LAN上Cisco IOS设备的企业最佳实践。

结论

- 请记住，客户端始终是决定漫游到特定AP的客户端，而WLC/AP无法决定该客户端的路由。漫游事件在无线客户端认为必须漫游时由它发起。
- WLC在同一个WLAN/SSID上支持大多数或所有FSR（快速安全漫游）方法的组合。但是，请注意，这通常不起作用，因为它高度依赖于客户端行为（不同移动设备之间差异很大），以便支持甚至理解WLC尝试将哪些内容通告为受支持。通常情况下，问题比预期修复的问题多，因此不建议这样做，而不是只用一个SSID即可实现互操作性。如果确实需要执行深入测试，必须完成对所有可能用于此WLAN的客户端的深入测试。
- 如果WLAN/SSID启用了安全功能，了解开发快速安全漫游方法是为了在您在AP之间移动时加速WLAN漫游过程，这一点非常重要。没有安全措施时，无需加速，因为客户端AP只需在发送数据帧之前交换AP之间漫游时始终需要的无线管理帧（来自客户端的开放系统身份验证、来自

AP的开放系统身份验证、重新关联请求和重新关联响应）。因此，这不会移动得更快。如果您遇到没有安全性的漫游问题，则没有快速漫游的方法来改进漫游，只有方法才能确认WLAN/SSID设置和设计是否适合无线客户端站点在AP覆盖信元之间相应地漫游。

- 802.11r/FT是通过WPA2-PSK实现的，目的是通过这种安全性加速漫游事件并避免四向握手，如802.11r一节所述。
- 所有方法都有各自的优缺点，但最后，您必须始终验证无线客户端站是否支持您要实施的特定方法，以及Cisco WLAN基础设施是否支持所有可用方法。因此，您必须选择连接到特定WLAN/SSID的无线客户端实际支持的最佳方法。例如，在某些部署中，您可以为思科无线IP电话（支持带CCKM的WPA2/AES，但不支持802.11r）创建一个带CCKM的WLAN/SSID，然后为支持此快速安全漫游方法的无线客户端创建一个带WPA2/AES的WLAN/SSID（或者使用OKC，如果支持）。
- 如果无线客户端不支持任何可用的快速安全漫游方法，则您可能需要接受以下事实：当具有802.1X/EAP安全的WLAN/SSID上的AP之间漫游时，这些客户端始终可以尝试本文档中介绍的延迟（这可能会造成客户端应用/服务中断）。
- 支持除SKC（WPA2 PMKID缓存）之外的所有方法，以便在由不同WLC管理的AP之间进行快速安全漫游（控制器间漫游），只要它们位于同一移动组。
- 当802.1X/EAP身份验证用于WPA/WPA2时，CUWN完全支持本文中介绍的所有不同快速安全漫游方法。当使用PSK(WPA2-Personal)时，CUWN不支持在与WPA2-RSN配合使用的方法上进行快速安全漫游（CCKM、PMKID缓存/SKC、OKC/PKC），其中大部分不需要快速漫游方法。但是，CUWN在采用PSK的WPA2-FT(802.11r)中支持快速安全漫游，本文也对此进行了说明。

相关信息

- 802.11r BSS快速过渡部署指南
- 思科技术支持和下载

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。