Determinación de Métodos para WLAN 802.11 y Roaming de Seguridad Rápida en CUWN

Contenido

Introducción

Este documento describe los tipos de roaming inalámbrico y de seguridad rápida disponibles para las LAN inalámbricas (WLAN) IEEE 802.11 en Unified Wireless Network (CUWN).

Prerequisites

Requirements

Cisco recomienda que tenga conocimiento sobre estos temas:

- Fundamentos de WLAN IEEE 802.11
- Seguridad WLAN IEEE 802.11
- Aspectos básicos de IEEE 802.1X/EAP

Componentes Utilizados

La información de este documento se basa en la versión 7.4 del software Cisco WLAN Controller.

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si tiene una red en vivo, asegúrese de entender el posible impacto de cualquier comando.

Antecedentes

La información de este documento se basa en la versión 7.4 del software Cisco WLAN Controller, pero la mayoría de las salidas y comportamientos de depuración descritos pueden aplicarse a cualquier versión de software que admita los métodos descritos. Los detalles de todos los métodos explicados aquí siguen siendo los mismos en los códigos de Cisco WLAN Controller posteriores (hasta la versión 8.3 en el momento en que se actualizó este artículo).

Este documento describe los diferentes tipos de itinerancia inalámbrica y los métodos de itinerancia rápida y segura disponibles para las LAN inalámbricas (WLAN) IEEE 802.11 compatibles con Cisco Unified Wireless Network (CUWN).

El documento no proporciona todos los detalles sobre cómo funciona cada método o cómo se configuran. El objetivo principal de este documento es describir las diferencias entre las diversas técnicas disponibles, sus ventajas y limitaciones, y el intercambio de tramas en cada método. Se proporcionan ejemplos de depuraciones del controlador WLAN (WLC) y se utilizan imágenes de paquetes inalámbricos para analizar y explicar los eventos que se producen para cada método de itinerancia descrito.

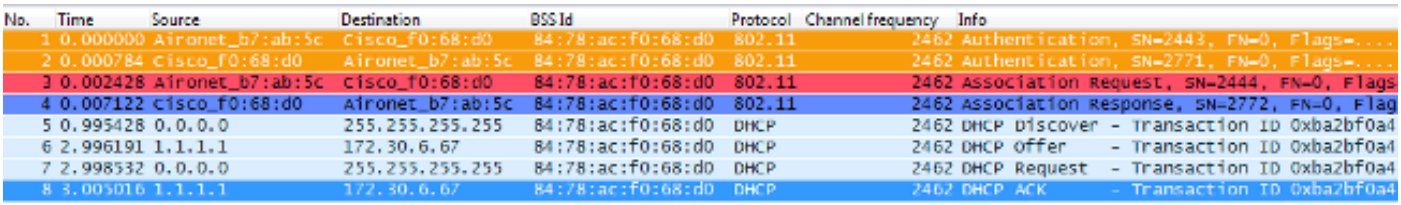
Antes de que se proporcione una descripción de los diferentes métodos de itinerancia de seguridad rápida disponibles para las WLAN, es importante comprender cómo funciona el proceso de asociación WLAN y cómo se produce un evento de itinerancia regular cuando no hay ninguna seguridad configurada en el identificador del conjunto de servicios (SSID).

Cuando un cliente inalámbrico 802.11 se conecta a un punto de acceso (AP), antes de que comience a pasar tráfico (tramas de datos inalámbricos), primero debe pasar el proceso de autenticación de sistema abierto 802.11 básico. A continuación, debe completarse el proceso de asociación. El proceso de autenticación de sistema abierto es como una conexión por cable en el

AP que el cliente selecciona. Este es un punto muy importante, ya que siempre es el cliente inalámbrico el que selecciona qué AP se prefiere, y basa la decisión en múltiples factores que varían entre los proveedores. Esta es la razón por la que el cliente comienza este proceso enviando la trama de autenticación al AP seleccionado, como se muestra más adelante en este documento. El AP no puede solicitar que usted establezca una conexión.

Una vez que el proceso de autenticación de sistema abierto se completa con éxito con una respuesta del AP ("cable conectado"), el proceso de asociación esencialmente termina la negociación de capa 2 (L2) 802.11 que establece el link entre el cliente y el AP. El AP asigna un ID de asociación al cliente si la conexión es exitosa, y lo prepara para pasar el tráfico o realizar un método de seguridad de nivel superior si está configurado en el SSID. El proceso de autenticación de sistema abierto consta de dos tramas de administración, así como el proceso de asociación. Las tramas de autenticación y asociación son **tramas de administración** inalámbrica, no tramas de datos, que son básicamente las utilizadas para el proceso de conexión con el AP.

A continuación se muestra una imagen de las tramas inalámbricas aéreas para este proceso:



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

> **Nota**: Si desea obtener información sobre el rastreo inalámbrico 802.11 y sobre los filtros/colores utilizados en Wireshark para las imágenes que aparecen en este documento, visite la publicación de la Comunidad de soporte de Cisco llamada 802.11 Sniffer image Analysis.

El cliente inalámbrico comienza con la trama de autenticación y el AP responde con otra trama de autenticación. El cliente luego envía la trama de Solicitud de asociación y el AP termina en una respuesta con la trama de Respuesta de asociación. Como se muestra en los paquetes DHCP, una vez que se pasan los procesos de autenticación y asociación del sistema abierto 802.11, el cliente comienza a pasar tramas de datos. En este caso, no hay ningún método de seguridad configurado en el SSID, por lo que el cliente comienza inmediatamente a enviar tramas de datos (en este caso, DHCP) que no están cifradas.

Como se muestra más adelante en este documento, si se habilita la seguridad en el SSID, existen tramas de entrada en contacto de autenticación y cifrado de nivel superior para el método de seguridad específico, justo después de la respuesta de asociación y antes de que se envíen las tramas de datos de tráfico del cliente, como DHCP, el protocolo de resolución de direcciones (ARP) y los paquetes de aplicaciones, que se cifran. Las tramas de datos sólo se pueden enviar hasta que el cliente esté totalmente autenticado y se negocien las claves de cifrado, según el método de seguridad configurado.

Basado en la imagen anterior, aquí están los mensajes que usted ve en las salidas del comando **debug client** del WLC cuando el cliente inalámbrico comienza una nueva asociación a la WLAN:

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

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

> **Nota**: El WLC debug utilizado para las salidas mostradas en este documento es el comando **debug client**, y los ejemplos sólo muestran algunos mensajes relevantes, no la salida completa. Para obtener más detalles sobre este comando de depuración, consulte el documento Introducción al cliente de depuración en controladores LAN inalámbricos (WLC).

Estos mensajes muestran las tramas de Solicitud de Asociación y Respuesta; las tramas de Autenticación iniciales no se registran en el WLC porque este intercambio de señales ocurre rápidamente en el nivel de AP en el CUWN.

¿Qué información aparece cuando el cliente se desplaza? El cliente siempre intercambia cuatro tramas de administración al establecer una conexión con un AP, que se debe al establecimiento de asociación del cliente o a un evento de roaming. El cliente sólo tiene una conexión establecida a un solo AP a la vez. La única diferencia en el intercambio de tramas entre una nueva conexión a la infraestructura WLAN y un evento de roaming es que las tramas de asociación de un evento de roaming se llaman tramas de **reasociación**, que indican que el cliente está en realidad en roaming desde otro AP sin intentos de establecer una nueva asociación con la WLAN. Estas tramas pueden contener diferentes elementos que se utilizan para negociar el evento de itinerancia; esto depende de la configuración, pero esos detalles están fuera del alcance de este documento.

Aquí hay un ejemplo del intercambio de tramas:

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

Estos mensajes aparecen en el resultado de la depuración:

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

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

Como se muestra, el cliente realiza correctamente un evento de roaming después de que se envía la solicitud de reasociación al nuevo AP y recibe la respuesta de reasociación del AP. Dado que el cliente ya tiene una dirección IP, las primeras tramas de datos son para paquetes ARP.

Si espera un evento de itinerancia, pero el cliente envía una Solicitud de asociación en lugar de una Solicitud de reasociación (que puede confirmar a partir de algunas imágenes y depuraciones similares a las explicadas anteriormente en este documento), entonces el cliente no está realmente en itinerancia. El cliente inicia una nueva asociación a la WLAN como si se hubiera

producido una desconexión e intenta volver a conectarse desde el principio. Esto puede suceder por varias razones, como cuando un cliente se aleja de las áreas de cobertura y luego encuentra un AP con suficiente calidad de señal para iniciar una asociación, pero normalmente indica un problema del cliente donde el cliente no inicia un evento de roaming debido a controladores, firmware o problemas de software.

> Nota: Puede consultar al proveedor del cliente inalámbrico para determinar la causa del problema.

Itinerancia con seguridad de nivel superior

Cuando el SSID se configura con una seguridad de nivel superior L2 sobre la autenticación básica de sistema abierto 802.11, se requieren más tramas para la asociación inicial y cuando se está en roaming. Los dos métodos de seguridad más comunes estandarizados e implementados para WLAN 802.11 se describen en este documento:

- WPA/WPA2-PSK (clave precompartida): autenticación de clientes con una clave precompartida.
- WPA/WPA2-EAP (protocolo de autenticación extensible): autenticación de clientes con un método 802.1X/EAP para validar credenciales más seguras mediante el uso de un servidor de autenticación, como certificados, nombre de usuario y contraseña, y tokens.

Es importante saber que, aunque estos dos métodos (PSK y EAP) autentican/validan los clientes de formas diferentes, ambos utilizan básicamente las mismas reglas WPA/WPA2 para el proceso de administración de claves. Tanto si la seguridad es WPA/WPA2-PSK o WPA/WPA2-EAP, el proceso conocido como protocolo de enlace de 4 vías WPA/WPA2 inicia la negociación de clave entre el WLC/AP y el cliente con una clave de sesión maestra (MSK) como material de clave original una vez que el cliente se valida con el método de autenticación específico utilizado.

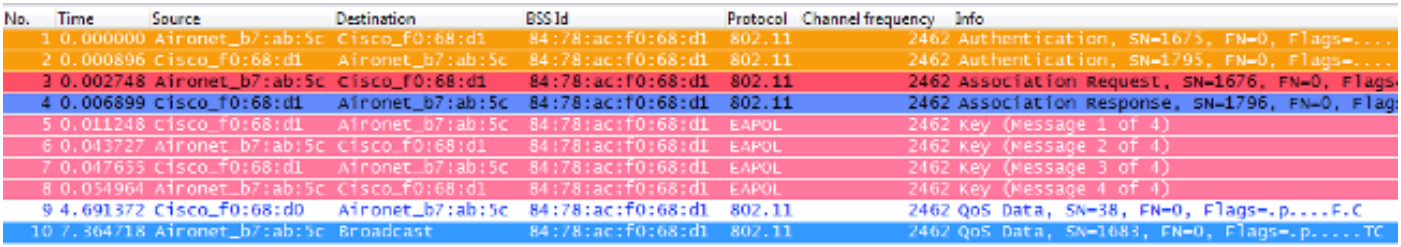
A continuación se muestra un resumen del proceso:

1. Una MSK se deriva de la fase de autenticación EAP cuando se utiliza la seguridad 802.1X/EAP, o de PSK cuando se utiliza WPA/WPA2-PSK como método de seguridad.
2. A partir de esta MSK, el cliente y el WLC/AP derivan la clave maestra en pares (PMK), y el WLC/AP genera una clave maestra en grupo (GMK).
3. Una vez que estas dos llaves maestras están listas, el cliente y el WLC/AP inician el intercambio de señales de 4 vías WPA/WPA2 (que se ilustra más adelante en este documento con algunas imágenes de la pantalla y los debugs) con las llaves maestras como las semillas para la negociación de las llaves de encripción reales.
4. Estas claves de encriptación finales se conocen como clave transitoria en pares (PTK) y clave transitoria de grupo (GTK). La PTK se deriva de la PMK y se utiliza para cifrar tramas de unidifusión con el cliente. La clave transitoria de grupo (GTK) se deriva del GMK y se utiliza para cifrar la multidifusión/difusión en este SSID/AP específico.

WPA/WPA2-PSK

Cuando WPA-PSK o WPA2-PSK se realiza mediante el protocolo de integridad de clave temporal (TKIP) o el estándar de cifrado avanzado (AES) para el cifrado, el cliente debe pasar por el proceso conocido como protocolo de enlace de 4 vías WPA tanto para la asociación inicial como

para la itinerancia. Como se ha explicado anteriormente, se trata básicamente del proceso de gestión de claves utilizado para que WPA/WPA2 derive las claves de encriptación. Sin embargo, cuando se realiza PSK, también se utiliza para verificar que el cliente tiene una clave previamente compartida válida para unirse a la WLAN. Esta imagen muestra el proceso de asociación inicial cuando se realiza WPA o WPA2 con PSK:



Como se muestra, después del proceso de asociación y autenticación del sistema abierto 802.11, hay cuatro tramas EAPOL del protocolo de enlace de 4 vías WPA, que son iniciadas por el AP con el **mensaje-1**, y terminadas por el cliente con el **mensaje-4**. Después de una entrada en contacto exitosa, el cliente comienza a pasar tramas de datos (como DHCP), que en este caso se cifran con las claves derivadas de la entrada en contacto de 4 vías (esta es la razón por la que no puede ver el contenido real y el tipo de tráfico de las imágenes inalámbricas).

> **Nota**: Las tramas EAPOL se utilizan para transportar todas las tramas de administración de claves y las tramas de autenticación 802.1X/EAP por el aire entre el AP y el cliente; se transmiten como tramas de datos inalámbricas.

Estos mensajes aparecen en los resultados de depuración:

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

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

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

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

*Dot1x_NW_MsgTask_4: Jun 21 19:30:05.309: 00:40:96:b7:ab:5c
```

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

Cuando está en roaming, el cliente básicamente realiza un seguimiento del mismo intercambio de tramas, donde se requiere el protocolo de enlace de 4 vías WPA para derivar nuevas claves de cifrado con el nuevo AP. Esto se debe a razones de seguridad establecidas por el estándar y al hecho de que el nuevo AP no conoce las claves originales. La única diferencia es que hay marcos de reasociación en lugar de marcos de asociación, como se muestra en esta imagen:

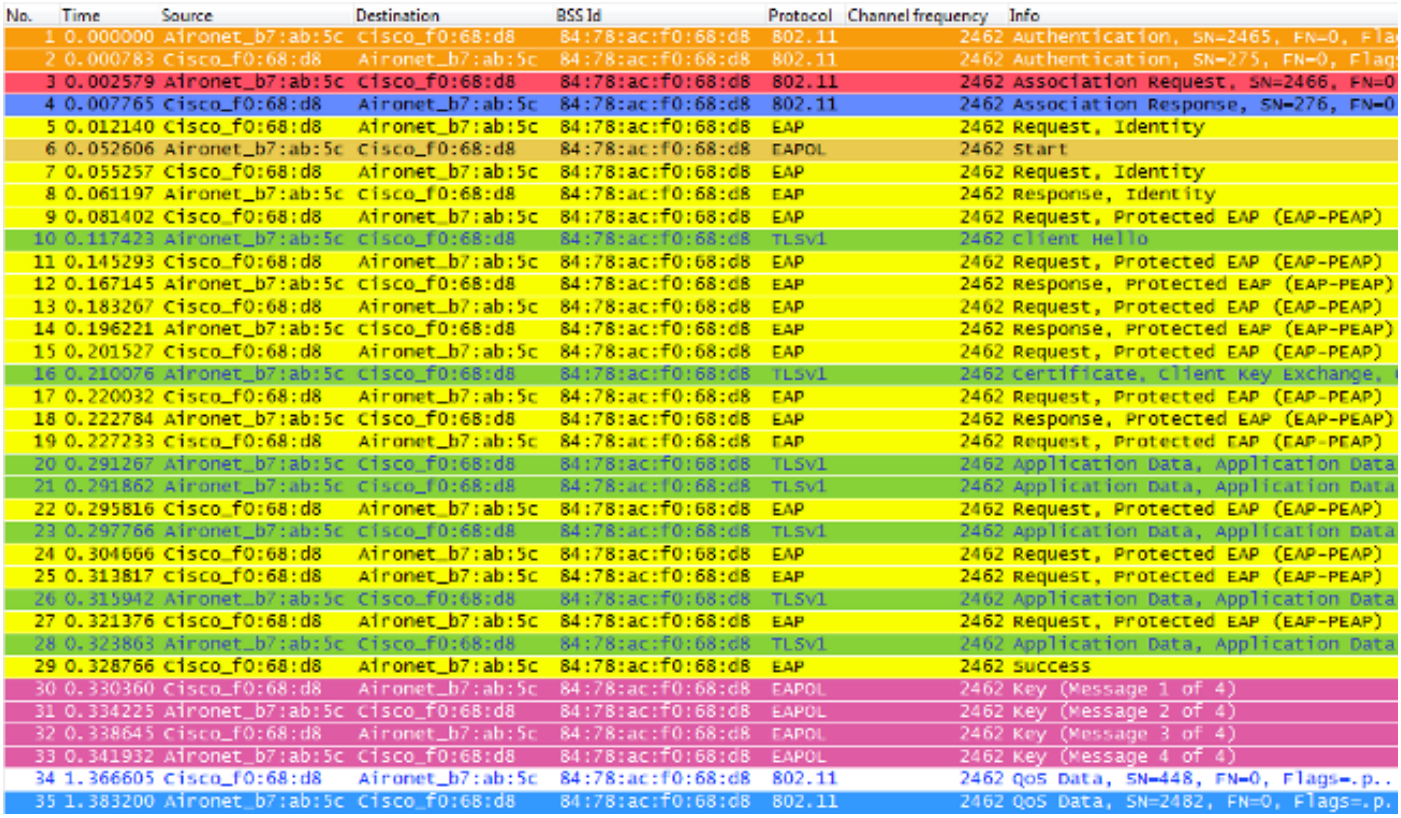
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3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

Verá los mismos mensajes en los resultados de depuración, pero el primer paquete del cliente es una reasociación en lugar de una asociación, como se ha mostrado y explicado anteriormente.

WPA/WPA2-EAP

Cuando se utiliza un método 802.1X/EAP para autenticar a los clientes en un SSID seguro, se requieren aún más tramas antes de que el cliente comience a pasar tráfico. Estas tramas adicionales se utilizan para autenticar las credenciales del cliente y, según el método EAP, puede haber entre cuatro y veinte tramas. Éstos se producen después de la asociación o reasociación, pero antes del protocolo de enlace de 4 vías WPA/WPA2, porque la fase de autenticación deriva la MSK utilizada como la semilla para la generación de la clave de cifrado final en el proceso de gestión de claves (protocolo de enlace de 4 vías).

Esta imagen muestra un ejemplo de las tramas intercambiadas por el aire entre el AP y el cliente inalámbrico en la asociación inicial cuando se realiza WPA con PEAPv0/EAP-MSCHAPv2:



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.

A veces este intercambio muestra más o menos tramas, que depende de múltiples factores, como el método EAP, las retransmisiones debido a problemas, el comportamiento del cliente (como las dos Solicitudes de identidad en este ejemplo, porque el cliente envía un **EAPOL START** después de que el AP envíe la primera Solicitud de identidad), o si el cliente ya intercambió el certificado con el servidor. Siempre que el SSID se configura para un método 802.1X/EAP, hay más tramas (para la autenticación) y, por lo tanto, se requiere más tiempo antes de que el cliente comience a enviar tramas de datos.

A continuación se muestra un resumen de los mensajes de depuración:

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

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

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

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

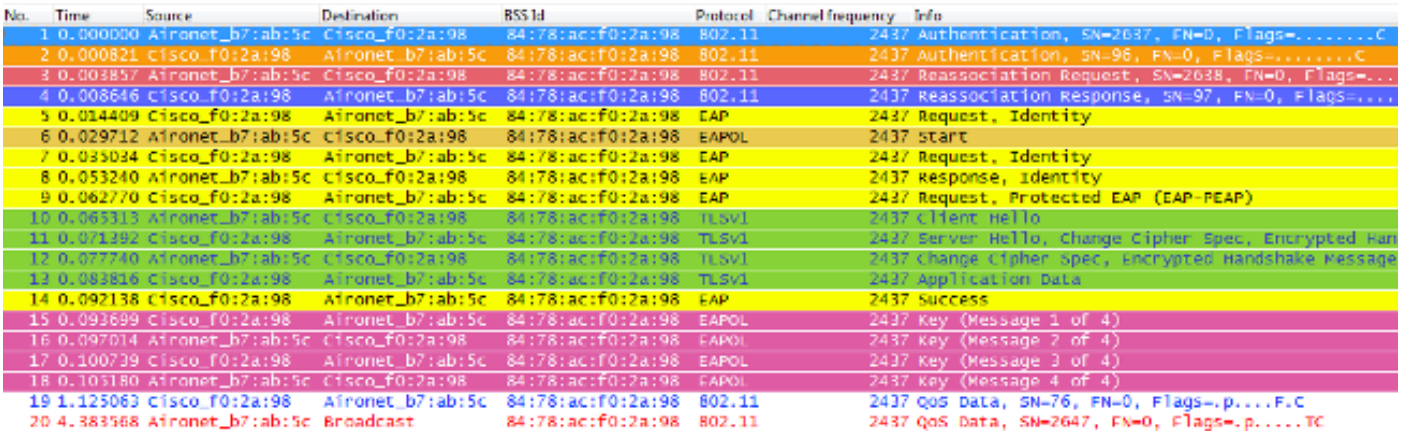
```
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.235: 00:40:96:b7:ab:5c
  Received EAPOL EAPPKT from mobile 00:40:96:b7:ab:5c
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.235: 00:40:96:b7:ab:5c
  Received Identity Response (count=2) from mobile 00:40:96:b7:ab:5c
!--- The client responds with an EAP Identity Response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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Cuando el cliente inalámbrico realiza un roaming regular aquí (el comportamiento normal, sin la implementación de un método de roaming seguro rápido), el cliente debe pasar exactamente por el mismo proceso y realizar una autenticación completa contra el servidor de autenticación, como se muestra en las imágenes. La única diferencia es que el cliente utiliza una Solicitud de Reasociación para informar al nuevo AP que está en realidad en roaming desde otro AP, pero el cliente todavía tiene que pasar por la validación completa y la generación de nueva clave:



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=2037, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=96, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2038, 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	0.033084	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	0.105180	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 4 of 4)
19	1.125063	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=76, FN=0, Flags=.p....F.C
20	4.383568	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=2647, FN=0, Flags=.p.....TC

Como se muestra, incluso cuando hay menos tramas que en la autenticación inicial (que es causada por múltiples factores, como se mencionó anteriormente), cuando el cliente se traslada a un nuevo AP, la autenticación EAP y los procesos de administración de claves WPA aún deben completarse para continuar pasando tramas de datos (incluso si el tráfico se envió activamente antes del roaming). Por lo tanto, si el cliente tiene una aplicación activa sensible a los retrasos (como aplicaciones de tráfico de voz o aplicaciones sensibles a los tiempos de espera), el usuario puede percibir problemas al desplazarse, como brechas de audio o desconexiones de aplicaciones. Esto depende del tiempo que tarda el proceso en que el cliente continúe enviando o recibiendo tramas de datos. Este retraso puede ser más largo, dependiendo de: el entorno de RF, la cantidad de clientes, el tiempo de ida y vuelta entre el WLC y los LAPs y con el servidor de autenticación, y otras razones.

A continuación se presenta un resumen de los mensajes de depuración para este evento de roaming (básicamente los mismos que los anteriores, por lo que estos mensajes no se describen con más detalle):

```
*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
```

Esta es la forma en que funcionan 802.1X/EAP y la estructura de seguridad WPA/WPA2. Con el fin de evitar el impacto de la aplicación/servicio en los retrasos de un evento de roaming regular, el sector de WiFi desarrolla e implementa varios métodos de roaming rápido seguro para acelerar el proceso de roaming cuando se utiliza seguridad en WLAN/SSID. Los clientes se enfrentan a cierta latencia cuando continúan pasando el tráfico mientras se desplazan entre los AP a través de la implementación de la seguridad de alto nivel en el WLAN. Esto se debe a la autenticación EAP y a los intercambios de tramas de administración de claves requeridos por la configuración de seguridad, como se explicó anteriormente.

Es importante comprender que el término "roaming seguro rápido" es simplemente el término que utiliza el sector en referencia a la implementación de un método o esquema que acelera el proceso de roaming cuando se configura la seguridad en la WLAN. En la siguiente sección se explican los diferentes métodos/esquemas de roaming de seguridad rápida que están disponibles para las WLAN y que son compatibles con CUWN.

Roaming seguro y rápido con CCKM

Cisco Centralized Key Management (CCKM) es el primer método de roaming rápido y seguro desarrollado e implementado en las WLAN empresariales, creado por Cisco como la solución utilizada para mitigar los retrasos explicados hasta el momento, cuando se utiliza la seguridad 802.1X/EAP en la WLAN. Al tratarse de un protocolo propiedad de Cisco, solo es compatible con los dispositivos de infraestructura WLAN de Cisco y los clientes inalámbricos (de varios proveedores) compatibles con Cisco Compatible Extension (CCX) para CCKM.

CCKM se puede implementar con los distintos métodos de encriptación disponibles para las

WLAN, entre los que se incluyen: WEP, TKIP y AES. También es compatible con la mayoría de los métodos de autenticación 802.1X/EAP utilizados para las WLAN, según la versión de CCX admitida por los dispositivos.

> **Nota**: Para obtener una descripción general del contenido de las funciones admitidas por las diferentes versiones de la especificación CCX (que incluye los métodos EAP admitidos), consulte el documento Versiones y funciones de CCX y verifique la versión exacta de CCX admitida por los clientes inalámbricos (si son compatibles con CCX), de modo que pueda confirmar si se puede implementar el método de seguridad que desea utilizar con CCKM.

Esta imagen inalámbrica proporciona un ejemplo de las tramas intercambiadas tras la asociación inicial cuando se realiza CCKM con TKIP como cifrado y PEAPv0/EAP-MSCHAPv2 como método 802.1X/EAP. Básicamente, se trata del mismo intercambio que si se realizara WPA/TKIP con PEAPv0/EAP-MSCHAPv2, pero esta vez CCKM entre el cliente y la infraestructura se negocia de modo que utilicen diferentes jerarquías de claves y métodos de caché para realizar la itinerancia segura rápida cuando el cliente debe desplazarse:

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

A continuación se muestra un resumen de los mensajes de depuración (con algunos intercambios EAP eliminados para reducir el resultado):

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

*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Processing WPA IE type 221, length 22 for mobile
  00:40:96:b7:ab:5c
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  CCKM: Mobile is using CCKM
!--- The WLC/AP finds an Information Element that claims CCKM
```

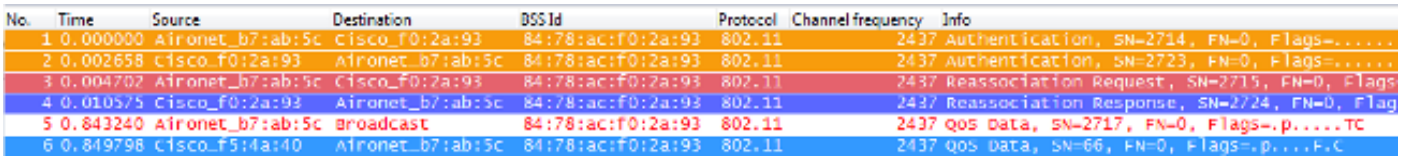
support on the Association request that is sent from the client.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Creating a PKC PMKID Cache entry for station 00:40:96:b7:ab:5c
  (RSN 0)<br/ >
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
```

```
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 0
*Dot1x_NW_MsgTask_4: Jun 25 15:41:41.841: 00:40:96:b7:ab:5c
  CCKM: Create a global PMK cache entry
!--- WLC creates a global PMK cache entry for this client,
     which is for CCKM in this case.

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

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

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

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

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

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

Con CCKM, la asociación inicial a la WLAN es similar a la WPA/WPA2 normal, donde una MSK (también conocida como clave de sesión de red [NSK]) se deriva mutuamente con el cliente y el servidor RADIUS. Esta clave primaria se envía del servidor al WLC después de una autenticación exitosa, y se almacena en caché como la base para la derivación de todas las claves subsiguientes durante la vida útil de la asociación del cliente con esta WLAN. A partir de aquí, el WLC y el cliente derivan la información semilla que se utiliza para el roaming seguro rápido basado en CCKM, esto pasa a través de un protocolo de enlace de 4 vías similar al de WPA/WPA2, para derivar las claves de cifrado de unidifusión (PTK) y multidifusión/difusión (GTK) con el primer AP.

La gran diferencia se nota cuando se está en roaming. En este caso, el cliente CCKM envía una sola trama de petición de reasociación al AP/WLC (que incluye un MIC y un número aleatorio que aumenta secuencialmente), y proporciona suficiente información (que incluye la nueva dirección MAC del AP **-BSSID-**) para derivar la nueva PTK. Con esta Solicitud de Reasociación, el WLC y el nuevo AP también tienen suficiente información para derivar la nueva PTK, por lo que simplemente responden con una Respuesta de Reasociación. El cliente ahora puede continuar pasando tráfico, como se muestra en esta imagen:



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

Aquí está un resumen de las depuraciones del WLC para este evento de roaming:

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

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

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

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

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

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

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

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

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 0

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Creating a PKC PMKID Cache entry for station
```

```
  00:40:96:b7:ab:5c (RSN 0) on BSSID 84:78:ac:f0:2a:93
!--- The new PMKID cache entry is created for this new
     AP-to-client association.

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

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

Como se muestra, se realiza un roaming de seguridad rápido mientras se evitan las tramas de autenticación EAP y se estrechan aún más las manos, porque las nuevas claves de cifrado aún se derivan, pero se basan en el esquema de negociación CCKM. Esto se completa con las tramas de reasociación de roaming y la información previamente almacenada en caché por el cliente y el WLC.

FlexConnect con CCKM

- Se admite la autenticación central. Esto incluye el switching de datos local y central. Los puntos de acceso deben formar parte del mismo grupo de FlexConnect.
- Se admite la autenticación local flexible. En el modo conectado, la memoria caché se puede distribuir desde el AP al controlador y luego al resto de los AP en el grupo FlexConnect.
- Se admite el modo independiente. Si la memoria caché ya está presente en el AP (debido a la distribución anterior), la itinerancia rápida funcionará. La nueva autenticación en modo independiente no admite el roaming de seguridad rápida.

Ventajas con CCKM

- CCKM es el método de roaming rápido y seguro más rápido implementado principalmente en WLANs empresariales. Los clientes no necesitan pasar por un protocolo de enlace de administración de claves para obtener nuevas claves cuando se produce un movimiento entre los AP, y nunca más se les requiere realizar una autenticación 802.1X/EAP completa con los nuevos AP durante la vida útil del cliente en esta WLAN.
- CCKM admite todos los métodos de encriptación disponibles en el estándar 802.11 (WEP, TKIP y AES), además de algunos métodos de propiedad heredados de Cisco que todavía se utilizan en clientes heredados.

Desventajas con CCKM

- CCKM es un método propiedad de Cisco que limita la implementación y la compatibilidad con la infraestructura WLAN de Cisco y los clientes inalámbricos CCX.
- La versión 5 de CCX no se ha adoptado de forma generalizada, por lo que muchos clientes inalámbricos CCX no admiten CCKM con WPA2/AES (principalmente porque la mayoría de

ellos ya admiten CCKM con WPA/TKIP, que sigue siendo muy seguro).

Itinerancia segura y rápida con almacenamiento en caché de PMKID/almacenamiento en caché de claves persistente

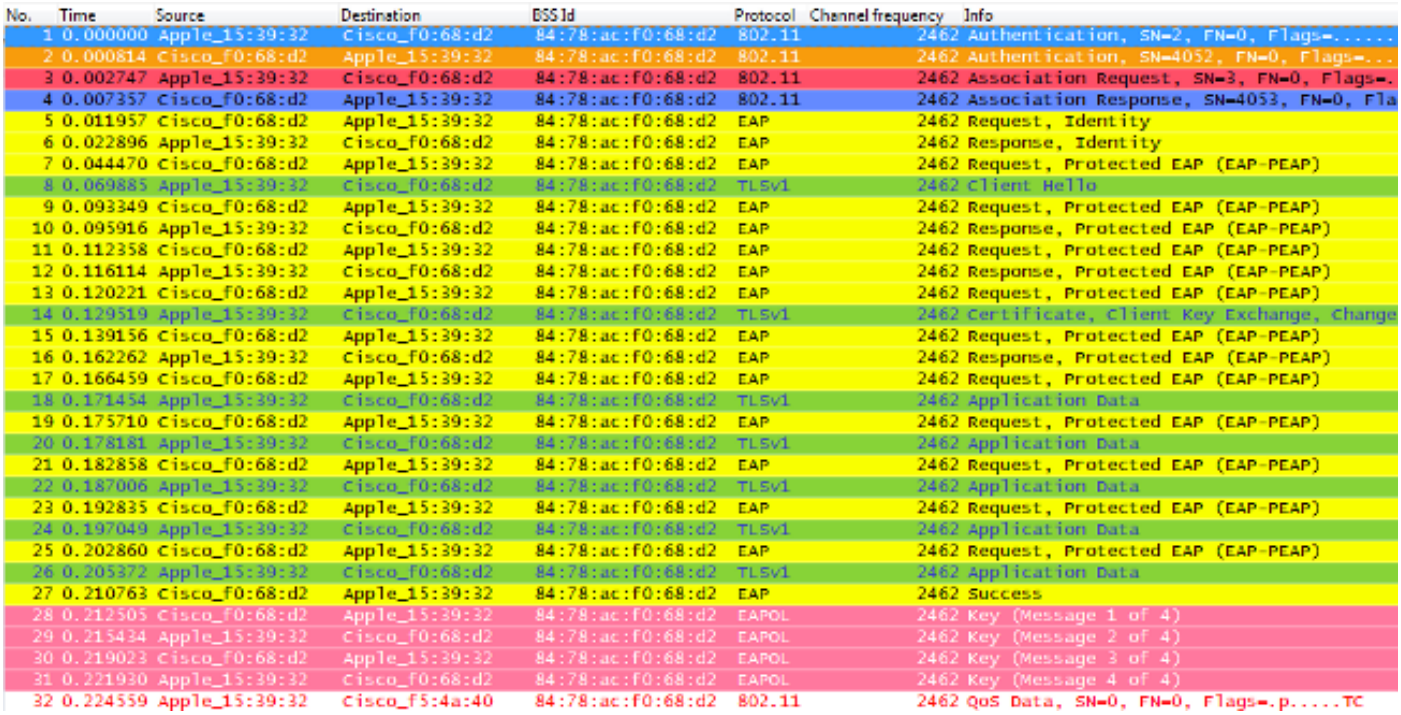
Pareja piensa que el almacenamiento en caché de Key ID (PMKID), o **Sticky Key Caching (SKC)**, es el primer método de roaming rápido y seguro sugerido por el estándar IEEE 802.11 dentro de la enmienda de seguridad 802.11i, donde el objetivo principal es estandarizar un alto nivel de seguridad para las WLAN. Esta técnica de roaming de seguridad rápida se agregó como un método opcional para los dispositivos WPA2 con el fin de mejorar el roaming cuando se implementó esta seguridad.

Esto es posible porque, cada vez que un cliente está completamente autenticado con EAP, el cliente y el Servidor de autenticación derivan un MSK, que se utiliza para derivar el PMK. Esto se utiliza como la semilla para el protocolo de enlace de 4 vías WPA2 para derivar la clave de cifrado de unidifusión (PTK) final que se utiliza para la sesión (hasta que el cliente se traslada a otro AP o caduca la sesión); por lo tanto, este método evita la fase de autenticación EAP al desplazarse porque reutiliza la PMK original almacenada en caché por el cliente y el AP. El cliente sólo tiene que pasar por el protocolo de enlace de 4 vías WPA2 para obtener nuevas claves de cifrado.

Este método no está ampliamente implementado como el método de roaming de seguridad rápida estándar 802.11 recomendado debido principalmente a estas razones:

- Este método es opcional y no es compatible con todos los dispositivos WPA2, ya que el propósito de la modificación de 802.11i no se refiere a la itinerancia segura y rápida, y el IEEE ya ha trabajado en otra modificación para estandarizar la itinerancia segura y rápida para WLAN (802.11r, que se trata más adelante en este documento).
- Este método tiene una gran limitación en su implementación: los clientes inalámbricos solo pueden realizar roaming seguro rápido cuando regresan a un AP donde previamente se habían autenticado/conectado.

Con este método, la asociación inicial a cualquier AP es como una autenticación regular de primera vez a la WLAN, donde la autenticación completa 802.1X/EAP contra el servidor de autenticación y el intercambio de señales de 4 vías para la generación de claves debe ocurrir antes de que el cliente pueda enviar tramas de datos, como se muestra en esta imagen de pantalla:



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

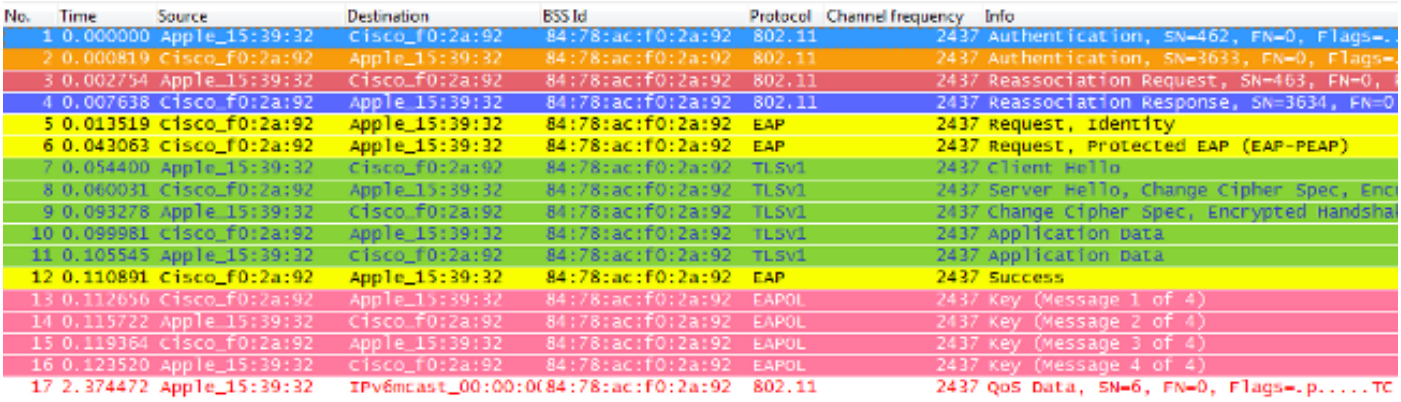
Las depuraciones revelan el mismo intercambio de tramas de autenticación EAP que el resto de los métodos en la autenticación inicial a la WLAN, con algunas salidas agregadas con respecto a las técnicas de almacenamiento en caché de claves utilizadas aquí. Estos resultados de depuración se cortan para mostrar principalmente la nueva información, no todo el intercambio de tramas EAP, porque básicamente la misma información se intercambia cada vez para la autenticación del cliente contra el servidor de autenticación. Esto se ha demostrado hasta ahora y se correlaciona con las tramas de autenticación EAP que se muestran en las imágenes de paquete, por lo que la mayoría de los mensajes EAP se eliminan de las salidas de depuración por motivos de simplicidad:

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

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

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

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

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

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

```
  (EAP Id 1)

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

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

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

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

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

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

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

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

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

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

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

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

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

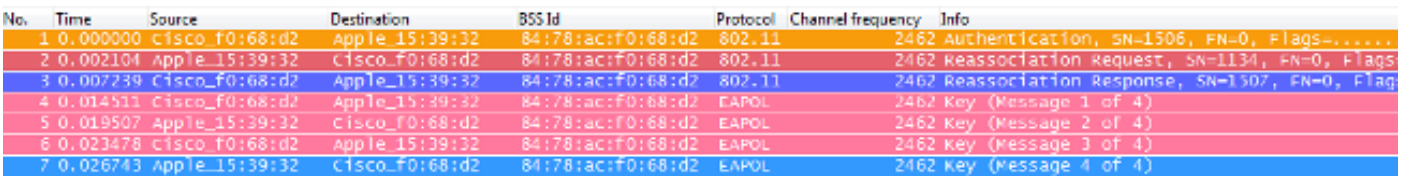
```
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from mobile
  ec:85:2f:15:39:32
!--- Message-2 of the WPA/WPA-2 4-Way handshake is successfully
      received from the client.

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

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

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

Con este método, el AP y el cliente inalámbrico almacenan en caché los PMK de las asociaciones seguras ya establecidas. Por lo tanto, si el cliente inalámbrico se traslada a un nuevo AP donde nunca se ha asociado, el cliente debe realizar una autenticación EAP completa nuevamente, como se muestra en esta imagen donde el cliente se traslada a un nuevo AP:



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

Sin embargo, si el cliente inalámbrico vuelve a un AP donde se realizó una asociación/autenticación previa, el cliente envía una trama de Solicitud de Reasociación que enumera varios PMKID, que informa al AP de los PMK almacenados en caché de todos los AP donde el cliente se ha autenticado previamente. Por lo tanto, dado que el cliente está regresando a un AP que también tiene un PMK almacenado en caché para este cliente, el cliente no necesita reautenticarse a través de EAP para derivar un nuevo PMK. El cliente simplemente pasa a través del protocolo de enlace WPA2 de 4 vías para derivar las nuevas claves de encriptación transitorias:



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)

> **Nota**: Esta imagen no muestra la primera trama de autenticación de sistema abierto 802.11 del cliente, pero esto no se debe al método implementado, ya que esta trama siempre es necesaria. La razón es que el adaptador o el software de imagen de paquete inalámbrico utilizado para detectar las tramas aéreas de este ejemplo no crea imágenes de esta trama específica, pero se deja así en el ejemplo con fines educativos. Tenga en cuenta que existe la posibilidad de que esto ocurra cuando realice imágenes de paquetes por aire; algunas tramas pueden ser perdidas por la imagen, pero en realidad se intercambian entre el cliente y el AP. De lo contrario, el roaming nunca comienza en este ejemplo.

Aquí está un resumen de las depuraciones del WLC para este método de roaming seguro rápido:

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

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

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

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

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

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

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

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Setting active key cache index 1 ---> 0

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:68:d2(status 0) ApVapId 3 Slot 0
```

```
!--- The Reassociation Response is sent to the client, which
      validates the fast-roam with SKC.

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

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

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

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

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

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

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

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

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

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

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

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

FlexConnect con almacenamiento en caché de PMKID/almacenamiento en caché de clave fija

- Cuando utiliza este método en una configuración de FlexConnect, podría funcionar y el comportamiento puede parecer similar a lo que se explicó anteriormente si utiliza la autenticación central de vuelta al WLC (con conmutación central o local); sin embargo, este método SKC no es compatible con FlexConnect.

- Este método sólo es oficialmente compatible en CUWN con AP de modo local, no en FlexConnect u otros modos.

Pros con PMKID Caching / Sticky Key Caching

Este método se puede implementar localmente mediante AP autónomos independientes, sin la necesidad de un dispositivo centralizado para administrar las claves almacenadas en caché.

Inconvenientes del almacenamiento en caché de PMKID/almacenamiento en caché de clave persistente

- Como se mencionó anteriormente en este documento, la principal limitación de este método es que el cliente solo puede realizar roaming seguro rápido cuando vuelve a un AP donde previamente se asoció/autenticó. Si se traslada a un nuevo AP, el cliente debe completar la autenticación EAP completa otra vez.
- El cliente inalámbrico y los AP deben recordar todos los PMK derivados en cada nueva autenticación, por lo que esta función normalmente se limita a una cierta cantidad de PMK que se almacenan en caché. Dado que este límite no está claramente definido por el estándar, los proveedores pueden definir diferentes límites en sus implementaciones de SKC. Por ejemplo, los controladores WLAN de Cisco pueden actualmente almacenar en caché los PMK de un cliente para hasta ocho AP. Si un cliente se traslada a más de ocho AP por sesión, los AP más antiguos se eliminan de la lista de caché para almacenar las entradas recién almacenadas en caché.
- Este método es opcional y todavía no es compatible con muchos dispositivos WPA2; por lo tanto, este método no se adopta ni implementa de forma generalizada.
- SKC no se soporta cuando realiza el roaming entre controladores, que ocurre cuando se mueve entre los AP administrados por diferentes WLC, incluso si están en el mismo grupo de movilidad.

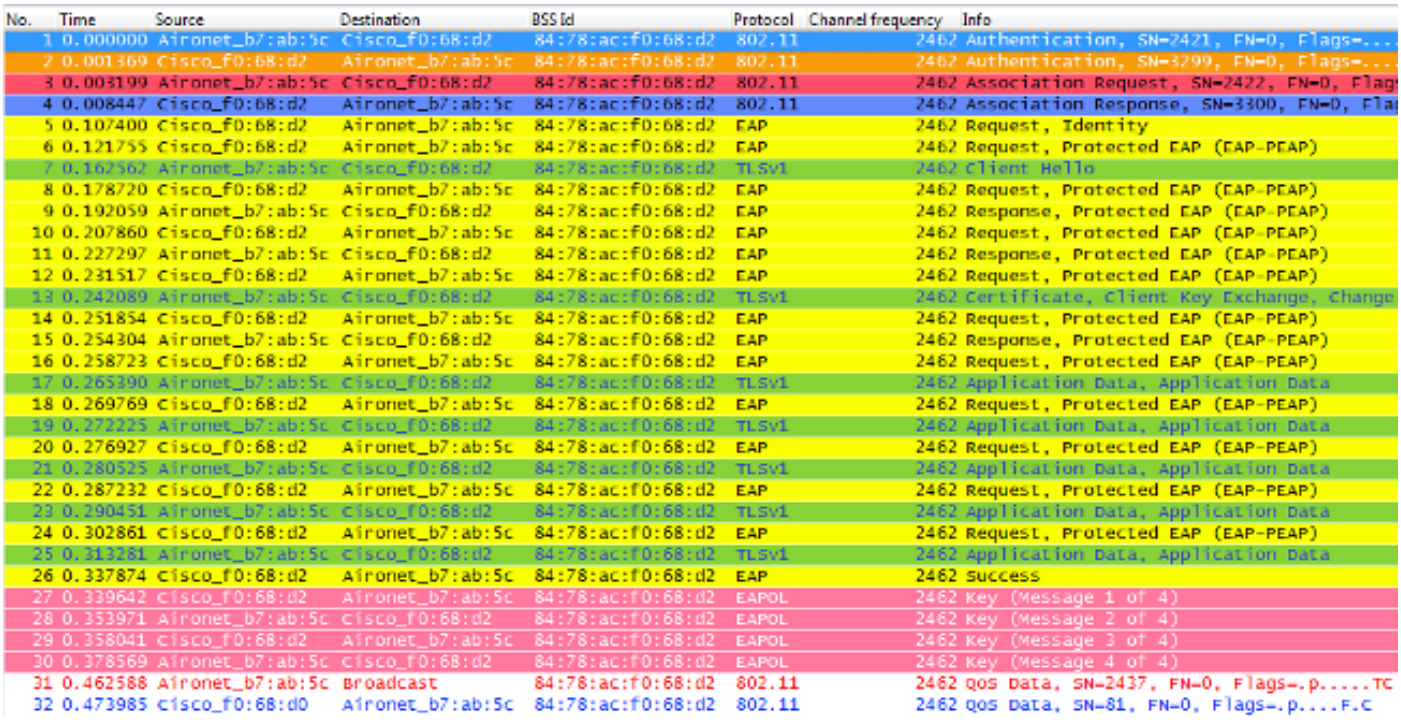
Itinerancia segura y rápida con almacenamiento en caché de claves oportunista

El almacenamiento en caché de claves oportunista (OKC), también conocido como almacenamiento en caché de claves proactivo (PKC) (este término se explica con más detalle en una nota que sigue), es básicamente una mejora del método de almacenamiento en caché PMKID WPA2 descrito anteriormente, por lo que también se denomina almacenamiento en caché PMKID proactivo/oportunista. Por lo tanto, es importante tener en cuenta que este no es un método de roaming de seguridad rápida definido por el estándar 802.11 y no es compatible con muchos dispositivos, pero al igual que el almacenamiento en caché PMKID, funciona con WPA2-EAP.

Esta técnica permite que el cliente inalámbrico y la infraestructura WLAN almacenen en caché sólo un PMK durante la duración de la asociación del cliente con esta WLAN (derivada de la MSK después de la autenticación 802.1X/EAP inicial con el servidor de autenticación), incluso cuando se desplaza entre varios AP, ya que todos comparten el PMK original que se utiliza como la semilla en todos los handshakes de 4 direcciones WPA2. Esto sigue siendo necesario, al igual que en SKC, para generar nuevas claves de cifrado cada vez que el cliente se reasocia con los

AP. Para que los AP compartan esta única PMK original de la sesión del cliente, todos deben estar bajo algún tipo de control administrativo, con un dispositivo centralizado que almacene en caché y distribuya la PMK original para todos los AP. Esto es similar al CUWN, donde el WLC realiza este trabajo para todos los LAPs bajo su control, y utiliza los grupos de movilidad para manejar este PMK entre los WLCs múltiples; por lo tanto, esto es una limitación en los entornos AP autónomos.

Con este método, al igual que en el almacenamiento en caché de PMKID (SKC), la asociación inicial a cualquier AP es una autenticación regular por primera vez a la WLAN, donde debe completar toda la autenticación 802.1X/EAP contra el servidor de autenticación y el protocolo de enlace de 4 vías para la generación de claves antes de poder enviar tramas de datos. Esta es una imagen de pantalla que ilustra esto:



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

Las salidas de debug muestran básicamente el mismo intercambio de tramas de autenticación EAP que el resto de los métodos descritos en este documento sobre la autenticación inicial a la WLAN (como se muestra en las imágenes), junto con la adición de algunas salidas que se refieren a las técnicas de almacenamiento en caché de claves utilizadas por el WLC aquí. Este resultado de depuración también se corta para mostrar solamente la información relevante:

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

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

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Received RSN IE with 0 PMKIDs from mobile
  00:40:96:b7:ab:5c
!--- Since this is an initial association, the Association
```

Request comes without any PMKID.

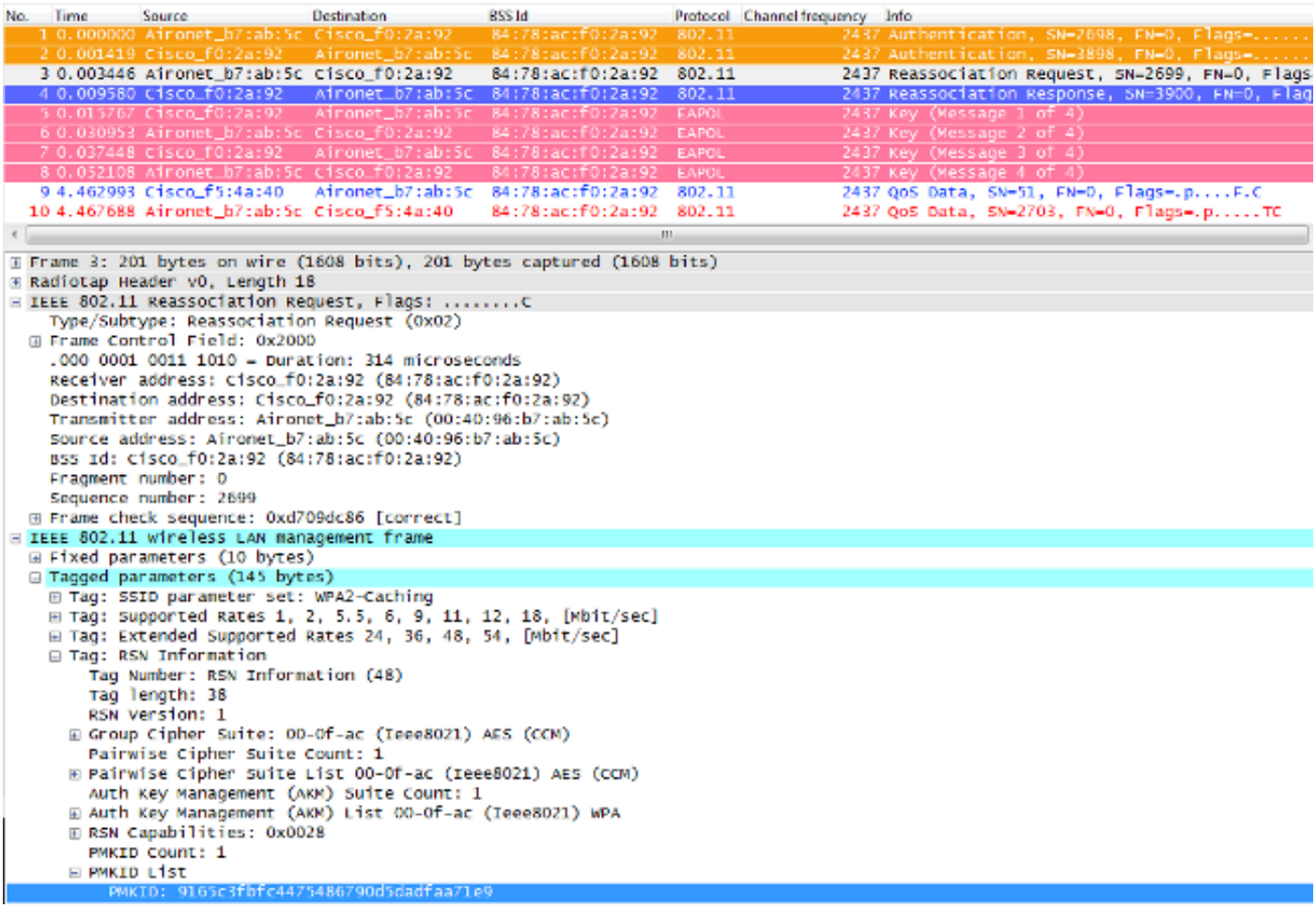
```
*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Setting active key cache index 0 ---> 8

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

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

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  PMK sent to mobility group
```

```
!--- The PMK cache entry for this client is shared with the
      WLCs on the mobility group.

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

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

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

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

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

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

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

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

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

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

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

Con este método, el cliente inalámbrico y el WLC (para todos los AP administrados) almacenan en caché el único PMK original de la asociación segura que se establece inicialmente. Básicamente, cada vez que el cliente inalámbrico se conecta a un AP específico, un PMKID se trocea en función de: la dirección MAC del cliente, la dirección MAC del AP (BSSID del WLAN), y el PMK derivado con ese AP. Por lo tanto, dado que OKC almacena en caché el mismo PMK original para todos los AP y el cliente específico, cuando este cliente (re)asocia a otro AP, el único valor que cambia para hacer hash con el nuevo PMKID es la nueva dirección MAC del AP.

Cuando el cliente inicia el roaming a un nuevo AP y envía la trama de Solicitud de reasociación, agrega el PMKID en el elemento de información RSN WPA2 si desea informar al AP que un PMK almacenado en caché se utiliza para el roaming seguro rápido. Ya conoce la dirección MAC del BSSID (AP) por donde se traslada, entonces el cliente simplemente hace un hashes con el nuevo PMKID que se utiliza en esta Solicitud de Reasociación. Cuando el AP recibe esta solicitud del cliente, también hace un hashes del PMKID con los valores que ya tiene (el PMK almacenado en

caché, la dirección MAC del cliente y su propia dirección MAC del AP), y responde con la respuesta de reasociación exitosa que confirma que los PMKIDs coinciden. El PMK almacenado en caché se puede utilizar como la semilla que inicia un protocolo de enlace de 4 vías WPA2 para derivar las nuevas claves de cifrado (y omitir EAP):



En esta imagen, se selecciona y expande el marco de solicitud de reasociación del cliente para que pueda ver más detalles del marco. La información de la dirección MAC y también el elemento de información Red de seguridad sólida (RSN, según 802.11i - WPA2), donde se muestra información sobre la configuración de WPA2 utilizada para esta asociación (resaltado es el PMKID obtenido de la fórmula con hash).

Aquí está un resumen de los debugs del WLC para este método de roaming seguro rápido con OKC:

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

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Processing RSN IE type 48, length 38 for mobile
  00:40:96:b7:ab:5c
!--- The WLC/AP finds and Information Element that claims
      PMKID Caching support on the Association request that
      is sent from the client.

*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Received RSN IE with 1 PMKIDs from mobile
  00:40:96:b7:ab:5c
```

```
!--- The Reassociation Request from the client comes with
      one PMKID.

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

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

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

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

*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: BSSID =  (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 90
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: realAA =  (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 92
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: Find PMK in cache: PMKID =  (16)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: AA (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 84 78 ac f0 2a 92
*apfMsConnTask_2: Jun 21 21:48:50.563:
  CCKM: SPA (6)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 00 40 96 b7 ab 5c
*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Adding BSSID 84:78:ac:f0:2a:92 to PMKID cache at
  index 0 for station 00:40:96:b7:ab:5c
*apfMsConnTask_2: Jun 21 21:48:50.563:
  New PMKID: (16)
*apfMsConnTask_2: Jun 21 21:48:50.563:
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9
*apfMsConnTask_2: Jun 21 21:48:50.563: 00:40:96:b7:ab:5c
  Computed a valid PMKID from MSCB PMK cache for mobile
  00:40:96:b7:ab:5c
!--- The new PMKID is computed and validated to match the
      one provided by the client, which is also computed with
      the same information. Hence, the fast-secure roam is
      possible.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Como se muestra al principio de las depuraciones, el PMKID debe calcularse después de que se reciba la solicitud de reasociación del cliente. Esto es necesario para validar el PMKID y confirmar que el PMK almacenado en caché se utiliza con el protocolo de enlace de 4 vías WPA2 para derivar las claves de cifrado y finalizar el roaming de seguridad rápida. No confunda las entradas de CCKM en los debugs; esto no se utiliza para realizar CCKM, sino OKC, como se explicó

anteriormente. CCKM aquí es simplemente un nombre utilizado por el WLC para esas salidas, como el nombre de una función que maneja los valores para calcular el PMKID.

FlexConnect con almacenamiento en caché de claves oportunista

- Se admite la autenticación central. Esto incluye conmutación de datos local y central. Si el AP es parte del mismo grupo de FlexConnect, la itinerancia de seguridad rápida es controlada por el AP; de lo contrario, la itinerancia de seguridad rápida es controlada por el controlador. **Nota**: Esta configuración puede funcionar si los AP no están en el mismo grupo FlexConnect, pero no es una configuración recomendada o admitida.
- Se admite la autenticación local flexible. En el modo conectado, la memoria caché se puede distribuir desde el AP al controlador y luego al resto de los AP en el grupo FlexConnect.
- Se admite el modo independiente. Si la memoria caché ya está presente en el AP (debido a la distribución anterior), el roaming seguro rápido funcionará. La nueva autenticación en modo independiente no admite el roaming de seguridad rápida.

Ventajas con Opportunistic Key Caching

- El cliente inalámbrico y la infraestructura WLAN no necesitan recordar varios PMKID, sino simplemente almacenar en caché el PMK original de la autenticación inicial a la WLAN. Luego, debe volver a hash el PMKID adecuado (utilizado en la Solicitud de Reasociación) requerido con cada asociación segura de AP para validar el roaming seguro rápido.
- Aquí, el cliente inalámbrico realiza un roaming seguro rápido a un nuevo AP en el mismo WLAN/SSID, incluso si nunca se asoció con ese AP (no es el caso en SKC). Mientras el cliente realice la autenticación 802.1X/EAP inicial con un AP administrado por la implementación centralizada que administra la memoria caché PMK para todos los AP para los que el cliente se desplaza, no se requieren más autenticaciones completas para el resto de la vida útil del cliente en esta WLAN.

Inconvenientes con el almacenamiento en caché de claves oportunista

- Este método sólo se implementa en un entorno centralizado donde todos los AP están bajo algún tipo de control administrativo (como un Controlador WLAN) que es responsable de almacenar en caché y compartir el PMK original de la sesión del cliente. Por lo tanto, esto es una limitación en los entornos AP autónomos.
- Las técnicas que se aplican en este método no se sugieren ni se describen en el estándar 802.11, por lo que la compatibilidad varía ampliamente de un dispositivo a otro. Sin embargo, este sigue siendo el método más adoptado mientras se esperaba 802.11r.

Nota sobre el término "Proactive Key Caching"

El almacenamiento en caché de claves proactivo (o PKC) se conoce como OKC (Opportunistic Key Caching), y los dos términos se utilizan indistintamente cuando describen el mismo método aquí explicado. Sin embargo, este era solo un término que Airspace utilizó en 2001 para un antiguo método de almacenamiento en caché de claves, que luego fue utilizado por el estándar 802.11i como base para la "Preautenticación" (otro método de roaming seguro rápido explicado brevemente a continuación). PKC no es Preauthentication ni OKC (Opportunistic Key Caching), pero cuando usted escucha o lee acerca de PKC, la referencia es básicamente a OKC, y no a

Preauthentication.

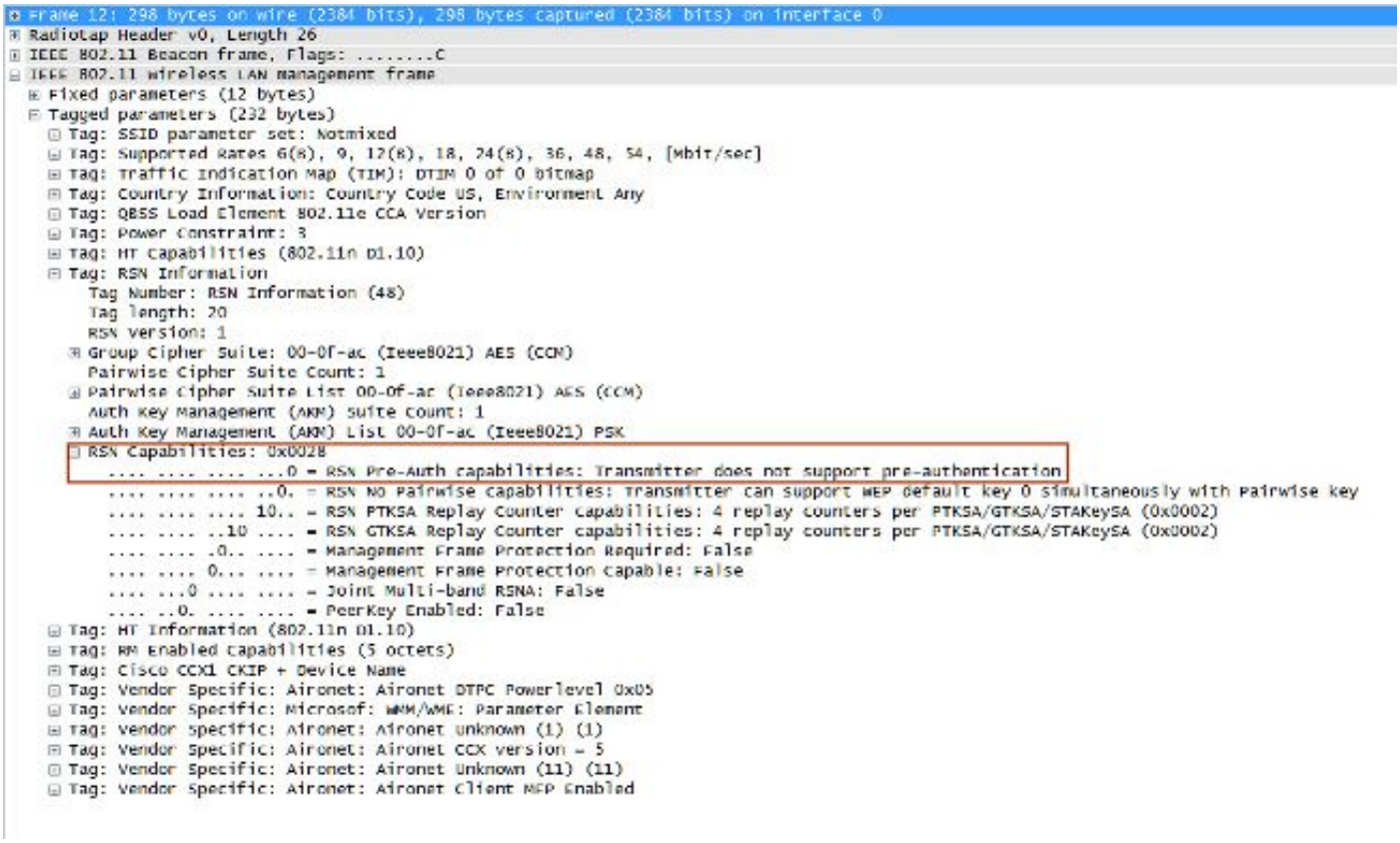
Itinerancia rápida y segura con autenticación previa

Este método también lo sugiere el estándar IEEE 802.11 en la enmienda de seguridad 802.11i, por lo que también funciona con WPA2, pero es el único método de roaming seguro rápido que no es compatible con la infraestructura WLAN de Cisco. Por esta razón, se explica brevemente aquí y sin resultados.

Con la autenticación previa, los clientes inalámbricos pueden autenticarse con varios AP a la vez mientras están asociados con el AP actual. Cuando esto ocurre, el cliente envía las tramas de autenticación EAP al AP actual donde está conectado, pero está destinado a los otros AP donde el cliente quiere realizar la autenticación previa (AP vecinos que son posibles candidatos para el roaming). El AP actual envía estas tramas a los AP de destino a través del sistema de distribución. El nuevo AP realiza una autenticación completa contra el servidor RADIUS para este cliente, por lo que se completa un nuevo protocolo de enlace de autenticación EAP, y este nuevo AP actúa como el autenticador.

La idea es realizar la autenticación y derivar el PMK con los AP vecinos antes de que el cliente realmente se traslade a ellos, por lo que cuando es el momento de trasladarse, el cliente ya está autenticado y con un PMK ya almacenado en caché para esta nueva asociación segura de AP a cliente, por lo que solo necesitan realizar el intercambio de señales de 4 vías y experimentar una itinerancia rápida después de que el cliente envíe su solicitud de reasociación inicial.

Esta es una imagen de una baliza AP que muestra el campo RSN IE que anuncia el soporte para la autenticación previa (esta es de un Cisco AP, donde se confirma que la autenticación previa no es compatible):



Pros con autenticación previa

Hay un PMK para cada asociación segura de AP a cliente, que podría considerarse una ventaja de seguridad en caso de que un AP se vea comprometido y las claves sean robadas (no se pueden utilizar con otros AP). Sin embargo, esta ventaja de seguridad es manejada por la infraestructura WLAN de diferentes maneras en otros métodos.

Contras con autenticación previa

- Debido a que hay un PMK por AP, los clientes tienen un límite en la cantidad de AP que se pueden autenticar previamente.
- Cada vez que un cliente realiza la preautenticación con un nuevo AP, hay un intercambio de autenticación EAP completo, lo que significa más carga en la red y en el servidor de autenticación.
- La mayoría de los clientes inalámbricos no admiten este método, ya que nunca se adoptó demasiado (OKC fue más adoptado).

Itinerancia rápida y segura con 802.11r

La técnica de roaming de seguridad rápida basada en la enmienda 802.11r (oficialmente denominada **Transición rápida de BSS** por el estándar 802.11 y conocida como **FT**) es el primer método ratificado oficialmente (en 2008) por el IEEE para el estándar 802.11 como la solución para realizar transiciones rápidas entre los AP (conjuntos de servicios básicos o BSS), que define claramente la jerarquía de claves que se utiliza cuando se manejan y almacenan en caché claves en una WLAN. Sin embargo, su adopción ha sido lenta, principalmente debido a las otras soluciones ya disponibles cuando se requerían transiciones rápidas, como con las implementaciones de VoWLAN cuando se utiliza con uno de los métodos previamente explicados en este documento. Solo hay unos pocos dispositivos que actualmente admitan algunas de las opciones de FT (en 2013).

Esta técnica es más compleja de explicar que los otros métodos, ya que introduce nuevos conceptos y varias capas de PMK que se almacenan en caché en diferentes dispositivos (cada dispositivo tiene una función diferente) y proporciona incluso más opciones para la itinerancia segura y rápida. Por lo tanto, se proporciona un breve resumen sobre este método y la forma en que se implementa con cada opción disponible.

802.11r es diferente de SKC y OKC, principalmente debido a estas razones:

- Los mensajes de intercambio de señales (intercambio PMKID, ANonce y SNonce, por ejemplo) ocurren en tramas de autenticación 802.11 o en tramas de acción en lugar de tramas de reasociación. A diferencia de los métodos de almacenamiento en caché PMKID, se evita la fase de entrada en contacto de 4 vías independiente, que se lleva a cabo después del intercambio de mensajes de (re)asociación. El intercambio de señales clave con el nuevo AP comienza antes de que el cliente se traslade/reasocie completamente con este nuevo AP.
- Proporciona dos métodos para el protocolo de enlace de itinerancia rápida: a través de AIR y a través del sistema de distribución (DS).
- 802.11r tiene más capas de jerarquía de claves.
- Dado que este protocolo evita el protocolo de enlace de 4 vías para la gestión de claves cuando un cliente se desplaza (genera nuevas claves de cifrado -PTK y GTK- sin necesidad de este protocolo de enlace), también se puede aplicar para configuraciones WPA2 con PSK, y no solo cuando se utiliza 802.1X/EAP para la autenticación. Esto acelera aún más el

roaming para estas configuraciones, donde no se producen intercambios de intercambio de señales de EAP o de 4 vías.

Con este método, el cliente inalámbrico realiza solamente una autenticación inicial contra la infraestructura WLAN cuando se establece una conexión con el primer AP, y realiza un roaming seguro rápido mientras se desplaza entre los AP del mismo dominio de movilidad FT.

Este es uno de los nuevos conceptos, que básicamente se refiere a los AP que utilizan el mismo SSID (conocido como conjunto de servicios extendido o ESS) y manejan las mismas claves FT. Esto es similar a los otros métodos explicados hasta ahora. La forma en que los AP manejan las claves de dominio de movilidad de FT se basa normalmente en una configuración centralizada, como el WLC o los grupos de movilidad; sin embargo, este método también se puede implementar en entornos AP autónomos.

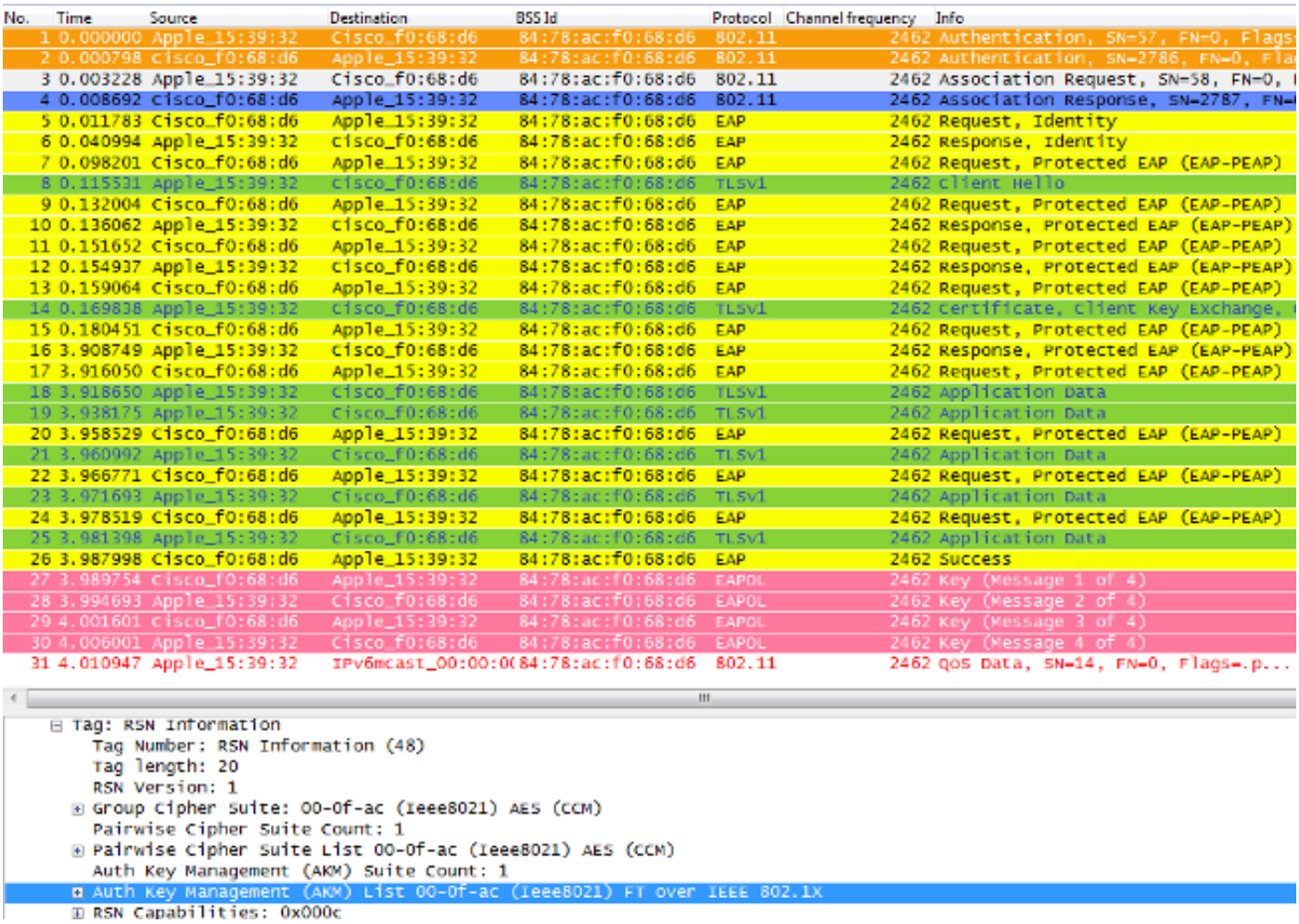
A continuación se muestra un resumen de la jerarquía de claves:

- Una MSK todavía se deriva en el suplicante del cliente y en el Servidor de autenticación desde la fase de autenticación inicial 802.1X/EAP (transferida del Servidor de autenticación al Autenticador (WLC) una vez que la autenticación es exitosa). Esta MSK, como en los otros métodos, se utiliza como la semilla para la jerarquía de claves FT. Cuando se utiliza WPA2-PSK en lugar de un método de autenticación EAP, PSK es básicamente este MSK.
- Una clave maestra en pares R0 (PMK-R0) se deriva de la MSK, que es la clave de primer nivel de la jerarquía de claves FT. Los titulares de claves para este PMK-R0 son el WLC y el cliente.
- Una clave de segundo nivel, llamada Pairwise Master Key R1 (PMK-R1), se deriva del PMK-R0, y los titulares de claves son el cliente y los AP administrados por el WLC que contiene el PMK-R0.
- La clave del tercer y último nivel de la jerarquía de claves FT es la PTK, que es la clave final utilizada para cifrar las tramas de datos de unidifusión 802.11 (similar a los otros métodos que utilizan WPA/TKIP o WPA2/AES). Esta PTK se deriva en FT del PMK-R1, y los titulares de claves son el cliente y los AP administrados por el WLC.

  Nota: En función del proveedor de WLAN y de las configuraciones de implementación (como los puntos de acceso autónomos, FlexConnect o Mesh), la infraestructura WLAN puede transferir y gestionar las claves de una forma diferente. Incluso puede cambiar las funciones de los titulares de las claves, pero como eso está fuera del alcance de este documento, los ejemplos basados en el resumen de jerarquía de claves que se dio anteriormente son el siguiente enfoque. En realidad, las diferencias no son tan relevantes para comprender el proceso, a menos que realmente necesite analizar en profundidad los dispositivos de infraestructura (y su código) para descubrir un problema de software.

Transición rápida de BSS en el aire

Con este método, la primera asociación a cualquier AP es una autenticación regular por primera vez a la WLAN, donde la autenticación completa 802.1X/EAP contra el servidor de autenticación y el intercambio de señales de 4 vías para la generación de claves debe ocurrir antes de que se envíen las tramas de datos, como se muestra en esta imagen de pantalla:



Las principales diferencias son:

- La negociación de la Administración de claves de autenticación es ligeramente diferente de la WPA/WPA2 normal, por lo que se utiliza información adicional para realizar esta negociación cuando ocurre la asociación a una infraestructura WLAN que soporta FT. Como se muestra en la imagen, se selecciona el marco de Solicitud de asociación del cliente y se resalta el campo AKM del elemento de información RNS para mostrar que este cliente desea realizar FT sobre 802.1X/EAP.
- También se muestra el Elemento de información del dominio de movilidad (parte de FT), donde el campo **Capacidad y política de FT** indica si la Transición rápida de BSS se completa por el aire o por el DS cuando se realiza la itinerancia rápida (esto indica por el aire en esta imagen).
- También se agrega otro elemento de información (Fast BSS Transition o FT IE, que se describe más adelante en este documento) con información necesaria para realizar la secuencia de autenticación de FT cuando FT roaming.
- La generación de claves es diferente debido a la jerarquía de claves, por lo que aunque el protocolo de enlace de 4 vías de FT tenga un aspecto similar al de WPA/WPA2 4-Way, en realidad su contenido es ligeramente diferente.

Las depuraciones muestran básicamente el mismo intercambio de tramas de autenticación EAP que el resto de los métodos en la autenticación inicial a la WLAN (como se observa en las imágenes), pero se agregan algunas salidas que se refieren a las técnicas de almacenamiento en caché de claves utilizadas por el WLC; por lo tanto, esta salida de depuración se corta para mostrar solamente la información relevante:

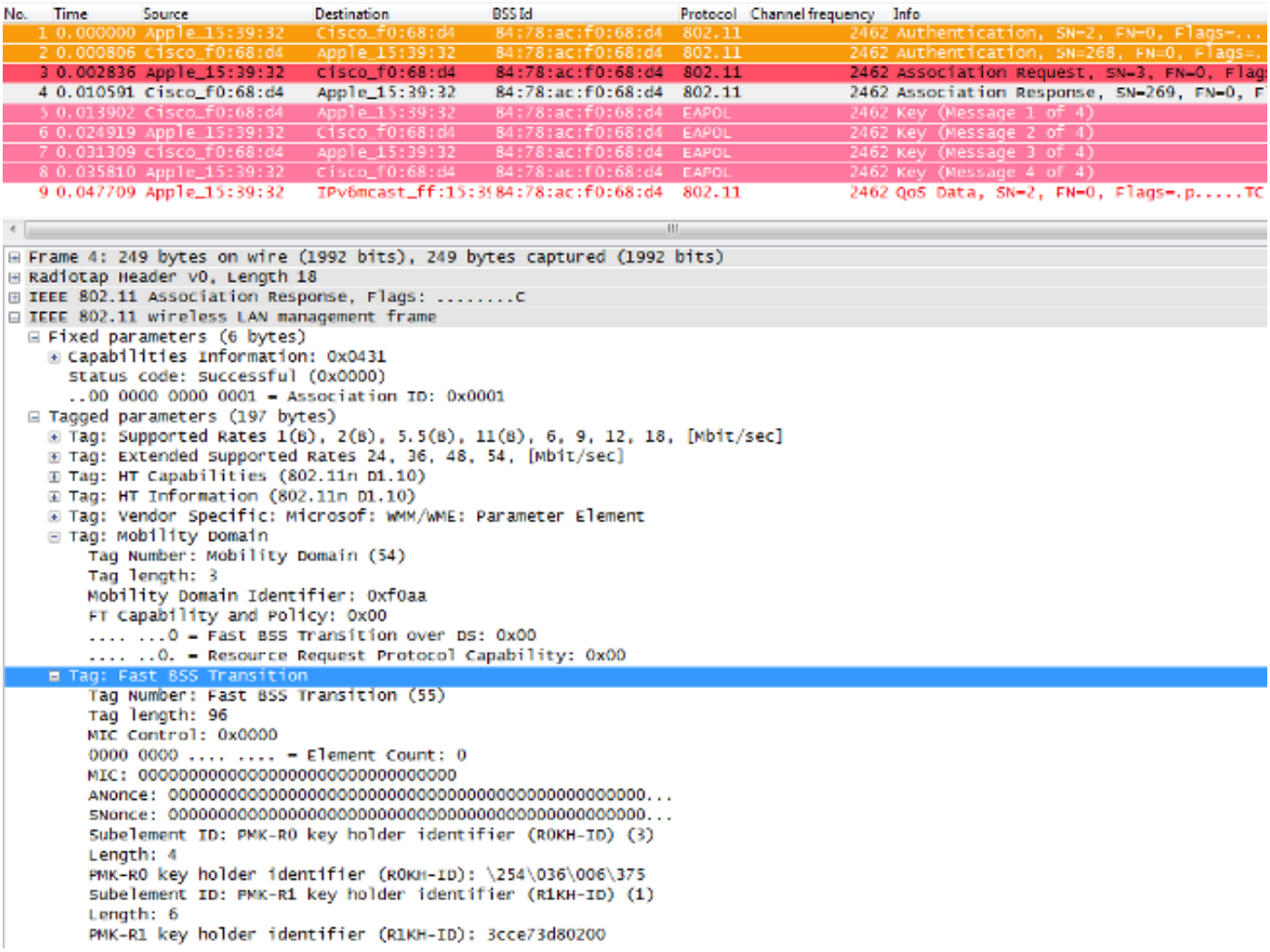
```
*apfMsConnTask_0: Jun 27 19:25:23.426: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d6
!--- This is the Association request from the client.

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.479: ec:85:2f:15:39:32
```

```
  Received EAP Response from mobile ec:85:2f:15:39:32
  (EAP Id 2, EAP Type 25)

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
```

```
  Received EAPOL-key in PTK_START state (message 2) from
  mobile ec:85:2f:15:39:32
!--- Message-2 of the FT 4-Way handshake is received
     successfully from the client.

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

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

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

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

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

> Nota: Para depurar este método y alcanzar los resultados adicionales de 802.11r/FT que se muestran aquí, se habilita una depuración adicional junto con el cliente de depuración, que es debug ft events enable.

Estas son las imágenes y depuraciones de una asociación inicial a la WLAN cuando se realiza FT con WPA2-PSK (en lugar de un método 802.1X/EAP), donde se selecciona la trama de respuesta de asociación del AP para mostrar el elemento de información de transición rápida de BSS (resaltado). También se muestra parte de la información clave necesaria para realizar el protocolo de enlace FT de 4 vías:



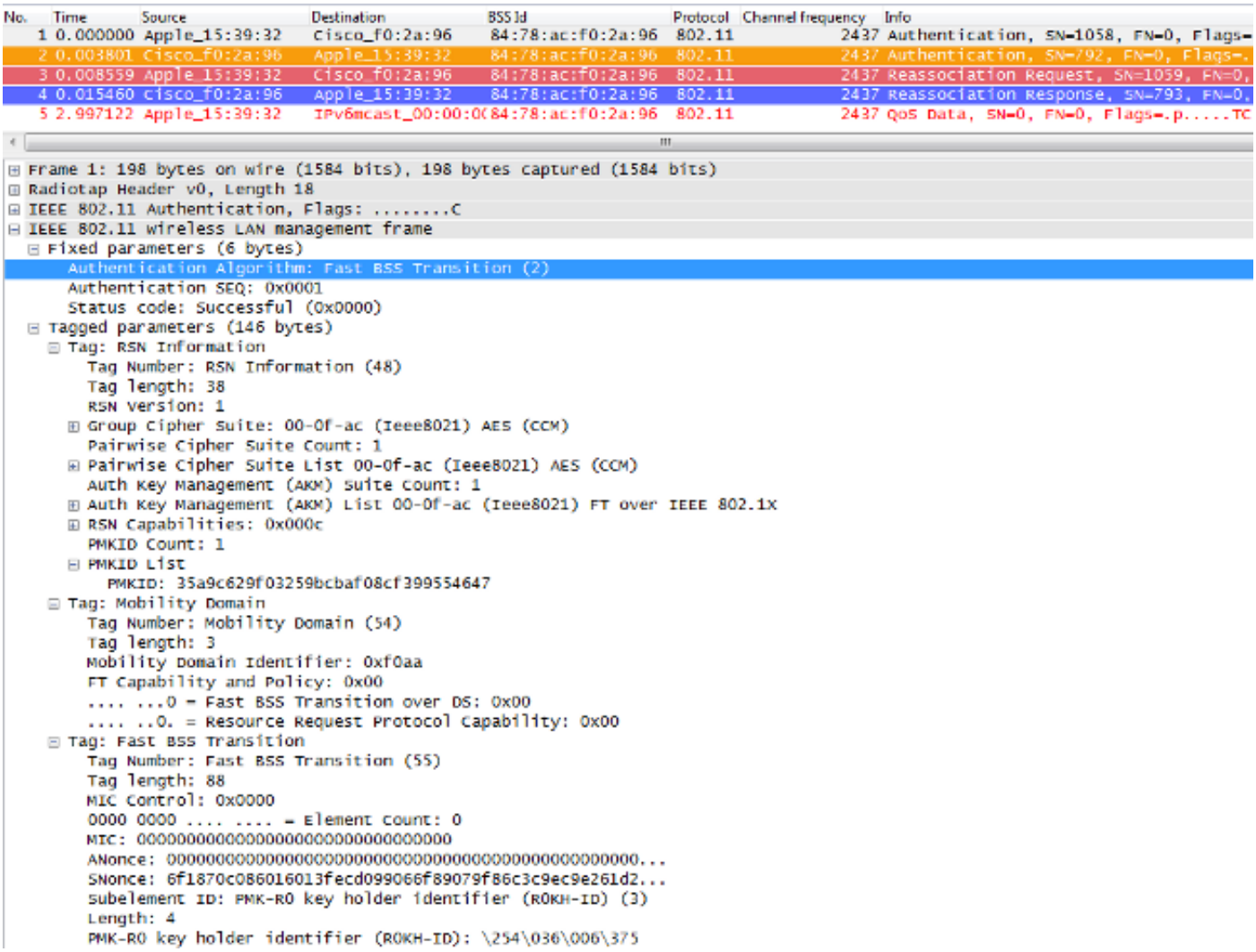

```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*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
```

Con 802.11r, la asociación inicial a la WLAN es la base utilizada para derivar las claves base utilizadas por esta técnica, al igual que en los otros métodos de roaming de seguridad rápida. Las principales diferencias se producen cuando el cliente comienza a vagar; FT no solo evita 802.1X/EAP cuando se utiliza, sino que en realidad realiza un método de roaming más eficiente que combina las tramas iniciales de autenticación y reasociación de sistema abierto 802.11 (que siempre se utilizan y requieren cuando se roaming entre AP) para intercambiar información FT y derivar nuevas claves de cifrado dinámico en lugar del protocolo de enlace de 4 vías.

La siguiente imagen muestra las tramas intercambiadas cuando se realiza una transición rápida de BSS en el aire con seguridad 802.1X/EAP. Se selecciona la trama de autenticación de sistema abierto desde el cliente al AP para ver los elementos de información del protocolo FT que se requieren para comenzar la negociación de la clave FT. Esto se utiliza para derivar la nueva PTK con el nuevo AP (basado en el PMK-R1). El campo que muestra el algoritmo de autenticación se resalta para mostrar que este cliente no realiza una simple autenticación de sistema abierto, sino una transición rápida de BSS:

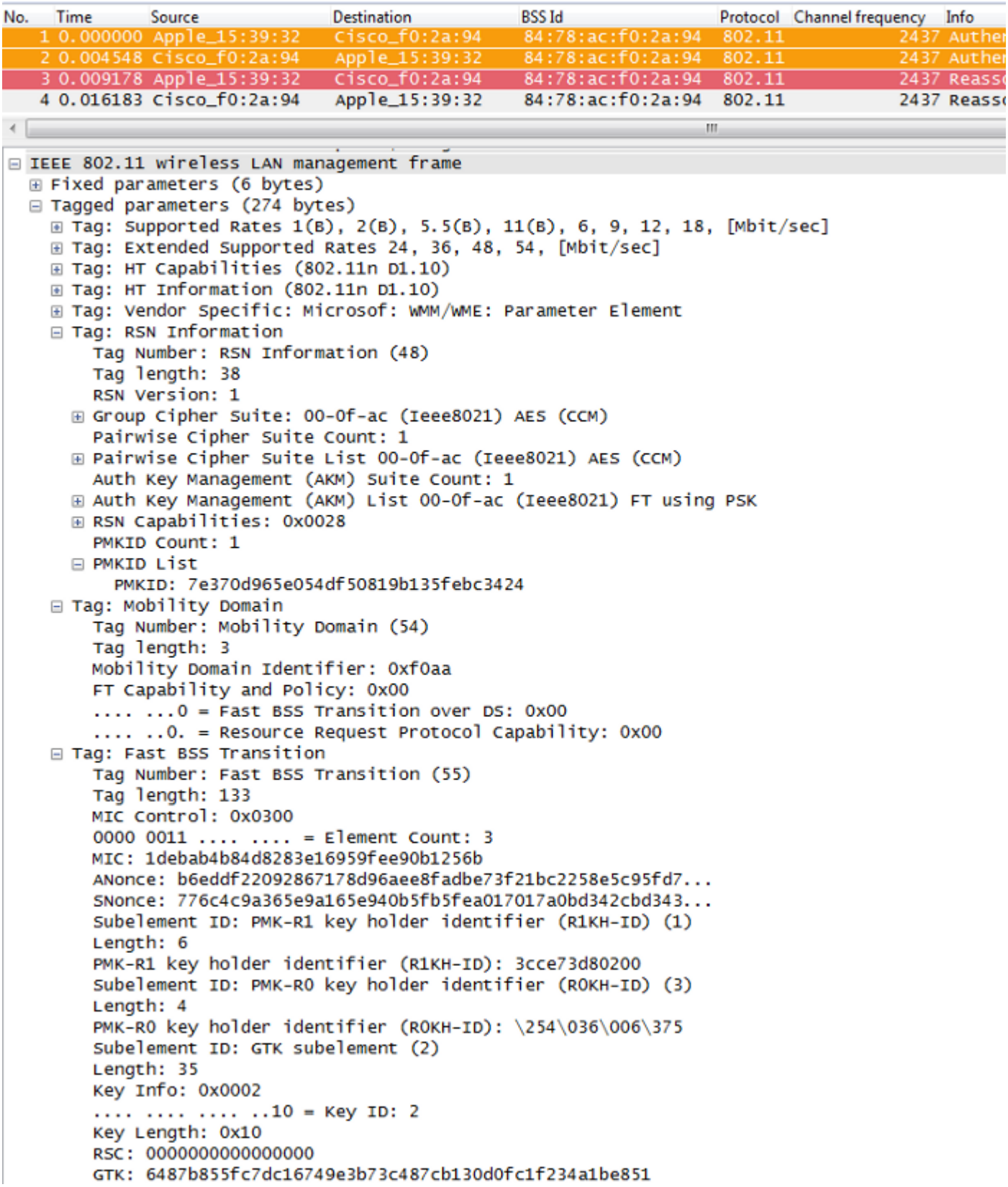


Éstos son los resultados de debug del WLC cuando este evento de roaming de FT ocurre con 802.1X/EAP:

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

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

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

*apfMsConnTask_2: Jun 27 19:25:48.751: ec:85:2f:15:39:32
  Created a new preauth entry for AP:84:78:ac:f0:2a:96
*apfMsConnTask_2: Jun 27 19:25:48.751: Adding MDIE,
  ID is:0xaaf0
```

```
!--- WLC creates a new preauth entry for this AP-and-Client pair,
      and adds the MDIE information.

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

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

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

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

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

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

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

Esta es una imagen que muestra una transición rápida de BSS en el aire con seguridad WPA2-PSK, donde se selecciona la trama de respuesta de reasociación final del AP al cliente para mostrar más detalles sobre este intercambio de FT:



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	Reassc
4	0.016183	Cisco_f0:2a:94	Apple_15:39:32	84:78:ac:f0:2a:94	802.11	2437	Reassc

Estas son las salidas de depuración cuando este evento de roaming de FT ocurre con PSK, que son similares a las que se utilizan cuando se utiliza 802.1X/EAP:

```
*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
```

Como se muestra en la imagen, una vez que se negocia la transición rápida de BSS tras la asociación inicial a la WLAN, las cuatro tramas que se utilizan y se requieren para el roaming (autenticación de sistema abierto del cliente, autenticación de sistema abierto del AP, solicitud de reasociación y respuesta de reasociación) se utilizan básicamente como un protocolo de enlace de 4 vías de FT para derivar la nueva PTK (clave de cifrado de unidifusión) y GTK (clave de cifrado de multidifusión/difusión).

Esto sustituye al protocolo de enlace de 4 vías que normalmente ocurre después de intercambiar estas tramas, y el contenido FT y la negociación de clave en estas tramas es básicamente el mismo si utiliza 802.1X/EAP o PSK como método de seguridad. Como se muestra en la imagen, el campo AKM es la diferencia principal, que confirma si el cliente realiza FT con PSK o 802.1X. Por lo tanto, es importante tener en cuenta que estas cuatro tramas normalmente no tienen este

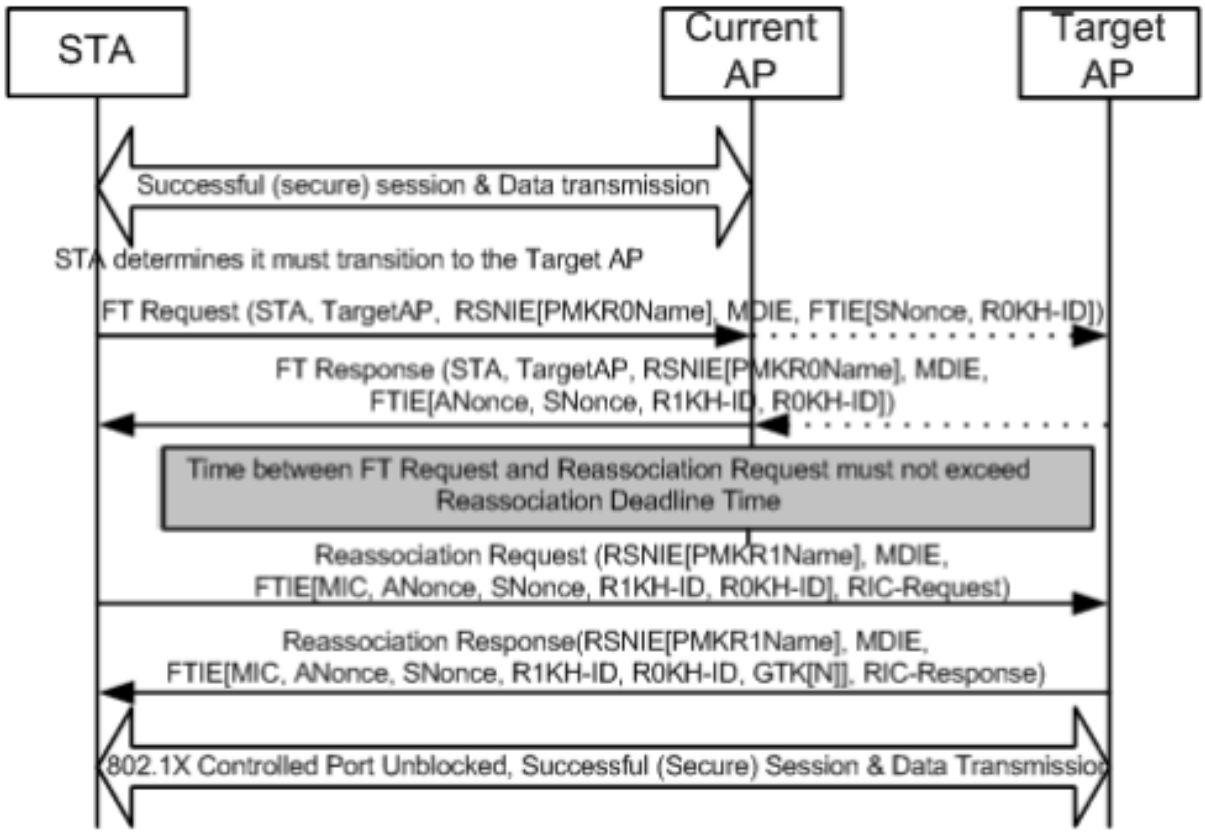
tipo de información de seguridad para la negociación de clave, sino solo cuando el cliente FT se desplaza si 802.11r se implementa y negocia entre el cliente y la infraestructura WLAN en la asociación inicial.

Transición rápida de BSS a través de DS

802.11r permite otra implementación de Fast BSS Transition, en la que el cliente inicia el roaming de FT con el nuevo AP para el que el cliente se desplaza sobre el DS (sistema de distribución), y no sobre la marcha. En este caso, las tramas de la acción FT se utilizan para iniciar la negociación de clave en lugar de las tramas de la autenticación de sistema abierto.

Básicamente, una vez que el cliente decide que puede vagar a un mejor AP, el cliente envía una trama de solicitud de acción de FT al AP original donde está conectado actualmente antes de la itinerancia. El cliente indica el BSSID (dirección MAC) del AP de destino en el que desea realizar el roaming de FT. El AP original reenvía esta trama de solicitud de acción de FT al AP de destino a través del sistema de distribución (normalmente la infraestructura cableada), y el AP de destino responde al cliente con una trama de respuesta de acción de FT (también a través del DS, por lo que finalmente puede enviarlo por el aire al cliente). Una vez que este intercambio de tramas de acción FT es exitoso, el cliente finaliza el roaming FT; el cliente envía la solicitud de reasociación al AP de destino (esta vez por el aire) y recibe una respuesta de reasociación del nuevo AP para confirmar la derivación de las claves de itinerancia y final.

En resumen, hay cuatro tramas para negociar la transición rápida de BSS y derivar nuevas claves de cifrado, pero aquí las tramas de autenticación de sistema abierto se sustituyen por las tramas de solicitud/respuesta de acción de FT, que se intercambian con el AP de destino sobre el sistema de distribución con el AP actual. Este método también es válido para los métodos de seguridad 802.1X/EAP y PSK, todos soportados por los controladores de LAN inalámbrica de Cisco; sin embargo, dado que esta transición Over-the-DS no es soportada e implementada por la mayoría de los clientes inalámbricos en la industria Wi-Fi (y dado que las salidas de intercambio de tramas y de depuración son básicamente las mismas), no se proporcionan ejemplos en este documento. En su lugar, esta imagen se utiliza para visualizar la transición rápida de BSS sobre el DS:



FlexConnect con 802.11r

- Se admite la autenticación central. Esto incluye conmutación de datos local y central. Los puntos de acceso deben formar parte del mismo grupo de FlexConnect.
- No se admite la autenticación local.
- No se admite el modo independiente.

Pros. con 802.11r

- Este método es el primero que utiliza una jerarquía de claves claramente definida por el IEEE en el estándar 802.11 como una enmienda (802.11r), por lo que la implementación de estas técnicas de FT es más compatible entre los proveedores y sin diferentes interpretaciones.
- 802.11r permite varias técnicas útiles, en función de sus necesidades (Over-the-Air y Over-the-DS, para la seguridad 802.1x/EAP y para la seguridad PSK).
- El cliente inalámbrico realiza un roaming seguro rápido a un nuevo AP en la misma WLAN/SSID, incluso si nunca se asoció con ese AP y sin la necesidad de guardar varios PMKID.
- Este es el primer método de roaming rápido y seguro que permite un roaming más rápido incluso con la seguridad PSK, y evita el protocolo de enlace de 4 vías que se requiere al roaming entre AP con WPA/WPA2 PSK. El objetivo principal de los métodos de roaming de seguridad rápida es evitar el protocolo de enlace 802.1X/EAP cuando se implementa este método de seguridad; sin embargo, para la seguridad PSK, el evento de roaming se acelera aún más con 802.11r cuando se evita el protocolo de enlace de 4 vías.

Desventajas de 802.11r

- Hay algunos dispositivos cliente inalámbricos que realmente admiten transiciones rápidas de BSS y, en la mayoría de los casos, no admiten todas las técnicas disponibles en 802.11r.
- Debido al hecho de que estas implementaciones son muy jóvenes, no hay suficientes resultados de prueba de entornos de producción reales o suficientes resultados de depuración para abordar las posibles advertencias que pueden aparecer.
- Cuando configura un WLAN/SSID para utilizar cualquiera de los métodos FT, solamente los clientes inalámbricos que soportan 802.11r pueden conectarse a este WLAN/SSID. Los parámetros de FT no son opcionales para los clientes, por lo que los clientes inalámbricos que no admiten 802.11r deben conectarse con una WLAN/SSID independiente donde FT no esté configurado en absoluto.

802.11r adaptable

- Algunos clientes heredados no pueden asociarse con una WLAN/SSID que tenga 802.11r habilitado incluso para el "modo mixto" (que espera que pueda tener en los mismos clientes SSID que admiten y que no admiten 802.11r). Esto ocurre cuando el controlador del suplicante del cliente que es responsable de analizar el Elemento de información de red de seguridad robusto (RSN IE) es antiguo y no conoce las suites AKM adicionales en el IE. Debido a esta limitación, los clientes no pueden enviar solicitudes de asociación a las WLAN que anuncian el soporte 802.11r y, por lo tanto, necesita configurar una WLAN/SSID para los clientes 802.11r y una WLAN/SSID separada para los clientes que no soportan 802.11r.
- Para solucionar este problema, la infraestructura de LAN inalámbrica de Cisco ha introducido la función 802.11r adaptable. Cuando el modo FT está configurado en Adaptive en el nivel WLAN, WLAN anuncia el ID de dominio de movilidad 802.11r en una WLAN habilitada para 802.11i. Algunos dispositivos cliente Apple iOS10 identifican la presencia de MDIE en una WLAN 80211i/WPA2 y realizan un intercambio de señales propietario para establecer una asociación 802.11r. Una vez que el cliente completa la asociación exitosa de 802.11r, puede realizar el roaming de FT como en una WLAN normal habilitada para 802.11r. El FT Adaptive solo es aplicable a los dispositivos Apple iOS10 (y posteriores) seleccionados. El resto de los clientes pueden seguir teniendo una asociación 802.11i/WPA2 en la WLAN y realizar el método FSR aplicable como se admita.
- Encontrará más documentación sobre esta nueva función introducida para que los dispositivos iOS10 ejecuten 802.11r en una WLAN/SSID donde 802.11r no está realmente habilitado (de modo que otros clientes que no sean 802.11r puedan conectarse correctamente) en Prácticas recomendadas empresariales para dispositivos Cisco IOS en LAN inalámbrica de Cisco.

Conclusiones

- Tenga en cuenta que el cliente es siempre el que decide vagar a un AP específico, y el WLC/AP no puede decidir esto para el cliente. El evento de itinerancia lo inicia el cliente inalámbrico una vez que considera que debe itinerar.
- El WLC soporta una combinación de la mayoría o de todos los métodos FSR (Fast-Secure Roaming) juntos en el mismo WLAN/SSID. Sin embargo, tenga en cuenta que esto normalmente no funciona, ya que depende mucho del comportamiento del cliente (muy diferente a través de diferentes dispositivos móviles) para soportar o incluso entender aquello que el WLC intenta anunciar como soportado. En lugar de lograr la interoperabilidad en un

solo SSID, normalmente hay más problemas que los que se espera solucionar, por lo que no se recomienda. Si esto es realmente necesario, deben realizarse pruebas exhaustivas con todos los clientes posibles que se utilizarán en esta WLAN.

- Es muy importante entender que los métodos de roaming de seguridad rápida se desarrollan para acelerar el proceso de roaming de WLAN cuando se mueve entre APs si el WLAN/SSID tiene la seguridad habilitada. Cuando no hay seguridad, no hay nada que acelerar, ya que el cliente-AP simplemente intercambia las tramas de administración inalámbrica que siempre se requieren cuando se está en roaming entre los AP antes de que se envíen las tramas de datos (autenticación de sistema abierto del cliente, autenticación de sistema abierto del AP, solicitud de reasociación y respuesta de reasociación). Por lo tanto, esto no puede moverse más rápido. Si encuentra problemas de roaming sin seguridad, entonces no hay métodos de roaming rápido para mejorar el roaming, solo métodos para confirmar si la configuración y el diseño de WLAN/SSID son apropiados para que las estaciones cliente inalámbricas se trasladen en consecuencia entre las celdas de cobertura de AP.
- 802.11r/FT se implementa con WPA2-PSK para acelerar los eventos de itinerancia con esta seguridad y evitar el protocolo de enlace de 4 vías, como se explica en la sección 802.11r.
- Todos los métodos tienen sus ventajas y desventajas, pero al final, siempre debe verificar si las estaciones cliente inalámbricas admiten el método específico que desea implementar y si la infraestructura WLAN de Cisco admite todos los métodos disponibles. Por lo tanto, debe seleccionar el mejor método que admitan realmente los clientes inalámbricos que se conectan al WLAN/SSID específico. Por ejemplo, en algunas implementaciones puede crear un WLAN/SSID con CCKM para los teléfonos IP inalámbricos de Cisco (que admiten WPA2/AES con CCKM, pero no 802.11r) y, a continuación, otro WLAN/SSID con WPA2/AES a través de 802.11r/FT para los clientes inalámbricos que admiten este método de itinerancia segura rápida (o utilice OKC, si es compatible).
- Si los clientes inalámbricos no admiten ninguno de los métodos de roaming de seguridad rápida disponibles, puede aceptar el hecho de que esos clientes siempre pueden experimentar los retrasos explicados en este documento cuando se desplazan entre AP en un WLAN/SSID con seguridad 802.1X/EAP (que puede causar interrupciones en las aplicaciones/servicios del cliente).
- Todos los métodos, excepto SKC (WPA2 PMKID Caching), son compatibles con la itinerancia de seguridad rápida entre los AP administrados por diferentes WLC (itinerancia entre controladores), siempre que estén en el mismo grupo de movilidad.
- CUWN es totalmente compatible con todos los diferentes métodos de roaming de seguridad rápida que se tratan en este artículo cuando se utiliza la autenticación 802.1X/EAP para WPA/WPA2. CUWN no admite la itinerancia segura rápida en métodos que funcionan con WPA2-RSN (CCKM, PMKID Caching/SKC, OKC/PKC) cuando se utiliza PSK (WPA2-Personal), donde los métodos de itinerancia rápida no son necesarios en su mayoría. Sin embargo, CUWN admite la itinerancia segura y rápida en el caso de WPA2-FT (802.11r) con PSK, como también se explica en este artículo.

Información Relacionada

- Guía de implementación de transición rápida a 802.11r BSS
- Asistencia técnica y descargas de Cisco

Acerca de esta traducción
Cisco ha traducido este documento combinando la traducción automática y los recursos humanos a fin de ofrecer a nuestros usuarios en todo el mundo contenido en su propio idioma.

Tenga en cuenta que incluso la mejor traducción automática podría no ser tan precisa como la proporcionada por un traductor profesional.

Cisco Systems, Inc. no asume ninguna responsabilidad por la precisión de estas traducciones y recomienda remitirse siempre al documento original escrito en inglés (insertar vínculo URL).