Controleer methoden voor 802.11 WLAN en Fast-Secure roaming op CUWN

Inhoud

Inleiding

Dit document beschrijft draadloze en snel beveiligde roamingtypen die beschikbaar zijn voor IEEE 802.11 Wireless LAN's (WLAN's) op Unified Wireless Network (CUWN).

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- IEEE 802.11 WLAN-fundamenten
- IEEE 802.1 WLAN-beveiliging
- Basisfuncties IEEE 802.1X/EAP

Gebruikte componenten

De informatie in dit document is gebaseerd op Software voor Cisco WLAN-controller, versie 7.4.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk live is, moet u zorgen dat u de potentiële impact van elke opdracht begrijpt.

Achtergrondinformatie

De informatie in dit document is gebaseerd op de softwareversie 7.4 van Cisco WLAN-controllers, maar de meeste van de beschreven debug-uitgangen en -gedrag kunnen van toepassing zijn op elke softwareversie die de besproken methoden ondersteunt. De specificaties van alle methoden die hier worden uitgelegd, blijven hetzelfde bij latere codes van Cisco WLAN-controllers (tot versie 8.3 tegen de tijd dat dit artikel is bijgewerkt).

Dit document beschrijft de verschillende typen draadloze roaming- en snel beveiligde roamingmethoden die beschikbaar zijn voor IEEE 802.11 draadloze LAN's (WLAN's) die worden ondersteund op het Cisco Unified Wireless Network (CUWN).

Het document bevat niet alle details over de manier waarop elke methode werkt of hoe deze zijn geconfigureerd. Het belangrijkste doel van dit document is om de verschillen te beschrijven tussen de verschillende beschikbare technieken, hun voordelen en beperkingen, en de frames-uitwisseling op elke methode. Er worden voorbeelden van WLAN-controllers (WLC) geboden en draadloze pakketafbeeldingen worden gebruikt om de gebeurtenissen te analyseren en uit te leggen die voor elke beschreven zwervende methode optreden.

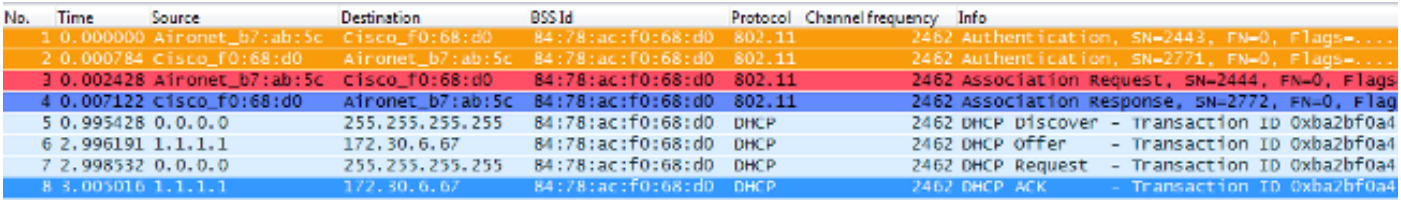
Alvorens een beschrijving van de verschillende snel-veilige het zwerven methodes beschikbaar voor WLANs wordt gegeven, is het belangrijk om te begrijpen hoe het WLAN associatieproces werkt, en hoe een regelmatige het zwerven gebeurtenis voorkomt wanneer er geen veiligheid die op het Vastgestelde Herkenningsteken van de Dienst (SSID) wordt gevormd is.

Wanneer een 802.11 draadloze client verbinding maakt met een access point (AP), moet deze eerst het 802.11 Open System-verificatieproces doorstaan voordat er verkeer (draadloze datakaders) wordt doorgegeven. Vervolgens moet het associatieproces worden voltooid. Het verificatieproces voor het Open System is vergelijkbaar met een kabelverbinding op het toegangspunt dat door de client wordt geselecteerd. Dit is een heel belangrijk punt, omdat het altijd de draadloze client is die selecteert welke AP wordt geprefereerd, en de beslissing baseert op meerdere factoren die variëren tussen leveranciers. Daarom begint de client met dit proces

door het verificatiekader naar het geselecteerde toegangspunt te verzenden, zoals later in dit document wordt getoond. Het toegangspunt kan niet vragen of u een verbinding wilt maken.

Zodra het Open System-verificatieproces met succes is voltooid met een respons van het AP ("met kabel verbonden"), voltooit het associatieproces in wezen de 802.11 Layer 2 (L2)-onderhandeling die de koppeling tussen de client en het AP tot stand brengt. AP wijst een vereniging-ID toe aan de client als de verbinding succesvol is, en bereidt deze voor om verkeer over te gaan of om een beveiligingsmethode op een hoger niveau uit te voeren indien geconfigureerd op de SSID. Het verificatieproces van het Open System bestaat uit twee beheerframes en het associatieproces. Verificatie- en associatiekaders zijn draadloze **beheerframes**, geen gegevenskaders, die in principe de frames zijn die worden gebruikt voor het verbindingsproces met het AP.

Hier is een afbeelding van de draadloze frames via de lucht voor dit proces:



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

> **Opmerking**: als u meer wilt weten over 802.11 draadloze snuffelen en over de filters/kleuren die op Wireshark worden gebruikt voor de afbeeldingen die in dit document worden weergegeven, gaat u naar de Cisco Support Community-post met de naam 802.11 Sniffer image Analysis.

De draadloze client begint met het verificatiekader en de AP antwoordt met een ander verificatiekader. De client verstuurt vervolgens het Associatie-verzoekframe en de AP eindigt in een antwoord met het Associatie-responsframe. Zoals wordt getoond in de DHCP-pakketten, begint de client, zodra de 802.11 Open System-verificatie- en associatieprocessen zijn doorgegeven, gegevenskaders te verzenden. In dit geval is er geen beveiligingsmethode geconfigureerd op de SSID, zodat de client onmiddellijk begint met het verzenden van gegevenskaders (in dit geval DHCP) die niet zijn versleuteld.

Zoals later in dit document wordt getoond, als de beveiliging op de SSID is ingeschakeld, zijn er hogere handshake-frames voor verificatie en codering voor de specifieke beveiligingsmethode, net nadat de Association Response en voorafgaand aan het verzenden van datakaders van clientverkeer zijn verzonden, zoals DHCP, Address Resolution Protocol (ARP) en applicatiepakketten, die zijn versleuteld. De kaders van gegevens kunnen slechts worden verzonden tot de cliënt volledig voor authentiek wordt verklaard, en de encryptiesleutels worden besproken, gebaseerd op de gevormde veiligheidsmethode.

Gebaseerd op het vorige beeld, zijn hier de berichten die u in de output van WLC ziet **debug cliëntbevel** wanneer de draadloze cliënt met een nieuwe vereniging aan WLAN begint:

```
*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.
```

> **Opmerking**: de WLC debug gebruikt voor de outputs in dit document is de **debug client** opdracht, en de voorbeelden tonen alleen enkele relevante berichten, niet de gehele output. Voor meer informatie over deze debug-opdracht raadpleegt u het document Understand the Debug Client on Wireless LAN Controllers (WLC's).

Deze berichten tonen de Associatie Vraag- en Reactieframes; de eerste Verificatieframes worden niet vastgelegd bij de WLC omdat deze handdruk snel gebeurt op het AP-niveau op de CUWN.

Welke informatie verschijnt wanneer de klant zwerft? De klant ruilt altijd vier beheerframes bij het instellen van een verbinding met een AP, die te wijten is aan hetzij de vestiging van de klant van vereniging, of een roaming-gebeurtenis. De client heeft slechts één verbinding met slechts één toegangspunt tegelijk. Het enige verschil in de kaderuitwisseling tussen een nieuwe verbinding aan de WLAN-infrastructuur en een zwervende gebeurtenis is dat de associatieframes van een zwervende gebeurtenis **Reassociation-**frames worden genoemd, wat aangeeft dat de client eigenlijk zwerft van een andere AP zonder pogingen om een nieuwe associatie met het WLAN te creëren. Deze frames kunnen verschillende elementen bevatten die worden gebruikt om te onderhandelen over de roaminggebeurtenis; dit hangt af van de installatie, maar die details vallen buiten het bereik van dit document.

Hier is een voorbeeld van de frameuitwisseling:

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

Deze berichten verschijnen in de debug uitvoer:

```
*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.
```

Zoals getoond, voert de client met succes een zwervende gebeurtenis uit nadat het Reassociation-verzoek naar het nieuwe toegangspunt is verzonden, en ontvangt de Reassociation Response van het toegangspunt. Aangezien de client al een IP-adres heeft, zijn de eerste gegevenskaders bestemd voor ARP-pakketten.

Als u een roamende gebeurtenis verwacht, maar de client stuurt een associatie-verzoek in plaats van een reassociatie-verzoek (dat u kunt bevestigen op basis van sommige afbeeldingen en debugs zoals eerder uitgelegd in dit document), dan zwerft de client niet echt. De client begint een nieuwe verbinding met het WLAN alsof er een verbinding wordt verbroken en probeert opnieuw

verbinding te maken vanuit het niets. Dit kan om meerdere redenen gebeuren, zoals wanneer een client zich van de dekkingsgebieden verwijdert en vervolgens een AP vindt met voldoende signaalkwaliteit om een associatie te starten, maar het duidt normaal gesproken op een clientprobleem waarbij de client geen roamende gebeurtenis start vanwege stuurprogramma's, firmware of software problemen.

> **Opmerking**: u kunt contact opnemen met de verkoper van de draadloze client om de oorzaak van het probleem vast te stellen.

Zwerven met security op hoger niveau

Wanneer de SSID is geconfigureerd met L2 hogere beveiligingsniveau bovenop basis 802.11 Open System-verificatie, zijn meer frames nodig voor de eerste associatie en bij roaming. De twee meest gebruikelijke beveiligingsmethoden die zijn gestandaardiseerd en geïmplementeerd voor 802.11 WLAN's worden in dit document beschreven:

- **WPA/WPA2-PSK (Pre-Shared Key)** - verificatie van clients met een Preshared-Key.
- **WPA/WPA2-EAP (Extensible Verification Protocol)** - verificatie van clients met een 802.1X/EAP-methode om veiligere referenties te valideren met behulp van een verificatieserver, zoals certificaten, gebruikersnaam en wachtwoord en tokens.

Het is belangrijk om te weten dat, hoewel deze twee methoden (PSK en EAP) de clients op verschillende manieren authenticeren/valideren, beide in wezen dezelfde WPA/WPA2-regels gebruiken voor het sleutelbeheerproces. Of de beveiliging nu WPA/WPA2-PSK of WPA/WPA2-EAP is, het proces dat bekend staat als de WPA/WPA2 4-Way-handdruk begint met de belangrijkste onderhandeling tussen de WLC/AP en de client met een Master Session Key (MSK) als het oorspronkelijke sleutelmateriaal zodra de client is gevalideerd met de specifieke gebruikte verificatiemethode.

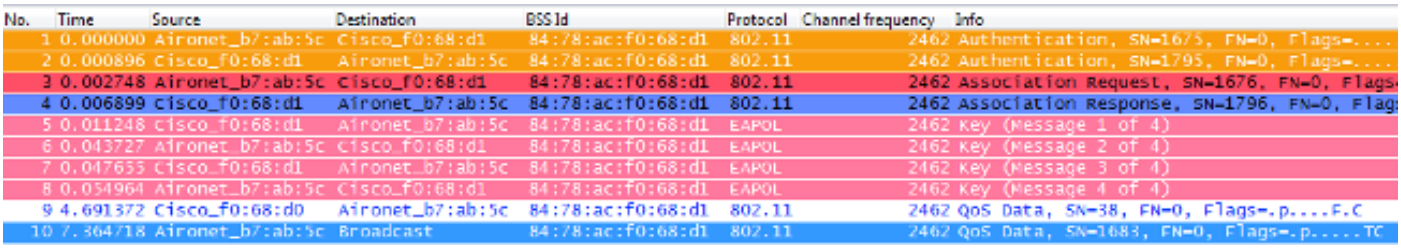
Hier volgt een samenvatting van het proces:

1. Een MSK wordt afgeleid van de EAP-verificatiefase wanneer 802.1X/EAP-beveiliging wordt gebruikt, of van de PSK wanneer WPA/WPA2-PSK als beveiligingsmethode wordt gebruikt.
2. Uit deze MSK, de client en WLC/AP halen de Pairwise Master Key (PMK) af, en de WLC/AP genereert een Group Master Key (GMK).
3. Zodra deze twee Master Keys klaar zijn, starten de client en de WLC/AP de WPA/WPA2 4-Way handshake (die later in dit document wordt geïllustreerd met een aantal schermafbeeldingen en debugs) met de Master Keys als de zaadjes voor onderhandeling van de daadwerkelijke encryptie sleutels.
4. Deze laatste coderingssleutels staan bekend als de Pairwise Transient Key (PTK) en de Group Transient Key (GTK). De PTK is afgeleid van de PMK en wordt gebruikt om unicastframes te versleutelen met de client. De Group Transient Key (GTK) is afgeleid van de GMK en wordt gebruikt om multicast/broadcast op deze specifieke SSID/AP te versleutelen.

WPA/WPA2-PSK

Wanneer WPA-PSK of WPA2-PSK wordt uitgevoerd via TKIP (Temporal Key Integrity Protocol) of Advanced Encryption Standard (AES) voor de codering, moet de client het proces doorlopen dat

bekend staat als de WPA 4-Way handdruk voor zowel de eerste associatie als tijdens het zwerven. Zoals eerder uitgelegd, is dit in principe het sleutelbeheerproces dat wordt gebruikt om WPA/WPA2 de coderingssleutels te laten afleiden. Wanneer echter PSK wordt uitgevoerd, wordt deze ook gebruikt om te verifiëren dat de client een geldige vooraf gedeelde sleutel heeft om zich aan te sluiten bij het WLAN. Dit beeld toont het eerste associatieproces wanneer WPA of WPA2 met PSK wordt uitgevoerd:



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

Zoals getoond, na het 802.11 Open System authenticatie en associatieproces, zijn er vier EAPOL frames van de WPA 4-Way handshake, die worden geïnitieerd door de AP met **bericht-1**, en geëindigd door de client met **bericht-4**. Na een succesvolle handdruk, begint de client gegevenskaders (zoals DHCP) door te geven, die in dit geval zijn versleuteld met de toetsen die zijn afgeleid van de 4-voudige handdruk (daarom kunt u de werkelijke inhoud en het type verkeer niet zien van de draadloze afbeeldingen).

> **Opmerking**: EAPOL-frames worden gebruikt om alle belangrijke beheerframes en 802.1X/EAP-verificatiekaders via de ether tussen het toegangspunt en de client te transporteren; ze worden verzonden als draadloze gegevenskaders.

Deze berichten verschijnen in de debug-uitgangen:

```
*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.
```

Tijdens het zwerven, volgt de client in principe dezelfde frameruiling, waar de WPA 4-Way handshake nodig is om nieuwe coderingssleutels af te leiden met de nieuwe AP. Dit komt door veiligheidsredenen die in de standaard zijn vastgelegd en door het feit dat het nieuwe toegangspunt de oorspronkelijke sleutels niet kent. Het enige verschil is dat er Reassociation-frames zijn in plaats van Association-frames, zoals in deze afbeelding wordt getoond:

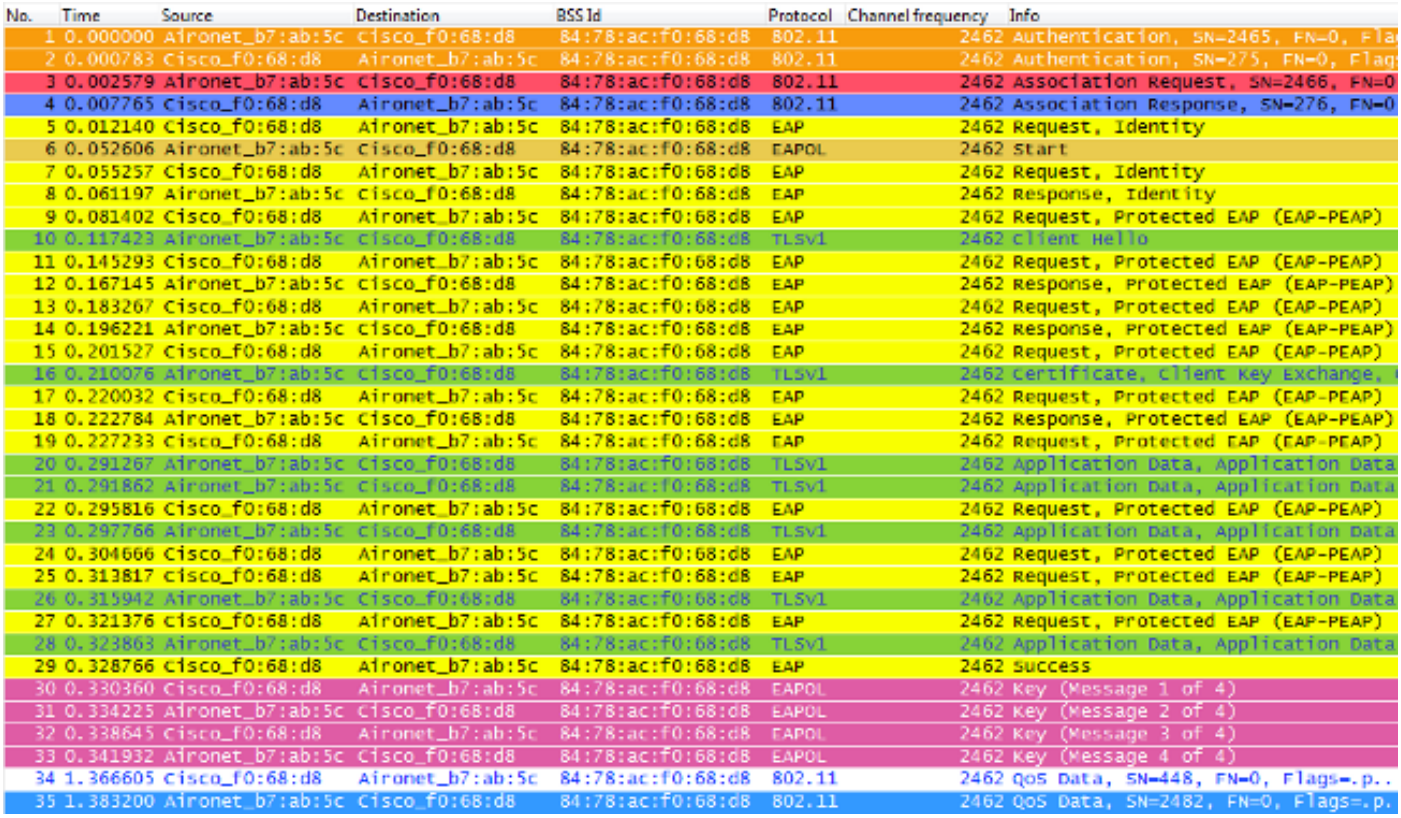
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3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

U ziet dezelfde berichten in de debug-uitgangen, maar het eerste pakket van de client is een Reassociation in plaats van een Association, zoals eerder getoond en uitgelegd.

WPA/WPA2-EAP

Wanneer een 802.1X/EAP-methode wordt gebruikt voor de verificatie van de clients op een beveiligde SSID, zijn er nog meer frames nodig voordat de client verkeer begint te passeren. Deze extra frames worden gebruikt om de aanmeldingsgegevens voor de client te verifiëren. Afhankelijk van de EAP-methode kunnen er tussen vier en twintig frames zijn. Deze komen na de Association/Reassociation, maar vóór de WPA/WPA2 4-Way handshake, omdat de authenticatiefase de MSK afleidt die als zaad voor de definitieve encryptie sleutelgeneratie in het sleutelbeheerproces wordt gebruikt (4-Way handdruk).

Deze afbeelding toont een voorbeeld van de frames die via de ether tussen het toegangspunt en de draadloze client zijn uitgewisseld bij de eerste koppeling wanneer WPA met PEAPv0/EAP-MSCHAPv2 wordt uitgevoerd:



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.

Soms toont deze uitwisseling min of meer frames, die afhankelijk zijn van meerdere factoren, zoals de EAP-methode, heruitzendingen door problemen, cliëntgedrag (zoals de twee Identity Requirements in dit voorbeeld, omdat de client een **EAPOL START** verstuurt nadat de AP de eerste Identity Query verstuurt), of als de client het certificaat al met de server heeft uitgewisseld. Wanneer de SSID is geconfigureerd voor een 802.1X/EAP-methode, zijn er meer frames (voor de verificatie) en is er dus meer tijd nodig voordat de client begint met het verzenden van gegevenskaders.

Hier is een samenvatting van de debug-berichten:

```
*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.
```

Wanneer de draadloze client hier regelmatig zwerft (het normale gedrag, zonder implementatie van een snel beveiligde zwervende methode), moet de client precies hetzelfde proces doorlopen en een volledige verificatie uitvoeren tegen de verificatieserver, zoals in de afbeeldingen wordt getoond. Het enige verschil is dat de client een Reassociation-verzoek gebruikt om het nieuwe AP te informeren dat het eigenlijk zwerft van een andere AP, maar de client moet nog steeds door volledige validatie en nieuwe sleutelgeneratie:

No.	Time	Source	Destination	BSSId	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=2037, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=96, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2038, FN=0, Flags=...
4	0.008646	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Reassociation Response, SN=97, FN=0, Flags=...
5	0.014409	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
6	0.029712	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Start
7	0.033084	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
8	0.053240	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAP	2437	Response, Identity
9	0.062770	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Protected EAP (EAP-PEAP)
10	0.065313	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Client Hello
11	0.071292	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Server Hello, Change Cipher Spec, Encrypted Han
12	0.077740	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Change Cipher Spec, Encrypted Handshake Message
13	0.083816	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	TLSv1	2437	Application Data
14	0.092138	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Success
15	0.093699	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 1 of 4)
16	0.097014	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 2 of 4)
17	0.100739	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 3 of 4)
18	0.105180	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Key (Message 4 of 4)
19	1.125063	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=76, FN=0, Flags=.p....F.C
20	4.383568	Aironet_b7:ab:5c	Broadcast	84:78:ac:f0:2a:98	802.11	2437	QoS Data, SN=2647, FN=0, Flags=.p.....TC

Zoals aangetoond, zelfs wanneer er minder frames zijn dan bij de eerste verificatie (die wordt veroorzaakt door meerdere factoren, zoals eerder vermeld), moeten, wanneer de client naar een nieuwe AP zwerft, de EAP-verificatie en de WPA-sleutelbeheerprocessen nog worden voltooid om door te kunnen gaan met het doorgeven van datakaders (zelfs als er actief verkeer werd verzonden voor het zwerven). Daarom, als de client een actieve toepassing heeft die gevoelig is voor vertragingen (zoals spraak-verkeer applicaties, of applicaties die gevoelig zijn voor onderbrekingen), dan kan de gebruiker problemen waarnemen bij het zwerven, zoals audio gaps of applicaties verbreken. Dit is afhankelijk van hoe lang het proces duurt voordat de client doorgaat met het verzenden/ontvangen van datakaders. Deze vertraging kan langer zijn, afhankelijk van: de RF-omgeving, de hoeveelheid clients, de round-trip tijd tussen de WLC en LAP's en met de verificatieserver, en andere redenen.

Hier is een samenvatting van de debug berichten voor deze roaming gebeurtenis (in principe hetzelfde als de vorige, dus deze berichten worden niet verder beschreven):

```
*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
```

Dit is de manier waarop 802.1X/EAP en het WPA/WPA2-beveiligingsframework werken. Om te voorkomen dat de toepassing/service-impact op vertragingen van een reguliere roaminggebeurtenis, worden meerdere snel beveiligde roamingmethoden ontwikkeld en geïmplementeerd door de WiFi-industrie om het roamingproces te versnellen wanneer de beveiliging wordt gebruikt op de WLAN/SSID. De clients hebben te maken met enige latentie wanneer ze tijdens het zwerven tussen AP's via de implementatie van high-level security op het WLAN verkeer blijven doorvoeren. Dit komt door de EAP-verificatie en de uitwisseling van sleutelbeheerframes die vereist zijn voor de beveiligingsinstellingen, zoals eerder is uitgelegd.

Het is belangrijk om te begrijpen dat snel beveiligde roaming slechts de term is die door de industrie wordt gebruikt in verwijzing naar de implementatie van een methode/schema die het roaming proces versnelt wanneer de beveiliging is geconfigureerd op het WLAN. De verschillende snelbeveiligde roamingmethoden/-schema's die beschikbaar zijn voor WLAN's en worden ondersteund door de CUWN, worden in de volgende sectie uitgelegd.

Snel beveiligde roaming met CCKM

Cisco Centralized Key Management (CCKM) is de eerste snel beveiligde roamingmethode die is ontwikkeld en geïmplementeerd op WLAN's van ondernemingen. Deze methode is door Cisco gecreëerd als de oplossing die wordt gebruikt om de tot nu toe verklaarde vertragingen te beperken, wanneer 802.1X/EAP-beveiliging wordt gebruikt op het WLAN. Aangezien dit een merkgebonden protocol van Cisco is, wordt het alleen ondersteund door Cisco WLAN-infrastructuurapparaten en draadloze clients (van meerdere leveranciers) die Cisco Compatible Extension (CCX)-compatibel zijn voor CCKM.

CCKM kan worden geïmplementeerd met alle verschillende coderingsmethoden die beschikbaar zijn voor WLAN's, zoals WEP, TKIP en AES. Het wordt ook ondersteund met de meeste

802.1X/EAP-verificatiemethoden die worden gebruikt voor WLAN's, afhankelijk van de CCX-versie die door de apparaten wordt ondersteund.

> **Opmerking**: voor een overzicht van de functieinhoud die wordt ondersteund door de verschillende versies van de CCX-specificatie (die EAP-methoden omvat die worden ondersteund), verwijzen we naar het document CCX versies en functies en controleren we de exacte CCX-versie die wordt ondersteund door uw draadloze clients (indien deze CCX-compatibel zijn), zodat u kunt bevestigen of de beveiligingsmethode die u met CCKM wilt gebruiken, kan worden geïmplementeerd.

Dit draadloze beeld geeft een voorbeeld van de frames die bij de eerste koppeling zijn uitgewisseld wanneer u CCKM uitvoert met TKIP als codering en PEAPv0/EAP-MSCHAPv2 als de 802.1X/EAP-methode. Dit is in principe dezelfde uitwisseling als wanneer WPA/TKIP met PEAPv0/EAP-MSCHAPv2 wordt uitgevoerd, maar deze keer wordt CCKM tussen de client en de infrastructuur besproken, zodat ze verschillende sleutelhiërarchie en cachemethoden gebruiken om Fast Secure Roaming uit te voeren wanneer de client moet zwerven:

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

Hier volgt een samenvatting van de debug-berichten (sommige EAP-uitwisselingen zijn verwijderd om de uitvoer te beperken):

```
*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.

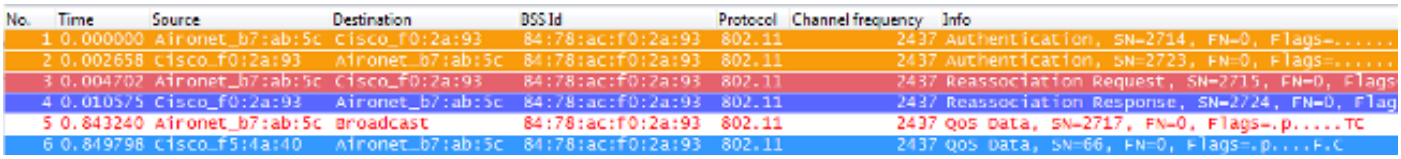
```
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
!--- This is the key cache index for this client, which is set temporally.

*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.
```

Met CCKM is de aanvankelijke koppeling aan het WLAN vergelijkbaar met de reguliere WPA/WPA2, waarbij een MSK (ook bekend als de Network Session Key (NSK)) wederzijds wordt afgeleid met de client en de RADIUS-server. Deze primaire sleutel wordt verzonden van de server naar de WLC na een succesvolle authenticatie, en wordt gecached als basis voor afleiding van alle volgende sleutels voor het leven van de client associatie met dit WLAN. Van hieruit halen de WLC en de client de zaadinformatie af die wordt gebruikt voor snel beveiligde roaming op basis van CCKM, dit gaat door een 4-weg handdruk vergelijkbaar met die van WPA/WPA2, om de unicast (PTK) en multicast/broadcast (GTK) encryptie sleutels af te leiden met de eerste AP.

Het grote verschil wordt opgemerkt bij roaming. In dit geval stuurt de CCKM-client één

Reassociation-aanvraagframe naar de AP/WLC (dat een MIC en een sequentieel stijgend willekeurig nummer omvat), en biedt voldoende informatie (dat het nieuwe AP MAC-adres **- BSSID-** omvat) om de nieuwe PTK af te leiden. Met dit Reassociation-verzoek hebben de WLC en de nieuwe AP ook genoeg informatie om de nieuwe PTK af te leiden, dus ze reageren gewoon met een Reassociation Response. De client kan nu doorgaan met het doorgeven van verkeer, zoals in deze afbeelding wordt getoond:



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

Hier is een samenvatting van de WLC debugs voor dit roaming evenement:

```
*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.
```

Zoals getoond, wordt snel-beveiligde roaming uitgevoerd terwijl de EAP-verificatieframes worden vermeden en nog meer 4-way handshakes, omdat de nieuwe encryptie sleutels nog steeds worden afgeleid, maar gebaseerd op het CCKM-onderhandelingsschema. Dit wordt voltooid met de zwervende Reassociation frames en de informatie die eerder door de client en de WLC is gecached.

FlexConnect met CCKM

- Centrale verificatie wordt ondersteund. Dit omvat Local and Central data switching. De AP's moeten deel uitmaken van dezelfde FlexConnect Group.
- Flex Lokale verificatie wordt ondersteund. In de verbonden modus kan het cachegeheugen worden gedistribueerd van het toegangspunt naar de controller en vervolgens naar de rest van de toegangspunten in de FlexConnect-groep.
- Standalone modus wordt ondersteund. Als de cache al aanwezig is op het toegangspunt (vanwege eerdere distributie), zal snel roamen werken. Nieuwe verificatie in standalone modus ondersteunt geen snel beveiligd zwerven.

Voordelen met CCKM

- CCKM is de snelst beveiligde roamingmethode die meestal wordt geïmplementeerd op WLAN's van ondernemingen. Clients hoeven niet over een sleutelbeheerhanddruk te gaan om nieuwe sleutels af te leiden wanneer er een verplaatsing tussen AP's plaatsvindt, en hoeven tijdens het clientleven op dit WLAN nooit meer een volledige 802.1X/EAP-verificatie met nieuwe AP's uit te voeren.
- CCKM ondersteunt alle encryptiemethoden die beschikbaar zijn binnen de 802.11-standaard (WEP, TKIP en AES), naast een aantal oudere bedrijfseigen methoden van Cisco die nog steeds worden gebruikt op oudere clients.

Nadelen met CCKM

- CCKM is een bedrijfseigen methode van Cisco die de implementatie en ondersteuning beperkt tot Cisco WLAN-infrastructuur en draadloze CCX-clients.
- CCX versie 5 wordt niet breed geaccepteerd, dus CCKM met WPA2/AES wordt niet ondersteund door veel draadloze CCX-clients (vooral omdat de meeste CCKM al

ondersteunen met WPA/TKIP, dat nog steeds zeer veilig is).

Fast-Secure roaming met PMKID-caching / tijdelijke toetscaching

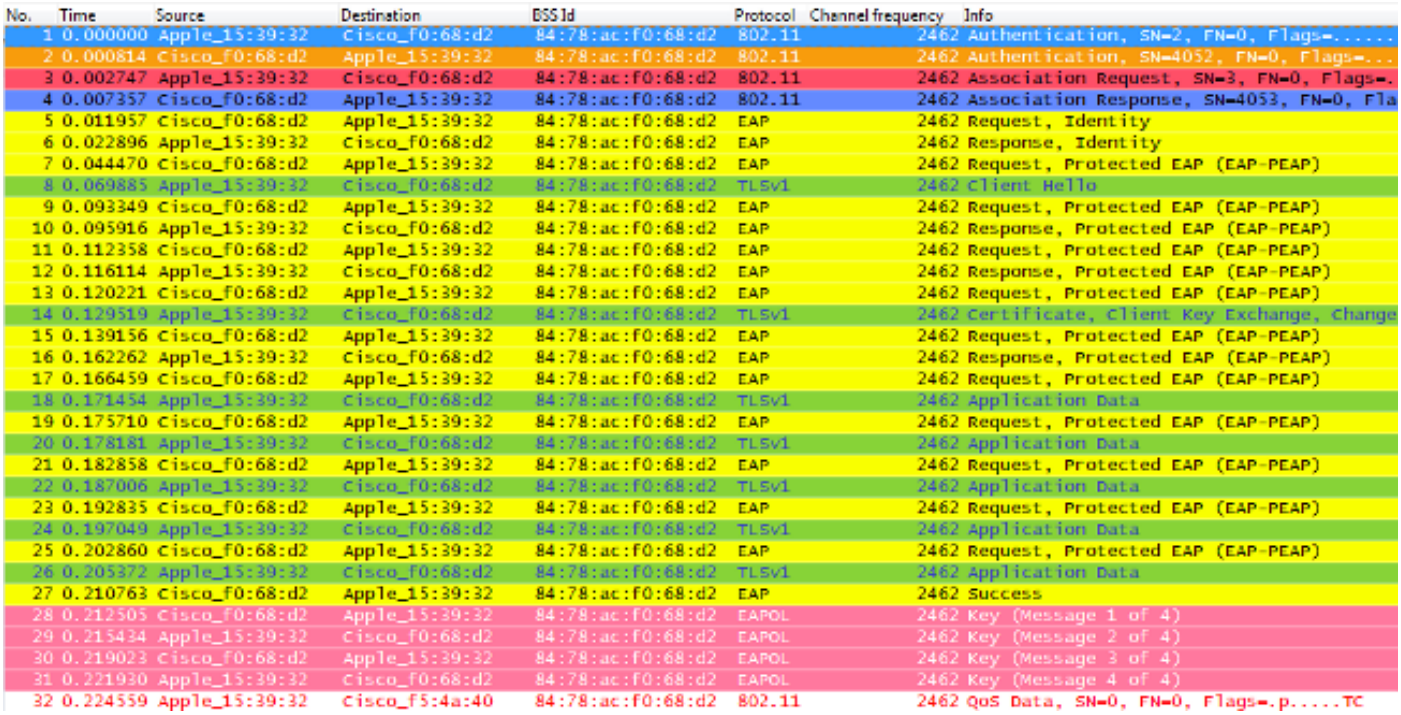
Pairwise think Key ID (PMKID) caching, of **Sticky Key Caching (SKC)**, is de eerste snel beveiligde roamingmethode die wordt voorgesteld door de IEEE 802.11-standaard binnen het 802.11i-beveiligingsamendement, waar het belangrijkste doel is om een hoog beveiligingsniveau voor WLAN's te standaardiseren. Deze snelle beveiligde zwervende techniek werd toegevoegd als een optionele methode voor WPA2-apparaten om zwerven te verbeteren wanneer deze beveiliging is geïmplementeerd.

Dit is mogelijk omdat, elke keer dat een client volledig EAP-geauthenticeerd is, de client en de verificatieserver een MSK afleiden, die wordt gebruikt om de PMK af te leiden. Dit wordt gebruikt als het zaad voor de WPA2 4-Way handshake om de definitieve unicast encryptie sleutel (PTK) af te leiden die wordt gebruikt voor de sessie (tot de client naar een andere AP zwerft of de sessie verloopt); vandaar, voorkomt deze methode de EAP-verificatiefase bij zwerven omdat het de oorspronkelijke PMK gebruikt die door de client en de AP wordt gecachet. De client hoeft alleen de WPA2 4-Way handshake te doorlopen om nieuwe coderingssleutels af te leiden.

Deze methode wordt niet op grote schaal toegepast als de aanbevolen 802.11-standaardmethode voor snel beveiligde roaming, voornamelijk om de volgende redenen:

- Deze methode is optioneel en wordt niet ondersteund door alle WPA2-apparaten, omdat het doel van het 802.11i-amendement niet betrekking heeft op snel beveiligde roaming en het IEEE al aan een ander amendement heeft gewerkt om snel beveiligde roaming voor WLAN's te standaardiseren (802.11r, dat later in dit document wordt besproken).
- Deze methode heeft een grote beperking op de implementatie: Draadloze klanten kunnen alleen snel beveiligde roaming uitvoeren wanneer ze terugzwerven naar een AP waar ze eerder geauthenticeerd/verbonden waren.

Met deze methode is de eerste koppeling naar elke AP net als een gewone verificatie van de eerste keer naar het WLAN, waar de gehele 802.1X/EAP-verificatie tegen de verificatieserver en de 4-voudige handdruk voor de sleutelgeneratie moeten gebeuren voordat de client in staat is om gegevenskaders te verzenden, zoals in dit schermbeeld wordt getoond:



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

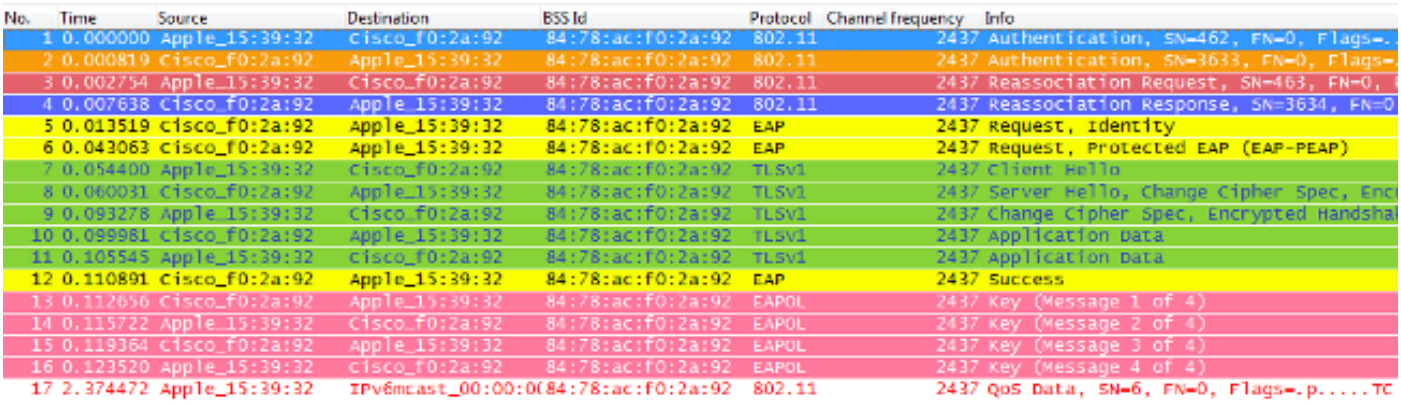
De debugs onthullen dezelfde EAP authenticatie frame uitwisseling als de rest van de methoden op de eerste verificatie aan het WLAN, met enkele uitgangen toegevoegd met betrekking tot de belangrijkste caching technieken hier gebruikt. Deze debug-uitgangen worden afgekapt om voornamelijk de nieuwe informatie te tonen, niet de gehele EAP-frameuitwisseling, omdat in principe dezelfde informatie elke keer wordt uitgewisseld voor verificatie van de client tegen de verificatieserver. Dit is tot nu toe aangetoond en is gecorreleerd met de EAP-verificatiekaders in de pakketafbeeldingen. De meeste EAP-berichten worden dus voor de eenvoud uit de debug-uitgangen verwijderd:

```
*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
```

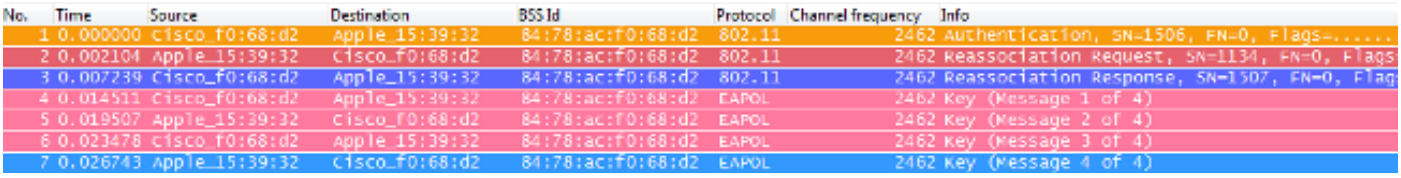
```
*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from mobile
  ec:85:2f:15:39:32
!--- Message-2 of the WPA/WPA-2 4-Way handshake is successfully
      received from the client.

*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.
```

Met deze methode, de AP en draadloze client cache de PMK's van de beveiligde associaties die al zijn ingesteld. Als de draadloze client daarom zwerft naar een nieuw toegangspunt waar het nog nooit is gekoppeld, moet de client opnieuw een volledige EAP-verificatie uitvoeren, zoals wordt getoond in dit beeld waar de client zwerft naar een nieuw toegangspunt:



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

Echter, als de draadloze client terugzwerft naar een AP waar een eerdere associatie/authenticatie heeft plaatsgevonden, dan stuurt de client een Reassociation request frame dat meerdere PMKID's opsomt, die de AP informeert over de PMK's die gecachet zijn vanaf alle AP's waar de client eerder geverifieerd heeft. Aangezien de client terugzwerft naar een toegangspunt dat ook een PMK heeft gecachet voor deze client, hoeft de client daarom niet opnieuw te worden geverifieerd via EAP om een nieuwe PMK af te leiden. De client gaat simpelweg door de WPA2 4-Way handshake om de nieuwe tijdelijke coderingssleutels af te leiden:



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)

Opmerking: Deze afbeelding toont het eerste 802.11 Open System-verificatiekader van de client niet, maar dit is niet te wijten aan de geïmplementeerde methode, omdat dit frame altijd vereist is. De reden hiervoor is dat dit specifieke frame niet wordt weergegeven door de adapter of de software voor het draadloze pakketbeeld die wordt gebruikt om de frames in de lucht bij dit voorbeeld op te sporen, maar dat het als dit in het voorbeeld blijft staan voor educatieve doeleinden. Houd er rekening mee dat dit kan gebeuren wanneer u over-the-air pakketafbeeldingen uitvoert. Sommige frames kunnen door de afbeelding gemist worden, maar worden in feite uitgewisseld tussen de client en het toegangspunt. Anders begint het zwerven nooit op dit voorbeeld.

Hier is een samenvatting van de WLC debugs voor deze fast-secure roaming methode:

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

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

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

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

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

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

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

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

*apfMsConnTask_0: Jun 22 00:26:40.788: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID
```

```
  84:78:ac:f0:68:d2(status 0) ApVapId 3 Slot 0
!--- The Reassociation Response is sent to the client, which
     validates the fast-roam with SKC.

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

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

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

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

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

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

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

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

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

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

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

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

FlexConnect met PMKID-caching / Sticky Key Caching

- Wanneer u deze methode gebruikt op een FlexConnect-installatie, kan dit werken en het gedrag kan vergelijkbaar lijken met wat eerder werd uitgelegd als u de centrale verificatie weer gebruikt in de WLC (met centrale of lokale switching); deze SKC-methode wordt echter niet ondersteund op FlexConnect.
- Deze methode wordt alleen officieel ondersteund op CUWN met Local Mode AP's, niet op

FlexConnect of andere modi.

Voordelen met PMKID Caching / Sticky Key Caching

Deze methode kan lokaal worden geïmplementeerd door autonome, onafhankelijke AP's, zonder dat er een gecentraliseerd apparaat nodig is om de gecachede sleutels te beheren.

Nadelen met PMKID Caching / Sticky Key Caching

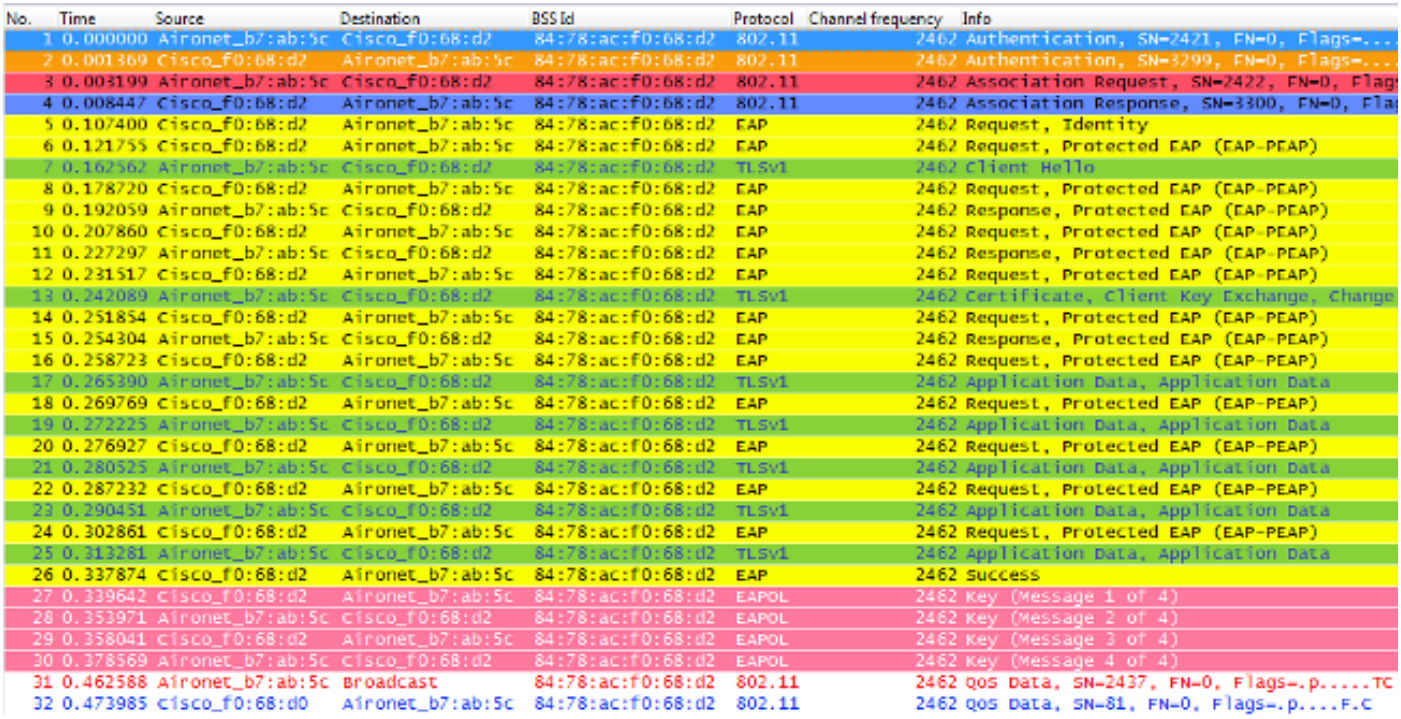
- Zoals eerder in dit document is vermeld, is de belangrijkste beperking van deze methode dat de client alleen snel beveiligde roaming kan uitvoeren wanneer hij terugzwerft naar een toegangspunt waar hij eerder gekoppeld/geverifieerd is. Als u zwerft naar een nieuw toegangspunt, moet de client de volledige EAP-verificatie opnieuw uitvoeren.
- De draadloze client en AP's moeten alle PMK's onthouden die bij elke nieuwe verificatie worden afgeleid, dus deze optie is normaal beperkt tot een bepaalde hoeveelheid PMK's die worden gecachet. Aangezien deze grens niet duidelijk door de norm wordt bepaald, kunnen de verkopers verschillende grenzen op hun SKC implementaties bepalen. Zo kunnen de Cisco WLAN-controllers op dit moment de PMK’s vanaf een client voor maximaal acht AP’s in een cache plaatsen. Als een client naar meer dan acht AP's per sessie zwerft, worden de oudste AP's uit de cachelijst verwijderd om de nieuwe gecachede vermeldingen op te slaan.
- Deze methode is optioneel en wordt nog steeds niet ondersteund door veel WPA2-apparaten; daarom wordt deze methode niet algemeen gebruikt en geïmplementeerd.
- SKC wordt niet ondersteund wanneer u intercontroller roaming uitvoert, wat gebeurt wanneer u zich tussen AP’s beweegt die worden beheerd door verschillende WLC’s, zelfs als ze op dezelfde mobiliteitsgroep zitten.

Fast-Secure Roaming met opportunistische toetscaching

Opportunistische Key Caching (OKC), ook bekend als Proactive Key Caching (PKC) (deze term wordt in meer detail uitgelegd in een notitie die hierna komt), is in feite een uitbreiding van de eerder beschreven WPA2 PMKID-caching methode, wat de reden is dat het ook Proactive/Opportunistic PMKID Caching wordt genoemd. Daarom is het belangrijk om op te merken dat dit geen snel beveiligde roamingmethode is die is gedefinieerd door de 802.11-standaard en niet wordt ondersteund door veel apparaten, maar net als PMKID-caching werkt het met WPA2-EAP.

Deze techniek stelt de draadloze client en de WLAN-infrastructuur in staat om slechts één PMK in het cachegeheugen op te slaan gedurende de levensduur van de clientassociatie met dit WLAN (afgeleid van de MSK na de initiële 802.1X/EAP-verificatie met de verificatieserver), zelfs bij roaming tussen meerdere AP’s, aangezien ze allemaal de oorspronkelijke PMK delen die als zaad wordt gebruikt op alle 4-voudige handgrepen van WPA2. Dit is nog steeds nodig, net zoals het in SKC is, om nieuwe coderingssleutels te genereren telkens wanneer de client zich opnieuw bij de AP’s voegt. Als de AP's deze oorspronkelijke PMK van de clientsessie willen delen, moeten ze allemaal onder een soort administratieve controle staan, met een gecentraliseerd apparaat dat de oorspronkelijke PMK voor alle AP's opslaat en verdeelt. Dit is vergelijkbaar met de CUWN, waar de WLC deze taak uitvoert voor alle LAP's onder zijn controle, en de mobiliteitsgroepen gebruikt om deze PMK tussen meerdere WLC's te verwerken; daarom is dit een beperking op autonome AP-omgevingen.

Met deze methode, net als in PMKID Caching (SKC), is de aanvankelijke associatie met elk AP een regelmatige eerste-keer verificatie naar het WLAN, waar u de volledige 802.1X/EAP-verificatie moet voltooien tegen de verificatieserver en de 4-voudige handdruk voor sleutelgeneratie voordat u gegevenskaders kunt verzenden. Dit is een afbeelding van een scherm dat dit illustreert:



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

De debug-uitgangen tonen in principe dezelfde EAP-verificatiekaderuitwisseling als de rest van de methoden die in dit document worden beschreven bij de eerste verificatie van het WLAN (zoals in de afbeeldingen wordt getoond), samen met de toevoeging van enkele uitgangen die betrekking hebben op de belangrijkste cachetechnieken die hier door de WLC worden gebruikt. Dit debug uitvoer wordt ook gesneden om alleen de relevante informatie te tonen:

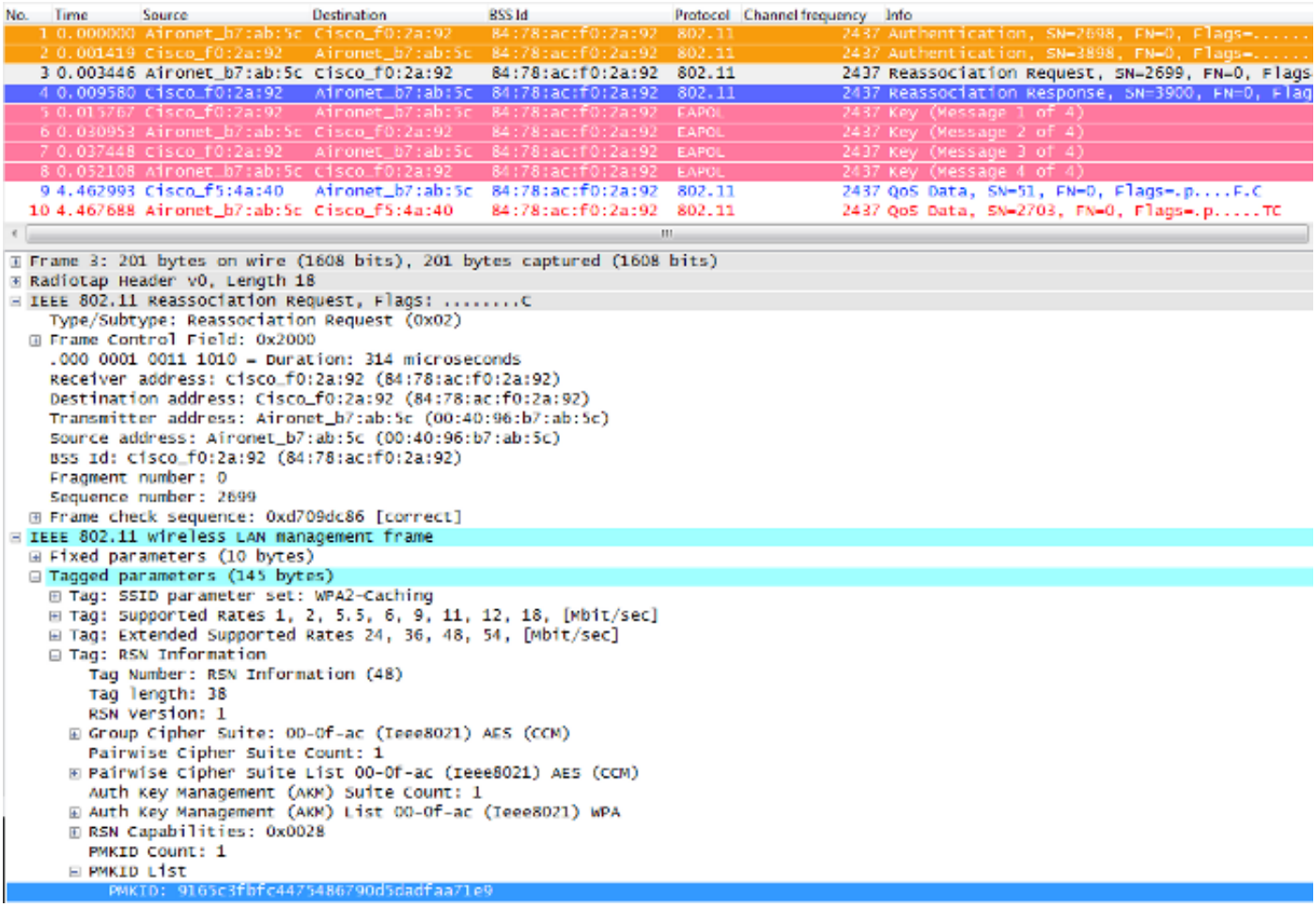
```
*apfMsConnTask_0: Jun 21 21:46:06.515: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID
  84:78:ac:f0:68:d2
!--- This is the Association Request from the client.

*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
```

Met deze methode, de draadloze client en de WLC (voor alle beheerde AP's) cachegeheugen de enige oorspronkelijke PMK van de beveiligde associatie die in eerste instantie is opgericht. Elke keer dat een draadloze client verbinding maakt met een specifieke AP, wordt een PMKID gehakt op basis van: het client-MAC-adres, het AP MAC-adres (BSSID van het WLAN) en de PMK die is afgeleid van die AP. Daarom, aangezien OKC dezelfde originele PMK voor alle APs en de specifieke client caches, wanneer deze client (opnieuw) associeert met een andere AP, is de enige waarde die verandert om de nieuwe PMKID te hakken het nieuwe AP MAC-adres.

Wanneer de client het zwerven op een nieuwe AP initieert en het Reassociation-verzoekframe verstuurt, voegt het de PMKID toe op het WPA2 RSN Information Element als het de AP wil informeren dat een gecachede PMK wordt gebruikt voor snel-veilig zwerven. Het kent al het MAC-adres van de BSSID (AP) voor waar het zwerft, dan de client gewoon hasht de nieuwe PMKID die wordt gebruikt op dit Reassociation-verzoek. Wanneer de AP dit verzoek van de client ontvangt, hasht het ook de PMKID met de waarden die het al heeft (het gecachede PMK, het client-MAC-adres en zijn eigen AP MAC-adres), en reageert met de succesvolle Reassociation Response die bevestigt dat de PMKID's overeenkwamen. De gecachede PMK kan worden gebruikt als het zaad dat een WPA2 4-Way handshake start om de nieuwe encryptiesleutels af te leiden (en EAP over te slaan):



In deze afbeelding wordt het Reassociatie-aanvraagkader van de client geselecteerd en uitgebreid, zodat u meer details van het frame kunt zien. De MAC-adresinformatie en ook het informatie-element van het robuuste beveiligingsnetwerk (RSN, zoals in 802.11i - WPA2), waar informatie over de WPA2-instellingen die voor deze koppeling worden gebruikt wordt weergegeven (gemarkeerd is de PMKID die wordt verkregen met de gehakte formule).

Hier is een samenvatting van WLC debugs voor deze snel-beveiligde roaming methode met OKC:

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

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

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

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

*apfMsConnTask_2: Jun 21 21:48:50.563:
```

```
  [0000] 91 65 c3 fb fc 44 75 48 67 90 d5 da df aa 71 e9

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

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

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

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

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

*dot1xMsgTask: Jun 21 21:48:50.570: 00:40:96:b7:ab:5c
```

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

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

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

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

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

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

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

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

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

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

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

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

Zoals getoond aan het begin van de debugs, moet de PMKID worden berekend nadat de Reassociation-aanvraag van de client is ontvangen. Dit is nodig om de PMKID te valideren en te bevestigen dat de gecachede PMK wordt gebruikt met de WPA2 4-Way handshake om de encryptiesleutels af te leiden en het snel-beveiligde zwerven te voltooien. Verwar de CCKM-vermeldingen niet met de debugs; dit wordt niet gebruikt om CCKM uit te voeren, maar OKC, zoals eerder uitgelegd. CCKM hier is simpelweg een naam die door de WLC wordt gebruikt voor die uitgangen, zoals de naam van een functie die de waarden verwerkt om de PMKID te berekenen.

FlexConnect met opportunistische toetscaching

- Centrale verificatie wordt ondersteund. Dit omvat Lokale en Centrale gegevensomschakeling. Als het toegangspunt deel uitmaakt van dezelfde FlexConnect-groep, wordt het snelbeveiligde zwerven uitgevoerd door het toegangspunt, anders wordt het snelbeveiligde zwerven uitgevoerd door de controller.
  Opmerking: deze instelling kan werken als de toegangspunten niet tot dezelfde FlexConnect-groep behoren, maar dit is geen aanbevolen of ondersteunde instelling.
- Flex Lokale verificatie wordt ondersteund. In de verbonden modus kan het cachegeheugen worden gedistribueerd van het toegangspunt naar de controller en vervolgens naar de rest van de toegangspunten in de FlexConnect-groep.
- Standalone modus wordt ondersteund. Als de cache al aanwezig is op het toegangspunt (als gevolg van eerdere distributie), zal snel beveiligde roaming werken. Nieuwe verificatie in standalone modus ondersteunt geen snel beveiligd zwerven.

Voordelen met opportunistische toetscaching

- De draadloze client en de WLAN-infrastructuur hoeven niet meerdere PMKID's te onthouden, maar cachen de oorspronkelijke PMK eenvoudig van de oorspronkelijke verificatie naar het WLAN. Dan moet u de juiste PMKID (gebruikt op de Reassociation-aanvraag) die vereist is bij elke beveiligde AP-associatie om het snel beveiligde zwerven te valideren.
- Hier voert de draadloze client snel beveiligde roaming uit naar een nieuwe AP op dezelfde WLAN/SSID, zelfs als deze nooit is gekoppeld aan die AP (niet het geval in SKC). Zolang de client de initiële 802.1X/EAP-verificatie uitvoert met één AP die wordt beheerd door de gecentraliseerde implementatie die het PMK-cache verwerkt voor alle AP's waar de client zwerft, zijn geen volledige verificaties meer vereist voor de rest van het clientleven op dit WLAN.

Nadelen met Opportunistische Key Caching

- Deze methode wordt alleen geïmplementeerd op een gecentraliseerde omgeving waar alle AP's onder een of ander soort administratieve controle staan (zoals een WLAN-controller) die verantwoordelijk is voor caching en het delen van de oorspronkelijke PMK van de clientsessie. Daarom is dit een beperking op autonome AP-omgevingen.
- De technieken die in deze methode worden toegepast worden niet voorgesteld of beschreven op de 802.11 standaard, dus de ondersteuning varieert sterk van het ene apparaat tot het andere. Niettemin is dit nog steeds de methode die meer werd aangenomen in afwachting van 802.11r.

Opmerking over de term "Proactive Key Caching"

Proactieve Key Caching (of PKC) is bekend als OKC (Opportunistische Key Caching), en de twee termen worden onderling verwisselbaar gebruikt wanneer ze dezelfde hier uiteengezette methode beschrijven. Dit was echter slechts een term die in 2001 door Airspace werd gebruikt voor een oude key caching methode, die vervolgens werd gebruikt door de 802.11i standaard als basis voor "pre-authenticatie" (een andere Fast Secure Roaming methode, hieronder kort uitgelegd). PKC is geen preauthenticatie of OKC (Opportunistische Key Caching), maar wanneer u hoort of leest over PKC, is de verwijzing in principe naar OKC, en niet naar Preauthenticatie.

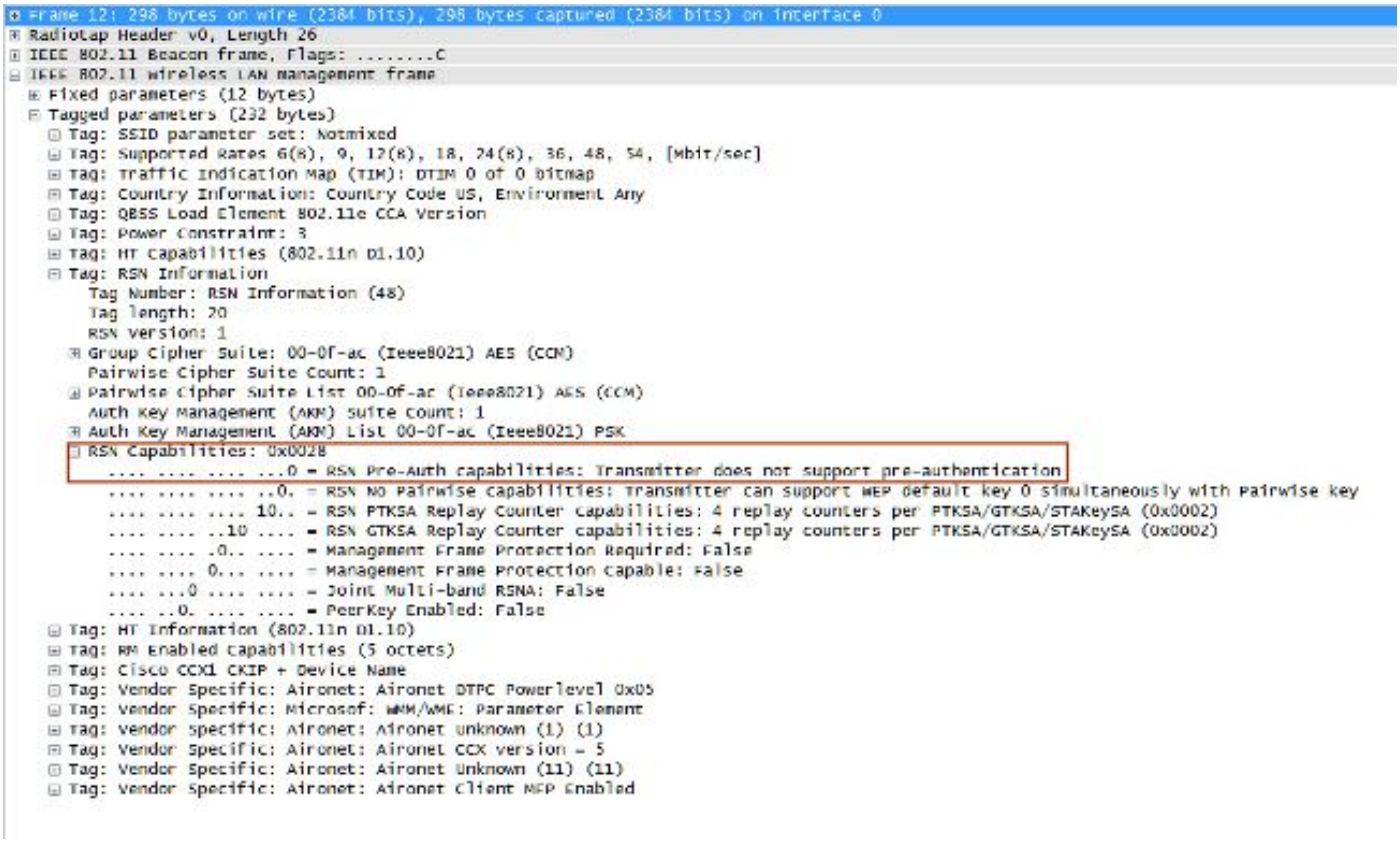
Fast-Secure Roaming met voorafgaande verificatie

Deze methode wordt ook gesuggereerd door de IEEE 802.11-standaard binnen het 802.11i-beveiligingsamendement, dus het werkt ook met WPA2, maar het is de enige Fast Secure Roaming-methode die niet wordt ondersteund door Cisco WLAN-infrastructuur. Daarom wordt het hier slechts kort en zonder output toegelicht.

Met Verificatie vooraf kunnen de draadloze clients met meerdere AP's tegelijk worden geverifieerd terwijl ze gekoppeld zijn aan het huidige AP. Wanneer dit gebeurt, stuurt de client de EAP-verificatiekaders naar het huidige toegangspunt waar de verbinding is gemaakt, maar het is bestemd voor de andere toegangspunt(en) waar de client de verificatie vooraf wil uitvoeren (naburige toegangspunten die mogelijke kandidaten voor zwerven zijn). Het huidige toegangspunt verzendt deze frames naar de doel-toegangspunt(en) via het distributiesysteem. Het nieuwe AP voert volledige verificatie uit tegen de RADIUS-server voor deze client, zodat een volledige nieuwe EAP-verificatie handshake is voltooid en dit nieuwe AP fungeert als de Authenticator.

Het idee is om authenticatie uit te voeren en PMK af te leiden met de naburige AP's voordat de client daadwerkelijk naar hen zwerft, dus wanneer het tijd is om te zwerven, de client is al geauthenticeerd en met een PMK al gecachet voor deze nieuwe veilige vereniging van AP-to-client, zodat ze alleen de 4-Way Handshake moeten uitvoeren en een snel zwerven ervaren nadat de client zijn eerste Reassociation-verzoek verstuurt.

Hier is een afbeelding van een AP-beacon die het RSN IE-veld toont dat ondersteuning voor verificatie vooraf adverteert (dit veld is afkomstig van een Cisco AP, waar is bevestigd dat verificatie vooraf niet wordt ondersteund):



Voordelen met verificatie vooraf

Er is één PMK voor elke AP-to-client beveiligde associatie, die kan worden beschouwd als een

veiligheidsvoordeel als een AP is gecompromitteerd en de sleutels worden gestolen (kan niet worden gebruikt met andere AP's). Dit beveiligingsvoordeel wordt echter op verschillende manieren op andere methoden door de WLAN-infrastructuur verwerkt.

Nadelen met voorafgaande verificatie

- Omdat er één PMK per AP is, hebben de cliënten een grens op de hoeveelheid APs die kan vooraf worden geverifieerd.
- Elke keer dat een client verificatie vooraf uitvoert met een nieuw toegangspunt, is er een volledige EAP-verificatie-uitwisseling, wat betekent dat het netwerk en de verificatieserver zwaarder worden belast.
- De meeste draadloze klanten ondersteunen deze methode niet, aangezien dit nooit hoog werd goedgekeurd (OKC was meer geadopteerd).

Fast-Secure Roaming met 802.11r

De Fast-Secure roaming techniek gebaseerd op de 802.11r-wijziging (officieel **Fast BSS Transition** genoemd door de 802.11-standaard, bekend als **FT**) is de eerste methode die officieel is geratificeerd (op 2008) door de IEEE voor de 802.11-standaard als de oplossing om snelle overgangen tussen AP's (Basic Service Sets of BSS's) uit te voeren, die duidelijk de sleutelhiërarchie definieert die wordt gebruikt wanneer u sleutels en cachesleutels op een WLAN behandelt. De adoptie is echter traag geweest, vooral door de andere oplossingen die al beschikbaar waren toen er eigenlijk snelle overgangen nodig waren, zoals met VoWLAN-implementaties bij gebruik met een van de methoden die eerder in dit document zijn uitgelegd. Er zijn slechts een paar apparaten die momenteel een aantal van de FT-opties ondersteunen (tegen 2013).

Deze techniek is complexer om te verklaren dan de andere methoden, aangezien het nieuwe concepten en meerdere lagen PMKs introduceert die op verschillende apparaten (elk apparaat met een andere rol) worden gecached, en biedt zelfs nog meer opties voor snel-veilig zwerven. Daarom wordt een korte samenvatting gegeven over deze methode en de manier waarop deze wordt toegepast met elke beschikbare optie.

802.11r verschilt van SKC en OKC, voornamelijk om de volgende redenen:

- Handshake messaging (bijvoorbeeld PMKID, ANonce en SNonce exchange) vindt plaats in 802.11 verificatie frames of in Action frames in plaats van Reassociation frames. Anders dan bij PMKID-caching, wordt de aparte 4-way handshake fase, die wordt uitgevoerd na de (re)associatie bericht uitwisseling, vermeden. De belangrijkste handdruk met het nieuwe AP begint voordat de client volledig zwerft/opnieuw koppelt aan dit nieuwe AP.
- Het biedt twee methoden voor de snelzwervende handdruk: via de AIR en via het distributiesysteem (DS).
- 802.11r heeft meer lagen in de sleutelhiërarchie.
- Aangezien dit protocol de 4-voudige handdruk voor het sleutelbeheer vermijdt wanneer een client zwerft (genereert nieuwe coderingssleutels -PTK en GTK- zonder de noodzaak van deze handdruk), kan het ook worden toegepast voor WPA2-instellingen met een PSK, en niet alleen wanneer 802.1X/EAP wordt gebruikt voor de verificatie. Dit versnelt het zwerven nog meer voor deze opstellingen, waar geen EAP of 4-Way handshake uitwisselingen voorkomen.

Met deze methode voert de draadloze client slechts één initiële verificatie uit tegen de WLAN-

infrastructuur wanneer een verbinding met het eerste toegangspunt tot stand wordt gebracht, en voert de client snel beveiligde roaming uit tussen toegangspunten van hetzelfde FT-mobiliteitsdomein.

Dit is een van de nieuwe concepten, die in principe verwijst naar de AP's die dezelfde SSID (bekend als Extended Service Set of ESS) gebruiken en dezelfde FT-toetsen afhandelen. Dit is vergelijkbaar met de andere methoden die tot nu toe zijn toegelicht. De manier waarop de AP's omgaan met de FT-mobiliteitsdomeinsleutels is normaal gesproken gebaseerd op een gecentraliseerde instelling, zoals de WLC of mobiliteitsgroepen; deze methode kan echter ook worden geïmplementeerd op autonome AP-omgevingen.

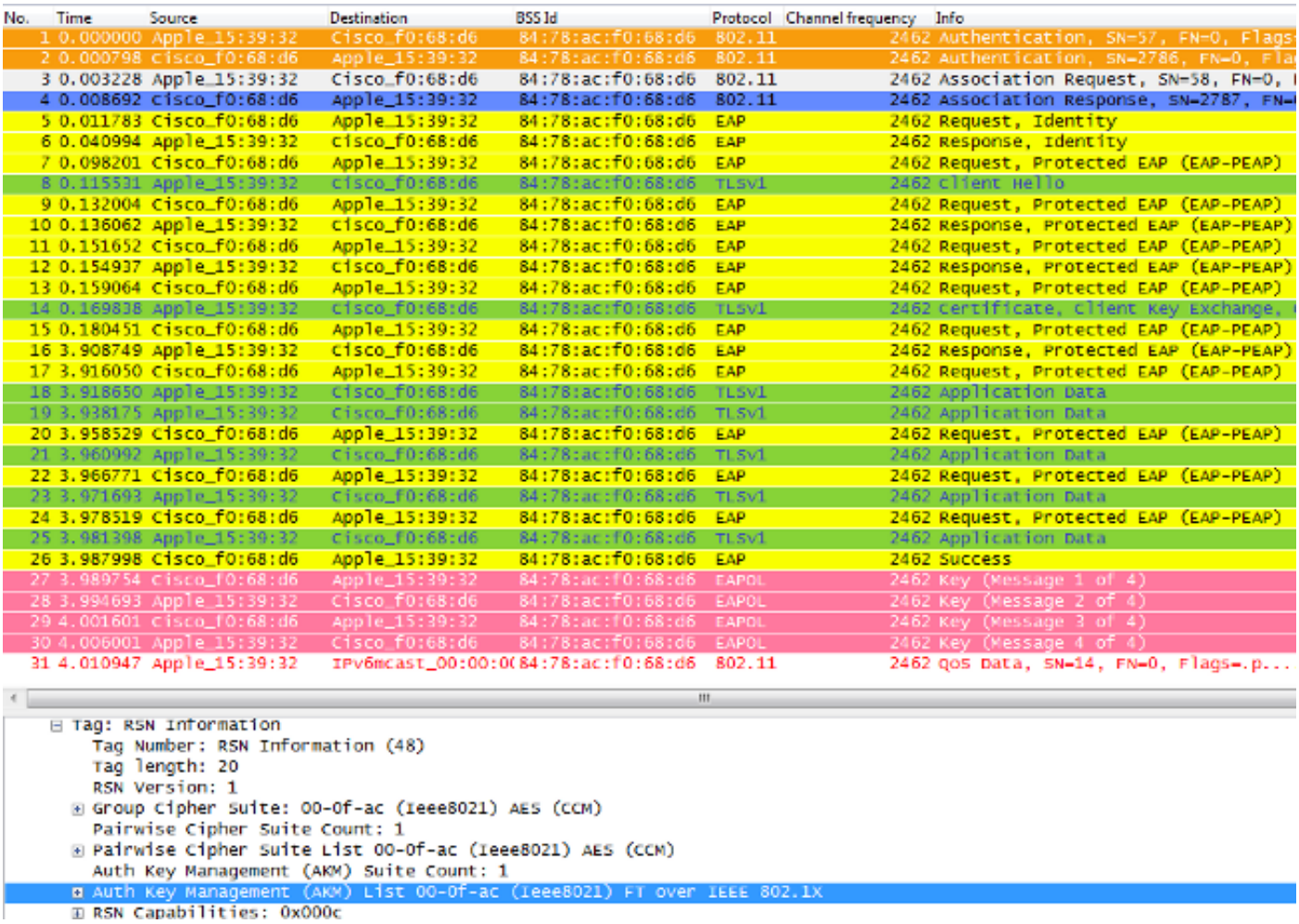
Hier is een samenvatting van de belangrijkste hiërarchie:

- Er wordt nog steeds een MSK afgeleid van de client supplicant en de verificatieserver van de initiële 802.1X/EAP-verificatiefase (overgedragen van de verificatieserver naar de verificator (WLC) zodra de verificatie succesvol is). Deze MSK wordt, net als bij de andere methoden, gebruikt als het zaad voor de FT-sleutelhiërarchie. Wanneer u WPA2-PSK gebruikt in plaats van een EAP-verificatiemethode, is de PSK in principe deze MSK.
- Een Pairwise Master Key R0 (PMK-R0) is afgeleid van de MSK, die de key op het eerste niveau van de FT-sleutelhiërarchie is. De belangrijkste houders voor deze PMK-R0 zijn de WLC en de client.
- Een tweede-niveau sleutel, genaamd een Pairwise Master Key R1 (PMK-R1), is afgeleid van de PMK-R0, en de belangrijkste houders zijn de client en de AP's beheerd door de WLC die de PMK-R0 houdt.
- De derde en laatste leveltoets van de FT-sleutelhiërarchie is de PTK, die de definitieve sleutel is die wordt gebruikt om de 802.11 unicast datakaders te versleutelen (vergelijkbaar met de andere methoden die WPA/TKIP of WPA2/AES gebruiken). Deze PTK is op FT afgeleid van de PMK-R1, en de belangrijkste houders zijn de client en de AP's beheerd door de WLC.

  Opmerking: afhankelijk van de WLAN-leverancier en de implementatiestations (zoals Autonomous APs, FlexConnect of Mesh) kan de WLAN-infrastructuur de sleutels op een andere manier overdragen en verwerken. Het kan zelfs de rollen van de sleutelhouders veranderen, maar aangezien dat buiten het bereik van dit document valt, zijn de voorbeelden op basis van de eerder gegeven samenvatting van de sleutelhiërarchie de volgende focus. De verschillen zijn eigenlijk niet zo relevant om het proces te begrijpen, tenzij je echt nodig hebt om de infrastructuur apparaten (en hun code) grondig te analyseren om een software probleem te ontdekken.

Snelle BSS-overgang via de lucht

Met deze methode is de eerste koppeling naar een AP een regelmatige eerste-keer verificatie naar het WLAN, waar de gehele 802.1X/EAP-verificatie tegen de verificatieserver en de 4-voudige handdruk voor de sleutelgeneratie moeten plaatsvinden voordat gegevensframes worden verzonden, zoals in deze afbeelding op het scherm wordt getoond:



De belangrijkste verschillen zijn:

- De onderhandelingen voor het beheer van verificatiesleutels verschillen enigszins van de reguliere WPA/WPA2, zodat enige extra informatie wordt gebruikt om deze onderhandeling uit te voeren wanneer de koppeling aan een WLAN-infrastructuur die FT ondersteunt, plaatsvindt. Zoals in de afbeelding wordt getoond, wordt het Associatie-aanvraagkader van de client geselecteerd en wordt het AKM-veld van het RNS Informatie Element gemarkeerd om aan te tonen dat deze client FT over 802.1X/EAP wil uitvoeren.
- Ook wordt het Mobility Domain Information Element (onderdeel van FT) getoond, waar het veld **FT Capability and Policy** aangeeft of de Fast BSS Transition wordt voltooid Over-the-Air of Over-the-DS bij snel zwerven (dit geeft Over-the-Air aan in deze afbeelding).
- Er wordt ook een ander informatieonderdeel toegevoegd (Fast BSS Transition of FT IE, dat later in dit document wordt beschreven) met informatie die nodig is om de FT-authenticatiereeks uit te voeren bij FT-roaming.
- De sleutelgeneratie is anders door de sleutelhiërarchie, dus ook al lijkt de FT 4-Way handdruk op de WPA/WPA2 4-Way handdruk, het is eigenlijk een beetje anders qua inhoud.

De debugs tonen in principe dezelfde EAP authenticatie frame uitwisseling als de rest van de methoden op de eerste authenticatie aan het WLAN (zoals opgemerkt van de beelden), maar sommige outputs die betrekking hebben op de belangrijkste caching technieken die worden gebruikt door de WLC worden toegevoegd; dus deze debug uitvoer wordt gesneden om alleen de relevante informatie te tonen:

```
*apfMsConnTask_0: Jun 27 19:25:23.426: ec:85:2f:15:39:32
  Association received from mobile on BSSID
```

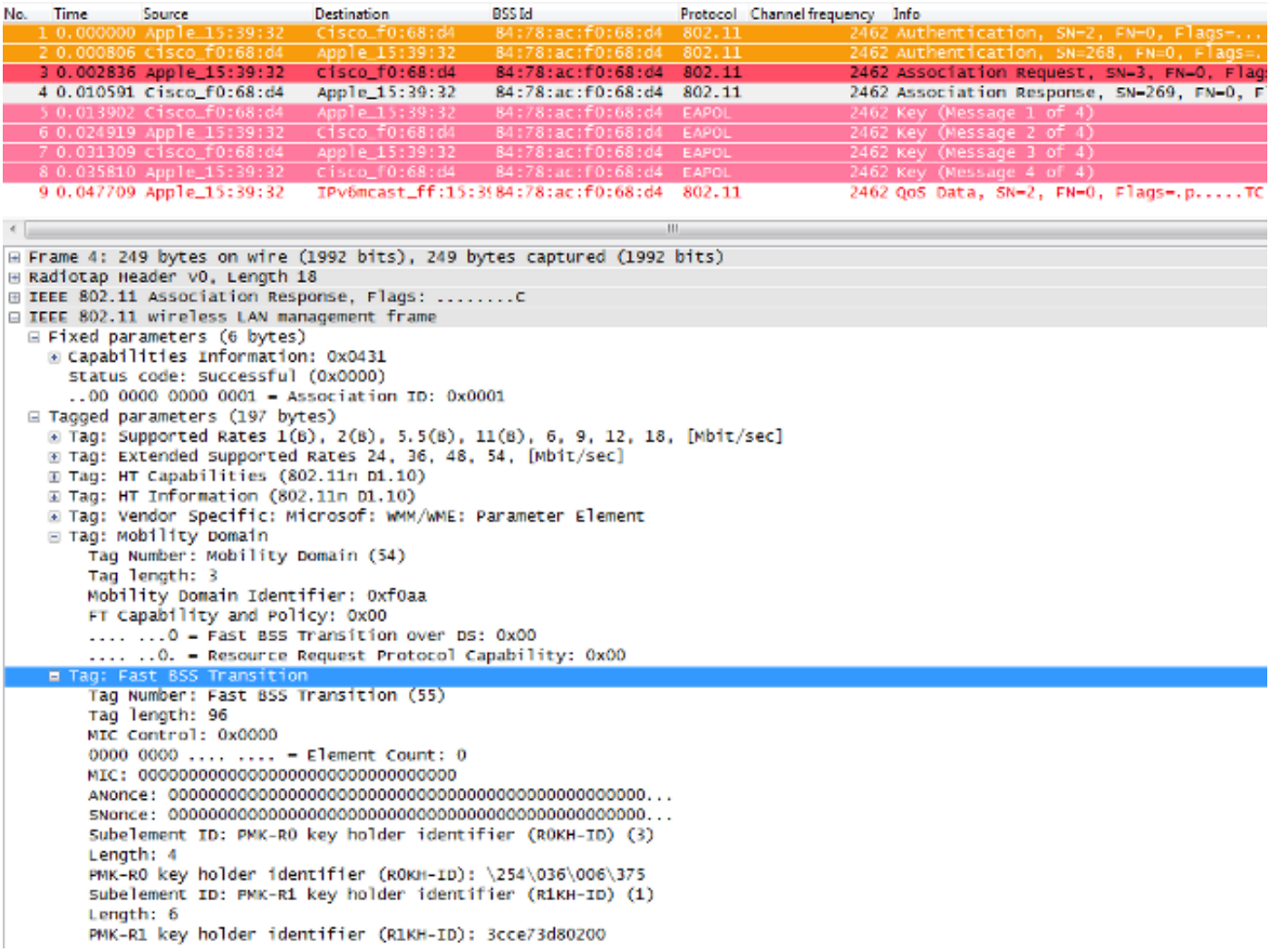
```
  84:78:ac:f0:68:d6
!--- This is the Association request from the client.

*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.
```

> Opmerking: om deze methode te debuggen en de extra 802.11r/FT-uitgangen te bereiken die hier worden getoond, wordt een extra debug ingeschakeld samen met de **debug client**, die de **debug ft-gebeurtenissen inschakelen** is.

Hier zijn de afbeeldingen en debugs van een eerste koppeling naar het WLAN wanneer u FT met WPA2-PSK uitvoert (in plaats van een 802.1X/EAP-methode), waar het Association Response frame van het AP is geselecteerd om het Fast BSS Transition Information Element (gemarkeerd) te tonen. Enkele van de belangrijkste informatie die nodig is om de FT 4-Way handshake uit te voeren wordt ook getoond:



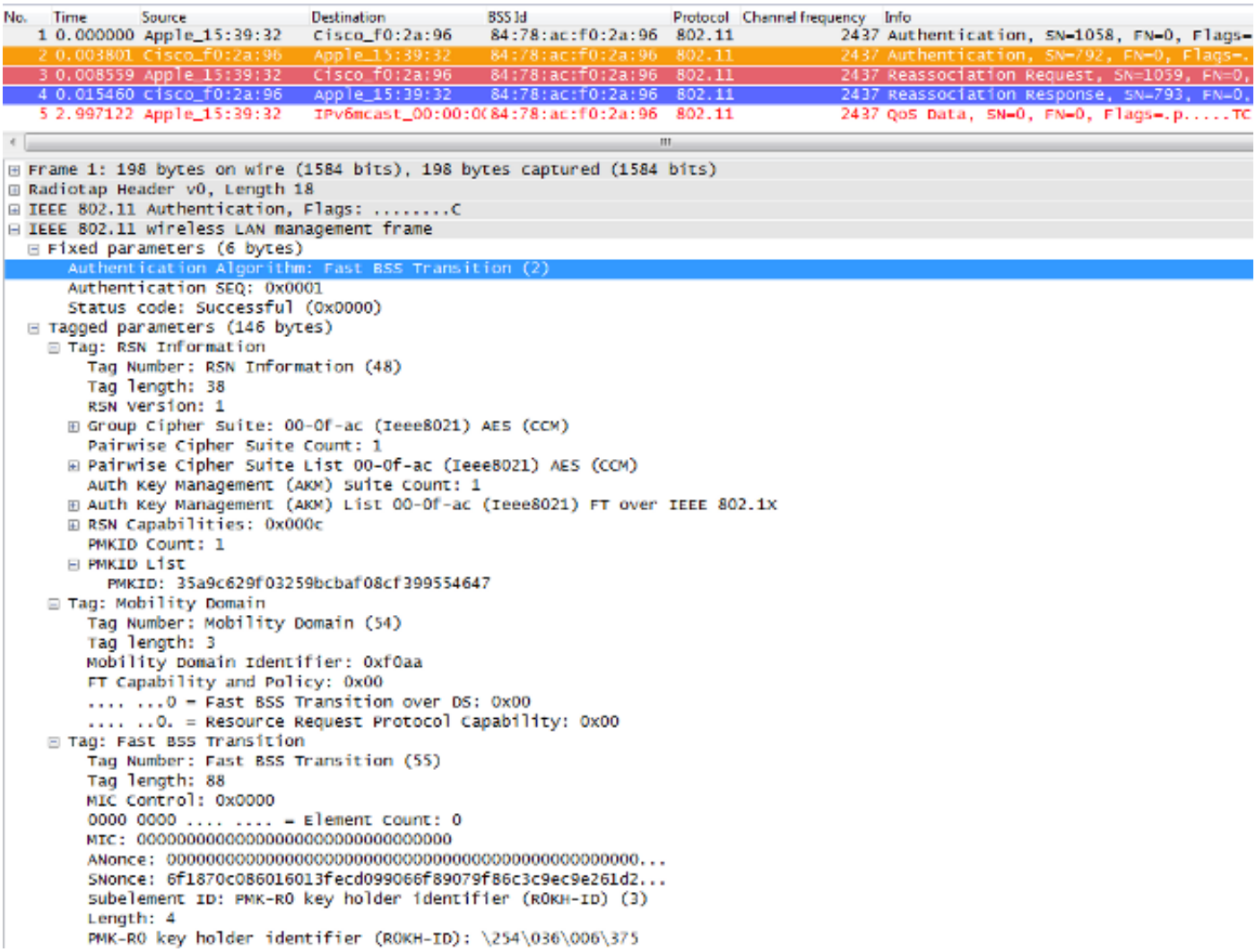

```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*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
```

Met 802.11r is de eerste verbinding met het WLAN de basis die wordt gebruikt om de basissleutels af te leiden die door deze techniek worden gebruikt, net als bij de andere snel beveiligde roamingmethoden. De belangrijkste verschillen komen wanneer de client begint te zwerven; FT vermijdt niet alleen 802.1X/EAP wanneer dit wordt gebruikt, maar voert in feite een efficiëntere zwervende methode uit die de initiële 802.11 Open System-verificatie- en reassociatieframes (die altijd worden gebruikt en vereist bij zwerven tussen AP's) combineert om FT-informatie uit te wisselen en nieuwe dynamische coderingssleutels af te leiden in plaats van de 4-way handshake.

De volgende afbeelding toont de frames die zijn uitgewisseld wanneer een Fast BSS Transition Over-the-Air met 802.1X/EAP-beveiliging wordt uitgevoerd. Het kader voor Open Systeemverificatie van client tot AP is geselecteerd om de FT-protocol informatie-elementen te zien die nodig zijn om te beginnen met de FT-toetsonderhandeling. Dit wordt gebruikt om de nieuwe PTK met de nieuwe AP af te leiden (gebaseerd op de PMK-R1). Het veld dat het verificatiealgoritme toont, wordt gemarkeerd om aan te tonen dat deze client geen eenvoudige Open System-verificatie uitvoert, maar een Fast BSS-overgang:

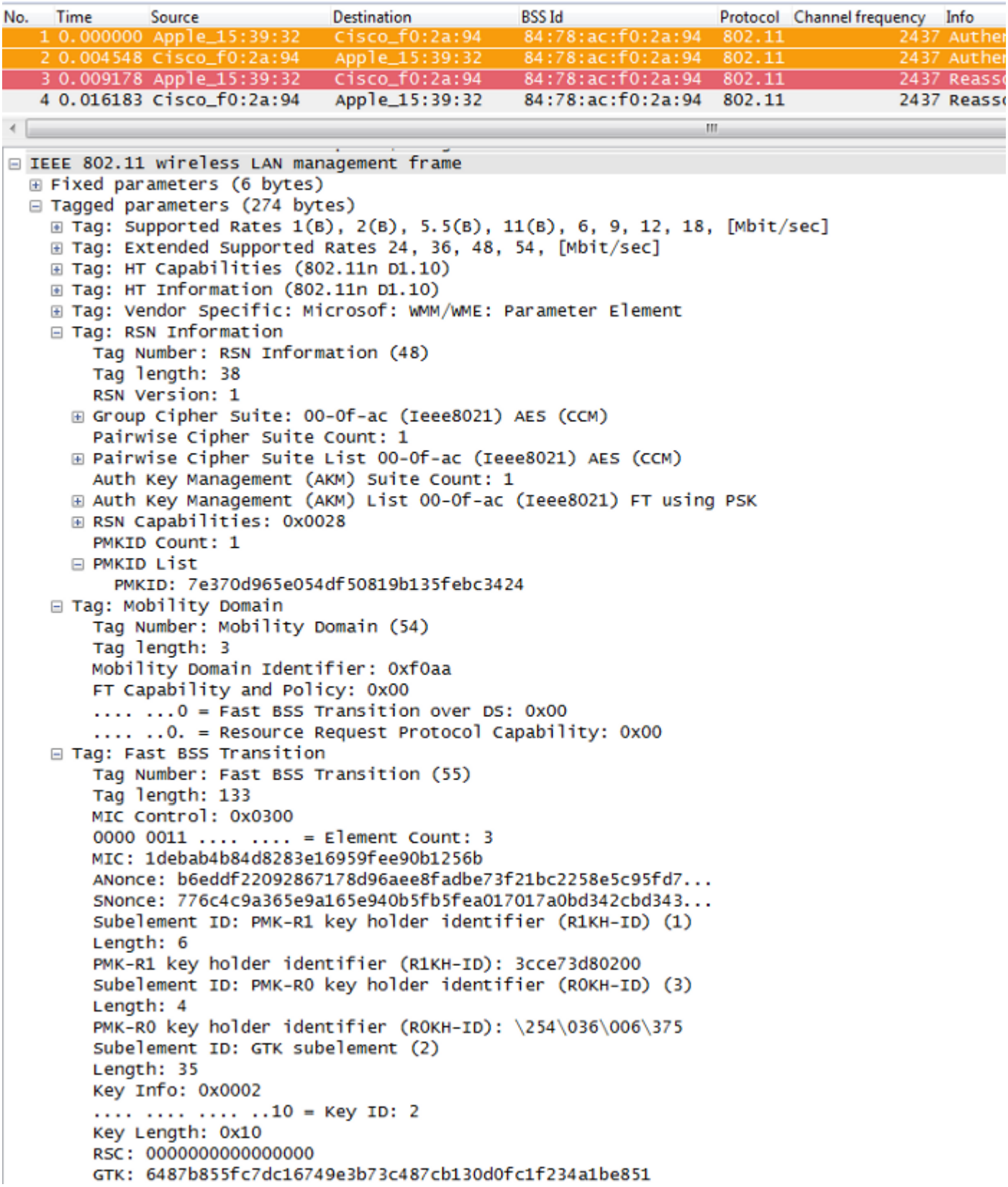


Hier zijn de debug-uitgangen van de WLC wanneer deze FT roaming-gebeurtenis plaatsvindt met 802.1X/EAP:

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

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

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

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

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

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

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

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

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

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

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

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

Hier is een afbeelding die een snelle BSS Transition Over-the-Air met WPA2-PSK-beveiliging toont, waar het uiteindelijke Reassociation Response frame van de AP op de client is geselecteerd om meer details over deze FT-uitwisseling te tonen:



Hier zijn de debug-uitgangen wanneer deze FT roaming-gebeurtenis plaatsvindt met PSK, die vergelijkbaar zijn met de uitgangen wanneer 802.1X/EAP wordt gebruikt:

```
*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
```

Zoals in de afbeelding wordt getoond, worden de vier frames die worden gebruikt en nodig zijn voor zwerven (Open Systeemverificatie van de client, Open Systeemverificatie van de AP, Reassociation Verzoek, en Reassociation Response), zodra de Fast BSS Transition wordt onderhandeld op de eerste verbinding met het WLAN, in principe gebruikt als een FT 4-way handshake om de nieuwe PTK (unicast encryptie sleutel) en GTK (multicast/broadcast encryptie sleutel) af te leiden.

Dit vervangt de 4-voudige handdruk die normaal optreedt nadat deze frames zijn uitgewisseld, en de FT-inhoud en de belangrijkste onderhandeling over deze frames is in principe hetzelfde of u 802.1X/EAP of PSK gebruikt als de beveiligingsmethode. Zoals in de afbeelding wordt getoond, is het AKM-veld het belangrijkste verschil, dat bevestigt als de client FT met PSK of 802.1X uitvoert. Daarom is het belangrijk om op te merken dat deze vier frames normaal gesproken niet dit type

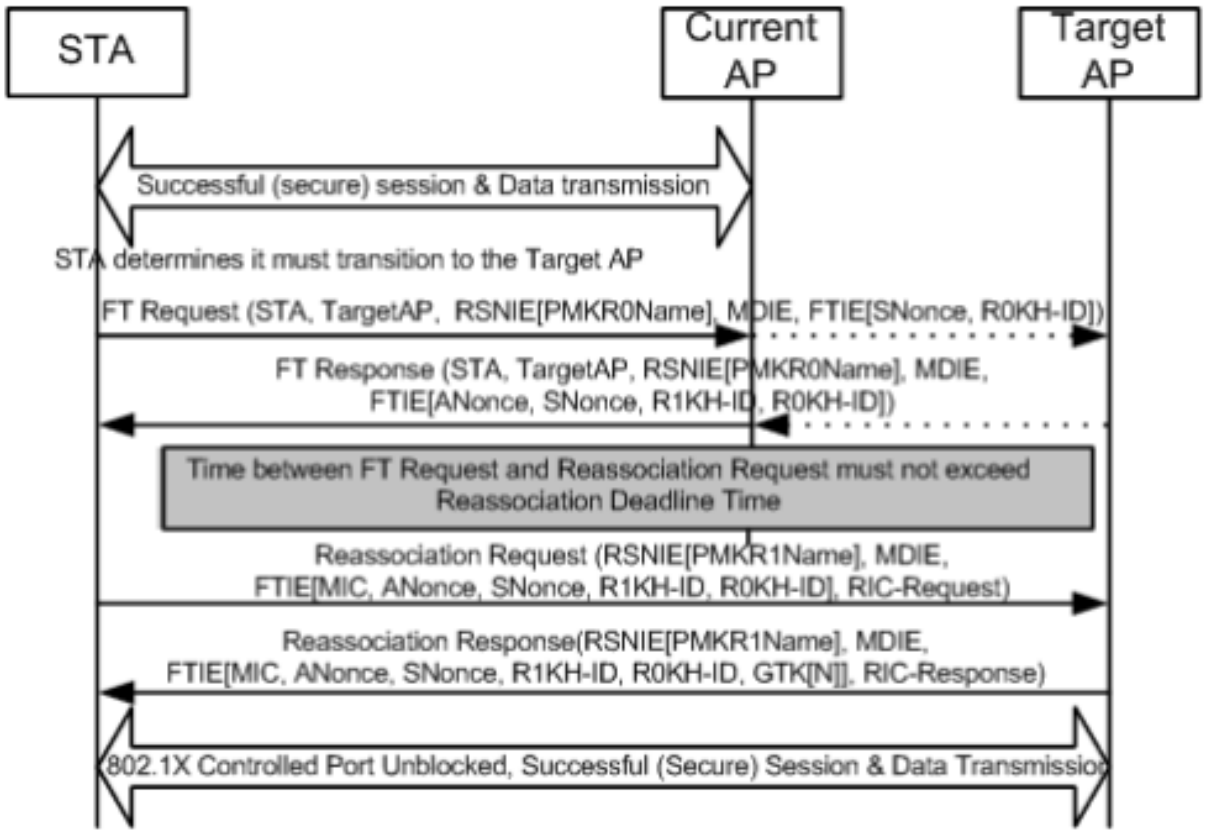
beveiligingsinformatie voor belangrijke onderhandeling hebben, maar alleen wanneer client-FT roamt als 802.11r is geïmplementeerd en onderhandeld tussen de client en de WLAN-infrastructuur bij eerste koppeling.

Snelle BSS-overgang over de DS

802.11r staat een andere implementatie van Fast BSS Transition toe, waarbij de FT roaming wordt geïnitieerd door de client met de nieuwe AP waarvoor de client over-the-DS (distributiesysteem) zwerft, en niet over-the-Air. In dit geval worden FT Action frames gebruikt om de belangrijkste onderhandeling te starten in plaats van de Open System Verification frames.

Wanneer de client besluit dat hij naar een betere AP kan zwerven, stuurt de client een FT Action request frame naar de oorspronkelijke AP waar hij op dit moment verbonden is voordat hij zwerft. De client geeft de BSSID (MAC-adres) van de doel-AP aan waar het wil roamen. De oorspronkelijke AP stuurt dit FT Action request frame door naar de doel-AP via het distributiesysteem (normaal de bekabelde infrastructuur), en de doel-AP reageert op de client met een FT Action Response frame (ook via de DS, zodat het eindelijk over-the-air naar de client kan sturen). Zodra deze FT Action frame exchange succesvol is, voltooit de client de FT roaming; de client stuurt het Reassociation-verzoek naar de doel-AP (dit keer via de lucht) en ontvangt een Reassociation Response van de nieuwe AP om de roaming en de uiteindelijke afleiding van de sleutels te bevestigen.

Samengevat zijn er vier frames om te onderhandelen over Fast BSS Transition en nieuwe coderingssleutels af te leiden, maar hier worden de Open System-verificatieframes gesubstitueerd door de FT Action request/response-frames, die via het distributiesysteem met het huidige AP worden uitgewisseld. Deze methode is ook geldig voor zowel de beveiligingsmethoden 802.1X/EAP en PSK, die allemaal worden ondersteund door de Cisco draadloze LAN-controllers. Aangezien deze overstap over-the-DS echter niet wordt ondersteund en geïmplementeerd door de meeste draadloze clients in de WiFi-industrie (en omdat de uitgangen voor frameuitwisseling en debug in principe hetzelfde zijn), worden in dit document geen voorbeelden gegeven. In plaats daarvan wordt deze afbeelding gebruikt om de Fast BSS Transition Over-the-DS te visualiseren:



FlexConnect met 802.11r

- Centrale verificatie wordt ondersteund. Dit omvat Lokale en Centrale gegevensomschakeling. De AP's moeten deel uitmaken van dezelfde FlexConnect Group.
- Lokale verificatie wordt niet ondersteund.
- Standalone modus wordt niet ondersteund.

Voordelen met 802.11r

- Deze methode is de eerste die een belangrijke hiërarchie gebruikt die duidelijk door IEEE op de 802.11-standaard als amendement (802.11r) is gedefinieerd, zodat de implementatie van deze FT-technieken beter compatibel is tussen leveranciers en zonder verschillende interpretaties.
- 802.11r maakt meerdere technieken mogelijk die nuttig zijn, afhankelijk van uw behoeften (Over-the-Air en Over-the-DS, voor 802.1x/EAP-beveiliging en voor PSK-beveiliging).
- De draadloze client voert snel beveiligde roaming uit naar een nieuwe AP op dezelfde WLAN/SSID, zelfs als deze nooit is gekoppeld aan die AP, en zonder dat meerdere PMKID’s moeten worden opgeslagen.
- Dit is de eerste snel-beveiligde zwervende methode die snellere zwerven mogelijk maakt, zelfs met PSK-beveiliging, en vermijdt de 4-voudige handdruk die nodig is bij zwerven tussen AP's met WPA/WPA2 PSK. Het belangrijkste doel van de snelbeveiligde roamingmethoden is om de 802.1X/EAP-handdruk te vermijden wanneer deze beveiligingsmethode is geïmplementeerd; voor PSK-beveiliging wordt de roaminggebeurtenis echter nog sneller met 802.11r wanneer de 4-voudige handdruk wordt vermeden.

Nadelen met 802.11r

- Er zijn een paar draadloze clientapparaten die Fast BSS-overgangen ondersteunen, en in de meeste gevallen ondersteunen ze niet alle technieken die beschikbaar zijn op 802.11r.
- Vanwege het feit dat deze implementaties erg jong zijn, zijn er niet genoeg testresultaten van real-productie omgevingen of genoeg debug resultaten om mogelijke voorbehouden die kunnen verschijnen aan te pakken.
- Wanneer u een WLAN/SSID configureert om een van de FT-methoden te gebruiken, kunnen alleen draadloze clients die 802.11r ondersteunen verbinding maken met dit WLAN/SSID. De FT-instellingen zijn niet optioneel voor de clients, zodat draadloze clients die 802.11r niet ondersteunen, verbinding moeten maken met een afzonderlijke WLAN/SSID waar FT helemaal niet is geconfigureerd.

Adaptief 802.11r

- Sommige legacy-clients kunnen geen verbinding maken met een WLAN/SSID met 802.11r ingeschakeld, zelfs voor "gemengde modus" (waarvan u hoopt dat u dat kunt doen op dezelfde SSID-clients die 802.11r ondersteunen en die 802.11r niet ondersteunen). Dit is wanneer de bestuurder van de cliëntaanvrager die voor het ontleden van het Robust Security Network Information Element (RSN IE) verantwoordelijk is oud is en zich niet bewust van de extra AKM reeksen in IE. Vanwege deze beperking kunnen clients geen associatieverzoeken verzenden naar WLAN's die 802.11r-ondersteuning adverteren, en moet u daarom één WLAN/SSID voor 802.11r-clients en een afzonderlijke WLAN/SSID voor clients die 802.11r niet ondersteunen, configureren.
- Om dit te overwinnen, introduceerde Cisco draadloze LAN-infrastructuur de Adaptieve 802.11r-functie. Wanneer de FT-modus is ingesteld op Adaptief op WLAN-niveau, adverteert WLAN met 802.11r Mobility Domain ID op een 802.11i-enabled WLAN. Sommige Apple iOS10-clientapparaten identificeren de aanwezigheid van MDIE op een 80211i/WPA2 WLAN en voeren een bedrijfseigen handdruk om 802.11r-associatie tot stand te brengen. Zodra de client een succesvolle 802.11r-associatie heeft voltooid, kan FT-roaming worden uitgevoerd zoals in een normaal 802.11r-enabled WLAN. De FT Adaptive is alleen van toepassing op geselecteerde Apple iOS10 (en hoger) apparaten. Alle andere clients kunnen 802.11i/WPA2-associatie blijven gebruiken op het WLAN en de toepasselijke FSR-methode uitvoeren zoals ondersteund.
- Meer documentatie over deze nieuwe functie die is geïntroduceerd voor iOS10-apparaten om 802.11r uit te voeren op een WLAN/SSID waar 802.11r niet echt is ingeschakeld (zodat andere clients die geen 802.11r zijn, met succes verbinding kunnen maken), is te vinden in Enterprise Best Practices voor Cisco IOS-apparaten op Cisco Wireless LAN.

Conclusies

- Houd in gedachten dat de client altijd degene is die beslist om te zwerven naar een specifieke AP, en de WLC/AP kan dit niet beslissen voor de client. De roaming-gebeurtenis wordt geïnitieerd door de draadloze client zodra het denkt dat het moet zwerven.
- De WLC ondersteunt een combinatie van de meeste of alle FSR (Fast-Secure Roaming) methoden samen op dezelfde WLAN/SSID. Houd er echter rekening mee dat dit normaal niet werkt, aangezien het sterk afhankelijk is van het gedrag van de klant (zeer verschillend over

verschillende mobiele apparaten) om te ondersteunen of zelfs te begrijpen dat wat de WLC probeert te adverteren als ondersteund. In plaats van het bereiken van interoperabiliteit in slechts één SSID, zijn er normaal gesproken meer kwesties dan de kwesties die naar verwachting zullen worden vastgesteld, dus dit wordt niet aanbevolen. Diepgaande tests met alle mogelijke clients die op dit WLAN moeten worden gebruikt, moeten worden voltooid als dit echt nodig is.

- Het is zeer belangrijk om te begrijpen dat snel-veilige het zwerven methodes worden ontwikkeld om het WLAN het zwerven proces te versnellen wanneer u zich tussen APs beweegt als WLAN/SSID toegelaten veiligheid heeft. Wanneer er geen beveiliging is, is er niets te versnellen, omdat client-AP simpelweg de draadloze beheerframes ruilt die altijd worden vereist wanneer zwervend tussen AP's voordat de dataframes worden verzonden (Open Systeemverificatie van de client, Open Systeemverificatie van de AP, Reassociation Verzoek, en Reassociation Response). Dit kan dus niet sneller gaan. Als u problemen ondervindt met zwerven zonder beveiliging, dan zijn er geen snelzwervende methoden om roaming te verbeteren, alleen methoden om te bevestigen of de WLAN/SSID-installatie en het ontwerp geschikt zijn voor de draadloze clientstations om dienovereenkomstig tussen de AP-dekkingscellen te zwerven.
- 802.11r/FT is geïmplementeerd met WPA2-PSK om roaminggebeurtenissen met deze beveiliging te versnellen en de 4-voudige handdruk te vermijden, zoals in de 802.11r-sectie wordt uitgelegd.
- Alle methoden hebben hun voordelen en nadelen, maar uiteindelijk moet u altijd controleren of de draadloze clientstations de specifieke methode ondersteunen die u wilt implementeren en of de Cisco WLAN-infrastructuur alle beschikbare methoden ondersteunt. Aldus, moet u de beste methode selecteren die eigenlijk door de draadloze cliënten wordt ondersteund die met specifieke WLAN/SSID verbinden. In sommige implementaties kunt u bijvoorbeeld een WLAN/SSID met CCKM maken voor Cisco draadloze IP-telefoons (die WPA2/AES met CCKM ondersteunen, maar niet 802.11r) en vervolgens een ander WLAN/SSID met WPA2/AES via 802.11r/FT voor draadloze clients die deze Fast Secure Roaming-methode ondersteunen (of OKC gebruiken als dit wordt ondersteund).
- Als de draadloze clients geen van de beschikbare, snel beveiligde roamingmethoden ondersteunen, moet u aanvaarden dat die clients altijd de in dit document beschreven vertragingen kunnen experimenteren bij het roamen tussen AP's op een WLAN/SSID met 802.1X/EAP-beveiliging (wat storingen kan veroorzaken op de client-apps/services).
- Alle methoden, behalve SKC (WPA2 PMKID Caching), worden ondersteund voor snel beveiligde roaming tussen AP's die worden beheerd door verschillende WLC's (intercontroller roaming), zolang ze zich op dezelfde mobiliteitsgroep bevinden.
- CUWN ondersteunt alle verschillende Fast-Secure Roaming methoden die in dit artikel worden besproken wanneer 802.1X/EAP-verificatie wordt gebruikt voor WPA/WPA2. CUWN ondersteunt Fast-Secure Roaming niet op methoden die werken met WPA2-RSN (CCKM, PMKID Caching/SKC, OKC/PKC) wanneer PSK (WPA2-Personal) wordt gebruikt, waar Fast-Roaming methoden meestal niet nodig zijn. Echter, CUWN ondersteunt Fast-Secure Roaming in het geval van WPA2-FT (802.11r) met PSK zoals ook uitgelegd in dit artikel.

Gerelateerde informatie

- Implementatiehandleiding voor 802.11r BSS Fast Transition
- Cisco technische ondersteuning en downloads

Over deze vertaling

Cisco heeft dit document vertaald via een combinatie van machine- en menselijke technologie om onze gebruikers wereldwijd ondersteuningscontent te bieden in hun eigen taal. Houd er rekening mee dat zelfs de beste machinevertaling niet net zo nauwkeurig is als die van een professionele vertaler. Cisco Systems, Inc. is niet aansprakelijk voor de nauwkeurigheid van deze vertalingen en raadt aan altijd het oorspronkelijke Engelstalige document (link) te raadplegen.