802.11 无线局域网漫游和思科统一无线网络快速安全漫游

介绍

本文档介绍了思科统一无线网络（CUWN）所支持的 IEEE 802.11 无线局域网（WLAN）可用的不同类型的无线漫游和快速安全漫游方法。

本文档并不提供所有关于每种方法的工作原理和如何配置的细节。本文档的主要目的是为了说明各种可用技术的优点和局限性，以及每种方法帧交换之间的差异。提供了无线控制器的 debug 示例，并且使用抓取的无线数据包用以分析和解释每种漫游方法所发生的事件。

先决条件

要求

思科建议您已具备以下知识：
* IEEE 802.11 无线局域网基础知识
* IEEE 802.11 无线局域网安全知识
* IEEE 802.1X/EAP 基础知识

使用的组件

本文档中的信息是基于思科无线控制器软件版本 7.4，但大部分所描述的调试输出和行为描述适用于支持这些办法的任何软件版本。

本文档中的信息是从一个特定实验室环境中的设备产生的。本文档中使用的所有设备都使用的缺省（默认）配置。如果您的网络在提供业务，请确保您了解所有命令的潜在影响。

背景资料

在给出可用于无线局域网的不同的快速安全漫游方法的描述之前，了解无线局域网的关联过程的工作原理以及了解没有在服务集标识符（SSID）上进行安全配置的常规漫游事件是非常重要的。

当一个 802.11 无线客户端连接到一个无线接入点（AP）并在它开始传递数据（无线数据帧）之前，它首先必须通过基本的 802.11 开放系统认证过程。然后必须完成关联过程。可以把开放系统认证过程看做是将客户端与选择的无线接入点进行“线缆连接”的过程。理解这一点是非常重要的，因为始终是无线客户端在做决定优选哪一个无线接入点，这些决定基于多种因素，对于不同设备制造商而不同。这就是为什么如本文档后面所示，是客户端在开始的时候会发送验证帧到选定的无线接入点。无线接入点不能去要求客户端建立连接。

当收到了无线接入点的认证回复的时候，开放系统认证过程就完成了，关联过程会自动地完成 802.11 的 2 层协商，在客户端和无线接入点之间建立一个连接。如果连接成功，无线接入点会分配一个关联 ID 给客户端，并且准备传送数据流量，或者是执行在 SSID 配置的更高层次的安全方法。开放系统认证过程包括两个管理帧交互，同样关联过程也需要交互两个管理帧。认证和关联帧是无线管理帧而不是数据帧，它们基本上用于客户端与无线接入点的连接过程。

以下是在无线环境里这一过程中抓取的无线帧：

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462	Authentication, SN=2443, FN=0, Flags=....
2	0.000784	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462	Authentication, SN=2771, FN=0, Flags=....
3	0.002428	Aironet_b7:ab:5c	Cisco_f0:68:d0	84:78:ac:f0:68:d0	802.11	2462	Association Request, SN=2444, FN=0, Flags
4	0.007122	Cisco_f0:68:d0	Aironet_b7:ab:5c	84:78:ac:f0:68:d0	802.11	2462	Association Response, SN=2772, FN=0, Flag
5	0.995428	0.0.0.0	255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462	DHCP Discover - Transaction ID 0xba2bf0a4
6	2.996191	1.1.1.1	172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462	DHCP Offer - Transaction ID 0xba2bf0a4
7	2.998332	0.0.0.0	255.255.255.255	84:78:ac:f0:68:d0	DHCP	2462	DHCP Request - Transaction ID 0xba2bf0a4
8	3.005016	1.1.1.1	172.30.6.67	84:78:ac:f0:68:d0	DHCP	2462	DHCP ACK - Transaction ID 0xba2bf0a4

注意 ：如果您想了解 802.11 无线嗅探，以及这个文件中的 Wireshark 抓取数据包所使用的过滤器/颜色，请访问思科支持社区中名为 802.11 嗅探器抓取数据包分析的文章。

无线客户端以发送认证帧作为开始，无线接入点会用另一认证帧答复。然后客户端就会发送关联请求帧，无线接入点用关联响应帧答复作为结束。从 DHCP 数据包中可以看出，一旦通过了 802.11 开放系统认证和关联进程，客户端就会开始传送数据帧。在这种情况下，对于没有进行安全方法配置的 SSID，客户端就会立即开始发送没有加密的数据帧（在本示例中发送 DHCP 数据帧）。

本文档后面有介绍到，如果在 SSID 上启用安全设置，对于某些特定的安全方法有更高级别的身份验证和加密的握手帧，这些都会发生在关联响应之后以及发送任何客户端的加密流量数据帧之前，这些加密的流量帧包括了如 DHCP，地址解析协议（ARP）和应用程序数据包。基于配置的安全方法，数据帧只能在客户端完成充分验证和协商了加密密钥之后才能发送。

基于前面抓取的数据包，可以中无线控制器中通过调试客户端命令看到当无线客户端开始一个新的关联到 WLAN 的消息的输出：

```
*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
!---这个是从无线客户端到无线接入点的关联请求。
```

*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
!---这个是从无线接入点到无线客户端的关联回复。

注意 ：本文档中所示的无线控制器调试信息是调试客户端命令，例子只显示一些相关的信息，而不是整个调试输出。有关此调试命令的详细信息，参考文档“理解无线控制器调试客户端”一文 。

这些消息显示了关联请求和响应帧；初始身份验证数据帧没有在无线控制器上记录，因为这一握手在 CUWN 中是无线接入点级的快速反应。

当客户端漫游会出现什么样的信息？建立与一个无线接入点的连接，客户端总是会交换四个管理数据帧，无论是客户端在建立关联，或在漫游时。客户端在同一时间只能与一个无线接入点建立一个连接。与 WLAN 基础设施建立新连接和一个漫游事件之间的帧交换的唯一区别在于，一个漫游事件的关联帧被称为重关联帧，这表明客户端实际上是从另一个无线接入点漫游过来的，不是要试图建立一个到 WLAN 的新的连接。这些数据帧可以包含不同的以协商漫游事件的元素；这取决于设置，但这些细节超出了本文档的范围。

以下是帧交换的一个例子：

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

以下是调试输出的信息：

*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
!---这个是从无线客户端到无线接入点的重关联请求。

*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
!---这个是从无线接入点到客户端的重关联回复。

如上所示，客户端在向新的无线接入点发送完重关联请求并从这个无线接入点接收到了重关联回复之后，就算成功的执行一个漫游事件。由于客户端已经有 IP 地址了，发送的第一个数据帧应该是 ARP 报文。

如果你期望的是一个漫游的事件，但客户端发送的是关联请求，而不是一个重关联请求（你可以从一些抓取数据包并进行类似于本文前面所述的调试就可以得到确认），则客户端不是真的漫游。客户端就会像是断开了连接一样，会开始一个新的到 WLAN 的关联，并且会试图从头开始重新连接。这种情况可能发生的原因有多种，比如当一个移动的客户端远离了覆盖

区域，然后寻找到一个有足够的信号质量来启动一个关联的无线接入点，但它通常因为客户端的问题而没有启动漫游，比如由于驱动程序，固件或软件的问题。

注意：您可以与无线客户端供应商确认问题的原因。

漫游与更高级别的安全

如果 SSID 在基本 802.11 开放系统认证之上还配置了 L2 更高级别的安全性时，那么在初始关联和漫游的时候都需要交换更多的帧。802.11 无线局域网的两种最常见的安全方法将在本文档中描述：

* WPA/WPA2-PSK（预共享密钥） - 客户端通过预共享密钥的进行认证。
* WPA/WPA2- EAP（可扩展认证协议） - 客户端认证使用 802.1X/EAP 方法，以便通过使用一个认证服务器来验证更安全的凭证，如证书、用户名和密码、令牌等。

需要知道的是，尽管这两种方法（PSK 和 EAP）以不同的方式认证/验证客户端，但是基本上都使用与 WPA /WPA2 规则相同的密钥管理流程。无论安全性是使用 WPA/WPA2-PSK 或 WPA/WPA2-EAP，这个过程都被称为 WPA/WPA2 的四次握手，最开始在无线控制器和无线接入点之间进行密钥协商，一旦客户端通过某种特定的认证方法进行了验证，主会话密钥（Master Session Key - MSK）就会成为原始密钥材料。

下面是这个过程的总结：

1.当使用 802.1X/EAP 安全方法的时候，会从 EAP 认证过程中产生一个 MSK，或当使用 WPA/WPA2-PSK 安全方法的时候，从 PSK 中产生 MSK。
2.从这个 MSK 中，客户端和无线控制器/无线接入点会导出成对主密钥（Pairwise Master Key - PMK）且无线控制器/无线接入点会产生一个组主密钥（Group Master Key - GMK）。
3.一旦这两个主密钥都准备好后，客户端和无线控制器/无线接入点就会启动 WPA/WPA2 的 4 次握手过程（本文档后面说明），并且会使用主密钥作为种子来协商实际的加密密钥。
4.那些最终加密的密钥被称为成对临时密钥（Pairwise Transient Key - PTK）和组临时密钥（Group Transient Key - GTK）。PTK 是从 PMK 衍生得出的，用来加密客户端的单播数据帧。组临时密钥（GTK）是从 GMK 衍生得出的，并且用于加密某个特定 SSID/无线接入点的组播/广播帧。

WPA/WPA2-PSK

当 WPA-PSK 或 WPA2- PSK 使用临时密钥完整性协议（TKIP）或高级加密标准（AES）加密，在初始关联的过程和漫游的时候，客户端必须经过被称为 WPA 四次握手的过程。如前面所

解释的，这基本上是为了 WPA/WPA2 用于导出加密密钥的密钥管理过程。当然，PSK 过程也被用来验证客户端是否具有有效的预共享密钥来加入无线网络。下面截图显示了 WPA 或 WPA2 PSK 执行时的初始关联过程：

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	7.564718	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 开放系统认证和关联过程之后，还有来自 WPA 四次握手过程的 4 个 EAPOL 帧，由无线接入的发起的以消息-1 为开始，以消息 4 结束。在成功握手之后，客户端开始发送数据帧（如 DHCP），在这种情况下会使用从四次握手产生的密钥进行加密（这就是为什么你不能从无线抓取数据中看到实际的内容和类型的原因）。

注意 ：EAPOL 帧是为了在无线接入点和客户端之间发送所有密钥管理帧和 802.1X/EAP 验证帧；它们作为无线数据帧发送。

在调试输出中出现的信息：

```
*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
```

!---关联握手完成

```
*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
```

!---从 WLC/AP 发送了 WPA/WPA2 四次握手的信息-1 到客户端

```
*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
```

!---客户端成功接收到了 WPA/WPA2 四次握手的信息-2

```
*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
```

!---从 WLC/AP 发送了 WPA/WPA2 四次握手的信息-3 到客户端

```
*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
```

!---客户端成功接收到了 WPA/WPA2 四次握手的信息-4，它确认了获取到的密钥的安装。它们可以用来加密现在的这个无线接入点的数据帧。

在漫游的时候，客户端基本上遵循相同的帧交换规则，WPA 四次握手会与新的无线接入点得出新的加密密钥。这是标准处于安全原因的规定，新的无线接入点不知道原始密钥的原因。唯一的区别是重关联帧与关联帧的不同，如下抓取的数据所示：

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	0.014109	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

可以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帧不是关联帧，而是一个重新关联帧，如上图所示。

WPA/WPA2-EAP

当在设置了 802.1X/EAP 安全方法的 SSID 上认证客户端，在客户端开始发送数据之前还需要更多的帧交换。这些额外的帧被用来验证客户端的认证信息，都是基于 EAP 方法，有可能是 4 到 20 个帧之间。这些发生在关联/重关联之后，但在 WPA/WPA2 4 次握手之前，因为正是这个认证过程派生了在密钥管理过程中（4 次握手）生成最终加密密钥所需要的 MSK。

以下无线抓取数据显示在初始关联的阶段无线接入点和无线客户端之间的帧交换例子，在这个时候执行了 PEAPv0/EAP-MSCHAPv2 的 WPA 认证：

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.

有的时候交换过程会显示或多或少的帧，这取决于多种因素，比如 EAP 方法，由于出现问题的重发，客户端行为（如本例中的两个身份请求，因为客户端在无线接入点发送了第一身份请求之后发送一条 EAPOL 启动），或者是客户端已与服务器交换了证书。每当 SSID 被配置了 802.1X/EAP 安全方法，都会有更多的帧（用于认证），因此在客户端开始发送数据帧之前它需要更多的时间。

下面是调试信息摘要：

```
*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
```

!---关联握手完成

```
*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)
```

!---为了进行更高层次的认证过程，一旦客户端关联上就会发送 EAP 身份请求。它会通知客户端，需要提供一个身份来进行 802.1X/EAP 类型的认证。

```
*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
```

!---无线客户端决定要开始 EAP 认证过程，并且通过发送一个 EAPOL START 数据帧通知 AP

```
*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 发送另一个 EAP 身份请求给客户端

```
*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
```

!---客户端使用在 EAPOL 帧上的 EAP 身份响应作为回复

```
*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)
```

!---一旦 WLC/AP 发送了在 RADIUS 接入请求包中的客户端回复到认证服务器,为了正式开启与客户端的协商，握手和认证（根据使用的 EAP 方法，有的时候使用相互认证），服务器会使用一个 RADIUS 接入挑战作为回复。这个由 WLC/AP 接收到的回复会发送给客户端。

```
*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)
```

!---客户端会使用一个 EAPOL 帧上的 EAP 回应做回复，它会在 RADIUS 接入请求包上发送给认证服务器。服务器会使用另一个 RADIUS 接入挑战作为回复。根据 EAP 安全方法（交换使用的证书，TLS 隧道的建立，客户端认证信息的确认，需要时服务器身份的确认），这个过程会持续。所以，在 WLC/AP 端来看下边的几个信息基本上都是相同的，它在客户端和认证服务器交换的之间作为一个“代理”。

```
*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
```

!---这个客户端的认证成功完成,所以 RADIUS 服务器发送一个 RADIUS 接入接受信息给 WLC/AP。这个 RADIUS 接入接受信息包含了专门分配给这个客户端的特殊属性（如果在认证服务器上针对这个客户端有相应的设置的话）。这个接入接受信息还包括了客户端在 EAP 认证过程中产生的 MSK 信息，所以 WLC/AP 使用它与无线客户端开始 WPA/WPA2 4 次握手过程。

```
*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)
```

!---认证的接受/通过使用 EAP 成功信息发送给客户端

```
*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
```

!--- WLC/AP 发送 WPA/WPA2 4 次握手的信息-1 到客户端

```
*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
```

!---客户端成功接收到 WPA/WPA2 4 次握手的信息-2

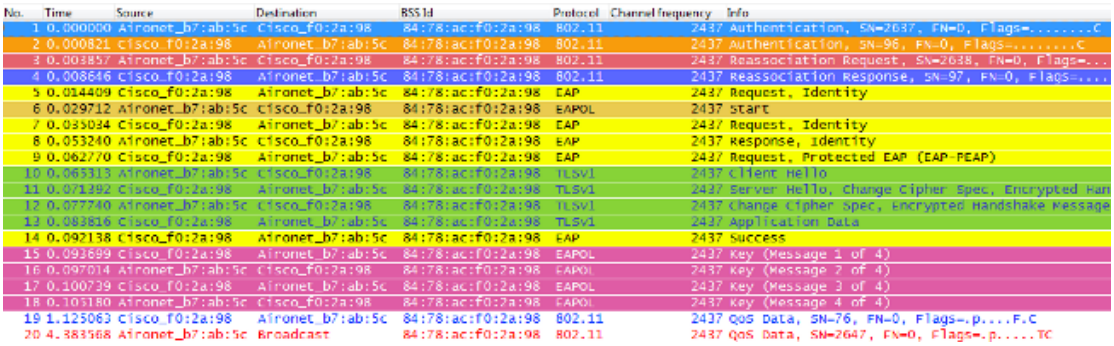
```
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.481: 00:40:96:b7:ab:5c
  Sending EAPOL-Key Message to mobile 00:40:96:b7:ab:5c
  state PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01
```

!--- WLC/AP 发送 WPA/WPA2 4 次握手的信息-3 到客户端

```
*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
```

!---客户端成功接收到 WPA/WPA2 4 次握手的信息-4（最后一个信息），它确认了生成密钥的成功安装。他们可以被用来加密现有 AP 数据帧。

当无线客户端在这个时候执行常规漫游（正常的没有使用的快速安全漫游的方法），客户端必须经过完全相同的过程，并执行对认证服务器的完整验证，如图所示。唯一不同的是，客户端使用的是重关联请求来告知新的无线接入点，它实际上是从另一个无线接入点漫游过来的，但客户端仍然要经过完全的验证和新的密钥生成过程：



No.	Time	Source	Destination	BSSId	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=2637, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=98, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2638, FN=0, Flags=...
4	0.008646	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Reassociation Response, SN=97, FN=0, Flags=....
5	0.014409	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
6	0.029712	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Start
7	0.033034	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
8	0.053240	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAP	2437	Response, Identity
9	0.062770	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Protected EAP (EAP-PEAP)
10	0.065313	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Client Hello
11	0.071292	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Server Hello, Change Cipher Spec, Encrypted Han
12	0.077740	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Change Cipher Spec, Encrypted Handshake Message
13	0.083816	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Application Data
14	0.092138	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Success
15	0.093699	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 1 of 4)
16	0.097014	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 2 of 4)
17	0.100739	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 3 of 4)
18	0.103180	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 4 of 4)
19	1.125063	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=76, FN=0, Flags=.p....F.C
20	4.383568	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=2647, FN=0, Flags=.p.....TC

如图所示，帧比在初始验证的时候少了（如前文所述这是由多种因素造成的），当客户端漫游到新的无线接入点时，为了继续发送数据帧（即使流量在漫游之前主动发送的），EAP 认证和 WPA 密钥管理过程仍然必须完成。因此，如果客户端有一个活跃的应用程序对延迟敏感（如语音流量应用，或者说是超时敏感的应用），那么用户在漫游时会感受到某些问题，比如音频间断或应用程序断开连接。这取决于客户端需要需要多少时间才能继续发送/接收数据帧。取决于不同因素，这种延迟可能会更长：射频环境，客户端的数量，在无线控制器和无线接入点以及与认证服务器之间的往返时间，以及其他原因。

以下是漫游行为的调试信息（基本上与以前的相同，因此这些消息将不进一步描述）：

```
*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
```

这就是 802.1X/EAP 和 WPA/WPA2 安全框架的工作方式。为了防止一个普通的漫游事件的延误对应用程序/服务的影响，为了在 WLAN/SSID 上使用了安全方法的情况下加速漫游进程，WiFi 行业定制并实施了多个快速安全漫游方法。在 WLAN 上部署了高级安全方法，客户端在无线接入点之间进行漫游并且在继续发送数据流量的时候，客户端会感受到一定的延迟。这是由于设置的安全方法中所需的 EAP 认证和密钥管理的帧交换过程，如前面所述。

需要了解的是快速安全漫游只是在 WLAN 上配置了安全方法的时候，所使用的行业术语，用来形容所使用的加速漫游过程的方法/计划。CUWN 所支持的可用于无线局域网的不同的快速安全漫游的方法/方案，将在下一节中做描述。

快速安全漫游与 CCKM

思科集中密钥管理（CCKM）是思科创建的在企业 WLAN 上开发和使用的第一个快速安全漫游方法，在 WLAN 上使用了 802.1X/EAP 安全方法的时候，为了减轻之前讲到的延迟思科所开发的解决方案。由于这是思科的专有协议，它仅支持思科 WLAN 基础设施设备和思科兼容扩展（CCX）CCKM 适配的无线客户端（众多供应商均支持）。

CCKM 适用于所有的可用于 WLAN 的不同加密方法，包括：WEP，TKIP 和 AES。它也支持大部分用于 WLAN 的 802.1X/EAP 认证办法，这取决于设备所支持的 CCX 版本。

注：对于不同版本的 CCX 规范（包括支持的 EAP 方法）所支持的功能特性内容概述，请参考 CCX 版本和功能文档，并验证你的无线客户端所支持的确切 CCX 版本（如果它们是 CCX 兼容的），这样就可以确认你想要使用的 CCKM 安全方法可以实现。

以下无线抓取的数据提供了在初始关联的时候，使用 TKIP 加密并且使用 PEAPv0/EAP-MSCHAPv2 作为 802.1X/EAP 方法的 CCKM 帧交换例子。这基本上与 WPA / TKIP 和 PEAPv0/EAP-MSCHAPv2 执行相同的交换过程，但是客户端和基础设施之间的 CCKM 是通过协商实现的，所以为了在客户端漫游中实现快速安全漫游的时候使用不同的密钥层次结构和缓存的方法：

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	0.176246	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)

下面是调试信息的摘要（删除某些 EAP 交换以减少输出信息）：

```
*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
```

!---这是客户端的关联请求

```
*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
```

!--- WLC/AP 找到了客户端发送在关联请求上宣告支持 CCKM 的信息元素

```
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
```

!---这是该客户端临时设置的密钥缓存索引

```
*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
```

!---关联回复发送给了客户端

```
*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)
```

!---一旦客户端关联了并开始了高级认证过程的时候，就会有一个 EAP 身份请求发送到客户端。

```
*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 为这个客户端创建一个全局 PMK 缓存条目，在这里指的是 CCKM

```
*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)
```

!---通知客户端 EAP 认证成功

```
*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
```

!--- WLC/AP 发送初始 4 次握手过程的信息-1 给客户端

```
*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
```

!---客户端成功接收到初始 4 次握手过程的信息-2

```
*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
```

!---该客户端的 CCKM 缓存目录在同一个移动组中的无线控制器中分享

```
*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
```

!--- WLC/AP 发送初始 4 次握手过程的信息-3 给客户端

```
*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
```

!--- 客户端成功接收到初始 4 次握手过程的信息-4（最后的信息），验证了生成密钥的成功安装。现在它们可以被用来加密现有无线接入点的数据帧。

使用 CCKM，到 WLAN 的初始关联过程与常规的 WPA/WPA2 过程类似，一个 MSK（也称为网络会话密钥（NSK））是由客户端和 RADIUS 服务器共同产生的。该主密钥是在身份验证成功

后由服务器发送到无线控制器的，并缓存了起来用作在该 WLAN 中客户端关联生命周期的所有后续密钥的产生。之后，为了获得第一个无线接入点的单播（PTK）和组播/广播（GTK）加密密钥，无线控制器和客户端派生出了用于基于 CCKM 的快速安全漫游和经过类似于 WPA/WPA2 的 4 次握手过程的种子信息。

这是漫游时的最大区别。在这种情况下以得出新的 PTK，CCKM 客户端会发送一个重关联请求帧到无线接入点/无线控制器（包括 MIC 和一个递增的随机数），并提供足够的信息（包括新的无线接入点的 MAC 地址，BSSID）以衍生新的 PTK。为了导出新的 PTK，利用这个重关联请求，无线控制器与新的无线接入点拥有了具有足够的信息，因此它们可以简单地用重关联响应作出回复。现在客户端可以继续发送数据，如下所示：

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

以下是这个漫游事件的无线控制器的调试输出摘要：

```
 *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
```

!---客户端接收到了重关联请求，并且提供了所需的 CCKM 信息来产生快速安全漫游的新密钥

```
*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 计算出该 CCKM 快速漫游交换中所使用的 MIC

```
*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
```

!--- 生成了新的 PTK

```
*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
```

!--- 为这个新的无线接入点到客户端的关联创建了一个 PMKID 缓存条目

```
*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
```

!--- WLC/AP 发送了重关联回复给客户端，其中包括了确认新的快速漫游和密钥生成所需的 CCKM 信息

```
*dot1xMsgTask: Jun 25 15:43:33.757: 00:40:96:b7:ab:5c
  Skipping EAP-Success to mobile 00:40:96:b7:ab:5c
```

!--- 由于快速漫游而跳过了 EAP 过程，CCKM 也不需要更进一步的密钥握手过程。客户端已经准备好发送加密的数据帧到新的无线接入点

如图所示，快速安全漫游时，同时避免了 EAP 认证帧，以及 4 次握手过程。因为仍然获得了新的加密密钥，是基于 CCKM 协商方案生成的。这个过程的完成包括了漫游重关联帧和由客户端和无线控制器先前缓存的信息。

FlexConnect 与 CCKM

* 当您在 FlexConnect 设置情况下使用 CCKM，行为与之前描述的完全一样（快速安全漫游的时候），只要客户端漫游到的无线接入点同属与一个 FlexConnect 组。
* 只要无线接入点处于连接模式（无论是集中或本地交换），CCKM 可与配置在集中或本地认证模式的 FlexConnect 无线接入的协同工作。

CCKM 的优点

* CCKM 是主要部署在企业无线局域网中最快的快速安全漫游的方法。在 WLAN 中的客户端生命周期内，当在无线接入点之间漫游的时候为了获得新的密钥，客户端不需要再进行一个密钥管理握手过程，并且永远不会再需要与新的无线接入点来执行完整 802.1X/EAP 认证。
* CCKM 支持所有的 802.11 标准中可用的加密方式（WEP，TKIP 和 AES），此外还支持传统的 Cisco 专有的方式，有一些目前还在传统客户端中使用。

CCKM 的缺点

* CCKM 是思科专有的方案，这限制了只有思科 WLAN 基础设施和 CCX 无线客户端才能使用和支持。
* CCX 版本 5 还没有被广泛采用，所以很多 CCX 无线客户端还不支持使用 WPA2/AES 的 CCKM（主要是因为大部分都已经支持采用 WPA / TKIP 的 CCKM，这仍然是非常安全的）。

快速安全漫游与 WPA2 PMKID 缓存

WPA2 成对主密钥 ID（PMKID）缓存，或称粘性密钥缓存（SKC），是由 802.11i 安全修正案提出的第一个 802.11 标准的快速安全漫游方法，最主要的目的是对高级的 WLAN 安全标准化。为了改善设置了安全的漫游，为 WPA2 设备添加了快速安全漫游技术作为可选方法。

这是有可能的，因为每次当每一个客户端进行了完全的 EAP 验证的时候，客户端和认证服务器就会推导出一个 MSK，并用它导出 PMK。这是用来作为 WPA2 4 次握手的种子，以得出用户会话的最终的单播加密密钥（PTK）（直到客户端漫游到另一个无线接入点或者会话过期）；因此，当漫游的时候这种方法可以防止 EAP 认证阶段，因为她重新使用了由客户端和 AP 缓存的原来的 PMK。客户端只需要经过 WPA2 4 次握手，就可以得出新的加密密钥。

与 802.11 标准快速安全漫游方法相比，这个方法没有被广泛部署，主要的原因是：

* 这种方法是可选的，并非所有的 WPA2 设备都支持，因为 802.11i 的修正案不涉及快速安全漫游，并且 IEEE 已经在另一项修正案中标准化 WLAN 的快速安全漫游（即 802.11r 标准，在本文档的后面会介绍）。
* 这种方法在其施行的时候有一个很大的限制：无线客户端只能漫游回到他们以前已经完成认证/连接的无线接入点的时候，才能执行快速安全漫游。

使用这种方法，与任何无线接入点的初始关联过程就像是一个到 WLAN 的普通的首次认证过

程，与认证服务器的整个 802.1X/EAP 认证过程和 4 次握手的密钥生成过程必须发生在客户端能够正常发送数据帧之前，如下抓取数据所示：

No.	Time	Source	Destination	BSSId	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

除了一些与密钥缓存技术相关的额外的输出，这个 debug 输出显示了与 WLAN 初始认证方法的都相同的 EAP 认证帧交换。这些调试输出都有删减以显示出新的信息，而不是整个 EAP 帧交换过程，因为每一次与认证服务器的客户端身份验证交换的信息都一样。这些抓取的数据包输出显示的都是与 EAP 相关的认证帧，所以为了简化输入，删除了大部分的 EAP 调试信息。

```
*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
```

!--- 这是客户端发送的关联请求

```
*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
```

!--- WLC/AP 在客户端发送的关联请求中发现关于支持 PMKID 缓存的信息元素

```
*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
```

!---因为这是初始的关联，关联请求没有任何 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
```

!--- 发送关联回复给客户端

```
*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 为这个客户端创建一个 PMK 缓存目录，在这里用来做 SKC，所以 PMKID 使用无线接入点的 MAC 地址计算（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)
```

!--- WPA/WPA2 4 次握手的信息-1 里边包括了做了 hash 算法的 PMKID

```
*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
```

!--- 这是做了 hash 的 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
```

!--- WLC/AP 发送 WPA/WPA2 4 次握手的信息-1 给客户端

```
*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
```

!--- 客户端成功接收到 WPA/WPA2 4 次握手的信息-2

```
*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
```

!--- WLC/AP 发送 WPA/WPA2 4 次握手的信息-3 给客户端

```
*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
```

!---客户端成功接收到初次 WPA/WPA2 4 次握手的信息-4（最后一个信息），验证了生成密钥的成功安装。可以用来对现目前无线接入点的数据帧进行加密。

使用这种方法，无线接入点和无线客户端会缓存已经建立安全关联的 PMK。因此，如果无

线客户端漫游到一个新的它之前没有关联过的无线接入点，则客户端必须重新执行完整的 EAP 认证过程，如下图所示的客户端漫游到新无线接入点的过程：

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	0.118364	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一个之前已经关联/认证过的无线接入点，那么客户端就会发送一个重新关联连接请求帧，在其中列出了多个 PMKID，它会把之前客户端认证过的无线接入点的所有的 PMK 通知给这个 AP。因此，由于客户端漫游回一个无线接入点，并且也有一个这个客户端的缓存 PMK，所以客户端不需要再通过 EAP 重新验证得出一个新的 PMK。客户端只需经过 WPA2 4 次握手以得出新的临时加密密钥：

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)

注意：这些抓取的数据不显示来自客户端的第一次 802.11 开放系统认证帧，但这不是由于实施的方法中的原因，因为这个帧总是必需的。在这个例子中，这个特定的帧没有被无线帧抓嗅适配器或者是无线包抓取软件抓取的原因是为了教育目的故意丢掉的。要知道，即当你在空中抓取数据包的时候这是有可能发生的；有些帧可能会被错过，但是在客户端和无线接入点之间确实交换了。否则，本例中漫游从来就没有开始。

是这个快速安全漫游方法的 WLC 调试信息：

```
 *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
```

!---这是客户端发送的关联请求

```
*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
```

!--- WLC/AP 在客户端发送的关联请求中发现关于支持 PMKID 缓存的信息元素

```
*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
```

!---客户端发送的重新关联请求中包含了 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
```

!---接收到的 PMKID

```
*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 在数据库中查找匹配的 PMKID

```
*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
```

!--- WLC 检验客户端提供的 PMKID，确认它是否为这一个客户端和 AP 对有有效的 PMK 缓存

```
*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
```

!---发送重新关联回复给客户端，同 SKC 验证了快速漫游

```
*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 会发起一个基于找到的缓存 PMK 的客户端与 AP 对的强健安全网络管理过程

```
*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)
```

!---缓存的 PMKID 会包含在 WPA/WPA2 4 次握手的信息-1 中

```
*dot1xMsgTask: Jun 22 00:26:40.795:
  [0000] c9 4d 0d 97 03 aa a9 0f 1b c8 33 73 01 f1 18 f5
```

!---对 PMKID 做 hash。接下来的信息与 WPA/WPA2 4 次握手信息相同，描述了用来完成加密密钥的生产和安装。

```
*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 WPA2 PMKID 缓存/粘性密钥缓存（Sticky Key Caching）

* 当你在应用了 FlexConnect 的场景中使用这个方法，其行为跟之前所述是完全一样的（当您在执行快速安全漫游的时候），并且客户端漫游的无线接入点不必属于同一个 FlexConnect 组。
* 这个方法只有当您在无线控制器中做集中的认证的时候才能使用(流量集中或者本地转发都可以)，如果你对 FlexConnect 配置进行本地认证的话这个方法就不能工作。这意味着全部的 802.1X/EAP 认证过程必须再次发生。因此，它是不支持 FlexConnect 无线接入点的独立模

式。

WPA2 PMKID 缓存/粘性密钥缓存的优点

这种方法可以在本地通过自主独立的无线接入点来实现，不需要一个集中的设备来管理缓存密钥。

WPA2 PMKID 缓存/粘性密钥缓存的缺点

* 正如本文前面所提到的，这种方法的主要限制是只有当漫游回到它以前关联/认证过的无线接入点的时候，客户端才能进行快速安全漫游。如果漫游到一个新的无线接入点，客户端必须再次完成整个 EAP 认证。
* 无线客户端和无线接入点必须记住所有的每一个新的认证生成的 PMK，所以这个功能通常受限于缓存 PMK 的数量。由于这个限制在标准中是没有明确定义的，厂商可以在自己的 SKC 应用中定义不同的限制。例如，思科无线控制器可以为一个无线客户端缓存多达八个无线接入点的 PMK。如果一个客户端在一个会话期间漫游到八个以上的无线接入点，最先连接的无线接入点就会从缓存列表中删除，以便存储新缓存条目。
* 这种方法是可选的，仍然有许多 WPA2 设备不支持，因此，这种方法没有被广泛采用和部署。
* SKC 不支持在不同无线控制器之间的漫游的，这种情况当你在不同的无线控制器管理的无线接入点之间移动的时候就会发生，即使他们是在同一个移动组中。

快速安全漫游与主动密钥缓存（Proactive Key Caching）

主动密钥缓存（PKC）或机会性密钥缓存（OKC）基本上是如前所述的 WPA2 PMKID 缓存方法的增强，这也就是为什么它被命名主动/机会 PMKID 缓存。要注意的是，它不是由 802.11 标准定义的快速安全漫游方法，有很多设备还是不支持的。

这种技术允许无线客户端和无线局域网基础设施在关联这个 WLAN 的客户端的生命周期内只缓存一个 PMK（由与认证服务器的初始 802.1X/EAP 认证的 MSK 生成），即使是在多个无线接入点之间漫游，他们在 WPA2 4 次握手的过程当中都分享使用这个原始的 PMK 作为种子。为了每次客户端与无线接入点重新关联的时候生成新的加密密钥，跟 SKC 一样，这仍然是必需的。为了让无线接入点能共享同一个客户端会话的原始的 PMK，他们必须是在某种管理

控制之下，有一个集中管理的设备能够缓存并且将所有无线接入点的原始 PMK 发布出去。这跟 CUWN 的情况是类似的，无线控制器为受它控制的无线接入点执行这个操作，并且在多个控制器之间的 PMK 的时候使用移动组来处理；因此，对于自主无线接入点情况是有限制的。

使用这种方法，就像在 PMKID 缓存（SKC）一样，与任何无线接入点的初始关联都是一个对 WLAN 的普通的首次认证过程，你必须跟认证服务器完成完整的 802.1X/EAP 认证过程和 4 次过程，才能生成发送数据帧需要的密钥。以下屏幕抓图说明了这一情况：

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, Fla
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

这些调试输出信息，与在本文档中描述的对 WLAN 初始认证的其他方法的 EAP 认证帧交换过程基本一样（如图中抓取信息一致），还有其他一些无线控制器使用的与 EAP 的密钥缓存技术相关的输出。该调试为了显示相关的信息对输出进行了删减：

```
*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
```

!---这是客户端发送的关联请求

```
*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
```

!--- WLC/AP 在客户端发送的关联请求中发现关于支持 PMKID 缓存的信息元素

```
*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
```

!---因为是第一次关联，关联请求不包含 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
```
!---发送关联回复信息给客户端

```
*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 为这个客户端创建一个 PMK 缓存条目，在这里用来做 OKC/PKC，所以 PMKID 使用无线接入点的 MAC 地址计算（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
```

!---这个客户端的 PMK 缓存条目共享给移动组里边的所有 WLC

```
*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)
```

!--- WPA/WPA2 4 次握手过程中的信息-1 包含了进行了 hash 算法的 PMKID

```
*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
```

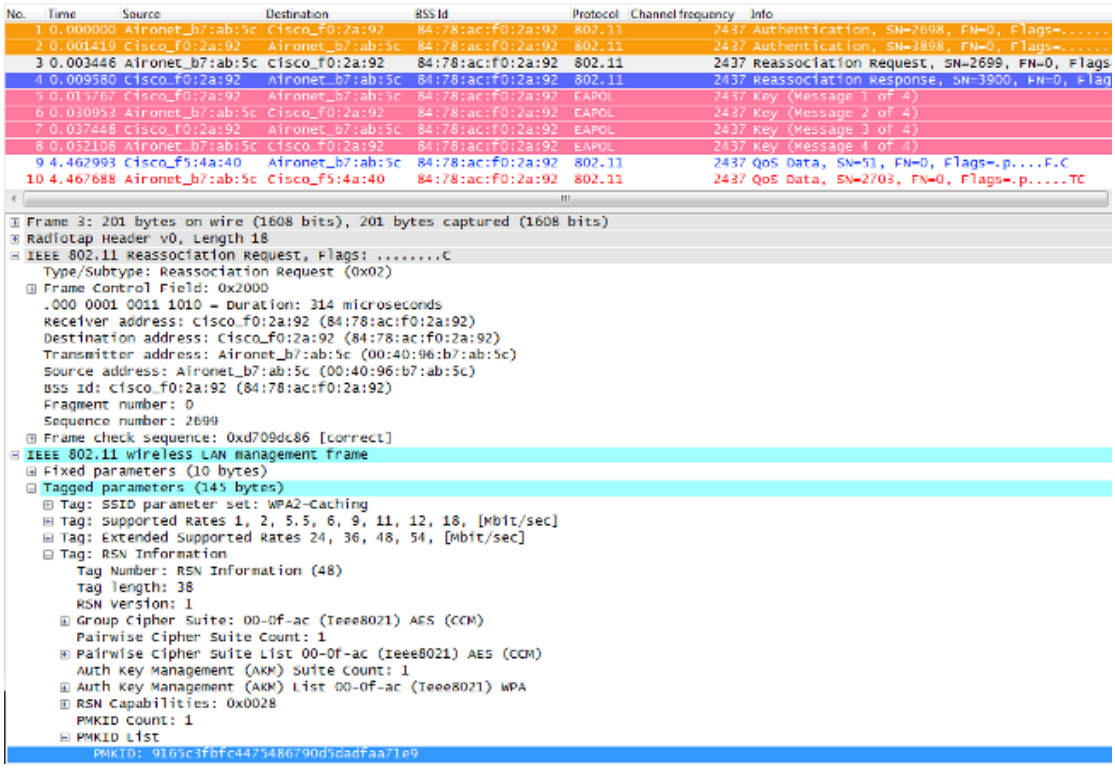
!---这个是 hash 算法之后的 PMKID。接下来的信息与 WPA/WPA2 4 次握手信息相同，描述了用来完成加密密钥的生成和安装。

```
*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
```

使用这种方法，无线客户端和无线控制器（适用于所有受管理的无线接入点）只缓存在初次建立了安全关联的原始的 PMK。基本上每次无线客户端连接到一个特定的无线接入点时，PMKID 的 hash 算法会基于：客户端的 MAC 地址，无线接入点的 MAC 地址（WLAN 的 BSSID），这个无线接入点生成的 PMK。因此，由于 PKC / OKC 缓存所有的无线接入点和某个特定客户的原始的 PMK，当该客户端（重新）关联到另一个无线接入点的时候，为了 hash 获得新的 PMKID，唯一改变的数据就是这个新的无线接入点的 MAC 地址。

当客户端漫游到一个新的无线接入点发送重新关联的请求帧的时候，如果这个客户端想要告知这个无线接入点有一个缓存的 PMK 可以用来做快速安全漫游，它会在 WPA2 RSN 信息元素中添加上 PMKID；因为客户端已经知道了它漫游到的 BSSID（无线接入点）的 MAC 地址，然后客户端只要简单地对 PMKID 进行 hash，可以用来做重新关联请求。当无线接入点从客户端接收到这个请求的时候，它也会用它已有的数据来 hash 这个 PMKID（缓存的 PMK，客户端的 MAC 地址和自己无线接入点的 MAC 地址），并发送重关联响应确认 PMKID 匹配。缓存的 PMK 可以作为启动一个为了得出新的加密密钥（跳过 EAP）的 WPA2 4 次握手过程的种子：



在这个抓取的数据中，选择并且扩展开了来自客户端的重连接请求帧，可以让你看到这个帧的更多细节。MAC 地址信息，也是强健安全网络（RSN）信息元素，其中显示了有关用于这个关联的 WPA2 设置信息（高亮选中的显示的是通过哈希公式得到的 PMKID）。

以下显示的是采用了 OKC/PKC 快速安全漫游方法时无线控制器的调试输出概要：

```
 *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
```

!---这是客户端发送的关联请求

```
*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
```

!--- WLC/AP 在客户端发送的关联请求中发现关于支持 PMKID 缓存的信息元素

```
*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
```

!---客户端的重关联请求包含了 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
```

!---因为客户端没有与这个新的无线接入点进行过认证，WLC 不能够找到与客户端提供的相匹配的 PMKID。但是，因为客户端执行的是 PKC/OKC 而不是 SKC（按照以下信息），WLC 会根据获得的信息计算一个新的 PMKID（缓存的 PMK，客户端 MAC 地址，新的无线接入点的 MAC 地址）

```
*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
```

!---新的 PMKID 计算并验证与客户端提供的相互匹配，因为客户端 提供的也是使用相同的信息计算出来的。所以，快速安全漫游可以执行。

```
*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
```

!---发送重关联回复给客户端，并且验证了 PKC/OKC 的快速漫游

```
*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 会发起一个与查找到的缓存 PMK 成对客户端和无线接入点的关联的强健安全网络。

```
*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)
```

!--- hash 的 PMKID 会包含在 WPA/WPA2 4 次握手的信息-1 中

```
*dot1xMsgTask: Jun 21 21:48:50.570:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
```

!---这个是 hash 算法之后的 PMKID。接下来的信息与 WPA/WPA2 4 次握手信息相同，描述了用来完成加密密钥的生成和安装。

```
*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
```

如图所示在 debug 的开始，PMKID 必须在接收到来自客户端的重关联请求之后被计算出来。上边的这个流程是必要的，为了证实 PMKID 和确认缓存的 PMK 用在了 WPA2 4 次握手的过程来生成加密密钥和成功完成快速安全漫游。不要混淆了 debug 信息中的 CCKM 条目；这不是为了要进行 CCKM，而是如前所述的 PKC / OKC。 CCKM 在这里仅仅是无线控制器使用的那些输出的名字，就像是用于计算 PMKID 的值的函数的名称一样。

FlexConnect 与主动/随机密钥缓存（Opportunistic Key Caching）

* 当您在开启了 FlexConnect 的环境中使用这个方法，它的行为如前所述是完全一样的（当您在执行快速安全漫游的时候），客户端漫游到的无线接入点并不需要属于同一个 FlexConnect 组中。
* 此方法仅能工作在当您的无线控制器做集中认证的情况下（可以使集中或本地转发）；如果你在启用了 FlexConnect 在本地认证，这意味着要进行一个完整的 802.1X/EAP 认证过程。因此，它是不支持独立模式情况下的 FlexConnect 无线接入点。

WPA2 主动/随机密钥缓存的优点

* 无线客户端和 WLAN 基础设施不需要记住多个 PMKIDs，而是只需缓存第一次对 WLAN 认证的初始 PMK。然后，你必须要重新 hash 正确的 PMKID（在重关联请求的时候使用），为了验证快速安全漫游需要，每个无线接入点的安全关联都需要这个步骤。

* 这样，无线客户端就可以执行快速安全漫游到同一 WLAN / SSID 中的一个新的无线接入点上，即使从来没有与这个无线接入点关联过（与使用 SKC 的情况不同）。只要是客户端执行初始 802.1X/EAP 认证过程的无线接入点受集中部署的管理，并且在客户端漫游的时候为所有的无线接入点执行 PMK 缓存过程的话，这样在这个 WLAN 中客户端的生命周期内就不需要再执行一个完整的认证过程了。

WPA2 主动/随机密钥缓存的缺点

* 这种方法仅适用于部署在无线接入点受集中控制和管理的环境下，无线控制器负责缓存和共享客户端会话的原始的 PMK。因此，对自主无线接入点的环境是有限制的。
* 该技术不是 802.11 标准定义的方法，所以还有许多 WPA2 设备不支持。因此，这种方法并未广泛采用和部署。

快速安全漫游与 802.11r 标准

基于 802.11r 修订标准的快速安全漫游技术（由 802.11 标准正式命名为快速 BSS 过渡，被称为 FT），是第一个由 IEEE 802.11 标准正式批准在无线接入点间（基本服务集或者是 BSS）执行快速转换的解决方案，它明确定义了用于处理和缓存 WLAN 密钥的密钥层次结构。然而，它的应用一直进展缓慢，主要是由于需要快速转换的时候已经有其他可用的解决方案，比如使用先前本文档中介绍的方法之一和 VoWLAN 的实现方式。目前只有少数设备支持某些 FT 功能。

这种技术因为引入了新的概念相比其他的方法要解释的话更为的复杂，不同层次的 PMKs 被缓存到不同设备上（每个设备有不同的角色）为快速安全漫游提供了更多的选项。因此，下面提供了一个简单的总结，以及每个可用选项的实现方法。

802.11r 标准与 SKC 和 PKC/OKC 不同，主要是因为以下这些原因：

* 握手消息（例如 PMKID，ANonce 和 SNonce 交换帧）发生在 802.11 身份验证帧或行动帧中，而不是重新关联帧。与 PMKID 缓存方法不同，在（重新）关联消息交换之后的 4 次握手阶段可以避免。与新的无线接入点的密钥握手过程开始在客户端完全漫游/重新关联到这个新的无线接入点之前。
* 它提供了两种快速漫游握手方法：使用无线空口，使用分布式系统（DS）。
* 802.11r 标准具有更多的密钥层次。
* 在客户端漫游的时候该协议避免了 4 次握手的密钥管理(生成新的加密密钥，PTK 和 GTK，不需要握手的过程)，它也可应用于对 WPA2 的 PSK 设置，并不只是在 802.1X/EAP 用于认证的时候。没有了 EAP 或者是 4 次握手交换过程，这为更快的漫游提供了便利。

使用这种方法，当对于 WLAN 基础设施中第一个无线接入点的连接建立的时候，无线客户端只执行一个初始的认证，并在属于同一个 FT 移动域中的不同无线接入点之间进行快速安全漫游。

这是其中一个新的概念，基本上是指使用相同的 SSID（称为一个扩展服务集 ESS 或），并处理相同的 FT 密钥的无线接入点。无线接入点处理的 FT 移动性域密钥的方式，通常是基于集中式的设置，如无线控制器或移动组；但是，这种方法也可以在独立的无线接入点的环境中实现。

以下是密钥层次结构的概要：

* 仍然会从客户端请求者和认证服务器的初次 802.1X/EAP 认证阶段中（一旦认证成功，就会从认证服务器转移到验证器（无线控制器））派生出一个 MSK。像其他方法一样，这个 MSK 作为 FT 密钥层次结构的种子。当您使用 WPA2-PSK 而不是 EAP 验证方法的时候，PSK 基本上就是这里的 MSK。
* 从这个 MSK 中生成一个成对主密钥 R0（PMK-R0），这是 FT 密钥层次结构的第一层次的密钥。这个 PMK-R0 中的密钥所有者是无线控制器和客户端。
* 第二个层次的密钥，叫做成对主密钥 R1（PMK-R1），从 PMK-R0 派生，密钥所有者是客户端和由拥有 RMK-R0 的无线控制器所管理的无线接入点。
* FT 密钥层次结构的第三个也是最后层次的密钥是 PTK，用来加密 802.11 的单播数据帧（类似于使用 WPA/TKIP 或 WPA2/AES 其他方法）。这个 PTK 从 PMK-R1 的 FT 上产生的，密钥所有者是客户端和由无线控制器管理的无线接入点。

注意：根据 WLAN 供应商和实施设置的不同（如自主的无线接入点、FlexConnect 或 Mesh），WLAN 基础设施传送和处理密钥的方式也会有所不同。它甚至可能会改变密钥所有人的角色，但因为本文的范围，前面给出的密钥层次总结的例子是接下来本文的重点。这些差异对于了解过程其实并不太相关，除非为了发现一个软件问题，你需要进行深入的基础设施设备（和他们的代码）的分析。

基于空口的快速 BSS 过渡

使用这种方法，任何无线接入点上的第一次关联都是一个普通的 WLAN 的首次认证过程，对认证服务器的 802.1X/EAP 认证过程和 4 次握手的密钥生成过程在发送数据帧之前都必须发生，如下抓取数据所示：

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	802.11	2462	Authentication, SN=57, FN=0, Flags
2	0.000798	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	802.11	2462	Authentication, SN=2786, FN=0, Fla
3	0.003228	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	802.11	2462	Association Request, SN=58, FN=0,
4	0.008692	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	802.11	2462	Association Response, SN=2787, FN=
5	0.011783	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Identity
6	0.040994	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAP	2462	Response, Identity
7	0.098201	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
8	0.115531	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Client Hello
9	0.132004	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
10	0.136062	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAP	2462	Response, Protected EAP (EAP-PEAP)
11	0.151652	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.154937	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.159064	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.169838	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Certificate, Client Key Exchange,
15	0.180451	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	3.908749	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAP	2462	Response, Protected EAP (EAP-PEAP)
17	3.916050	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	3.918650	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Application Data
19	3.938175	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Application Data
20	3.958529	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
21	3.960992	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Application Data
22	3.966771	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	3.971693	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Application Data
24	3.978519	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	3.981398	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	TLSv1	2462	Application Data
26	3.987998	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAP	2462	Success
27	3.989754	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAPOL	2462	Key (Message 1 of 4)
28	3.994693	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAPOL	2462	Key (Message 2 of 4)
29	4.001601	Cisco_f0:68:d6	Apple_15:39:32	84:78:ac:f0:68:d6	EAPOL	2462	Key (Message 3 of 4)
30	4.006001	Apple_15:39:32	Cisco_f0:68:d6	84:78:ac:f0:68:d6	EAPOL	2462	Key (Message 4 of 4)
31	4.010947	Apple_15:39:32	IPv6mcast_00:00:0(	84:78:ac:f0:68:d6	802.11	2462	QoS Data, SN=14, FN=0, Flags=.p...

```
⊟ Tag: RSN Information
    Tag Number: RSN Information (48)
    Tag length: 20
    RSN Version: 1
  ⊞ Group Cipher Suite: 00-0f-ac (Ieee8021) AES (CCM)
    Pairwise Cipher Suite Count: 1
  ⊞ Pairwise Cipher Suite List 00-0f-ac (Ieee8021) AES (CCM)
    Auth Key Management (AKM) Suite Count: 1
  ⊞ Auth Key Management (AKM) List 00-0f-ac (Ieee8021) FT over IEEE 802.1X
  ⊞ RSN Capabilities: 0x000c
```

主要的区别在于：

* 认证密钥管理协商与常规的 WPA/WPA2 略有不同，所以为了要执行关联到支持 FT 的 WLAN 基础网络的协商的时候，需要一些额外的信息。如上图所示选中的客户端的关联请求帧，为了表明客户端想要在 802.1X/EAP 上执行 FT，高亮标注了 RNS 信息元素的 AKM 字段。
* 如移动域信息元素显示（FT 的一部分），其中 FT 能力和策略字段表明了在快速漫游的时候快速 BSS 过渡完成是通过空口介质还是通过分布系统 DS（在本例中显示是通过空口）。
* 为了在 FT 漫游的时候执行 FT 的认证序列，还需要另外一个信息元素（在本文后边会介绍到的快速 BSS 过渡或者 FT IE）。
* 由于密钥层次结构不同，密钥生成过程也不一样，所以即使 FT 4 次握手过程看起来与 WPA/WPA2 4 次握手类似，但它实际上的内容还是略有不同。

这些调试输出信息，与在本文档中描述的对 WLAN 初始认证的其他方法的 EAP 认证帧交换过程基本一样（如图中抓取信息一致），添加了其他一些无线控制器使用的与 EAP 的密钥缓存技术相关的输出。该调试为了显示相关的信息对输出进行了删减：

```
*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
```

!---这是客户端发送的关联请求

```
*apfMsConnTask_0: Jun 27 19:25:23.427: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.
```

!--- WLC 确认客户端具有 802.11r 能力

```
*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
```

!--- WLC/AP 在客户端发送的关联请求中发现了申明支持 FT 能力的信息元素

```
*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
```

!---一旦 FT 信息计算完毕，就会发送关联回复给客户端（如前边的信息），该信息也添加到了回复中

```
*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 过程开始，然后发生跟前边一样的交换过程

```
*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
```

!--- RADIUS 服务器验证/认证客户端

```
*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 为这个客户端生成一个 PMK 缓存目录，在本例中用来做 802.1X 的 FT，所以 PMKID 的计算使用了无线接入点的 MAC 地址(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
```

!---定义了 R0KH-ID 和 R1KH-ID，以及 PMK 缓存有效期限

```
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.630: ec:85:2f:15:39:32
  PMK sent to mobility group
```

!---这个客户端的 FT PMK 缓存条目在 WLC 所在的移动组中共享

```
*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)
```

!--- hash 后的 PMKID 包括在了最初 FT 4 次握手的信息-1 中

```
*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
```

!--- WLC/AP 发送 FT 4 次握手的信息-1 给客户端

```
*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
```

!---客户端成功接收到了 FT4 次握手的信息-2

```
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Calculating PMKR0Name
```

!---计算出了 PMKR0Name

```
*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
```

!---在重新关联死亡时间 MDIE,TIE，和 R0Key 数据有效时间计算之后，WLC/AP 发送 FT 4 次握手包含了这

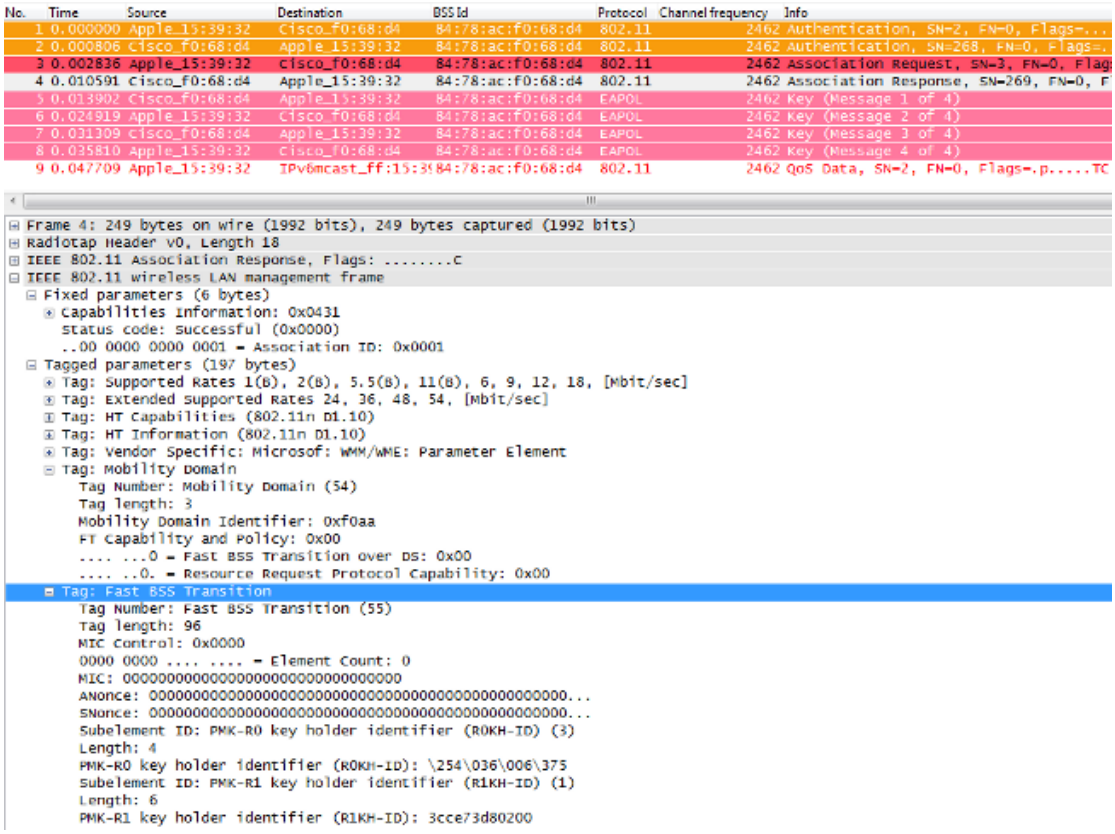
些数据的信息-3 给客户端

```
*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
```

!---客户端成功接收到最初的 FT 4 次握手的信息-4，确认了生成密钥的安装。可以用作本无线接入点的数据帧的加密。

注意：为了调试这个方法并显示的额外 802.11r/FT 的输出，除了调试客户端，还需要附加的是开启 FT 事件调试。

下面是对 WLAN 的初始关联使用 WPA2-PSK 执行 FT 的时候的抓取信息和调试输出（而不是 802.1X/EAP 方法），选中了无线接入点的关联回复帧来显示快速 BSS 过渡信息元素（高亮）。一些在执行 FT 4 次握手的必要的密钥信息也显示出来：




```
*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
```

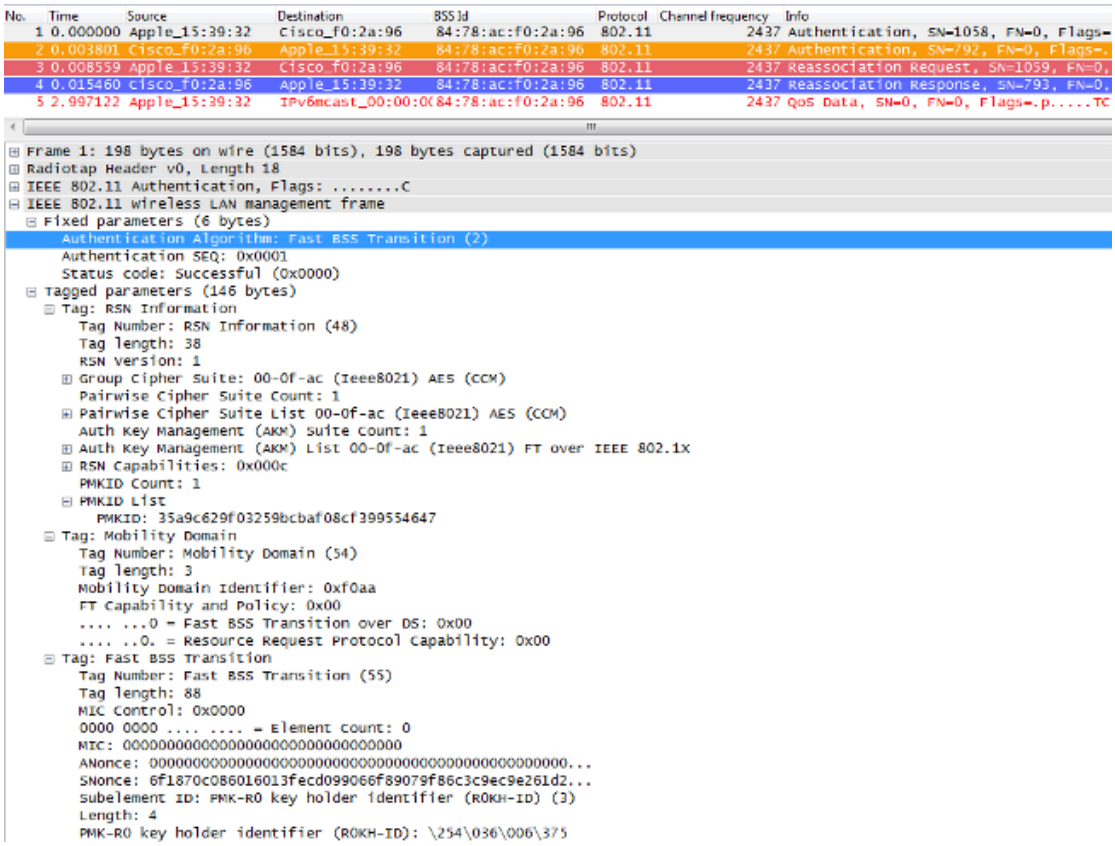
```
  Marking this mobile as TGr capable.

*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
```

使用 802.11r，对 WLAN 的初始关联是为了生成用于该技术的基本密钥，正如其它快速安全漫游方法。主要的区别在当客户端开始漫游的时候；使用 FT 的时候不仅避免了 802.1X/EAP，实际上它还执行更有效的漫游，结合了最初的 802.11 开放系统认证和重新关联帧（在不同无线接入点之间漫游都需要的），目的是为了在 4 次握手过程中交换 FT 信息以及生成新的动态加密密钥。

下图显示了通过空口执行使用 802.1X/EAP 安全的一个快速 BSS 过渡。选中了客户端发送到无线接入点的开放系统认证帧，来查看开始 FT 密钥协商过程需要的 FT 协议信息元素。这是为了生成与新无线接入点的新 PTK（基于 PMK-R1）。高亮突出显示了认证算法字段，为了表明该客户端不是执行的简单的开放系统身份验证，而是快速 BSS 过渡：



下面是 802.1X/EAP 的 FT 漫游事件发生时的调试输出：

```
*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Doing preauth for this client over the Air
```

!---无线控制器和这个客户端通过空口开始 FT 快速安全漫游，执行一种预认证，因为客户端通过空口发送给新的无线接入点的认证帧里边请求了 FT（在发送重新关联请求之前）

```
*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
```

!---无线控制器与客户端要漫游到的新 AP 执行一个本地漫游事件

```
*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 从客户端接收到一个 PMK（在这里叫做 AKM），与缓存的这个客户端的 PMK 匹配

```
*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 为这个无线接入点和客户端对创建一个新的预认证条目，并且添加上 MDIE 信息

```
*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
```

!---一旦客户端从 WLC/AP 接收到认证帧回复，就会发送重新关联请求，客户端漫游到的新无线接入点会接收到这个请求。

```
*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.
```

!---客户端确认这个客户端的 FT 快速安全漫游是成功的

```
*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
```

!---发送重新关联回复给客户端，包含 FT 移动域 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 漫游完成，跳过了 EAP 过程（还有其他的密钥管理握手过程），客户端已经可以发送加密数据帧到这个 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
```

下图显示了通过空口使用 WAP2-PSK 安全的快速 BSS 过渡，选中了无线接入点发送给客户端

的最后重新关联回复帧，来显示更多的有关 FT 交换的细节：

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Apple_15:39:32	Cisco_f0:2a:94	84:78:ac:f0:2a:94	802.11	2437	Auther
2	0.004548	Cisco_f0:2a:94	Apple_15:39:32	84:78:ac:f0:2a:94	802.11	2437	Auther
3	0.009178	Apple_15:39:32	Cisco_f0:2a:94	84:78:ac:f0:2a:94	802.11	2437	Reasso
4	0.016183	Cisco_f0:2a:94	Apple_15:39:32	84:78:ac:f0:2a:94	802.11	2437	Reasso

```
⊟ IEEE 802.11 wireless LAN management frame
  ⊞ Fixed parameters (6 bytes)
  ⊟ Tagged parameters (274 bytes)
    ⊞ Tag: Supported Rates 1(B), 2(B), 5.5(B), 11(B), 6, 9, 12, 18, [Mbit/sec]
    ⊞ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    ⊞ Tag: HT Capabilities (802.11n D1.10)
    ⊞ Tag: HT Information (802.11n D1.10)
    ⊞ Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
    ⊟ Tag: RSN Information
        Tag Number: RSN Information (48)
        Tag length: 38
        RSN Version: 1
      ⊞ Group Cipher Suite: 00-0f-ac (Ieee8021) AES (CCM)
        Pairwise Cipher Suite Count: 1
      ⊞ Pairwise Cipher Suite List 00-0f-ac (Ieee8021) AES (CCM)
        Auth Key Management (AKM) Suite Count: 1
      ⊞ Auth Key Management (AKM) List 00-0f-ac (Ieee8021) FT using PSK
      ⊞ RSN Capabilities: 0x0028
        PMKID Count: 1
      ⊟ PMKID List
          PMKID: 7e370d965e054df50819b135febc3424
    ⊟ Tag: Mobility Domain
        Tag Number: Mobility Domain (54)
        Tag length: 3
        Mobility Domain Identifier: 0xf0aa
        FT Capability and Policy: 0x00
        .... ...0 = Fast BSS Transition over DS: 0x00
        .... ..0. = Resource Request Protocol Capability: 0x00
    ⊟ Tag: Fast BSS Transition
        Tag Number: Fast BSS Transition (55)
        Tag length: 133
        MIC Control: 0x0300
        0000 0011 .... .... = Element Count: 3
        MIC: 1debab4b84d8283e16959fee90b1256b
        ANonce: b6eddf22092867178d96aee8fadbe73f21bc2258e5c95fd7...
        SNonce: 776c4c9a365e9a165e940b5fb5fea017017a0bd342cbd343...
        Subelement ID: PMK-R1 key holder identifier (R1KH-ID) (1)
        Length: 6
        PMK-R1 key holder identifier (R1KH-ID): 3cce73d80200
        Subelement ID: PMK-R0 key holder identifier (R0KH-ID) (3)
        Length: 4
        PMK-R0 key holder identifier (R0KH-ID): \254\036\006\375
        Subelement ID: GTK subelement (2)
        Length: 35
        Key Info: 0x0002
        .... .... .... ..10 = Key ID: 2
        Key Length: 0x10
        RSC: 0000000000000000
        GTK: 6487b855fc7dc16749e3b73c487cb130d0fc1f234a1be851
```

下面是使用 PSK 的 FT 漫游事件发生的调试输出，这与使用 802.1X/EAP 时类似：

```
*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，一旦与 WLAN 的初次关联的快速 BSS 过渡开始协商，漫游使用和需要的的 4 个帧（客户端的开放系统认证，无线接入点的开放系统认证，重关联请求，重新关联回复）基本上用作为 FT 4 次握手，以产生新的 PTK（单播加密密钥）和 GTK（组播/广播加密密钥）。

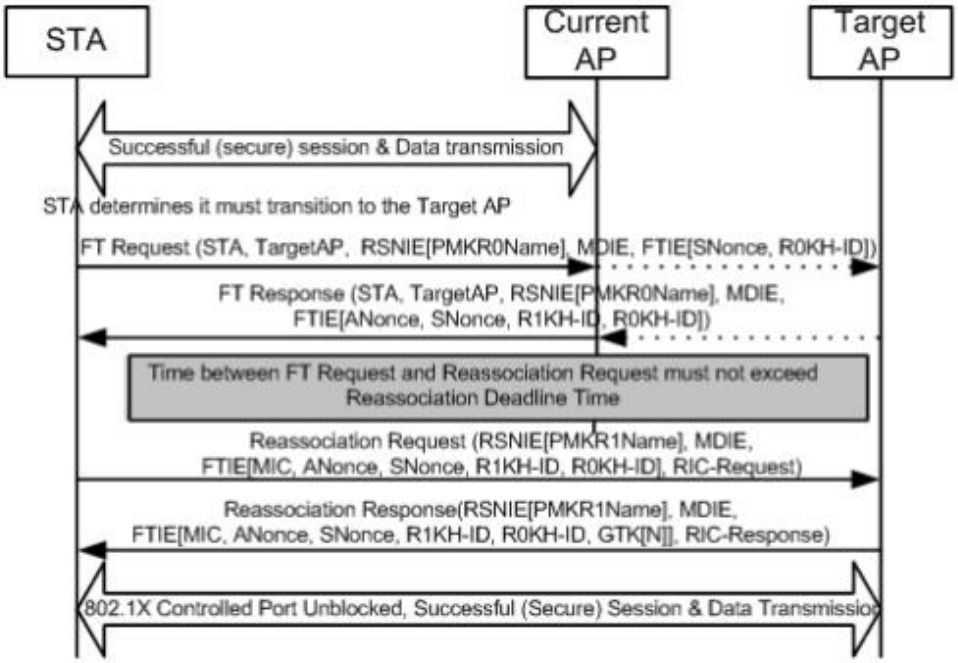
这种方式就代替了这些帧交换之后通常发生的 4 次握手过程，并且这些帧的 FT 内容和密钥协商基本上与使用 802.1X/EAP 或 PSK 的安全性方法的一样。如图所示，最主要的区别是 AKM 字段，它能够确认客户端执行的 FT 是 PSK 或 802.1X。因此，要注意的是这四个帧中通常没有这种类型的密钥协商的安全信息，但只有当客户端在 FT 漫游并且在客户端和 WLAN 基础设施的初次关联实施和协商了 802.11r 的情况下。

基于分布系统的快速 BSS 过渡

802.11r 标准允许另一种快速 BSS 过渡的部署，其中 FT 漫游是由客户端与新的无线接入点发起的，而客户端漫游通过的是 DS（分布式系统），而不是通过空口。在这种情况下，使用了 FT 行动帧，用来发起密钥协商而不是开放系统认证帧。

基本上，一旦客户端决定它可能漫游到一个较好的无线接入点上，客户端在它漫游之前就会发送一个 FT 行动请求帧到它当前连接的无线接入点。客户端表示目标无线接入点的 BSSID 就是它想要 FT 漫游到的地方。原始的无线接入点会通过 DS 系统转发此 FT 行动请求帧到目标无线接入点（通常是通过有线基础设施），目标无线接入点会发送一个 FT 行动响应帧到客户端（也通过 DS，最终通过空气发送到客户端）。一旦这种 FT 行动帧交换是成功的，客户端就完成了 FT 漫游；客户端会发送重新关联请求到目标 AP（这个时候通过空口），并从新的无线接入点接收到重新关联回复，以确认漫游和最终密钥。

总之，四个帧协商完成快速 BSS 过渡并生成新的加密密钥，但这里的开放系统认证帧被 FT 行动请求/响应帧取代，并且通过分布系统在本无线接入点和目标无线接入点之间进行交换。这种方法也适用于 802.1X/EAP 和 PSK 这两种安全方法，所有思科无线控制器都支持；但是，因为通过 DS 的交换没有被无线行业的大多数无线客户端支持和执行（而且帧交换和调试输出基本上是相同的），所以在本文档中不提供的相应的例子。可以使用下图对通过 DS 的快速 BSS 过渡进行可视化的讲解：



FlexConnect 与 802.11r 标准

* 当您在启用了 FlexConnect 时使用此方法，行为如前所述是完全一样的（当您在执行快速安全漫游的时候），并且客户端要漫游到的无线接入点并不需要属于同一个 FlexConnect 组。
* 这个方法只有当您在无线控制器中做集中的认证的时候才能使用(流量集中或者本地转发都可以)，如果你对 FlexConnect 配置为进行本地认证的话，这个方法就不能工作。因此不支持无线接入点工作在 FlexConnect 独立模式。

802.11r 标准的优点

* 这种方法是 IEEE 在 802.11 标准作为一个修正案（802.11r 标准）明确定义的第一个使用密钥层次结构的方法，因此这些 FT 技术的实现能兼容不同的厂商，不会有不同的解释。
* 802.11r 标准有多个有用的技术适应你的需求（通过空口和通过 DS，使用 802.1x/EAP 的安全性和 PSK 安全性）。
* 无线客户端执行快速安全漫游到相同 WLAN / SSID 下的一个新的无线接入点，即使从来没有与这个 AP 关联过也可以，不需要保存多个 PMKIDs。
* 这是第一个快速安全漫游的方法，即使是使用 PSK 也允许快速漫游，避免在 WPA/WPA2 PSK 之间漫游时所需要的 4 次握手。这个快速安全漫游方法的主要目的是为部署了安全方法的时候避免 802.1X/EAP 的握手过程；但是，对于 PSK 安全漫游事件来说，使用 802.11r 更是避免了 4 次握手过程。

802.11r 标准的缺点

* 实际上支持快速 BSS 过渡的无线客户端设备不多，并在大多数情况下，它们并不支持所有的 802.11r 标准中可用的技术。
* 因为这些部署都还非常的新，没有足够的真实生产环境的测试结果，或者是足够的调试结果，来应对可能出现的警告。
* 当您要配置 WLAN / SSID 来使用任何的 FT 方法的时候，那么只有支持 802.11r 标准的无线客户端才可以连接到这个 WLAN/SSID。FT 的设置对于客户端来说不是可选的，因此那些不支持 802.11r 标准的无线客户端必须连接到一个单独的没有配置 FT 的 WLAN/SSID。

结论

* 请记住，客户端才能决定漫游到某个特定的无线接入点，无线控制器/无线接入点无法决定。一旦客户端认为它应该漫游的时候，漫游事件才会由无线客户端启动。

* 非常重要的一点是，在 WLAN/SSID 部署了安全策略的时候，为了加速 WLAN 的漫游过程才有了快速安全漫游方法的开发。如果没有设置安全的情况下，没有什么加速的操作可以做，当在无线接入点之间漫游的时候，在发送数据帧之前，客户端和无线接入点只需要简单地交换漫游所需的无线管理帧就可以了（客户端的开放系统认证帧，无线接入点的开放系统认证帧，重关联请求，和重新关联回复）。因此，过程得不到加速。如果您在没有部署安全的时候遇到漫游问题，那么没有任何快速漫游方法可以加速漫游，能做的只是确认 WLAN /SSID 的设置和设计是否适合无线客户端工作站在无线接入点覆盖蜂窝之间漫游。

* 802.11r/FT 与 WPA2-PSK 一起，从而避免了 4 次握手，加快了安全漫游过程，正如文档中 802.11r 标准部分中所解释的。

* 没有快速安全漫游方法能在 FlexConnect 无线接入点的独立模式下工作。

* 所有的方法都有各自的优点和缺点，但最终你必须始终要确保无线客户端工作站是否支持要实现的具体方法，是否思科 WLAN 基础设施支持所有可用的方法。因此，你必须为连接到特定的 WLAN/SSID 的无线客户端选择所支持的最好的方法。例如，在某些部署中，您可能需要为思科无线 IP 电话创建一个 CCKM 的 WLAN/SSID（它支持 WPA2/AES 和 CCKM，但不支持 802.11r 标准），然后为支持 802.11r/FT 快速安全漫游方法的客户端创建一个 WPA2/AES 的 WLAN/SSID（或使用 OKC / PKC，如果支持的话）。

* 如果无线客户端不支持任何可用的快速安全漫游的方法，那么你可能需要接受这么一个事实，就是在使用了 802.1X/EAP 安全的 WLAN/SSID 下的无线接入点间漫游，这些客户端将会经受到前文提到的延迟（这可能会导致客户端应用程序/服务的中断）。

* 所有的方法，除了 SKC（WPA2 PMKID 缓存），都是支持不同的无线控制器管理的无线接入点间的快速安全漫游（无线控制器间漫游），只要它们是在同一个移动组中。

原文链接：http://www.cisco.com/en/US/tech/tk722/tk809/technologies_tech_note09186a0080c1a517.shtml
翻译人：冯博
校对：谢清
译于 2013 年 12 月