CUWN에서 802.11 WLAN 및 Fast-Secure Roaming에 대한 방법 확인

목차

소개

이 문서에서는 CUWN(Unified Wireless Network)의 IEEE 802.11 WLAN(Wireless LAN)에 사용할 수 있는 무선 및 고속 보안 로밍 유형에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- IEEE 802.11 WLAN 기본 사항
- IEEE 802.11 WLAN 보안
- IEEE 802.1X/EAP 기본 사항

사용되는 구성 요소

이 문서의 정보는 Cisco WLAN Controller Software 버전 7.4를 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 현재 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

이 문서의 정보는 Cisco WLAN Controller Software Version 7.4를 기반으로 하지만 설명된 대부분의 디버그 출력 및 동작은 앞서 설명한 방법을 지원하는 모든 소프트웨어 버전에 적용할 수 있습니다. 여기에 설명된 모든 방법의 세부 사항은 이후 Cisco WLAN Controller 코드(이 문서를 업데이트한 시점까지 버전 8.3까지)에서 동일합니다.

이 문서에서는 CUWN(Cisco Unified Wireless Network)에서 지원되는 IEEE 802.11 WLAN(Wireless LAN)에 사용할 수 있는 다양한 유형의 무선 로밍 및 빠른 보안 로밍 방법에 대해 설명합니다.

이 문서에서는 각 방법의 작동 방식 또는 구성 방식에 대한 모든 세부 사항을 제공하지 않습니다. 이 문서의 주요 목적은 사용 가능한 다양한 기술 간의 차이, 장점 및 제한 사항, 각 방법에 대한 프레임 교환을 설명하는 것입니다. WLC(WLAN Controller) 디버그의 예가 제공되며, 무선 패킷 이미지를 사용하여 설명된 각 로밍 방법에 대해 발생하는 이벤트를 분석하고 설명합니다.

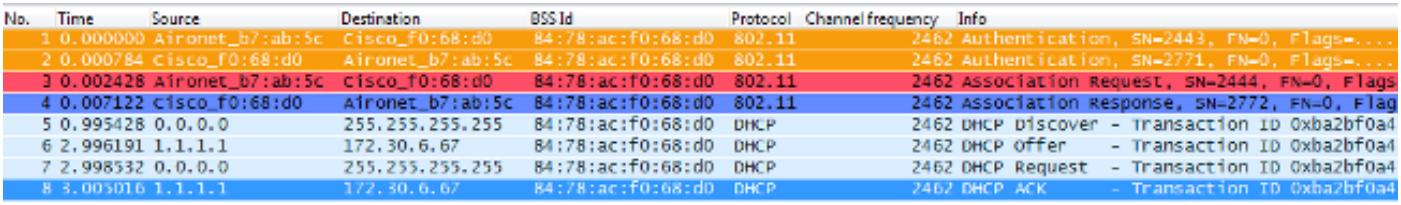
WLAN에 사용할 수 있는 다양한 고속 보안 로밍 방법에 대한 설명을 드리기 전에 WLAN 연결 프로세스가 작동하는 방식 및 SSID(Service Set Identifier)에 보안이 구성되지 않은 경우 일반 로밍 이벤트가 발생하는 방식을 이해하는 것이 중요합니다.

802.11 무선 클라이언트가 액세스 포인트(AP)에 연결되면 트래픽(무선 데이터 프레임)을 전달하기 전에 먼저 기본 802.11 개방형 시스템 인증 프로세스를 통과해야 합니다. 그런 다음 연결 프로세스를 완료해야 합니다. 오픈 시스템 인증 프로세스는 클라이언트가 선택하는 AP의 케이블 연결과 같습니다. 이 점은 매우 중요한 점입니다. 어떤 AP를 선호하는지 선택하는 무선 클라이언트이며 벤더 간에 다양한 여러 요인을 기반으로 결정을 내리기 때문입니다. 따라서 이 문서의 뒷부분에 나와 있는 것처럼 클라이언트가 인증 프레임을 선택한 AP에 전송하여 이 프로세스를 시작합니다. AP에서 연결을 설정하도록 요청할 수 없습니다.

AP("케이블 연결됨")의 응답과 함께 오픈 시스템 인증 프로세스가 성공적으로 완료되면 연결 프로세스는 클라이언트와 AP 간의 링크를 설정하는 802.11 L2(Layer 2) 협상을 기본적으로 종료합니다. AP는 연결에 성공할 경우 클라이언트에 연결 ID를 할당하고, SSID에 구성된 경우 트래픽을 전달하거나 더 높은 수준의 보안 방법을 수행하도록 준비합니다. 오픈 시스템 인증 프로세스는 연결 프로

세스뿐만 아니라 두 개의 관리 프레임으로 구성됩니다. 인증 및 연결 프레임은 데이터 **프레임**이 아니라 무선 관리 프레임이며, 기본적으로 AP와의 연결 프로세스에 사용됩니다.

다음은 이 프로세스를 위한 무선 프레임 상공의 이미지입니다.



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

> **참고**: 802.11 무선 스니핑 및 이 문서에 나온 이미지에 대해 Wireshark에서 사용되는 필터/색상에 대해 알아보려면 802.11 스니퍼 이미지 분석이라는 Cisco 지원 커뮤니티 게시물을 방문하십시오.

무선 클라이언트는 인증 프레임으로 시작하고 AP는 다른 인증 프레임으로 응답합니다. 그러면 클라이언트가 연결 요청 프레임을 전송하고 AP가 연결 응답 프레임의 응답을 완료합니다. DHCP 패킷에서 보여주는 것처럼 802.11 Open System 인증 및 연결 프로세스가 통과되면 클라이언트는 데이터 프레임 통과를 시작합니다. 이 경우 SSID에 보안 방법이 구성되어 있지 않으므로 클라이언트는 암호화되지 않은 데이터 프레임(이 경우 DHCP)을 즉시 보내기 시작합니다.

이 문서의 뒷부분에 나와 있는 것처럼, SSID에서 보안이 활성화된 경우, 특정 보안 방법에 대해 더 높은 수준의 인증 및 암호화 핸드셰이크 프레임이 있습니다. 이는 연결 응답 직후와 클라이언트 트래픽 데이터 프레임(예: 암호화된 DHCP, ARP(Address Resolution Protocol) 및 애플리케이션 패킷)이 전송되기 전입니다. 데이터 프레임은 클라이언트가 완전히 인증되고 구성된 보안 방법에 따라 암호화 키가 협상될 때까지만 전송할 수 있습니다.

이전 이미지를 기반으로, 무선 클라이언트가 WLAN에 대한 새 연결을 시작할 때 WLC **debug client** 명령의 출력에 표시되는 메시지는 다음과 같습니다.

```
*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.
```

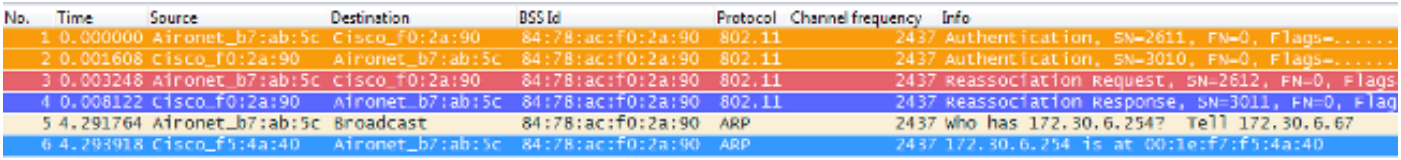
> **참고**: 이 문서에 표시된 출력에 사용되는 WLC 디버그는 debug **client** 명령이며, 예는 전체 출력이 아닌 일부 관련 메시지만 표시합니다. 이 debug 명령에 대한 자세한 내용은 WLC(무선 LAN 컨트롤러)의 디버그 클라이언트 이해 문서를 참조하십시오.

이러한 메시지는 연결 요청 및 응답 프레임을 보여줍니다. 이 핸드셰이크는 CUWN의 AP 레벨에서 빠르게 수행되므로 WLC에서 초기 인증 프레임이 로깅되지 않습니다.

클라이언트가 로밍할 때 표시되는 정보는 무엇입니까? 클라이언트는 클라이언트 연결 설정 또는

로밍 이벤트 때문에 AP에 대한 연결이 설정되면 항상 4개의 관리 프레임을 교환합니다. 클라이언트에는 한 번에 하나의 AP에만 설정된 연결이 하나만 있습니다. WLAN 인프라에 대한 새 연결과 로밍 이벤트 간의 프레임 교환의 유일한 차이점은 로밍 이벤트의 연결 프레임을 재연결 프레임이라고 하는데, 이는 클라이언트가 WLAN에 대한 새 연결을 설정하려는 시도 없이 다른 AP에서 실제로 로밍하고 있음을 나타냅니다. 이러한 프레임에는 로밍 이벤트를 협상하기 위해 사용되는 다양한 요소가 포함될 수 있습니다. 이 요소는 설정에 따라 다르지만 이러한 세부 정보는 이 문서의 범위에 포함되지 않습니다.

다음은 프레임 교환의 예입니다.



다음 메시지는 디버그 출력에 나타납니다.

```
*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.
```

그림과 같이 클라이언트는 새 AP에 재연결 요청이 전송된 후 로밍 이벤트를 성공적으로 수행하고 AP로부터 재연결 응답을 수신합니다. 클라이언트에 이미 IP 주소가 있으므로 첫 번째 데이터 프레임은 ARP 패킷에 대한 것입니다.

로밍 이벤트가 예상되지만 클라이언트가 Reassociation Request(재연결 요청) 대신 Association Request(연결 요청)를 전송하는 경우(이 문서의 앞부분에서 설명한 이미지와 유사한 일부 이미지 및 디버그에서 확인할 수 있음), 클라이언트는 실제로 로밍되지 않습니다. 클라이언트는 연결이 끊긴 것처럼 WLAN에 대한 새 연결을 시작하고 처음부터 다시 연결하려고 시도합니다. 이는 여러 가지 이유로 인해 발생할 수 있습니다. 예를 들어, 클라이언트가 서비스 지역을 벗어난 후 연결을 시작할 수 있는 신호 품질이 충분한 AP를 찾을 때 발생할 수 있지만 일반적으로 클라이언트가 드라이버, 펌웨어 또는 소프트웨어 문제로 인해 로밍 이벤트를 시작하지 않는 클라이언트 문제를 나타냅니다.

> **참고**: 무선 클라이언트 공급업체에 문의하여 문제의 원인을 확인할 수 있습니다.

더 높은 수준의 보안으로 로밍

기본 802.11 개방형 시스템 인증 위에 SSID가 L2 고급 보안으로 구성된 경우, 초기 연결 및 로밍 시 더 많은 프레임이 필요합니다. 802.11 WLAN에 대해 표준화되고 구현된 가장 일반적인 두 가지 보안 방법은 이 문서에서 설명합니다.

- **WPA/WPA2-PSK(사전 공유 키)** - 사전 공유 키를 사용하는 클라이언트 인증
- **WPA/WPA2-EAP(Extensible Authentication Protocol) -** 인증서, 사용자 이름 및 비밀번호, 토큰과 같은 인증 서버를 사용하여 보다 안전한 자격 증명을 검증하기 위해 802.1X/EAP 방법으로 클라이언트를 인증합니다.

이 두 가지 방법(PSK와 EAP)이 서로 다른 방법으로 클라이언트를 인증/검증하더라도 키 관리 프로

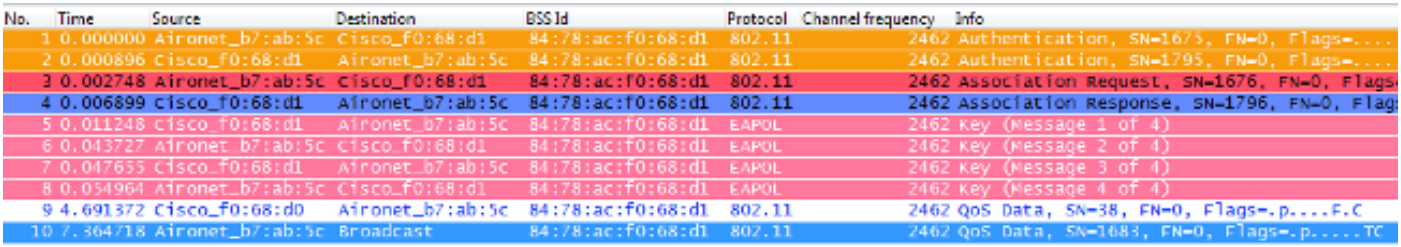
세스에 기본적으로 동일한 WPA/WPA2 규칙을 사용한다는 점을 알아야 합니다. 보안이 WPA/WPA2-PSK이든 WPA/WPA2-EAP이든 WPA/WPA2 4-Way 핸드셰이크라고 하는 프로세스는 사용된 특정 인증 방법으로 클라이언트를 검증한 후 원래 키 재료로 MSK(Master Session Key)를 사용하여 WLC/AP와 클라이언트 간의 키 협상을 시작합니다.

프로세스의 요약은 다음과 같습니다.

1. MSK는 802.1X/EAP 보안을 사용하는 경우 EAP 인증 단계에서 파생되거나 WPA/WPA2-PSK를 보안 방법으로 사용하는 경우 PSK에서 파생됩니다.
2. 이 MSK에서 클라이언트와 WLC/AP는 PMK(Pairwise Master Key)를 유도하고 WLC/AP는 GMK(Group Master Key)를 생성한다.
3. 이 두 마스터 키가 준비되면 클라이언트와 WLC/AP는 실제 암호화 키의 협상을 위한 시드로 마스터 키를 사용하여 WPA/WPA2 4-Way 핸드셰이크(이 문서의 뒷부분에 일부 화면 이미지 및 디버그를 통해 설명되어 있음)를 시작합니다.
4. 이러한 최종 암호화 키는 PTK(Pairwise Transient Key)와 GTK(Group Transient Key)라고 합니다. PTK는 PMK에서 파생되며 클라이언트와 유니캐스트 프레임을 암호화하기 위해 사용됩니다. GTK(Group Transient Key)는 GMK에서 파생되며, 이 특정 SSID/AP에서 멀티캐스트/브로드캐스트를 암호화하는 데 사용됩니다.

WPA/WPA2-PSK

암호화를 위해 TKIP(Temporal Key Integrity Protocol) 또는 AES(Advanced Encryption Standard)를 통해 WPA-PSK 또는 WPA2-PSK를 수행하는 경우 클라이언트는 초기 연결 및 로밍 시 모두 WPA 4-Way 핸드셰이크라는 프로세스를 거쳐야 합니다. 앞서 설명한 대로 이는 기본적으로 WPA/WPA2가 암호화 키를 파생하기 위해 사용하는 키 관리 프로세스입니다. 그러나 PSK를 수행할 때 클라이언트가 WLAN에 조인할 유효한 사전 공유 키를 가지고 있는지 확인하는 데에도 사용됩니다. 이 그림에서는 PSK가 있는 WPA 또는 WPA2를 수행할 때 초기 연결 프로세스를 보여 줍니다.



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

표시된 대로 802.11 개방형 시스템 인증 및 연결 프로세스 후에는 WPA 4-방향 핸드셰이크의 EAPOL 프레임이 4개 있습니다. 이 프레임은 **message-1**을 사용하는 AP에서 시작되고, **message-4**를 사용하는 클라이언트에서 끝납니다. 핸드셰이크가 성공하면 클라이언트는 데이터 프레임(예: DHCP)을 전달하기 시작합니다. 이 경우 4방향 핸드셰이크에서 파생된 키로 암호화됩니다(따라서 무선 이미지에서 실제 콘텐츠 및 트래픽 유형을 볼 수 없음).

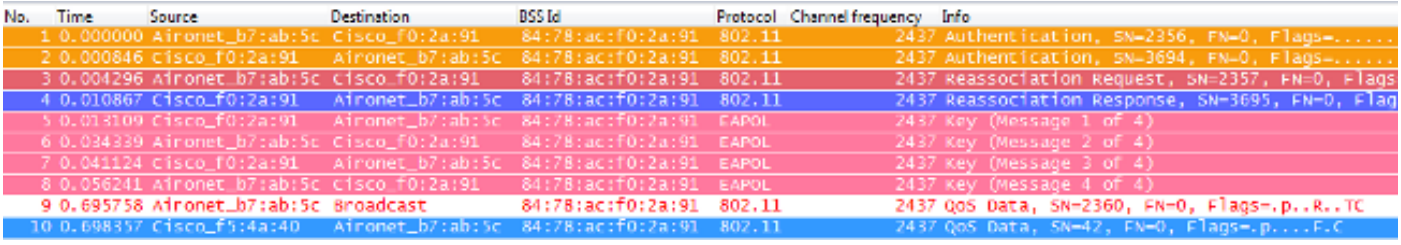
> **참고**: EAPOL 프레임은 모든 키 관리 프레임 및 802.1X/EAP 인증 프레임을 AP와 클라이언트 간에 무선으로 전송하기 위해 사용되며, 무선 데이터 프레임으로 전송됩니다.

이러한 메시지는 디버그 출력에 나타납니다.

```
*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.
```

로밍 시 클라이언트는 기본적으로 동일한 프레임 교환을 추적합니다. 이 경우 새 AP로 새 암호화 키를 파생하려면 WPA 4방향 핸드셰이크가 필요합니다. 이는 표준이 정한 보안 이유와 새 AP가 원래 키를 알지 못하기 때문이다. 유일한 차이점은 이 이미지에 표시된 것처럼 연결 프레임 대신 재연결 프레임이 있다는 것입니다.



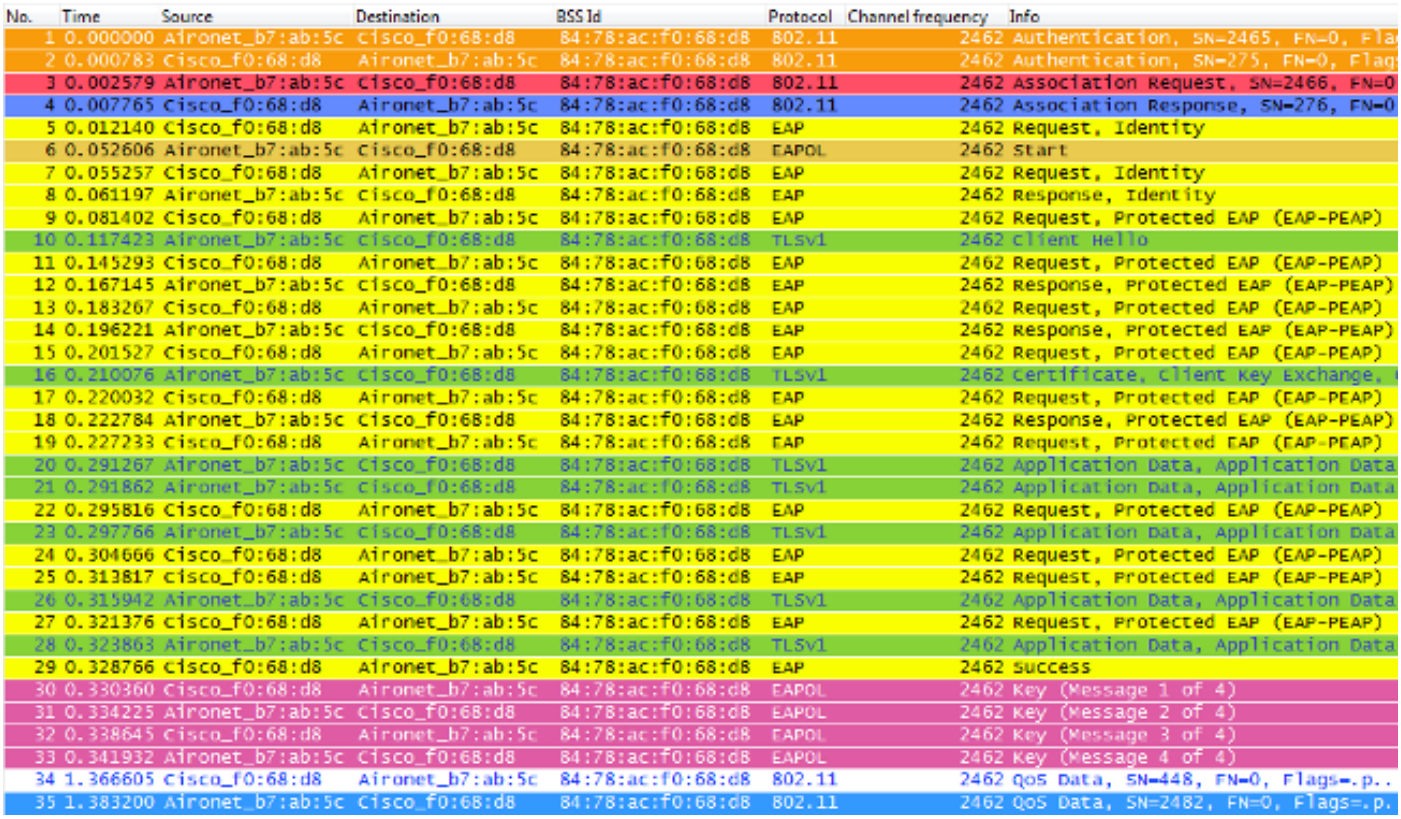
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3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

디버그 출력에서 동일한 메시지가 표시되지만, 앞서 보여주고 설명한 것처럼 클라이언트의 첫 번째 패킷이 연결 대신 재연결입니다.

WPA/WPA2-EAP

보안 SSID에서 클라이언트를 인증하는 데 802.1X/EAP 방법을 사용할 경우, 클라이언트가 트래픽 전달을 시작하기 전에 더 많은 프레임이 필요합니다. 이러한 추가 프레임은 클라이언트 자격 증명을 인증하는 데 사용되며 EAP 방법에 따라 4개에서 20개 사이의 프레임이 있을 수 있습니다. 연결/재연결 후 WPA/WPA2 4-Way 핸드셰이크 전에 이 단계가 발생합니다. 인증 단계에서 키 관리 프로세스(4-Way 핸드셰이크)에서 최종 암호화 키 생성에 대한 시드로 사용되는 MSK가 파생되기 때문입니다.

이 그림에서는 PEAPv0/EAP-MSCHAPv2와의 WPA를 수행할 때 초기 연결 시 AP와 무선 클라이언트 간에 공기로 교환되는 프레임의 예를 보여 줍니다.



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.

때때로 이 교환은 EAP 방법, 문제로 인한 재전송, 클라이언트 동작(예: AP가 첫 번째 ID 요청을 전송한 후 클라이언트가 EAPOL START를 보내기 때문에 이 예에서 두 가지 ID 요청) 또는 클라이언트가 서버와 인증서를 이미 교환한 경우 등 여러 요인에 따라 다소 많은 프레임을 표시합니다. SSID가 802.1X/EAP 방법으로 구성될 때마다 (인증을 위해) 더 많은 프레임이 있으므로 클라이언트가 데이터 프레임을 보내기 전에 더 많은 시간이 필요합니다.

다음은 디버그 메시지의 요약입니다.

```
*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

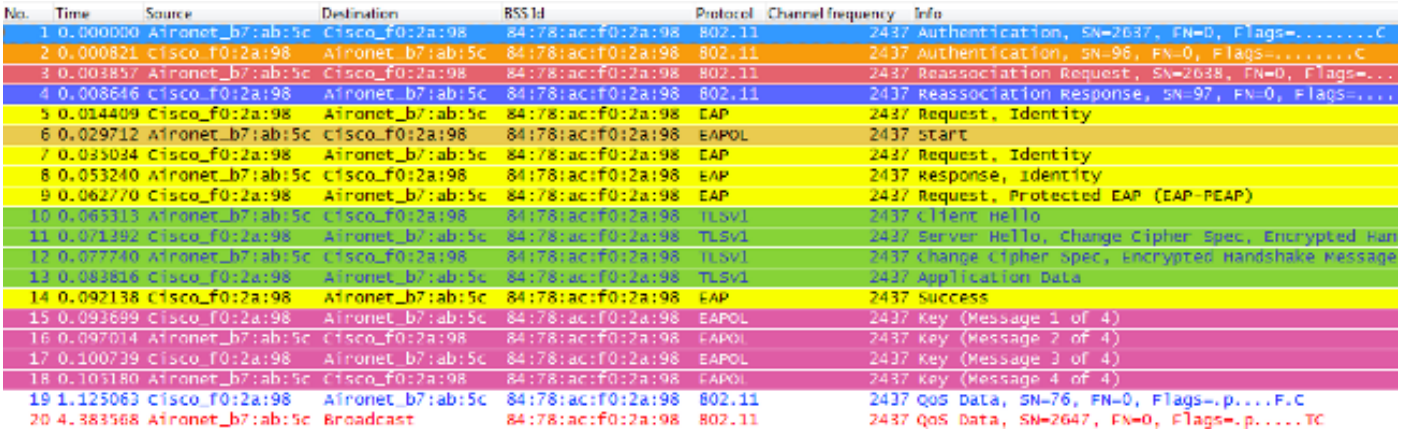
*Dot1x_NW_MsgTask_4: Jun 21 23:41:19.403: 00:40:96:b7:ab:5c
  Received EAP Response from mobile 00:40:96:b7:ab:5c
  (EAP Id 5, EAP Type 25)

```
*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.
```

무선 클라이언트가 여기에서 일반 로밍을 수행하는 경우(빠른 보안 로밍 방법을 구현하지 않고 정상적인 동작), 클라이언트는 그림과 같이 동일한 프로세스를 거쳐 인증 서버에 대해 전체 인증을 수행해야 합니다. 유일한 차이점은 클라이언트가 새 AP에 다른 AP에서 실제로 로밍하고 있음을 알리기 위해 Reassociation Request(재연결 요청)를 사용한다는 것이지만 클라이언트는 전체 검증과 새 키 생성을 거쳐야 한다는 것입니다.



No.	Time	Source	Destination	BSSId	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=2637, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=96, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2638, FN=0, Flags=...
4	0.008646	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Reassociation Response, SN=97, FN=0, Flags=....
5	0.014409	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
6	0.029712	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Start
7	0.035084	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
8	0.053240	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAP	2437	Response, Identity
9	0.062770	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Protected EAP (EAP-PEAP)
10	0.065313	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Client Hello
11	0.071292	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Server Hello, Change Cipher Spec, Encrypted Han
12	0.077740	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Change Cipher Spec, Encrypted Handshake Message
13	0.083816	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Application Data
14	0.092138	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Success
15	0.093699	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 1 of 4)
16	0.097014	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 2 of 4)
17	0.100739	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 3 of 4)
18	0.103180	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 4 of 4)
19	1.125063	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=76, FN=0, Flags=.p....F.C
20	4.383568	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=2647, FN=0, Flags=.p.....TC

그림과 같이, 초기 인증보다 프레임 수가 적은 경우에도(앞에서 언급한 것처럼 여러 요인으로 인해 발생하는) 클라이언트가 새 AP로 로밍할 때 데이터 프레임을 계속 통과하려면(로밍하기 전에 트래픽이 활발하게 전송된 경우에도) EAP 인증 및 WPA 키 관리 프로세스를 완료해야 합니다. 따라서 클라이언트에 지연에 민감한 활성 애플리케이션(예: 음성 트래픽 애플리케이션 또는 시간 초과에 민감한 애플리케이션)이 있는 경우, 사용자는 로밍 시 오디오 공백 또는 애플리케이션 연결 끊김과 같은 문제를 인지할 수 있습니다. 이는 클라이언트가 데이터 프레임을 계속 송수신하는 데 걸리는 프로세스에 따라 달라집니다. 이러한 지연은 RF 환경, 클라이언트의 양, WLC와 LAP 간의 왕복 시간, 인증 서버와의 왕복 시간 및 기타 이유에 따라 더 길어질 수 있습니다.

다음은 이 로밍 이벤트에 대한 디버그 메시지의 요약입니다(기본적으로 이전 메시지와 동일하므로 이러한 메시지에 대해서는 더 이상 설명하지 않음).

```
*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
```

이는 802.1X/EAP 및 WPA/WPA2 보안 프레임워크가 작동하는 방식입니다. 정기적인 로밍 이벤트에서 애플리케이션/서비스가 지연에 미치는 영향을 방지하기 위해, WLAN/SSID에서 보안을 사용할 때 로밍 프로세스를 가속화하기 위해 WiFi 업계에서 여러 가지 신속 보안 로밍 방법을 개발하여 구현하고 있습니다. 클라이언트가 WLAN에 높은 수준의 보안을 구축하여 AP 간에 로밍하는 동안 트래픽을 계속 전달할 경우 약간의 레이턴시가 발생합니다. 이는 앞서 설명한 대로 보안 설정에 필요한 EAP 인증 및 키 관리 프레임 교환 때문입니다.

WLAN에 보안을 구성할 때 로밍 프로세스를 가속화하는 방법/체계를 구현하는 것과 관련하여 빠른 보안 로밍이 업계에서 사용되는 용어임을 이해하는 것이 중요합니다. WLAN에 사용할 수 있으며 CUWN에서 지원하는 다양한 고속 보안 로밍 방법/체계에 대해서는 다음 섹션에서 설명합니다.

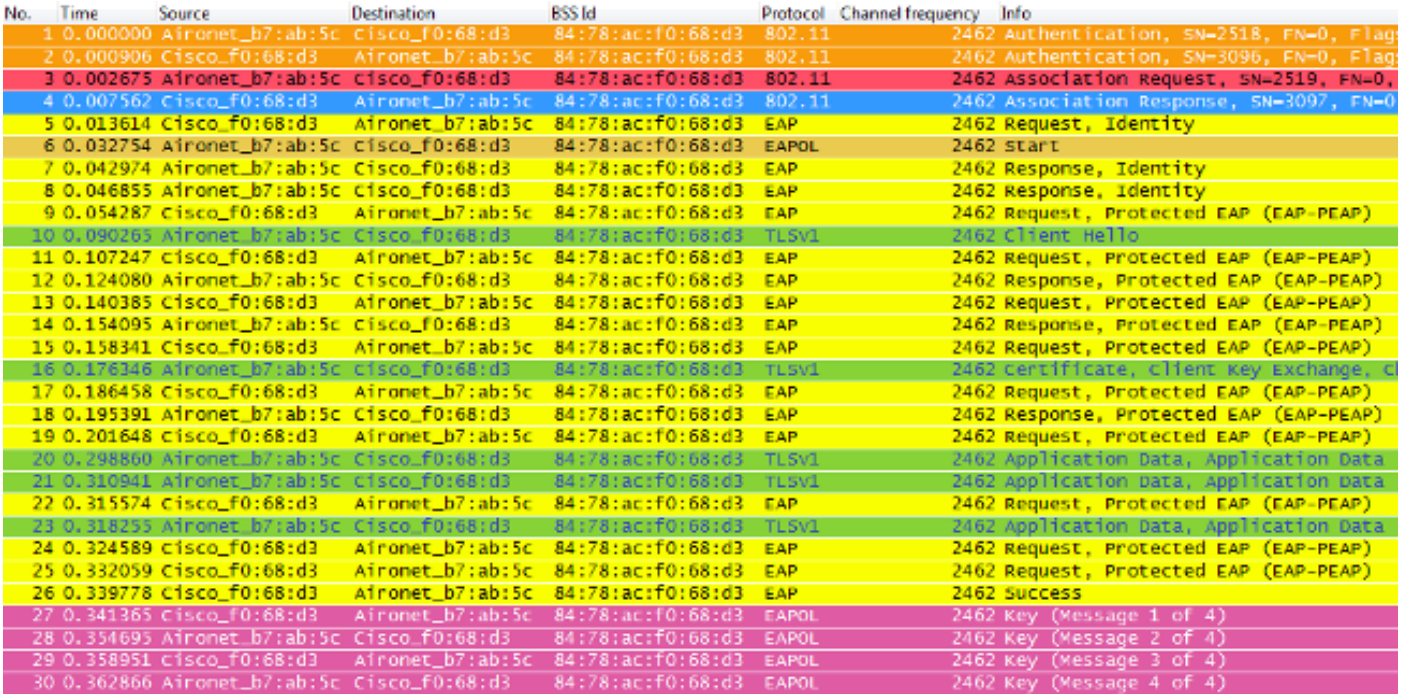
CCKM을 통한 빠른 보안 로밍

CCKM(Cisco Centralized Key Management)은 엔터프라이즈 WLAN에서 개발 및 구현된 최초의 고속 보안 로밍 방식으로서, WLAN에서 802.1X/EAP 보안을 사용할 때 지금까지 설명한 지연을 완화하기 위해 사용된 솔루션으로 Cisco가 개발했습니다. 이 프로토콜은 Cisco 독점 프로토콜이므로 CCKM에 대해 CCX(Cisco Compatible Extension)와 호환되는 Cisco WLAN 인프라 디바이스 및 무선 클라이언트(여러 공급업체)에서만 지원됩니다.

CCKM은 WLAN에 사용할 수 있는 모든 암호화 방법(WEP, TKIP 및 AES)으로 구현할 수 있습니다. 또한 디바이스에서 지원하는 CCX 버전에 따라 WLAN에 사용되는 대부분의 802.1X/EAP 인증 방법을 지원합니다.

> **참고**: 지원되는 EAP 방법을 포함하여 CCX 사양의 다른 버전에서 지원되는 기능 콘텐츠에 대한 개요는 CCX 버전 및 기능 문서를 참조하고 무선 클라이언트(CCX와 호환되는 경우)에서 지원하는 정확한 CCX 버전을 확인하여 CCKM에서 사용하려는 보안 방법을 구현할 수 있는지 확인할 수 있습니다.

이 무선 이미지는 암호화로 TKIP를 사용하고 802.1X/EAP 방법으로 PEAPv0/EAP-MSCHAPv2를 사용하여 CCKM을 수행할 때 초기 연결 시 교환되는 프레임의 예를 제공합니다. 이는 기본적으로 PEAPv0/EAP-MSCHAPv2를 사용하는 WPA/TKIP가 수행되는 것과 동일한 교환이지만 이번에는 클라이언트와 인프라 간의 CCKM이 협상되므로 클라이언트가 로밍해야 할 때 Fast Secure Roaming을 수행하기 위해 서로 다른 키 계층 및 캐시 방법을 사용합니다.



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)

다음은 디버그 메시지의 요약입니다(출력을 줄이기 위해 일부 EAP 교환이 제거됨).

```
*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.
```

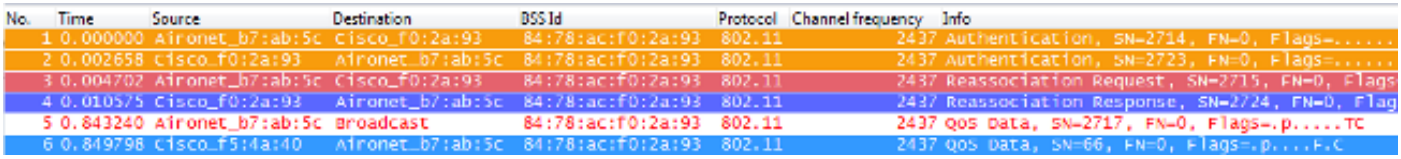
Further EAP messages are not described, as they are basically the same as the ones previously-explained.

```
*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.
```

CCKM에서 WLAN과의 초기 연결은 일반 WPA/WPA2와 비슷합니다. 여기서 MSK(Network Session Key(NSK)라고도 함)는 클라이언트 및 RADIUS 서버와 상호 파생됩니다. 이 기본 키는 성공적인 인증 후 서버에서 WLC로 전송되며, 이 WLAN과의 클라이언트 연결 수명 동안 모든 후속 키를 파생하기 위한 기반으로 캐시됩니다. 여기서 WLC와 클라이언트는 CCKM을 기반으로 신속 보안 로밍에 사용되는 시드 정보를 도출하며, 이는 WPA/WPA2와 유사한 4방향 핸드셰이크를 통해 첫 번째 AP로 유니캐스트(PTK) 및 멀티캐스트/브로드캐스트(GTK) 암호화 키를 도출합니다.

로밍할 때 큰 차이가 느껴집니다. 이 경우 CCKM 클라이언트는 AP/WLC(MIC 및 순차적으로 증가하는 난수 포함)에 단일 재연결 요청 프레임을 전송하고, 새 PTK를 유도하기 위해 충분한 정보(새 AP MAC 주소 **-BSSID-** 포함)를 제공합니다. 이 Reassociation Request를 사용하면 WLC와 새 AP도 새 PTK를 유도하기에 충분한 정보를 가지므로 Reassociation Response로 간단히 응답합니다. 이제 클라이언트는 다음 이미지에 표시된 대로 트래픽을 계속 전달할 수 있습니다.



다음은 이 로밍 이벤트에 대한 WLC 디버그의 요약입니다.

```
*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.
```

그림과 같이 EAP 인증 프레임을 피하고 더 많은 4방향 핸드셰이크를 수행하는 동안 빠른 보안 로밍이 수행됩니다. 새 암호화 키가 아직 파생되지만 CCKM 협상 체계를 기반으로 하기 때문입니다. 이 작업은 로밍 재연결 프레임 및 클라이언트와 WLC에서 이전에 캐시한 정보로 완료됩니다.

FlexConnect와 CCKM

- 중앙 인증이 지원됩니다. 여기에는 로컬 및 중앙 데이터 스위칭이 포함됩니다. AP는 동일한 FlexConnect 그룹에 속해야 합니다.
- Flex Local Authentication이 지원됩니다. 연결 모드에서는 캐시가 AP에서 컨트롤러로, 그런 다음 FlexConnect 그룹의 나머지 AP로 분산될 수 있습니다.
- 독립형 모드가 지원됩니다. 캐시가 이미 AP에 있는 경우(이전 배포로 인해) 빠른 로밍 기능이 작동합니다. 독립형 모드의 새 인증은 빠른 보안 로밍을 지원하지 않습니다.

CCKM의 장점

- CCKM은 가장 빠른 속도의 보안 로밍 방법으로 대부분 엔터프라이즈 WLAN에 구축됩니다. 클라이언트는 AP 간 이동이 발생할 때 새 키를 얻기 위해 키 관리 핸드셰이크를 진행할 필요가 없으며 이 WLAN에서 클라이언트 수명 동안 새 AP로 전체 802.1X/EAP 인증을 수행할 필요가 없습니다.
- CCKM은 레거시 클라이언트에서 계속 사용되는 일부 레거시 Cisco 독점 방법 외에 802.11 표준(WEP, TKIP, AES) 내에서 사용 가능한 모든 암호화 방법을 지원합니다.

CCKM의 단점

- CCKM은 Cisco의 독점적인 방법으로, 구현 및 지원을 Cisco WLAN 인프라 및 CCX 무선 클라이언트로 제한합니다.
- CCX 버전 5는 널리 채택되지 않으므로 WPA2/AES를 사용하는 CCKM은 많은 CCX 무선 클라이언트에서 지원하지 않습니다(주로 대부분의 CCKM이 WPA/TKIP를 사용하는 CCKM을 이미 지원하므로 여전히 매우 안전합니다).

PMKID 캐싱/고정 키 캐싱을 통한 빠른 보안 로밍

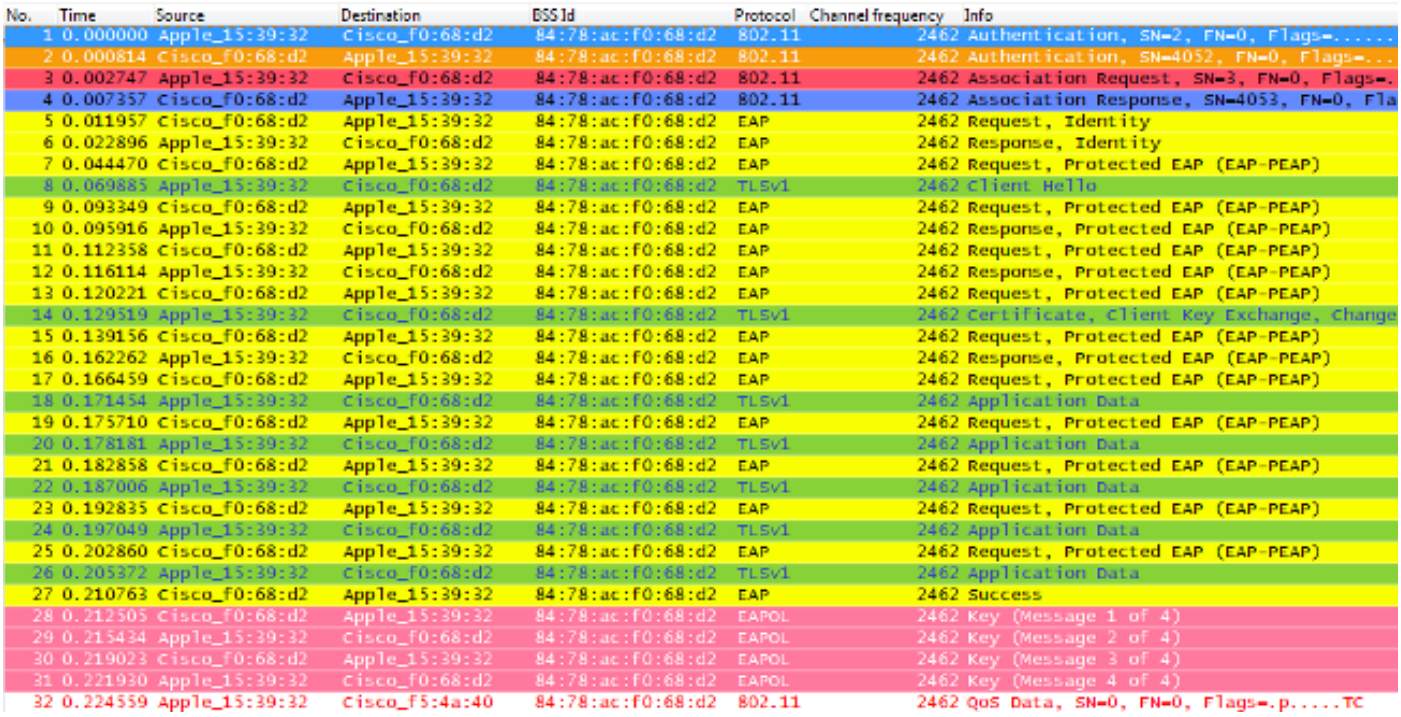
PMKID(Pairwise Think Key ID) 캐싱 또는 **SKC(Sticky Key Caching)**는 802.11i 보안 수정안에서 IEEE 802.11 표준에서 제안하는 첫 번째 고속 보안 로밍 방법입니다. 이 경우 주요 목적은 WLAN에 대한 높은 수준의 보안을 표준화하는 것입니다. 이 빠른 보안 로밍 기술은 이 보안 구현 시 로밍을 개선하기 위해 WPA2 장치에 대한 선택적 방법으로 추가되었습니다.

이는 클라이언트가 완전히 EAP 인증을 받을 때마다 클라이언트와 인증 서버에서 MSK를 파생하기 때문에 가능합니다. MSK는 PMK를 파생하기 위해 사용됩니다. 이 방법은 클라이언트가 다른 AP로 로밍하거나 세션이 만료될 때까지 세션에 사용되는 최종 유니캐스트 암호화 키(PTK)를 파생하기 위해 WPA2 4방향 핸드셰이크의 초기값으로 사용됩니다. 따라서 이 방법은 클라이언트와 AP에 의해 캐시된 원래 PMK를 재사용하므로 로밍할 때 EAP 인증 단계를 방지합니다. 클라이언트는 새 암호화 키를 파생시키기 위해 WPA2 4-Way 핸드셰이크만 통과하면 됩니다.

이 방법은 주로 다음과 같은 이유로 인해 권장되는 802.11 표준 고속 보안 로밍 방법으로 널리 구축되지 않습니다.

- 802.11i 수정안의 목적은 빠른 보안 로밍과 관련이 없으며 IEEE는 WLAN에 대한 빠른 보안 로밍을 표준화하기 위해 이미 다른 수정안에 노력했기 때문에 이 방법은 선택 사항이며 모든 WPA2 디바이스에서 지원되지 않습니다(이 문서의 뒷부분에서 다루는 802.11r).
- 이 방법은 구현에 큰 제한이 있습니다. 무선 클라이언트는 이전에 인증/연결한 AP로 다시 로밍할 때만 빠른 보안 로밍을 수행할 수 있습니다.

이 방법을 사용하면 AP에 대한 초기 연결이 WLAN에 대한 일반적인 첫 번째 인증과 같습니다. 여기서 인증 서버에 대한 전체 802.1X/EAP 인증 및 키 생성을 위한 4-Way 핸드셰이크가 수행되어야 클라이언트가 다음 화면 이미지에 표시된 것처럼 데이터 프레임을 전송할 수 있습니다.



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

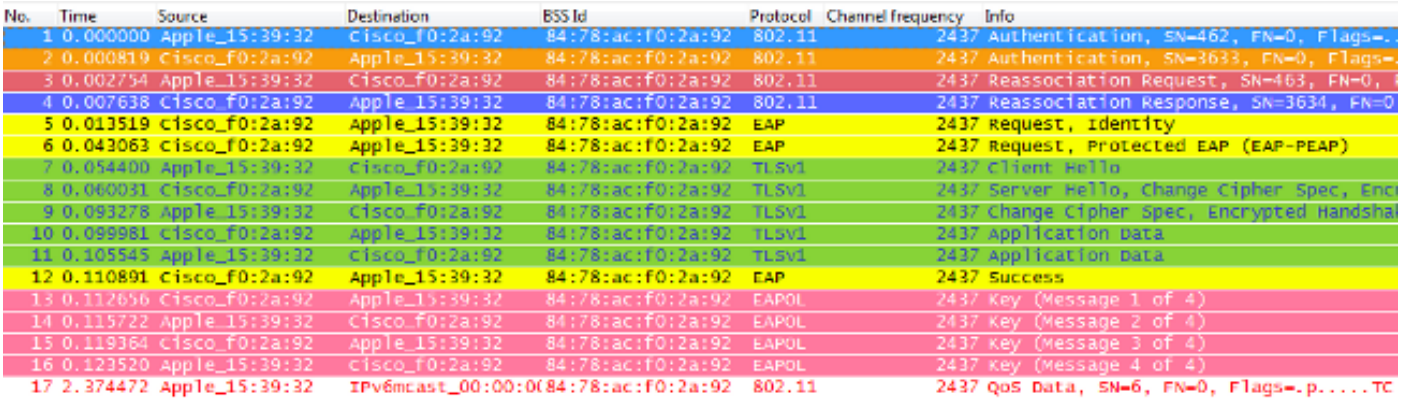
디버그는 WLAN에 대한 초기 인증 시 나머지 방법과 동일한 EAP 인증 프레임 교환을 보여주며, 여기에 사용된 키 캐싱 기술과 관련하여 일부 출력이 추가됩니다. 이러한 디버그 출력은 전체 EAP 프레임 교환이 아니라 주로 새 정보를 표시하기 위해 잘립니다. 기본적으로 인증 서버에 대한 클라이언트 인증을 위해 매번 동일한 정보가 교환되기 때문입니다. 이는 지금까지 설명되었으며 패킷 이미지에 표시된 EAP 인증 프레임과 상관관계가 있으므로 간소화를 위해 대부분의 EAP 메시지가 디버그 출력에서 제거됩니다.

```
*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.
```

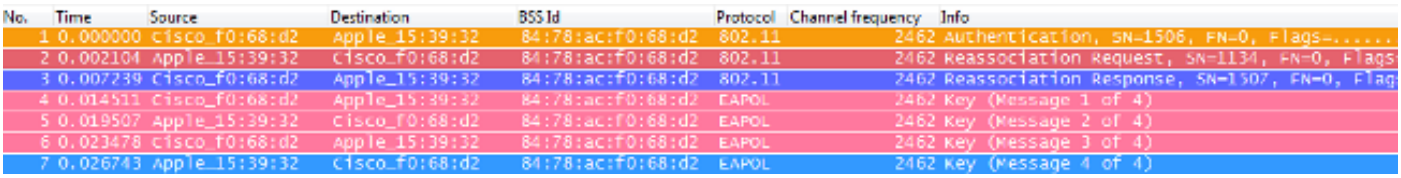
```
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from mobile
  ec:85:2f:15:39:32
!--- Message-2 of the WPA/WPA-2 4-Way handshake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

이 방법을 사용하면 AP 및 무선 클라이언트가 이미 설정된 보안 연결의 PMK를 캐시합니다. 따라서 무선 클라이언트가 연결되지 않은 새 AP로 로밍하는 경우 클라이언트는 다음 그림과 같이 전체 EAP 인증을 다시 수행해야 합니다. 여기서 클라이언트는 새 AP로 로밍합니다.



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

그러나 무선 클라이언트가 이전 연결/인증이 발생한 AP로 다시 로밍하면 클라이언트는 여러 PMKID를 나열하는 Reassociation Request(재연결 요청) 프레임을 전송합니다. 그러면 클라이언트가 이전에 인증된 모든 AP에서 캐시된 PMK를 AP에 알립니다. 따라서 클라이언트가 이 클라이언트에 대해 캐시된 PMK가 있는 AP로 다시 로밍되므로 새 PMK를 파생하기 위해 클라이언트가 EAP를 통해 다시 인증할 필요가 없습니다. 클라이언트는 WPA2 4-Way 핸드셰이크를 통해 새 임시 암호화 키를 파생시킵니다.



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)

참고: 이 이미지는 클라이언트의 첫 번째 802.11 Open System 인증 프레임을 표시하지 않지만, 이 프레임이 항상 필요하므로 구현된 방법 때문은 아닙니다. 그 이유는 이 특정 프레임이 이 예의 경우 무선 프레임을 스니핑하기 위해 사용되는 어댑터 또는 무선 패킷 이미지 소프트웨어에서 이미징되지 않지만, 교육적인 목적을 위해 이 예에서는 이와 같이 남겨두기 때문입니다. OTA(over-the-air) 패킷 이미지를 수행할 때 이러한 현상이 발생할 수 있습니다. 이미지에서 일부 프레임을 놓칠 수 있지만 실제로 클라이언트와 AP 간에 교환됩니다. 그렇지 않으면 이 예에서는 로밍이 시작되지 않습니다.

다음은 이 빠른 보안 로밍 방법에 대한 WLC 디버그의 요약입니다.

```
*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
```

PMKID 캐싱/고정 키 캐싱을 사용하는 FlexConnect

- FlexConnect 설정에서 이 방법을 사용하면 작동할 수 있으며, 중앙 인증을 WLC로 다시 사용하는 경우(중앙 또는 로컬 스위칭 사용) 동작이 이전에 설명한 것과 비슷할 수 있습니다. 그러나 이 SKC 방법은 FlexConnect에서 지원되지 않습니다.
- 이 방법은 FlexConnect 또는 기타 모드가 아닌 로컬 모드 AP가 있는 CUWN에서만 공식적으로 지원됩니다.

PMKID 캐싱/고정 키 캐싱의 장점

이 방법은 캐시된 키를 관리하기 위해 중앙 집중식 디바이스가 필요 없이 자동 독립적인 AP에 의해 로컬로 구현될 수 있습니다.

PMKID 캐싱/고정 키 캐싱의 단점

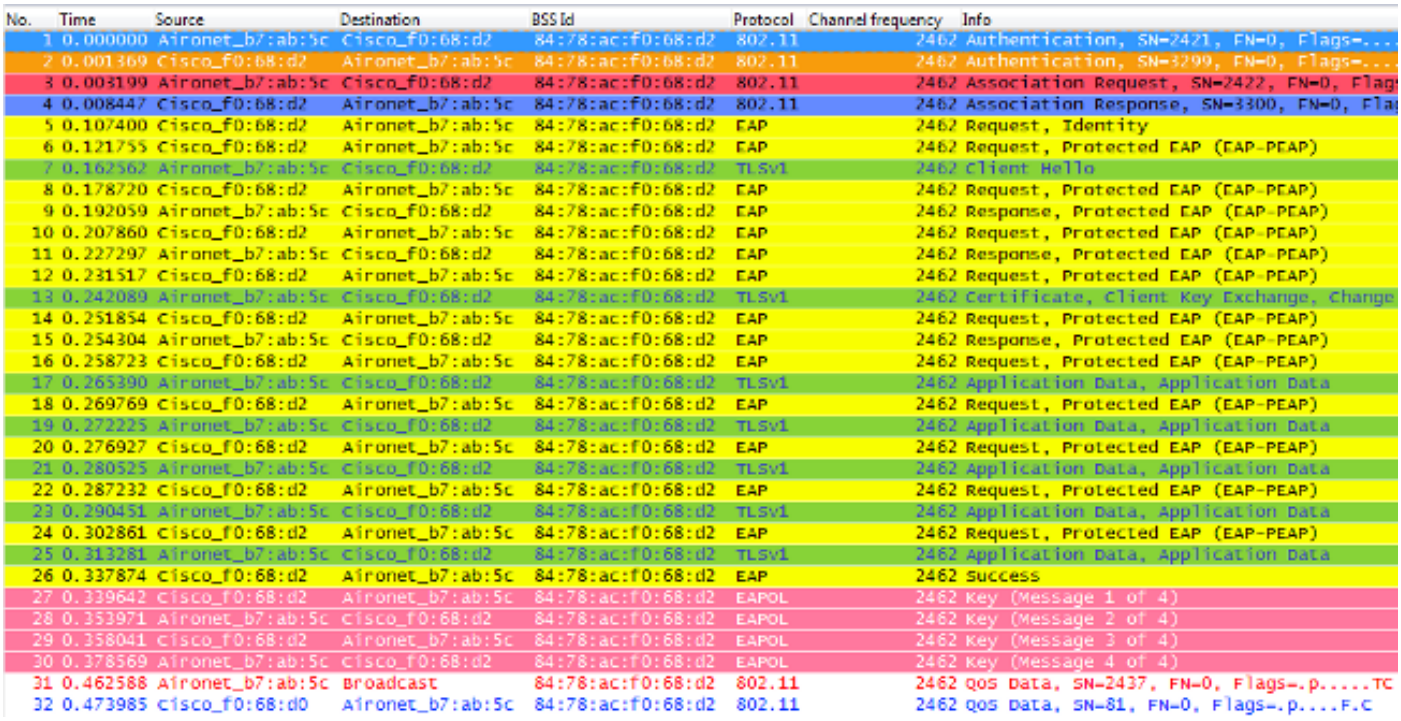
- 이 문서의 앞부분에서 설명한 것처럼, 클라이언트가 이전에 연결/인증된 AP로 다시 로밍할 때만 빠른 보안 로밍을 수행할 수 있다는 점이 이 방법의 주요 한계입니다. 새 AP로 로밍하는 경우 클라이언트는 전체 EAP 인증을 다시 완료해야 합니다.
- 무선 클라이언트 및 AP는 모든 새 인증에서 파생된 모든 PMK를 기억해야 합니다. 따라서 이 기능은 일반적으로 캐시된 특정 양의 PMK로 제한됩니다. 이 제한은 표준에 의해 명확하게 정의되어 있지 않으므로 공급업체는 SKC 구현에 다른 제한을 정의할 수 있습니다. 예를 들어 Cisco WLAN Controller는 현재 최대 8개의 AP에 대해 클라이언트에서 PMK를 캐시할 수 있습니다. 클라이언트가 세션당 8개 이상의 AP로 로밍하는 경우 새로 캐시된 항목을 저장하기 위해 가장 오래된 AP가 캐시 목록에서 제거됩니다.
- 이 방법은 선택 사항이며 여전히 많은 WPA2 장치에서 지원되지 않으므로 널리 채택되고 구축되지 않습니다.
- SKC는 컨트롤러 간 로밍을 수행할 때 지원되지 않습니다. 이는 서로 다른 WLC에서 관리하는 AP 간에 이동할 때 발생하며, 동일한 모빌리티 그룹에 있는 경우에도 마찬가지입니다.

기회주의적 키 캐싱을 통한 빠른 보안 로밍

PKC(Proactive Key Caching)라고도 하는 OKC(Opportunistic Key Caching)는 기본적으로 앞에서 설명한 WPA2 PMKID 캐싱 방법을 개선한 것입니다. 이 방법을 Proactive/Opportunistic PMKID 캐싱이라고도 합니다. 따라서 802.11 표준에 정의된 빠른 보안 로밍 방법이 아니며 많은 디바이스에서 지원되지 않지만 PMKID 캐싱과 마찬가지로 WPA2-EAP에서 작동합니다.

이 기술을 사용하면 무선 클라이언트와 WLAN 인프라에서 이 WLAN과의 클라이언트 연결 수명 동안 하나의 PMK만 캐시할 수 있습니다(인증 서버와 함께 초기 802.1X/EAP 인증 후 MSK에서 파생됨). 여러 AP 간에 로밍할 경우에도 모든 WPA2 4방향 핸드셰이크에서 시드로 사용되는 원래 PMK를 공유하므로 이러한 AP가 모두 AP를 공유합니다. 클라이언트가 AP와 다시 연결할 때마다 새 암호화 키를 생성하려면 SKC에서와 마찬가지로 이 작업이 필요합니다. AP가 클라이언트 세션에서 이 원본 PMK를 공유하려면 모든 AP에 대해 원본 PMK를 캐시하고 배포하는 중앙 집중식 디바이스를 사용하여 일종의 관리 제어를 받아야 합니다. 이는 WLC가 제어 중인 모든 LAP에 대해 이 작업을 수행하고 여러 WLC 간에 이 PMK를 처리하기 위해 모빌리티 그룹을 사용하는 CUWN과 유사합니다. 따라서 이는 자율 AP 환경에 대한 제한입니다.

이 방법을 사용하면 PMKID 캐싱(SKC)에서와 마찬가지로 AP에 대한 초기 연결은 WLAN에 대한 일반적인 첫 번째 인증이며, 여기서 인증 서버에 대한 전체 802.1X/EAP 인증 및 키 생성을 위한 4방향 핸드셰이크를 완료해야 데이터 프레임을 전송할 수 있습니다. 다음은 이를 보여 주는 화면 이미지입니다.



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

디버그 출력은 기본적으로 WLAN에 대한 초기 인증 시(이미지에 표시된 것처럼) 이 문서에 설명된 나머지 방법과 동일한 EAP 인증 프레임 교환을 보여주며, 여기에 WLC에서 사용하는 키 캐싱 기술과 관련된 일부 출력을 추가합니다. 관련 정보만 표시하기 위해 이 디버그 출력도 잘라냅니다.

```
*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
```

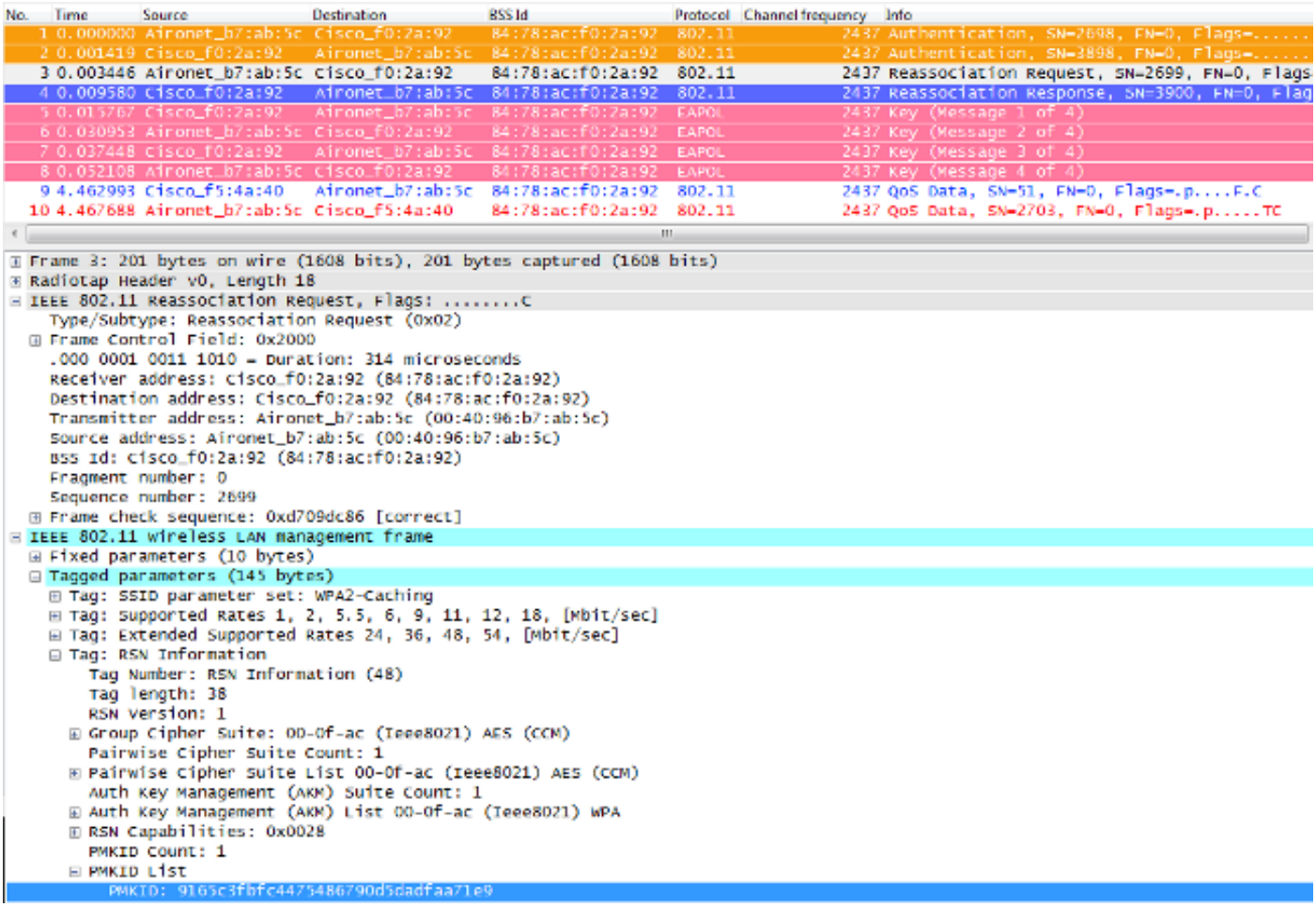
```
  Sending EAP-Request/Identity to mobile 00:40:96:b7:ab:5c
  (EAP Id 2)

*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
```

이 방법을 사용하면 무선 클라이언트와 WLC(모든 관리 대상 AP에 해당)는 초기에 설정된 보안 연결의 원래 PMK를 캐시합니다. 기본적으로 무선 클라이언트가 특정 AP에 연결할 때마다 클라이언트 MAC 주소, AP MAC 주소(WLAN의 BSSID) 및 해당 AP에서 파생된 PMK를 기반으로 PMKID가 해시됩니다. 따라서 OKC는 모든 AP 및 특정 클라이언트에 대해 동일한 원본 PMK를 캐시하므로 이 클라이언트가 다른 AP에 다시 연결할 때 새 PMKID를 해시하기 위해 변경되는 유일한 값은 새 AP MAC 주소입니다.

클라이언트가 새 AP로 로밍을 시작하고 재연결 요청 프레임을 전송하면, 캐시된 PMK가 빠른 보안 로밍에 사용됨을 AP에 알리려면 WPA2 RSN 정보 요소에 PMKID를 추가합니다. 로밍할 BSSID(AP)의 MAC 주소를 이미 알고 있으므로 클라이언트는 이 재연결 요청에 사용되는 새 PMKID를 해시하기만 합니다. AP는 클라이언트로부터 이 요청을 받으면 이미 가지고 있는 값(캐시된 PMK, 클라이언트 MAC 주소 및 자체 AP MAC 주소)으로 PMKID를 해시하고 PMKID가 일치하는지 확인하는 성공적인 재연결 응답으로 응답합니다. 캐시된 PMK는 새 암호화 키를 파생시키고 EAP를 건너뛰기 위해 WPA2 4방향 핸드셰이크를 시작하는 시드로 사용할 수 있습니다.



이 그림에서는 클라이언트의 Reassociation Request(재연결 요청) 프레임이 선택되어 확장되므로 프레임의 세부 사항을 더 자세히 볼 수 있습니다. MAC 주소 정보 및 RSN(Robust Security Network, 802.11i - WPA2) 정보 요소. 여기서 이 연결에 사용되는 WPA2 설정에 대한 정보가 표시됩니다(해시된 공식에서 얻은 PMKID가 강조 표시되어 있음).

다음은 OKC를 사용하는 이 빠른 보안 로밍 방법에 대한 WLC 디버그의 요약입니다.

```
*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
```

디버그의 시작 부분에 표시된 것처럼 PMKID는 클라이언트로부터 재연결 요청을 받은 후에 계산되어야 합니다. 이는 PMKID를 검증하고 캐시된 PMK가 WPA2 4-Way 핸드셰이크와 함께 사용되어 암호화 키를 파생시키고 빠른 보안 로밍을 완료하는지 확인하기 위해 필요합니다. 디버그의 CCKM 항목을 혼동하지 마십시오. 이는 CCKM을 수행하기 위해 사용되지 않지만 앞서 설명한 대로 OKC입니다. 여기서 CCKM은 PMKID를 계산하기 위해 값을 처리하는 함수의 이름과 같이 해당 출력에 대해 WLC에서 사용하는 이름입니다.

FlexConnect(기회주의적 키 캐싱)

- 중앙 인증이 지원됩니다. 여기에는 로컬 및 중앙 데이터 스위칭이 포함됩니다. AP가 동일한 FlexConnect 그룹에 속해 있는 경우 AP가 빠른 보안 로밍 기능을 제공하며, 그렇지 않으면 컨

트롤러가 빠른 보안 로밍 기능을 제공합니다.
참고: AP가 동일한 FlexConnect 그룹에 없는 경우 이 설정이 작동할 수 있지만, 권장 또는 지원되는 설정이 아닙니다.
- Flex Local Authentication이 지원됩니다. 연결 모드에서는 캐시가 AP에서 컨트롤러로, 그런 다음 FlexConnect 그룹의 나머지 AP로 분산될 수 있습니다.
- 독립형 모드가 지원됩니다. 캐시가 이미 AP에 있는 경우(이전 배포로 인해) 빠른 보안 로밍 기능이 작동합니다. 독립형 모드의 새 인증은 빠른 보안 로밍을 지원하지 않습니다.

기회주의적 키 캐싱의 장점

- 무선 클라이언트 및 WLAN 인프라는 여러 PMKID를 기억할 필요가 없으며, 단순히 초기 인증에서 WLAN으로 하나의 원래 PMK를 캐시하기만 하면 됩니다. 그런 다음 각 AP 보안 연결에 필요한 적절한 PMKID(재연결 요청에 사용됨)를 재해시하여 빠른 보안 로밍을 검증해야 합니다.
- 여기서 무선 클라이언트는 동일한 WLAN/SSID의 새 AP에 대해 빠른 보안 로밍을 수행합니다(SKC의 경우 아님). 클라이언트가 클라이언트가 로밍하는 모든 AP에 대해 PMK 캐시를 처리하는 중앙 집중식 구축에 의해 관리되는 하나의 AP로 초기 802.1X/EAP 인증을 수행하는 한 이 WLAN의 나머지 클라이언트 수명 동안 더 이상 전체 인증이 필요하지 않습니다.

기회주의적 키 캐싱의 단점

- 이 방법은 모든 AP가 클라이언트 세션에서 원래 PMK를 캐시하고 공유하는 일종의 관리 제어(예: WLAN 컨트롤러)를 담당하는 중앙 집중식 환경에서만 구축됩니다. 따라서 이는 자동 AP 환경에 대한 제한입니다.
- 이 방법에서 적용되는 기술은 802.11 표준에 제안되거나 설명되어 있지 않으므로, 각 장치마다 지원이 폭넓게 다릅니다. 그럼에도 불구하고 802.11r을 기다리면서 더 채택된 방법은 여전히 이것이다.

"Proactive Key Caching" 용어 참고

Proactive Key Caching(또는 PKC)은 OKC(Opportunistic Key Caching)로 알려져 있으며, 여기에서 설명한 것과 동일한 방법을 설명할 때 두 용어를 혼용하여 사용합니다. 그러나 이는 2001년 Airspace에서 기존 키 캐싱 방법을 위해 사용한 용어일 뿐이며, 802.11i 표준에서 "사전 인증"(아래에서 간략하게 설명하는 또 다른 빠른 보안 로밍 방법)의 기반으로 사용되었습니다. PKC는 Preauthentication 또는 OKC(Opportunistic Key Caching)가 아니지만, PKC에 대해 듣거나 읽을 때 기본적으로 참조는 Preauthentication이 아니라 OKC입니다.

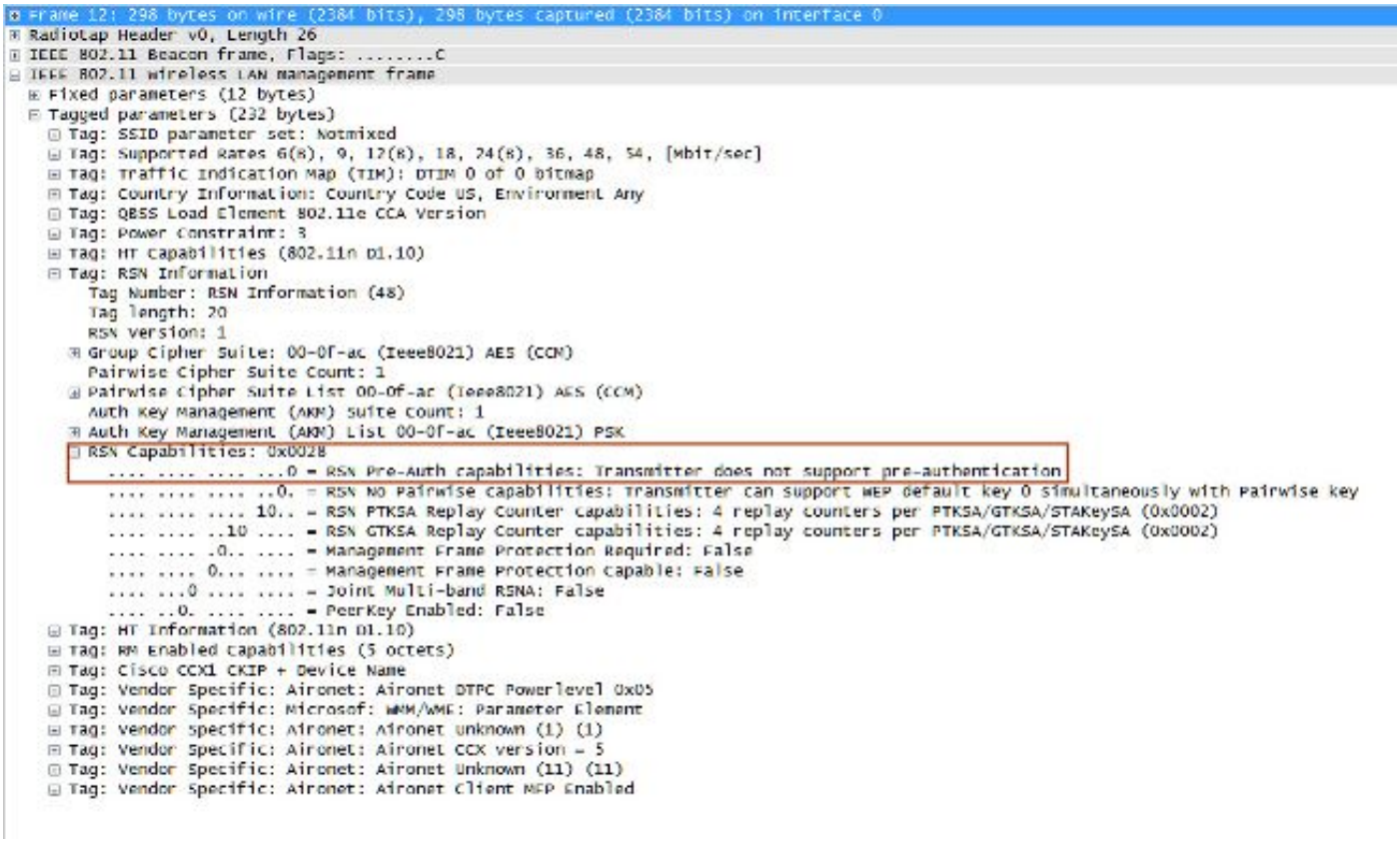
사전 인증을 통한 빠른 보안 로밍

이 방법은 802.11i 보안 수정 내에서 IEEE 802.11 표준에서도 제안하므로 WPA2에서도 작동하지만 Cisco WLAN 인프라에서 지원하지 않는 유일한 고속 보안 로밍 방법입니다. 이러한 이유로 여기서는 간략히 설명하고 산출물 없이 설명한다.

사전 인증을 사용하면 무선 클라이언트가 현재 AP와 연결되어 있는 동안 한 번에 여러 AP로 인증할 수 있습니다. 이 경우 클라이언트는 연결된 현재 AP에 EAP 인증 프레임을 전송하지만, 클라이언트가 사전 인증을 수행하려는 다른 AP(로밍할 수 있는 인접 AP)로 전달됩니다. 현재 AP는 이러한 프레임을 배포 시스템을 통해 대상 AP에 전송합니다. 새 AP는 이 클라이언트에 대해 RADIUS 서버에 대해 전체 인증을 수행하므로 전체 새 EAP 인증 핸드셰이크가 완료되며 이 새 AP는 인증자 역할을 합니다.

이 아이디어는 클라이언트가 실제로 로밍하기 전에 인접 AP와 인증을 수행하고 PMK를 파생시키는 것입니다. 따라서 로밍할 시간이 되면 클라이언트는 이미 인증되고 이 새로운 AP-클라이언트 보안 연결을 위해 이미 캐시된 PMK를 사용하므로 클라이언트가 초기 재연결 요청을 보낸 후 4-Way Handshake를 수행하고 빠른 로밍을 경험하면 됩니다.

다음은 사전 인증에 대한 지원을 광고하는 RSN IE 필드를 표시하는 AP 비컨의 이미지입니다(이 필드는 사전 인증이 지원되지 않는 것으로 확인된 Cisco AP의 것입니다).



사전 인증을 사용한 장점

각 AP-클라이언트 보안 연결에는 PMK가 하나씩 있습니다. 이는 AP가 손상되어 키가 도난(다른 AP와 함께 사용할 수 없음)될 경우 보안 장점으로 간주될 수 있습니다. 그러나 이러한 보안 이점은 WLAN 인프라에서 다른 방법으로 처리합니다.

사전 인증을 통한 단점

- AP당 하나의 PMK가 있으므로 클라이언트는 사전 인증할 수 있는 AP의 양에 제한이 있습니다.
- 클라이언트가 새 AP로 사전 인증을 수행할 때마다 완전한 EAP 인증 교환이 이루어지므로 네트워크 및 인증 서버에 더 많은 로드가 발생합니다.
- 대부분의 무선 클라이언트는 이 방법을 지원하지 않습니다. 이는 고도로 채택된 적이 없기 때문입니다(OKC가 더 채택되었습니다).

802.11r을 통한 빠른 보안 로밍

802.11r 수정(공식적으로 802.11 표준에 의한 **Fast BSS Transition**, **FT**로 알려짐)에 기반한 고속 보안 로밍 기술은 AP(Basic Service Sets 또는 BSS) 간에 고속 전환을 수행하는 솔루션으로서 IEEE에서 802.11 표준에 대해 공식적으로 비준한 첫 번째 방법입니다(2008년 기준). 이 방법은 WLAN에서 키를 처리하고 캐시할 때 사용되는 키 계층 구조를 명확하게 정의합니다. 그러나 이 문

서에서 앞서 설명한 방법 중 하나와 함께 사용할 경우 VoWLAN 구현과 같이 빠른 전환이 실제로 필요할 때 이미 사용 가능한 다른 솔루션 때문에 채택이 느려졌습니다. 현재 일부 FT 옵션을 지원하는 디바이스는 몇 개에 불과합니다(2013년 기준).

이 기술은 다른 방법보다 설명하기가 더 복잡합니다. 새로운 개념과 여러 PMK 레이어를 도입하여 서로 다른 장치(각각 다른 역할을 가진 장치)에 캐시하고 빠른 보안 로밍을 위한 더 많은 옵션을 제공하기 때문입니다. 따라서 이 방법 및 각 옵션을 사용하여 이 방법을 구현하는 방법에 대한 간략한 요약이 제공됩니다.

802.11r은 SKC 및 OKC와 다르며 주로 다음과 같은 이유 때문입니다.

- 핸드셰이크 메시징(예: PMKID, ANonce 및 SNonce 교환)은 802.11 인증 프레임 또는 Reassociation 프레임 대신 Action 프레임에서 발생합니다. PMKID 캐싱 방법과 달리, (재)연결 메시지 교환 이후에 수행되는 별도의 4방향 핸드셰이크 단계는 피합니다. 클라이언트가 새 AP를 완전히 로밍/재연결하기 전에 새 AP와의 키 핸드셰이크가 시작됩니다.
- 빠른 로밍 핸드셰이크에 대한 두 가지 방법, 즉 AIR를 통한 핸드셰이크와 DS(Distribution System)를 통한 핸드셰이크를 제공합니다.
- 802.11r에는 더 많은 키 계층 구조가 있습니다.
- 이 프로토콜은 클라이언트가 로밍할 때 키 관리를 위해 4방향 핸드셰이크를 피하므로(이 핸드셰이크의 필요 없이 새 암호화 키 -PTK 및 GTK- 생성), PSK를 사용하는 WPA2 설정에 적용할 수 있으며 802.1X/EAP가 인증에 사용되는 경우에만 적용할 수 있습니다. 이렇게 하면 EAP 또는 4-Way 핸드셰이크 교환이 발생하지 않는 이러한 설정에서 로밍이 더욱 가속화됩니다.

이 방법을 사용하면 무선 클라이언트는 첫 번째 AP에 연결이 설정되면 WLAN 인프라에 대해 하나의 초기 인증만 수행하며 동일한 FT 모빌리티 도메인의 AP 간에 로밍하는 동안 빠른 보안 로밍을 수행합니다.

이는 새로운 개념 중 하나로, 기본적으로 동일한 SSID(Extended Service Set 또는 ESS라고 함)를 사용하고 동일한 FT 키를 처리하는 AP를 가리킵니다. 이는 지금까지 설명한 다른 방법들과 유사하다. AP가 FT 모빌리티 도메인 키를 처리하는 방식은 일반적으로 WLC 또는 모빌리티 그룹과 같은 중앙 집중식 설정을 기반으로 합니다. 그러나 이 방법은 자동 AP 환경에서도 구현할 수 있습니다.

다음은 주요 계층 구조의 요약입니다.

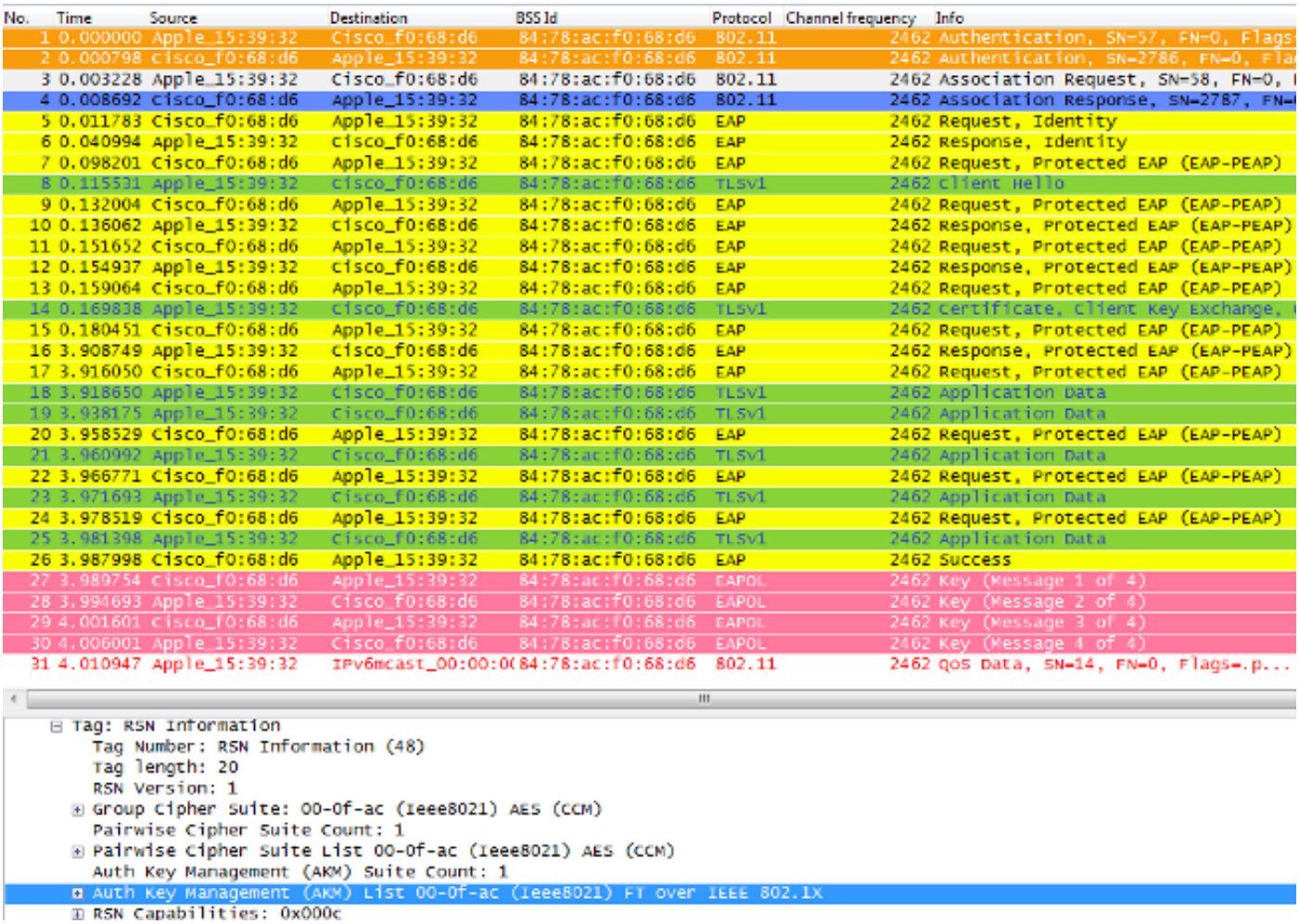
- MSK는 초기 802.1X/EAP 인증 단계(인증이 성공하면 인증 서버에서 인증자(WLC)로 전송됨)의 클라이언트 신청자 및 인증 서버에서 계속 파생됩니다. 이 MSK는 다른 방법과 마찬가지로 FT 키 계층 구조의 초기값으로 사용됩니다. EAP 인증 방법 대신 WPA2-PSK를 사용하는 경우 PSK는 기본적으로 이 MSK입니다.
- FT 키 계층의 첫 번째 레벨 키인 MSK에서 Pairwise Master Key R0(PMK-R0)이 파생된다. 이 PMK-R0의 키 홀더는 WLC와 클라이언트입니다.
- PMK-R1(Pairwise Master Key R1)이라고 하는 두 번째 레벨 키는 PMK-R0에서 파생되며, 키 홀더는 PMK-R0을 보유하는 WLC에 의해 관리되는 클라이언트 및 AP입니다.
- FT 키 계층의 세 번째 및 최종 레벨 키는 PTK이며, 이는 802.11 유니캐스트 데이터 프레임을 암호화하는 데 사용되는 최종 키입니다(WPA/TKIP 또는 WPA2/AES를 사용하는 다른 방법과 유사). 이 PTK는 PMK-R1의 FT에서 파생되며, 키 홀더는 WLC에서 관리하는 클라이언트와 AP입니다.

  참고: WLAN 공급업체 및 구현 설정(예: 자동 AP, FlexConnect 또는 메시)에 따라 WLAN 인프라는 다른 방식으로 키를 전송하고 처리할 수 있습니다. 핵심 보유자의 역할도 변경할 수 있지만, 이는 이 문서의 범위를 벗어나므로 앞에서 설명한 주요 계층 요약에 기반한 예가 다음 주안점입니다. 소프트웨어 문제를 발견하기 위해 인프라 장치(및 해당 코드)를 심층적으로 분석

할 필요가 없는 한, 그 차이점은 실제로 프로세스를 이해하는 데 그다지 관련이 없습니다.

빠른 BSS 무선 전환

이 방법을 사용하는 경우 AP에 대한 첫 번째 연결은 WLAN에 대한 첫 번째 정기 인증이며, 여기서 이 화면 이미지에 표시된 대로 데이터 프레임을 전송하기 전에 인증 서버에 대한 전체 802.1X/EAP 인증 및 키 생성을 위한 4-Way 핸드셰이크가 발생해야 합니다.



주요 차이점은 다음과 같습니다.

- 인증 키 관리 협상은 일반 WPA/WPA2와 약간 다르므로 FT를 지원하는 WLAN 인프라에 연결할 때 이 협상을 수행하기 위해 몇 가지 추가 정보가 사용됩니다. 그림과 같이 클라이언트의 Association Request(연결 요청) 프레임이 선택되고 RNS Information Element(RNS 정보 요소)의 AKM 필드가 강조 표시되어 이 클라이언트가 802.1X/EAP를 통해 FT를 수행하려고 함을 보여 줍니다.
- 또한 Mobility Domain Information Element(FT의 일부)도 표시되어 있습니다. 여기서 **FT Capability and Policy(FT 기능 및 정책)** 필드는 빠른 로밍 시 Fast BSS Transition이 Over-the-Air 또는 Over-the-DS로 완료되었는지 여부를 나타냅니다(이 이미지에서는 Over-the-Air를 나타냄).
- FT 로밍 시에 FT 인증 시퀀스를 수행하기 위해 필요한 정보와 함께 또 다른 정보 요소(Fast BSS Transition 또는 FT IE, 본 문서에서 나중에 설명됨)도 추가된다.
- 키 계층 구조상 키 생성이 다르므로 FT 4Way 핸드셰이크가 WPA/WPA2 4Way 핸드셰이크와 비슷하게 보이지만 실제로는 내용이 약간 다릅니다.

디버그는 기본적으로 WLAN에 대한 초기 인증 시 나머지 방법과 동일한 EAP 인증 프레임 교환을

보여줍니다(이미지에서 알 수 있듯이). 그러나 WLC에서 사용하는 키 캐싱 기술과 관련된 일부 출력이 추가되므로 이 디버그 출력은 관련 정보만 표시하기 위해 잘립니다.

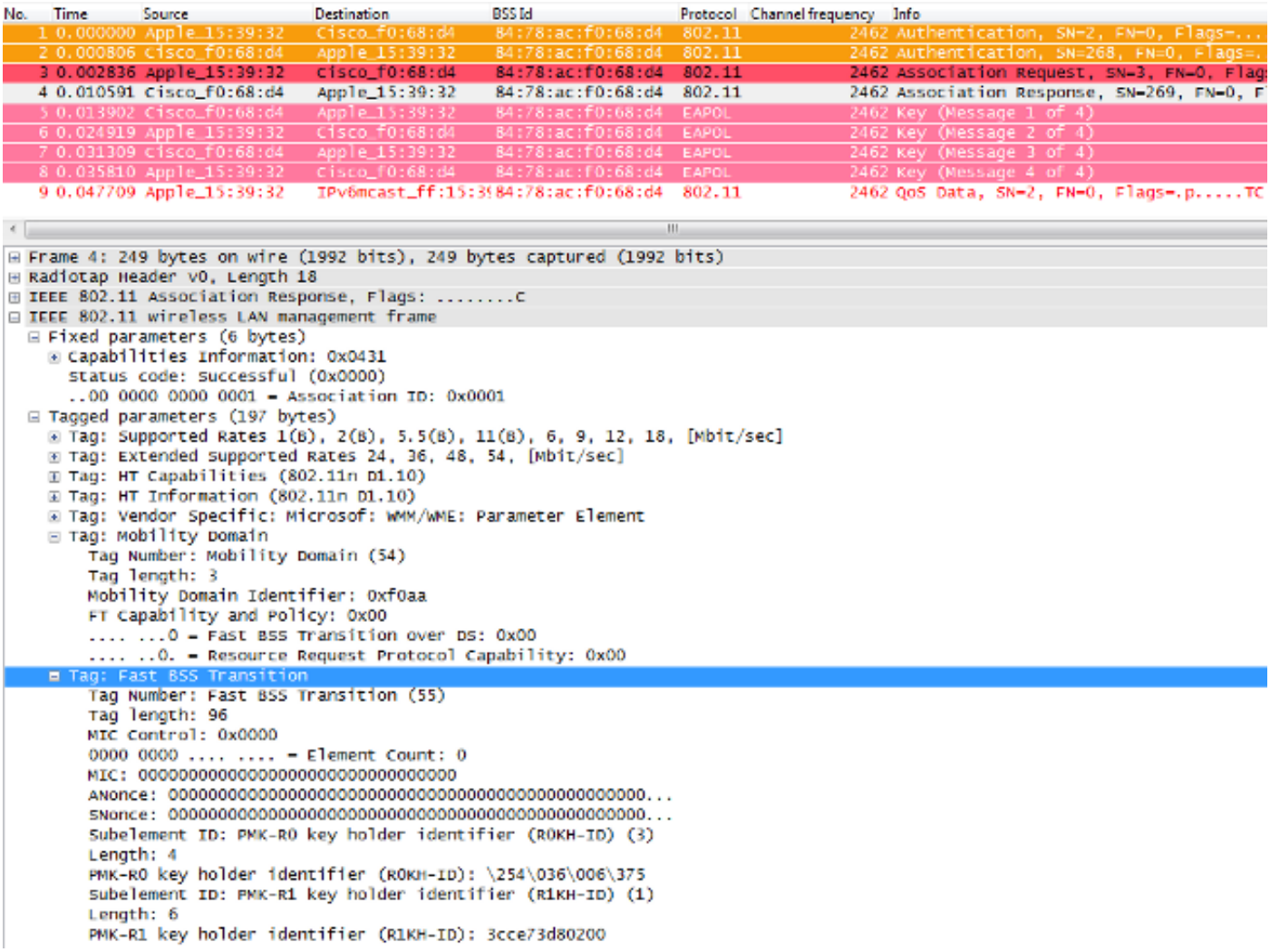
```
*apfMsConnTask_0: Jun 27 19:25:23.426: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d6
!--- This is the Association request from the client.

*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.
```

> **참고**: 이 메서드를 디버깅하고 여기에 표시된 추가 802.11r/FT 출력에 도달하기 위해 디버그 **ft 이벤트**가 활성화되는 **디버그 클라이언트**와 함께 **추가 디버그가 활성화됩니다**.

다음은 WPA2-PSK를 사용하여 FT를 수행할 때(802.1X/EAP 방식 대신) WLAN에 대한 초기 연결의 이미지 및 디버깅입니다. 여기서는 Fast BSS Transition Information Element(강조 표시)를 표시하기 위해 AP의 연결 응답 프레임이 선택됩니다. FT 4-Way 핸드셰이크를 수행하는 데 필요한 몇 가지 주요 정보도 표시됩니다.



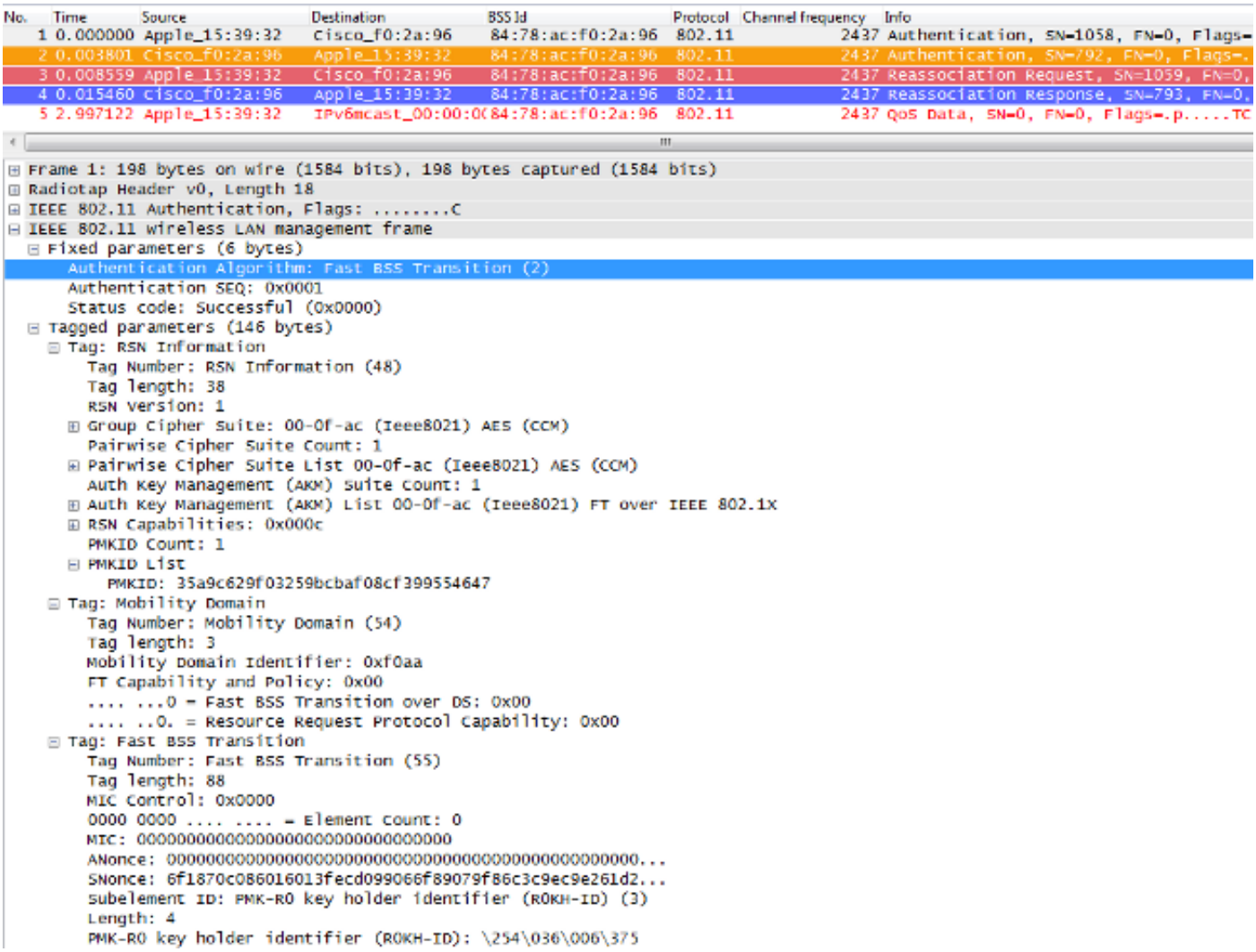

```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*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
```

802.11r에서는 다른 고속 보안 로밍 방법과 마찬가지로 WLAN에 대한 초기 연결을 기반으로 이 기술에 사용되는 기본 키를 파생시킵니다. 주요 차이점은 클라이언트가 로밍을 시작할 때 발생합니다. FT는 이를 사용할 때 802.1X/EAP를 피할 뿐만 아니라, 실제로 더 효율적인 로밍 방법을 수행하여 초기 802.11 Open System Authentication 및 Reassociation 프레임(AP 간 로밍할 때 항상 사용되고 필요함)을 결합하여 FT 정보를 교환하고 4방향 핸드셰이크 대신 새로운 동적 암호화 키를 파생시킵니다.

다음 이미지는 802.1X/EAP 보안을 사용하는 Fast BSS Transition Over-the-Air가 수행될 때 교환되는 프레임을 보여줍니다. FT 키 협상을 시작하는 데 필요한 FT 프로토콜 정보 요소를 보기 위해 클라이언트에서 AP로의 Open System Authentication 프레임이 선택됩니다. 이는 새로운 AP로 새로운 PTK를 유도하기 위해 사용된다(PMK-R1 기준). 이 클라이언트가 단순 개방형 시스템 인증이 아니라 고속 BSS 전환을 수행함을 나타내기 위해 인증 알고리즘을 표시하는 필드가 강조 표시됩니다.

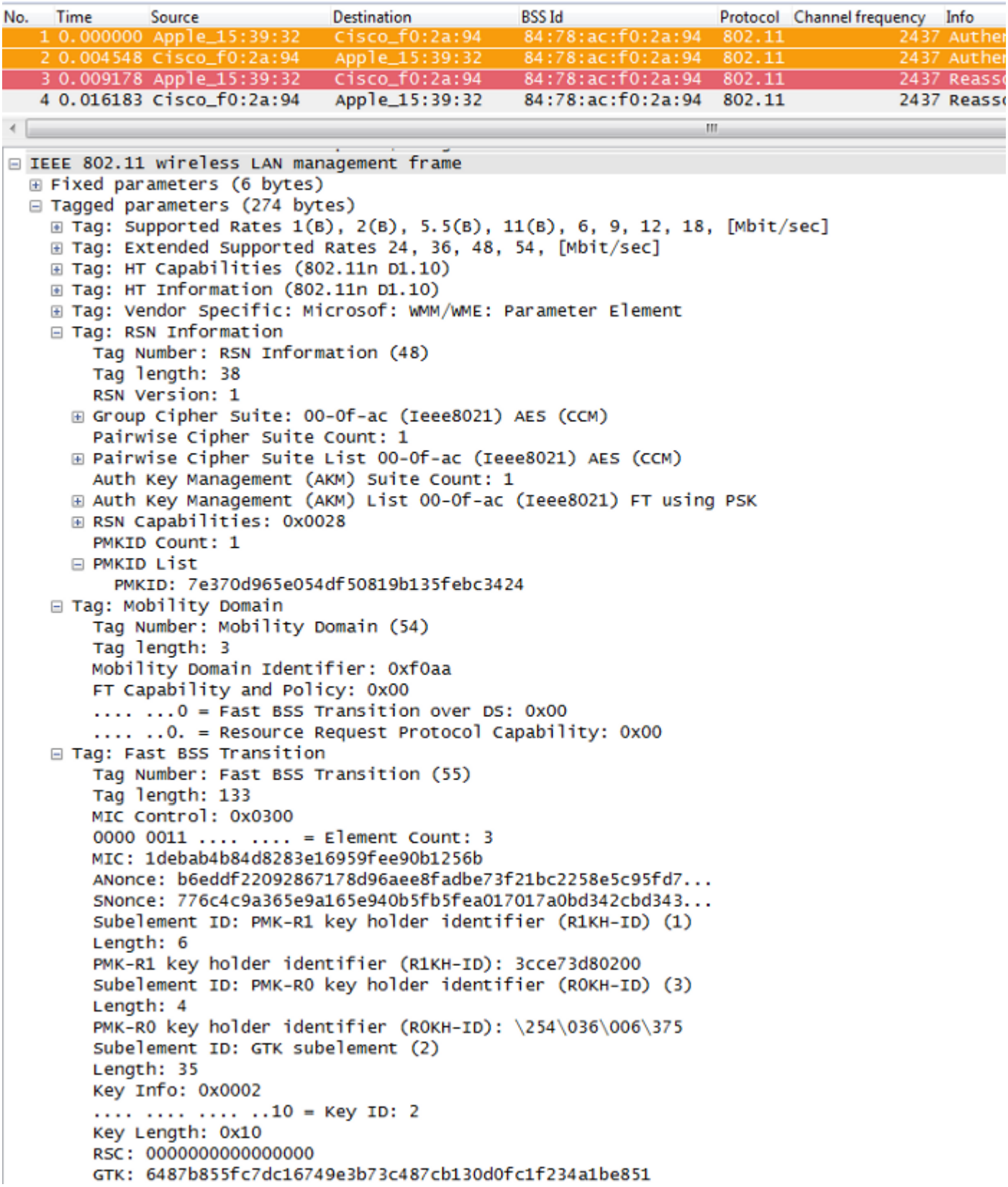


다음은 802.1X/EAP로 이 FT 로밍 이벤트가 발생할 때 WLC의 디버그 출력입니다.

```
*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
```

다음은 WPA2-PSK 보안과 함께 Fast BSS Transition Over-the-Air를 보여 주는 이미지입니다. 여기서는 이 FT 교환에 대한 자세한 내용을 보여 주기 위해 AP에서 클라이언트로의 최종 재연결 응답 프레임이 선택됩니다.



다음은 PSK에서 이 FT 로밍 이벤트가 발생할 때의 디버그 출력입니다. 이는 802.1X/EAP가 사용될 때의 출력과 유사합니다.

```
*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
```

이미지에 표시된 것처럼, WLAN에 대한 초기 연결 시 Fast BSS Transition이 협상되면, 새로운 PTK(유니캐스트 암호화 키) 및 GTK(멀티캐스트/브로드캐스트 암호화 키)를 유도하기 위해 로밍에 사용되고 필요한 4개의 프레임(클라이언트에서 오픈 시스템 인증, AP에서 오픈 시스템 인증, 재연결 요청 및 재연결 응답)이 기본적으로 FT 4방향 핸드셰이크로 사용됩니다.

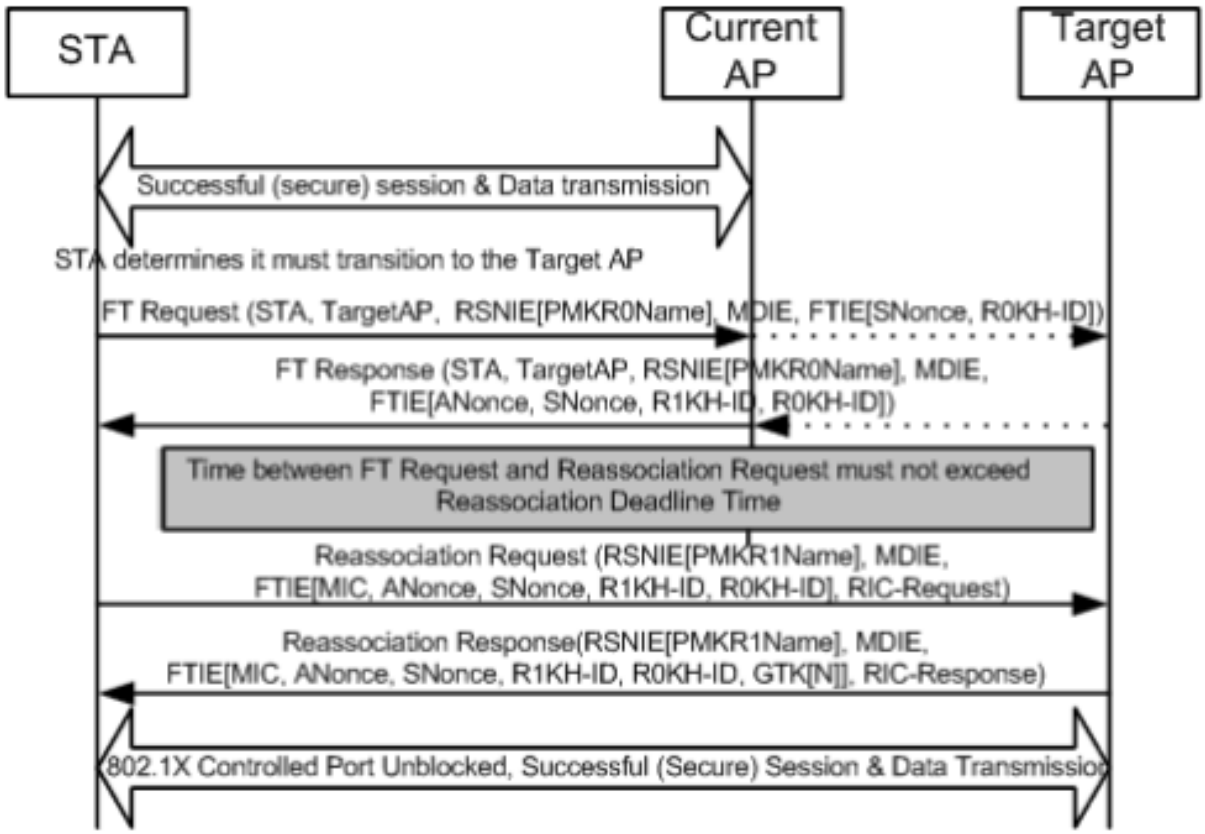
이는 일반적으로 이러한 프레임이 교환된 후 발생하는 4방향 핸드셰이크를 대체하며, 802.1X/EAP 또는 PSK를 보안 방법으로 사용하든 이러한 프레임에 대한 FT 내용 및 키 협상이 기본적으로 동일합니다. 그림에서 보듯이 AKM 필드는 주요 차이점으로서 클라이언트가 PSK나 802.1X로 FT를 수행하는지 확인한다. 따라서 이 4개의 프레임에는 일반적으로 키 협상을 위한 이러한 유형의 보안 정보가 없지만, 802.11r이 구현되어 클라이언트와 WLAN 인프라 간에 초기 연결 시 협상하는 경우 클라이언트 FT가 로밍하는 경우에만 이러한 유형의 보안 정보가 포함됩니다.

DS를 통한 빠른 BSS 전환

802.11r을 사용하면 Fast BSS Transition의 또 다른 구현이 가능합니다. 즉 클라이언트가 Over-the-DS(Distribution System)를 통해 로밍하고 Over-the-Air를 통해 로밍하지 않는 새 AP를 사용하여 클라이언트가 FT 로밍을 시작합니다. 이 경우 Open System Authentication 프레임 대신 키 협상을 시작하기 위해 FT Action 프레임이 사용됩니다.

기본적으로 클라이언트가 더 나은 AP로 로밍할 수 있다고 결정하면 로밍하기 전에 현재 연결되어 있는 원래 AP에 FT Action Request 프레임을 보냅니다. 클라이언트는 FT 로밍할 대상 AP의 BSSID(MAC 주소)를 나타냅니다. 원래 AP는 이 FT Action Request 프레임을 배포 시스템(일반적으로 유선 인프라)을 통해 대상 AP에 전달하며, 대상 AP는 FT Action Response 프레임으로 클라이언트에 응답합니다(또한 DS를 통해 전송하므로 최종적으로 클라이언트에 무선으로 전송할 수 있음). 이 FT Action 프레임 교환에 성공하면 클라이언트는 FT 로밍을 완료하고, 클라이언트는 대상 AP에 Reassociation Request(재연결 요청)를 전송하고(이번에는 OTA), 로밍 및 최종 키 파생을 확인하기 위해 새 AP로부터 Reassociation Response(재연결 응답)를 수신합니다.

요약하면, Fast BSS Transition을 협상하고 새 암호화 키를 파생하기 위한 네 개의 프레임이 있지만, 여기서는 Open System Authentication 프레임이 FT Action Request/Response 프레임으로 대체되며, 이는 배포 시스템을 통해 현재 AP와 대상 AP와 교환됩니다. 이 방법은 Cisco Wireless LAN Controller에서 모두 지원하는 보안 방법 802.1X/EAP 및 PSK에도 모두 유효합니다. 그러나 WiFi 업계의 대부분의 무선 클라이언트에서는 ODS(Over-the-DS) 전환이 지원되지 않으며 구현되지 않으므로(그리고 프레임 교환과 디버그 출력이 기본적으로 동일하므로) 이 문서에서는 예를 제공하지 않습니다. 대신 이 이미지는 Fast BSS Transition Over-the-DS를 시각화하기 위해 사용됩니다.



802.11r을 사용하는 FlexConnect

- 중앙 인증이 지원됩니다. 여기에는 로컬 및 중앙 데이터 스위칭이 포함됩니다. AP는 동일한 FlexConnect 그룹에 속해야 합니다.

- 로컬 인증은 지원되지 않습니다.
- 독립형 모드는 지원되지 않습니다.

802.11r의 장점

- 이 방법은 802.11 표준에서 IEEE에 의해 명확하게 정의된 키 계층 구조를 수정(802.11r)으로 사용하는 첫 번째 방법이므로, 이러한 FT 기술의 구현은 공급업체 간에 그리고 다른 해석 없이 더 호환됩니다.
- 802.11r은 사용자의 요구 사항에 따라 도움이 되는 여러 가지 기술을 허용합니다(802.1x/EAP 보안 및 PSK 보안용 Over-the-Air 및 Over-the-DS).
- 무선 클라이언트는 동일한 WLAN/SSID의 새 AP에 대해 빠른 보안 로밍을 수행합니다(해당 AP와 연결되지 않은 경우에도). 여러 PMKID를 저장할 필요 없습니다.
- 이는 PSK 보안을 사용해도 더 빠른 로밍이 가능한 최초의 빠른 보안 로밍 방식이며, WPA/WPA2 PSK를 사용하여 AP 간에 로밍할 때 필요한 4방향 핸드셰이크를 방지합니다. 고속 보안 로밍 방법의 주요 목적은 이 보안 방법을 구현할 때 802.1X/EAP 핸드셰이크를 방지하는 것입니다. 그러나 PSK 보안의 경우 4방향 핸드셰이크를 사용하지 않을 때 802.11r로 로밍 이벤트가 더욱 가속화됩니다.

802.11r의 단점

- Fast BSS Transitions를 실제로 지원하는 몇 가지 무선 클라이언트 장치가 있으며, 대부분의 경우 802.11r에서 사용 가능한 모든 기술을 지원하지는 않습니다.
- 이러한 구현이 매우 어려우므로 실제 프로덕션 환경의 테스트 결과가 충분하지 않거나 디버그 결과가 부족하여 나타날 수 있는 주의 사항을 해결할 수 없습니다.
- FT 방법을 사용하기 위해 WLAN/SSID를 구성하면 802.11r을 지원하는 무선 클라이언트만 이 WLAN/SSID에 연결할 수 있습니다. FT 설정은 클라이언트에 대해 선택 사항이 아니므로 802.11r을 지원하지 않는 무선 클라이언트는 FT가 전혀 구성되지 않은 별도의 WLAN/SSID에 연결해야 합니다.

적응형 802.11r

- 일부 레거시 클라이언트는 "혼합 모드"에 대해서도 802.11r이 활성화된 WLAN/SSID에 연결할 수 없습니다. 이는 802.11r을 지원하지만 지원하지 않는 동일한 SSID 클라이언트에 연결할 수 있기를 희망합니다. 이는 RSN IE(Robust Security Network Information Element)를 구문 분석하는 클라이언트 신청자의 드라이버가 오래되고 IE의 추가 AKM 제품군을 인식하지 못하는 경우입니다. 이러한 제한 때문에 클라이언트는 802.11r 지원을 알리는 WLAN에 연결 요청을 보낼 수 없으므로 802.11r 클라이언트에 대해 하나의 WLAN/SSID를 구성하고 802.11r을 지원하지 않는 클라이언트에 대해서는 별도의 WLAN/SSID를 구성해야 합니다.
- 이를 극복하기 위해 Cisco Wireless LAN Infrastructure에는 Adaptive 802.11r 기능이 도입되었습니다. FT 모드가 WLAN 레벨에서 Adaptive(적응)로 설정된 경우 WLAN은 802.11i 지원 WLAN에 802.11r 모빌리티 도메인 ID를 광고합니다. 일부 Apple iOS10 클라이언트 디바이스는 80211i/WPA2 WLAN에서 MDIE의 존재를 식별하고 802.11r 연결을 설정하기 위해 독점 핸드셰이크를 수행합니다. 클라이언트가 성공적인 802.11r 연결을 완료하면 일반적인 802.11r 지원 WLAN에서와 같이 FT 로밍을 수행할 수 있습니다. FT Adaptive는 선택된 Apple iOS10 이상 디바이스에만 적용됩니다. 다른 모든 클라이언트는 WLAN에서 계속 802.11i/WPA2 연결을 가질 수 있으며 지원되는 대로 적용 가능한 FSR 방법을 수행할 수 있습니다.
- 802.11r이 실제로 활성화되지 않은 WLAN/SSID에서 802.11r을 수행하기 위해 iOS10 디바이스

에 도입된 이 새로운 기능에 대한 추가 설명서는 Cisco Wireless LAN의 Enterprise Best Practices for Cisco IOS Devices에서 찾을 수 있습니다.

결론

- 항상 클라이언트가 특정 AP로 로밍하기로 결정하며 WLC/AP가 클라이언트에 대해 이를 결정할 수 없다는 점에 유의하십시오. 로밍 이벤트는 로밍해야 하는 것으로 간주되면 무선 클라이언트에 의해 시작됩니다.
- WLC는 동일한 WLAN/SSID에서 FSR(Fast-Secure Roaming) 방법의 대부분 또는 전체를 함께 지원합니다. 그러나 WLC가 지원되는 대로 광고를 시도한다는 것을 지원하거나 이해하기 위해 클라이언트 동작(여러 모바일 장치에서 매우 다름)에 크게 의존하기 때문에 이 기능이 정상적으로 작동하지 않습니다. SSID 하나만으로 상호운용성을 구현하는 대신, 일반적으로 수정될 것으로 예상되는 문제보다 더 많은 문제가 있으므로 이 방법은 권장되지 않습니다. 이 WLAN에서 사용할 수 있는 모든 클라이언트와 함께 심층 테스트를 수행해야 합니다. 이 작업이 정말 필요한 경우 완료해야 합니다.
- WLAN/SSID에 보안이 활성화된 경우 AP 간에 이동할 때 WLAN 로밍 프로세스를 가속화하기 위해 빠른 보안 로밍 방법이 개발된다는 사실을 이해하는 것이 매우 중요합니다. 보안이 적용되지 않을 경우, 데이터 프레임이 전송되기 전에 클라이언트-AP가 AP 간에 로밍할 때 항상 필요한 무선 관리 프레임을 교환하므로 가속화할 필요가 없습니다(클라이언트에서 시스템 인증 열기, AP에서 시스템 인증 열기, 재연결 요청 및 재연결 응답). 따라서 이 작업은 더 빠르게 이동할 수 없습니다. 보안 없이 로밍 문제가 발생하면 로밍을 개선하기 위한 빠른 로밍 방법은 없으며, WLAN/SSID 설정 및 설계가 무선 클라이언트 스테이션에서 AP 커버리지 셀 간에 적절하게 로밍하는 데 적합한지 확인하기 위한 방법만 있습니다.
- 802.11r 섹션에 설명된 대로 이 보안으로 로밍 이벤트를 가속화하고 4방향 핸드셰이크를 방지하기 위해 802.11r/FT는 WPA2-PSK로 구현됩니다.
- 모든 방법에는 장단점이 있지만, 결국 무선 클라이언트 스테이션이 구현하려는 특정 방법을 지원하는지, 그리고 Cisco WLAN 인프라가 사용 가능한 모든 방법을 지원하는지 항상 확인해야 합니다. 따라서 특정 WLAN/SSID에 연결하는 무선 클라이언트가 실제로 지원하는 최상의 방법을 선택해야 합니다. 예를 들어, 일부 구축에서는 Cisco 무선 IP Phone용 CCKM이 포함된 WLAN/SSID를 생성한 다음(CCKM은 WPA2/AES를 지원하지만 802.11r은 지원 안 함), 802.11r/FT를 통해 WPA2/AES가 포함된 또 다른 WLAN/SSID를 생성하여 이 Fast Secure Roaming 방법을 지원하는 무선 클라이언트를 지원할 수 있습니다(지원되는 경우 OKC 사용).
- 무선 클라이언트가 사용 가능한 빠른 보안 로밍 방법을 지원하지 않는 경우, 802.1X/EAP 보안을 사용하는 WLAN/SSID의 AP 간에 로밍할 때 해당 클라이언트가 항상 이 문서에 설명된 지연을 시험할 수 있다는 사실을 수락해야 합니다(클라이언트 앱/서비스에 중단을 초래할 수 있음).
- SKC(WPA2 PMKID 캐싱)를 제외한 모든 방법은 서로 다른 WLC에 의해 관리되는 AP 간의 빠른 보안 로밍(컨트롤러 간 로밍)을 지원합니다. 단, 동일한 모빌리티 그룹에 있어야 합니다.
- 802.1X/EAP 인증이 WPA/WPA2에 사용되는 경우 CUWN은 이 문서에서 설명하는 모든 다른 빠른 보안 로밍 방법을 완벽하게 지원합니다. PSK(WPA2-Personal)를 사용하는 경우 CUWN은 WPA2-RSN(CCKM, PMKID 캐싱/SKC, OKC/PKC)과 함께 작동하는 메서드에서 Fast-Secure 로밍을 지원하지 않습니다. 이 경우 Fast-Roaming 메서드는 대부분 필요하지 않습니다. 그러나 PSK를 사용하는 WPA2-FT(802.11r)의 경우 이 문서에서 설명한 대로 CUWN은 Fast-Secure 로밍을 지원합니다.

관련 정보

- 802.11r BSS 빠른 전환 구축 설명서
- Cisco 기술 지원 및 다운로드

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.