Verifica dei metodi per la WLAN 802.11 e il roaming Fast-Secure su CUWN

Sommario

Introduzione

In questo documento vengono descritti i tipi di roaming wireless e a sicurezza rapida disponibili per le LAN wireless (WLAN) IEEE 802.11 su Unified Wireless Network (CUWN).

Prerequisiti

Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Nozioni fondamentali sulle WLAN IEEE 802.11
- Sicurezza WLAN IEEE 802.11
- Nozioni di base di IEEE 802.1X/EAP

Componenti usati

Il riferimento delle informazioni contenute in questo documento è il software Cisco WLAN Controller versione 7.4.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Premesse

Le informazioni di questo documento si basano sul software Cisco WLAN Controller versione 7.4, ma la maggior parte degli output e dei comportamenti di debug descritti possono essere applicati a qualsiasi versione del software che supporta i metodi discussi. Le specifiche di tutti i metodi spiegati qui rimangono le stesse sui codici Cisco WLAN Controller successivi (fino alla versione 8.3 al momento dell'aggiornamento di questo articolo).

Questo documento descrive i diversi tipi di roaming wireless e i metodi di roaming veloce e sicuro disponibili per le LAN wireless (WLAN) IEEE 802.11 supportate sulla rete Cisco Unified Wireless Network (CUWN).

Nel documento non vengono fornite tutte le specifiche relative al funzionamento o alla configurazione di ogni metodo. Lo scopo principale di questo documento è descrivere le differenze tra le varie tecniche disponibili, i loro vantaggi e le loro limitazioni, e lo scambio di frame su ogni metodo. Vengono forniti esempi di debug di controller WLAN (WLC) e vengono usate immagini di pacchetti wireless per analizzare e spiegare gli eventi che si verificano per ogni metodo di roaming descritto.

Prima di fornire una descrizione dei diversi metodi di roaming a sicurezza rapida disponibili per le WLAN, è importante comprendere come funziona il processo di associazione WLAN e come si verifica un evento di roaming regolare quando non è configurata alcuna sicurezza sull'SSID (Service Set Identifier).

Quando un client wireless 802.11 si connette a un Access Point (AP), prima di iniziare a trasmettere il traffico (frame di dati wireless), deve prima superare il processo di autenticazione di base 802.11 Open System. Quindi, il processo di associazione deve essere completato. Il processo di autenticazione del sistema aperto è simile a una connessione via cavo sull'access point selezionato dal client. Questo è un punto molto importante, perché è sempre il client wireless a selezionare l'access point preferito e a decidere in base a diversi fattori che variano da fornitore

a fornitore. Ecco perché il client inizia questo processo inviando il frame di autenticazione all'access point selezionato, come mostrato più avanti in questo documento. L'access point non può richiedere di stabilire una connessione.

Una volta completato il processo di autenticazione del sistema aperto con una risposta dell'access point ("cavo connesso"), il processo di associazione conclude essenzialmente la negoziazione 802.11 Layer 2 (L2) che stabilisce il collegamento tra il client e l'access point. L'access point assegna un ID di associazione al client se la connessione viene stabilita correttamente e lo prepara in modo da trasmettere il traffico o eseguire un metodo di sicurezza di livello superiore, se configurato sull'SSID. Il processo di autenticazione Open System è costituito da due frame di gestione e dal processo di associazione. I frame di autenticazione e associazione sono **frame di gestione** wireless, non frame di dati, che sono fondamentalmente quelli utilizzati per il processo di connessione con l'access point.

Di seguito è riportata un'immagine dei frame wireless over-the-air per questo processo:

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

> **Nota**: per informazioni sullo sniffing wireless 802.11 e sui filtri/colori usati su Wireshark per le immagini visualizzate in questo documento, visitare il sito Web Cisco Support Community post intitolato 802.11 Sniffer image Analysis.

Il client wireless inizia con il frame di autenticazione e l'access point risponde con un altro frame di autenticazione. Il client invia quindi il frame Richiesta associazione e l'access point finisce in una risposta con il frame Risposta associazione. Come mostrato dai pacchetti DHCP, una volta passati i processi di autenticazione e associazione 802.11 Open System, il client inizia a passare i frame di dati. In questo caso, non esiste alcun metodo di sicurezza configurato sull'SSID, quindi il client inizia immediatamente a inviare frame di dati (in questo caso DHCP) non crittografati.

Come mostrato più avanti in questo documento, se la protezione è abilitata sul SSID, esistono frame di handshake di autenticazione e crittografia di livello superiore per il metodo di protezione specifico, subito dopo l'Association Response e prima dell'invio di qualsiasi frame di dati sul traffico del client, ad esempio DHCP, Address Resolution Protocol (ARP) e pacchetti di applicazioni, che vengono crittografati. I frame di dati possono essere inviati solo fino a quando il client non viene autenticato completamente e le chiavi di crittografia non vengono negoziate in base al metodo di protezione configurato.

A seconda dell'immagine precedente, di seguito sono riportati i messaggi che vengono visualizzati negli output del comando WLC **debug client** quando il client wireless inizia una nuova associazione alla WLAN:

```
*apfMsConnTask_0: Jun 21 18:55:14.221: 00:40:96:b7:ab:5c
  Association received from mobile on BSSID 84:78:ac:f0:68:d0
```

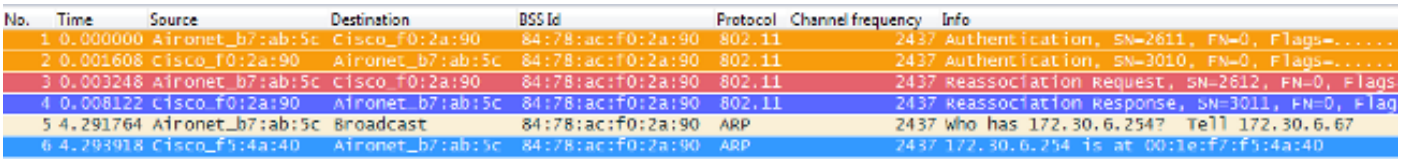
```
!--- This is the Association Request from the wireless client
      to the selected AP.

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

> **Nota**: il comando debug WLC usato per gli output mostrati in questo documento è il comando **debug client** e gli esempi mostrano solo alcuni messaggi importanti, non l'intero output. Per ulteriori informazioni sul comando debug, consultare il documento relativo al debug del client sui controller WLC (Wireless LAN Controller).

Questi messaggi mostrano i frame Richiesta associazione e Risposta; i frame Autenticazione iniziale non vengono registrati sul WLC perché questo handshake si verifica rapidamente a livello AP sul CUWN.

Quali informazioni vengono visualizzate quando il client esegue il roaming? Il client scambia sempre quattro frame di gestione quando stabilisce una connessione a un punto di accesso, a causa della creazione dell'associazione del client o di un evento di roaming. Il client ha una sola connessione stabilita a un solo access point alla volta. L'unica differenza nello scambio di frame tra una nuova connessione all'infrastruttura WLAN e un evento di roaming è che i frame di associazione di un evento di roaming sono chiamati frame di **riassociazione**, che indicano che il client sta effettivamente usando il roaming da un altro access point senza tentare di stabilire una nuova associazione con la WLAN. Questi frame possono contenere diversi elementi utilizzati per negoziare l'evento di roaming. Ciò dipende dall'impostazione, ma tali dettagli esulano dall'ambito del presente documento.

Di seguito è riportato un esempio di scambio di frame:



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

I messaggi seguenti vengono visualizzati nell'output del comando debug:

```
*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.
```

Come mostrato, il client esegue correttamente un evento roaming dopo l'invio della richiesta di riassociazione al nuovo punto di accesso e riceve la risposta di riassociazione dall'punto di accesso. Poiché il client ha già un indirizzo IP, i primi frame dati sono per i pacchetti ARP.

Se si prevede un evento di roaming, ma il client invia una richiesta di associazione invece di una richiesta di riassociazione (che è possibile confermare da alcune immagini e debug simili a quelli illustrati in precedenza in questo documento), il client non è realmente in roaming. Il client inizia una nuova associazione alla WLAN come se si fosse verificata una disconnessione e tenta di

riconnettersi da zero. Questo può accadere per diversi motivi, ad esempio quando un client si allontana dalle aree di copertura e quindi trova un punto di accesso con una qualità del segnale sufficiente per avviare un'associazione, ma in genere indica un problema del client in cui il client non avvia un evento di roaming a causa di driver, firmware o problemi software.

> **Nota**: è possibile contattare il fornitore del client wireless per determinare la causa del problema.

Roaming con sicurezza di livello superiore

Quando il SSID è configurato con la protezione di livello superiore L2 oltre all'autenticazione di base 802.11 Open System, sono necessari più frame per l'associazione iniziale e durante il roaming. I due metodi di sicurezza più comuni standardizzati e implementati per le WLAN 802.11 sono descritti nel presente documento:

- **WPA/WPA2-PSK (chiave già condivisa)** - autenticazione dei client con una chiave già condivisa.
- **WPA/WPA2-EAP (Extensible Authentication Protocol)**: autenticazione dei client con un metodo 802.1X/EAP per convalidare credenziali più sicure tramite un server di autenticazione, ad esempio certificati, nome utente e password e token.

Èimportante sapere che, anche se questi due metodi (PSK e EAP) autenticano/convalidano i client in modi diversi, entrambi utilizzano sostanzialmente le stesse regole WPA/WPA2 per il processo di gestione delle chiavi. Se la protezione è WPA/WPA2-PSK o WPA/WPA2-EAP, il processo noto come handshake WPA/WPA2 a 4 vie avvia la negoziazione della chiave tra il WLC/AP e il client con una chiave di sessione master (MSK) come materiale della chiave originale una volta che il client è convalidato con il metodo di autenticazione specifico utilizzato.

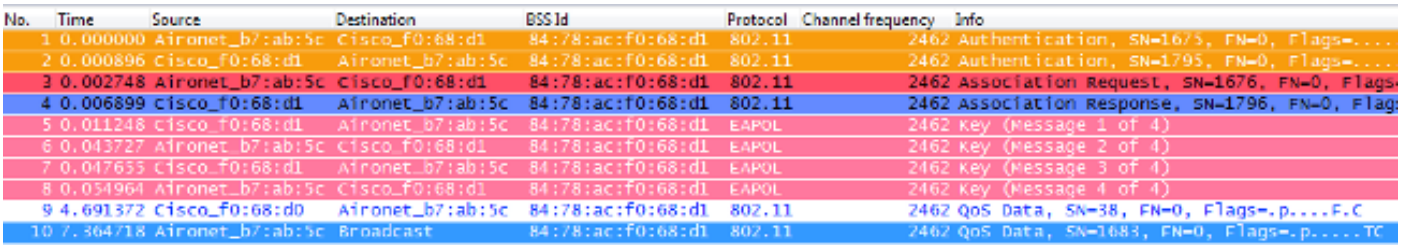
Di seguito è riportato un riepilogo del processo:

1. Il codice MSK è derivato dalla fase di autenticazione EAP quando si utilizza la protezione 802.1X/EAP oppure dal codice PSK quando si utilizza il metodo di protezione WPA/WPA2-PSK.
2. Da questo MSK, il client e il WLC/AP derivano la PMK (Pairwise Master Key) e il WLC/AP genera una GMK (Group Master Key).
3. Quando queste due chiavi master sono pronte, il client e il WLC/AP avviano l'handshake a 4 vie WPA/WPA2 (illustrato più avanti in questo documento con alcune immagini dello schermo e debug) con le chiavi master come base per la negoziazione delle chiavi di crittografia effettive.
4. Queste chiavi di crittografia finali sono note come Pairwise Transient Key (PTK) e Group Transient Key (GTK). La chiave PTK viene derivata dalla chiave PMK e utilizzata per crittografare i frame unicast con il client. La chiave GTK (Group Transient Key) deriva dalla chiave GMK e viene utilizzata per crittografare multicast/broadcast su questo SSID/AP specifico.

WPA/WPA2-PSK

Quando si esegue WPA-PSK o WPA2-PSK tramite TKIP (Temporal Key Integrity Protocol) o AES (Advanced Encryption Standard) per la crittografia, il client deve eseguire il processo noto come

handshake WPA a 4 vie sia per l'associazione iniziale che per il roaming. Come spiegato in precedenza, si tratta essenzialmente del processo di gestione delle chiavi utilizzato per consentire a WPA/WPA2 di derivare le chiavi di crittografia. Tuttavia, quando si esegue la chiave già condivisa, questa viene usata anche per verificare che il client abbia una chiave già condivisa valida per collegarsi alla WLAN. Nell'immagine è illustrato il processo di associazione iniziale quando viene eseguito WPA o WPA2 con PSK:



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

Come mostrato, dopo il processo di autenticazione e associazione 802.11 Open System, ci sono quattro frame EAPOL dall'handshake WPA a 4 vie, che vengono inizializzati dall'access point con **message-1**, e completati dal client con **message-4**. Dopo un handshake riuscito, il client inizia a passare i frame di dati (ad esempio DHCP), che in questo caso vengono crittografati con le chiavi derivate dall'handshake a 4 vie (per questo motivo non è possibile visualizzare il contenuto effettivo e il tipo di traffico proveniente dalle immagini wireless).

> **Nota**: i frame EAPOL vengono usati per trasportare tutti i frame di gestione delle chiavi e i frame di autenticazione 802.1X/EAP via etere tra l'access point e il client; vengono trasmessi come frame di dati wireless.

I messaggi riportati di seguito vengono visualizzati negli output del comando debug:

```
*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.
```

Durante il roaming, il client in pratica tiene traccia dello stesso scambio di frame, in cui è necessario l'handshake WPA a 4 vie per derivare nuove chiavi di crittografia con il nuovo access point. Ciò è dovuto a motivi di sicurezza stabiliti dallo standard e al fatto che il nuovo punto di accesso non conosce le chiavi originali. L'unica differenza è che esistono frame di riassociazione invece di frame di associazione, come mostrato nell'immagine seguente:

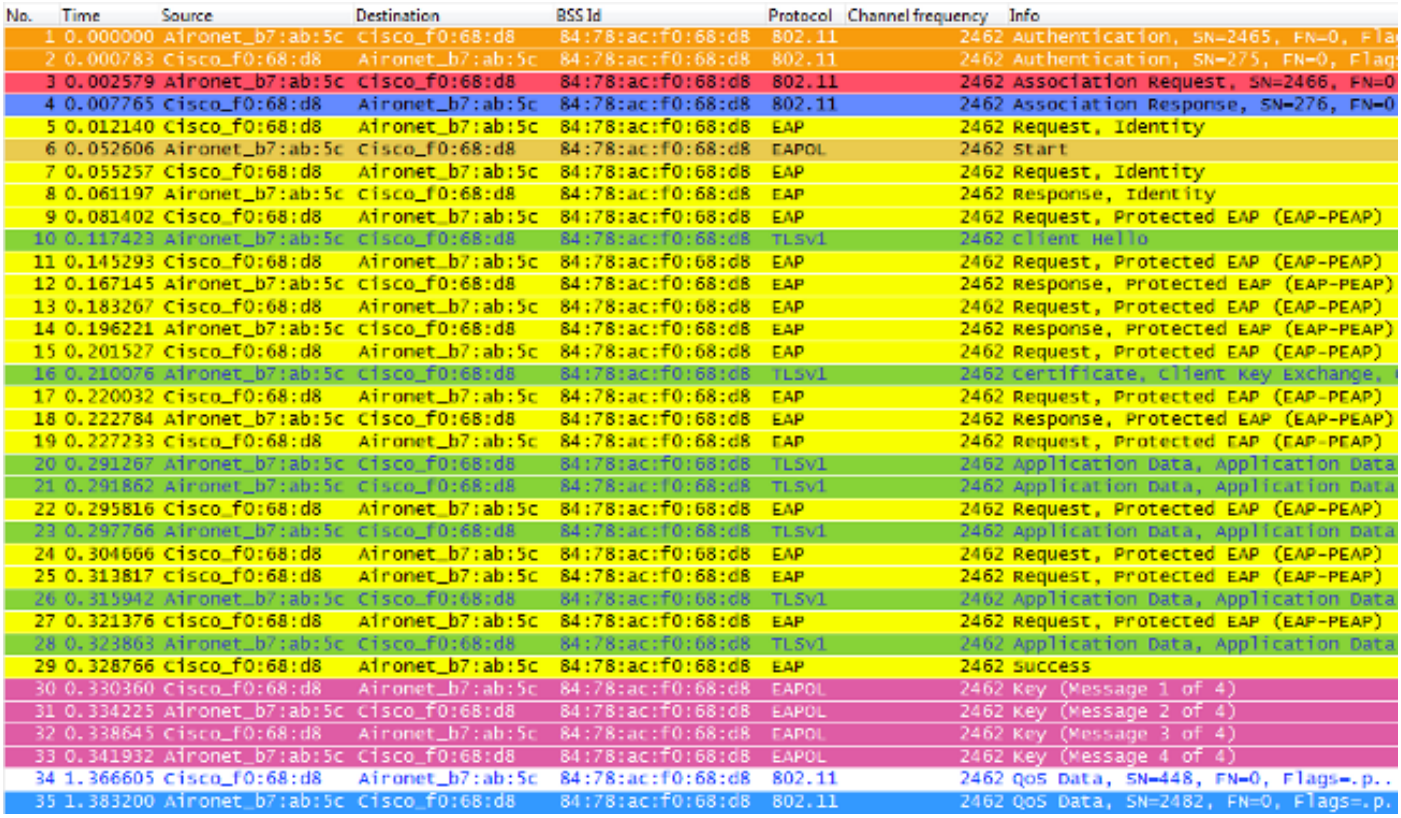
No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:91	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=2356, FN=0, Flags=.....
2	0.000846	Cisco_f0:2a:91	Aironet_b7:ab:5c	84:78:ac:f0:2a:91	802.11	2437	Authentication, SN=3694, FN=0, Flags=.....
3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

Gli stessi messaggi vengono visualizzati negli output di debug, ma il primo pacchetto proveniente dal client è una riassociazione anziché un'associazione, come mostrato e spiegato in precedenza.

WPA/WPA2-EAP

Quando si usa un metodo 802.1X/EAP per autenticare i client su un SSID sicuro, sono necessari ancora più frame prima che il client inizi a trasmettere il traffico. Questi frame aggiuntivi vengono utilizzati per autenticare le credenziali del client e, a seconda del metodo EAP, possono essere presenti da quattro a venti frame. Questi vengono forniti dopo l'associazione/riassociazione, ma prima dell'handshake a 4 vie WPA/WPA2, in quanto la fase di autenticazione deriva il MSK utilizzato come valore di inizializzazione per la generazione finale della chiave di crittografia nel processo di gestione delle chiavi (handshake a 4 vie).

Nell'immagine è mostrato un esempio di frame scambiati via etere tra l'access point e il client wireless durante l'associazione iniziale quando viene eseguito WPA con PEAPv0/EAP-MSCHAPv2:



No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	Authentication, SN=2465, FN=0, Fla
2	0.000783	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	Authentication, SN=275, FN=0, Flag
3	0.002579	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	Association Request, SN=2466, FN=0
4	0.007765	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	Association Response, SN=276, FN=0
5	0.012140	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Identity
6	0.052606	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Start
7	0.055257	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Identity
8	0.061197	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Identity
9	0.081402	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
10	0.117423	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Client Hello
11	0.145293	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
12	0.167145	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
13	0.183267	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
14	0.196221	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
15	0.201527	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
16	0.210076	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Certificate, Client Key Exchange,
17	0.220032	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
18	0.222784	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAP	2462	Response, Protected EAP (EAP-PEAP)
19	0.227233	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
20	0.291267	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
21	0.291862	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
22	0.295816	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
23	0.297766	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
24	0.304666	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
25	0.313817	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
26	0.315942	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
27	0.321376	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Request, Protected EAP (EAP-PEAP)
28	0.323863	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	TLSv1	2462	Application Data, Application Data
29	0.328766	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAP	2462	Success
30	0.330360	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 1 of 4)
31	0.334225	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 2 of 4)
32	0.338645	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 3 of 4)
33	0.341932	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	EAPOL	2462	Key (Message 4 of 4)
34	1.366605	Cisco_f0:68:d8	Aironet_b7:ab:5c	84:78:ac:f0:68:d8	802.11	2462	QoS Data, SN=448, FN=0, Flags=.p..
35	1.383200	Aironet_b7:ab:5c	Cisco_f0:68:d8	84:78:ac:f0:68:d8	802.11	2462	QoS Data, SN=2482, FN=0, Flags=.p.

A volte questo scambio mostra più o meno frame, che dipendono da più fattori, come il metodo EAP, ritrasmissioni a causa di problemi, comportamento del client (come le due richieste di identità in questo esempio, perché il client invia un **EAPOL START** dopo che l'AP invia la prima richiesta di identità), o se il client ha già scambiato il certificato con il server. Ogni volta che l'SSID è configurato per un metodo 802.1X/EAP, sono presenti più frame (per l'autenticazione) e quindi è necessario più tempo prima che il client inizi a inviare i frame dati.

Di seguito è riportato un riepilogo dei messaggi di debug:

```
*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.
```

Quando il client wireless esegue un roaming regolare (il comportamento normale, senza implementare un metodo di roaming veloce e sicuro), deve eseguire lo stesso processo ed eseguire un'autenticazione completa sul server di autenticazione, come mostrato nelle immagini. L'unica differenza consiste nel fatto che il client utilizza una richiesta di riassociazione per informare il nuovo access point che è in roaming da un altro access point, ma il client deve comunque eseguire la convalida completa e la generazione di nuove chiavi:

No.	Time	Source	Destination	BSSId	Protocol	Channel frequency	Info
1	0.000000	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=2637, FN=0, Flags=.....C
2	0.000821	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Authentication, SN=96, FN=0, Flags=.....C
3	0.003857	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	802.11	2437	Reassociation Request, SN=2638, FN=0, Flags=...
4	0.008646	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	802.11	2437	Reassociation Response, SN=97, FN=0, Flags=....
5	0.014409	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
6	0.029712	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAPOL	2437	Start
7	0.035084	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Identity
8	0.053240	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	EAP	2437	Response, Identity
9	0.062770	Cisco_f0:2a:98	Aironet_b7:ab:5c	84:78:ac:f0:2a:98	EAP	2437	Request, Protected EAP (EAP-PEAP)
10	0.065313	Aironet_b7:ab:5c	Cisco_f0:2a:98	84:78:ac:f0:2a:98	TLSv1	2437	Client Hello
11	0.071282	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

Come mostrato, anche quando ci sono meno frame rispetto all'autenticazione iniziale (che è causata da diversi fattori, come accennato in precedenza), quando il client esegue il roaming verso un nuovo access point, l'autenticazione EAP e i processi di gestione delle chiavi WPA devono ancora essere completati per continuare a passare i frame di dati (anche se il traffico è stato attivamente inviato prima del roaming). Pertanto, se il client dispone di un'applicazione attiva sensibile ai ritardi (ad esempio applicazioni per il traffico vocale o applicazioni sensibili ai timeout), l'utente può rilevare problemi durante il roaming, ad esempio interruzioni audio o disconnessioni dell'applicazione. Dipende dalla durata del processo affinché il client continui a inviare/ricevere frame di dati. Questo ritardo può essere maggiore, a seconda dell'ambiente RF, della quantità di client, del tempo di andata e ritorno tra i WLC e i LAP e con il server di autenticazione e di altri motivi.

Di seguito è riportato un riepilogo dei messaggi di debug per questo evento di roaming (fondamentalmente gli stessi dei messaggi precedenti, quindi questi messaggi non vengono descritti ulteriormente):

```
*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
```

In questo modo funzionano 802.1X/EAP e la struttura di protezione WPA/WPA2. Per evitare l'impatto di applicazioni/servizi sui ritardi dovuti a un evento di roaming regolare, il settore WiFi sviluppa e implementa diversi metodi di roaming veloci e sicuri al fine di accelerare il processo di roaming quando viene utilizzata la sicurezza sulla WLAN/SSID. I client devono affrontare una certa latenza quando continuano a passare il traffico mentre sono in roaming tra i punti di accesso tramite l'implementazione di un livello di sicurezza elevato sulla WLAN. Ciò è dovuto all'autenticazione EAP e agli scambi di frame di gestione delle chiavi richiesti dall'impostazione della sicurezza, come spiegato in precedenza.

Èimportante comprendere che il roaming veloce e sicuro è semplicemente il termine utilizzato dal settore in riferimento all'implementazione di un metodo/schema che accelera il processo di roaming quando la sicurezza è configurata sulla WLAN. I diversi metodi/schemi di roaming a sicurezza rapida disponibili per le WLAN e supportati dal CUWN sono spiegati nella sezione successiva.

Roaming veloce e sicuro con CCKM

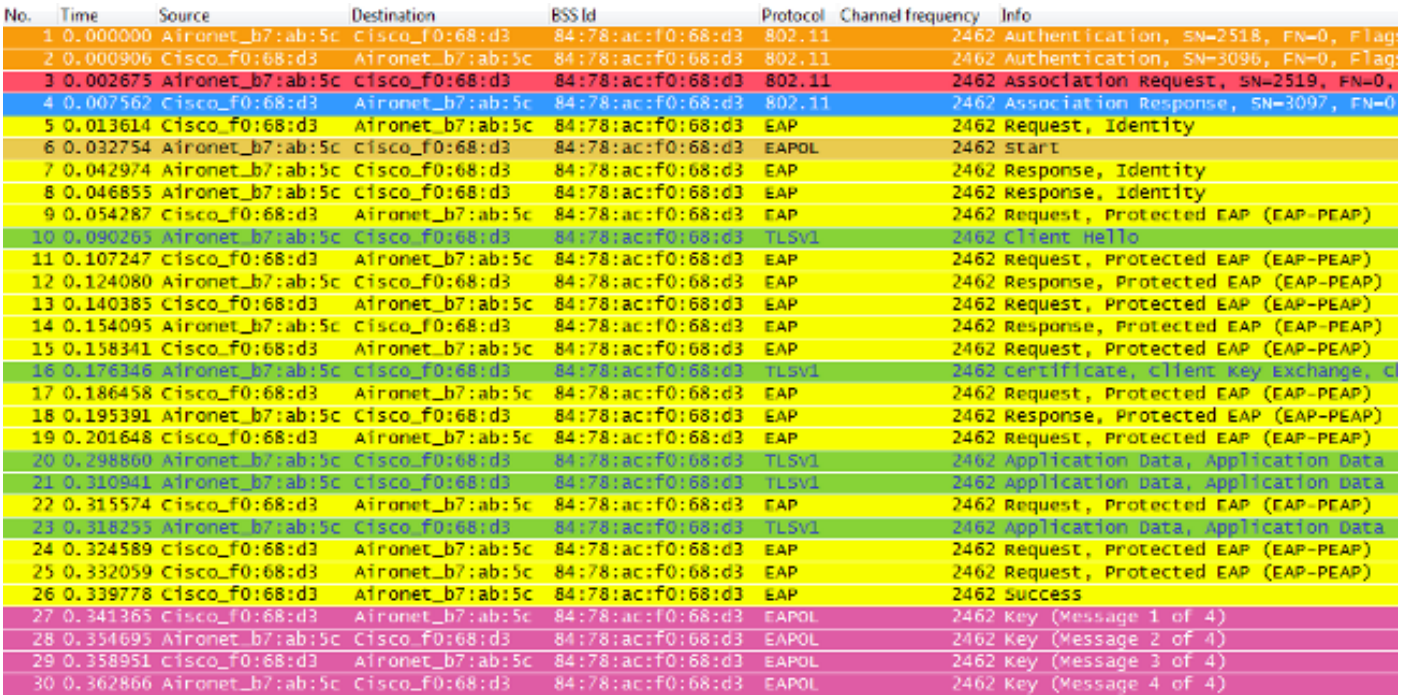
La gestione centralizzata delle chiavi (CCKM) di Cisco è il primo metodo di roaming veloce e sicuro sviluppato e implementato sulle WLAN aziendali, creato da Cisco come soluzione per ridurre i ritardi spiegati finora, quando si usa la sicurezza 802.1X/EAP sulla WLAN. Poiché si tratta di un protocollo proprietario di Cisco, è supportato solo dai dispositivi dell'infrastruttura WLAN e dai client wireless Cisco (di più fornitori) compatibili con Cisco Compatible Extension (CCX) per CCKM.

La funzione CCKM può essere implementata con tutti i diversi metodi di crittografia disponibili per le WLAN, tra cui WEP, TKIP e AES. È inoltre supportato dalla maggior parte dei metodi di

autenticazione 802.1X/EAP utilizzati per le WLAN, a seconda della versione CCX supportata dai dispositivi.

> **Nota**: per una panoramica sul contenuto delle funzioni supportate dalle diverse versioni della specifica CCX (che include i metodi EAP supportati), fare riferimento al documento Versioni e funzionalità CCX e verificare l'esatta versione CCX supportata dai client wireless (se sono compatibili con CCX), in modo da poter confermare se il metodo di sicurezza che si desidera utilizzare con CCKM può essere implementato.

Questa immagine wireless fornisce un esempio dei frame scambiati durante l'associazione iniziale quando si esegue la crittografia CCKM con TKIP e il metodo 802.1X/EAP-MSCHAPv2 con PEAPv0/EAP-MSCHAPv2. In pratica, lo scambio è lo stesso che avviene se viene eseguito WPA/TKIP con PEAPv0/EAP-MSCHAPv2, ma questa volta CCKM tra il client e l'infrastruttura viene negoziato in modo che utilizzino diversi metodi di cache e gerarchia di chiavi per eseguire il roaming protetto rapido quando il client deve eseguire il roaming:



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)

Di seguito è riportato un riepilogo dei messaggi di debug (con alcuni scambi EAP rimossi per ridurre l'output):

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

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

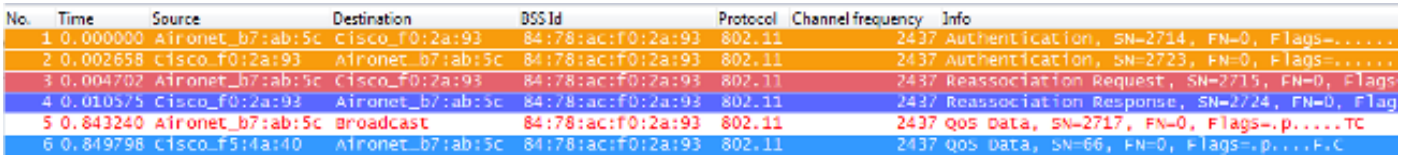
```
*apfMsConnTask_0: Jun 25 15:41:41.507: 00:40:96:b7:ab:5c
  Setting active key cache index 8 ---> 8
!--- This is the key cache index for this client, which is set temporally.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Con CCKM, l'associazione iniziale alla WLAN è simile alla normale WPA/WPA2, in cui un MSK (noto anche come NSK (Network Session Key)) viene derivato reciprocamente con il client e il server RADIUS. Questa chiave primaria viene inviata dal server al WLC dopo un'autenticazione riuscita e viene memorizzata nella cache come base per la derivazione di tutte le chiavi successive per la durata dell'associazione del client alla WLAN. Da qui, il WLC e il client derivano le informazioni di base che vengono utilizzate per il roaming veloce-sicuro basato su CCKM, questo passa attraverso un handshake a 4 vie simile a quello di WPA/WPA2, al fine di derivare le chiavi di crittografia unicast (PTK) e multicast/broadcast (GTK) con il primo AP.

La grande differenza si nota quando si utilizza il roaming. In questo caso, il client CCKM invia un singolo frame di richiesta di riassociazione all'AP/WLC (che include un MIC e un numero casuale

ad incremento sequenziale) e fornisce informazioni sufficienti (che includono il nuovo indirizzo MAC dell'AP **-BSSID-**) per derivare il nuovo PTK. Con questa richiesta di riassociazione, il WLC e la nuova AP hanno anche informazioni sufficienti per derivare la nuova PTK, quindi rispondono semplicemente con una risposta di riassociazione. Il client può continuare a trasmettere il traffico, come mostrato nell'immagine:



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

Di seguito è riportato un riepilogo dei debug WLC per questo evento di roaming:

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

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

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

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

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

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

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

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

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

*apfMsConnTask_2: Jun 25 15:43:33.751: 00:40:96:b7:ab:5c
  Creating a PKC PMKID Cache entry for station
  00:40:96:b7:ab:5c (RSN 0) on BSSID 84:78:ac:f0:2a:93
!--- The new PMKID cache entry is created for this new
      AP-to-client association.
```

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

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

Come mostrato in precedenza, il roaming viene eseguito senza frame di autenticazione EAP e con handshake a 4 vie, in quanto le nuove chiavi di crittografia sono ancora derivate, ma basate sullo schema di negoziazione CCKM. Questa operazione viene completata con i frame di riassociazione mobili e le informazioni precedentemente memorizzate nella cache dal client e dal WLC.

FlexConnect con CCKM

- L'autenticazione centrale è supportata. Ciò include la commutazione dei dati locale e centrale. Gli access point devono appartenere allo stesso gruppo FlexConnect.
- L'autenticazione locale Flex è supportata. In modalità connessa, la cache può essere distribuita dall'access point al controller e quindi agli altri access point nel gruppo FlexConnect.
- Èsupportata la modalità standalone. Se la cache è già presente nell'access point (a causa della distribuzione precedente), il roaming veloce funzionerà. La nuova autenticazione in modalità autonoma non supporta il roaming protetto veloce.

Vantaggi della tecnologia CCKM

- CCKM è il metodo di roaming più veloce e sicuro implementato principalmente sulle WLAN aziendali. Non è necessario che i client superino un handshake di gestione delle chiavi per ottenere nuove chiavi quando si verifica uno spostamento tra i punti di accesso e non sono più richiesti per eseguire un'autenticazione 802.1X/EAP completa con nuovi punti di accesso per tutta la durata del client su questa WLAN.
- CCKM supporta tutti i metodi di crittografia disponibili nello standard 802.11 (WEP, TKIP e AES), oltre ad alcuni metodi proprietari Cisco legacy ancora utilizzati sui client legacy.

Svantaggi con CCKM

- CCKM è un metodo proprietario di Cisco che limita l'implementazione e il supporto all'infrastruttura WLAN Cisco e ai client wireless CCX.
- CCX versione 5 non è ampiamente adottato, quindi CCKM con WPA2/AES non è supportato da molti client wireless CCX (principalmente perché la maggior parte di essi supporta già CCKM con WPA/TKIP, che è ancora molto sicuro).

Roaming sicuro e veloce con memorizzazione nella cache PMKID/Sticky Key Caching

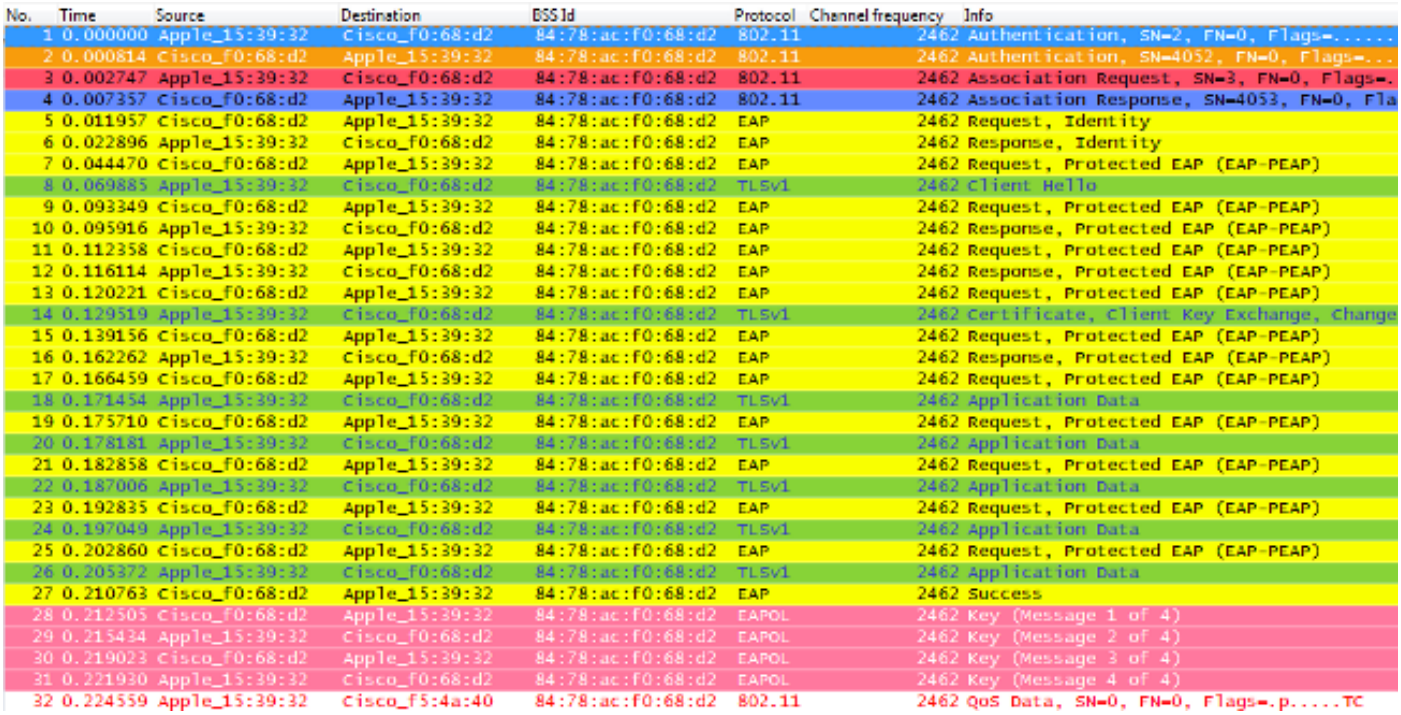
Pairwise think Key ID (PMKID) caching, o **Sticky Key Caching (SKC)**, è il primo metodo di roaming veloce e sicuro suggerito dallo standard IEEE 802.11 nell'ambito della modifica della sicurezza 802.11i, in cui lo scopo principale è quello di standardizzare un elevato livello di sicurezza per le WLAN. Questa tecnica di roaming a sicurezza rapida è stata aggiunta come metodo opzionale per i dispositivi WPA2 al fine di migliorare il roaming quando è stata implementata questa sicurezza.

Ciò è possibile perché, ogni volta che un client è completamente autenticato EAP, il client e il server di autenticazione derivano un MSK, che viene utilizzato per derivare la chiave PMK. Questo metodo viene utilizzato come base per l'handshake a 4 vie WPA2 per derivare la chiave di crittografia unicast finale (PTK) utilizzata per la sessione (finché il client non esegue il roaming a un altro punto di accesso o la sessione non scade); pertanto, questo metodo impedisce la fase di autenticazione EAP durante il roaming, in quanto riutilizza la chiave PMK originale memorizzata nella cache dal client e dall'punto di accesso. Il client deve solo eseguire l'handshake WPA2 a 4 vie per derivare nuove chiavi di crittografia.

Questo metodo non è ampiamente utilizzato come il metodo di roaming veloce sicuro standard 802.11 raccomandato principalmente per i seguenti motivi:

- Questo metodo è facoltativo e non è supportato da tutti i dispositivi WPA2, in quanto lo scopo della modifica 802.11i non riguarda il roaming veloce sicuro e l'IEEE ha già lavorato su un'altra modifica per standardizzare il roaming veloce sicuro per le WLAN (802.11r, che è trattata più avanti in questo documento).
- Questo metodo ha un grande limite nella sua implementazione: i client wireless possono eseguire il roaming veloce e sicuro solo quando ritornano a un access point in cui erano stati precedentemente autenticati/connessi.

Con questo metodo, l'associazione iniziale a qualsiasi access point è simile a una normale prima autenticazione alla WLAN, in cui l'intera autenticazione 802.1X/EAP contro il server di autenticazione e l'handshake a 4 vie per la generazione di chiavi devono avvenire prima che il client sia in grado di inviare frame di dati, come mostrato in questa immagine della schermata:



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

I debug rivelano lo stesso scambio di frame di autenticazione EAP degli altri metodi al momento dell'autenticazione iniziale sulla WLAN, con alcuni output aggiunti rispetto alle tecniche di memorizzazione nella cache delle chiavi usate qui. Questi output di debug vengono tagliati per mostrare principalmente le nuove informazioni, non l'intero scambio di frame EAP, perché fondamentalmente le stesse informazioni vengono scambiate ogni volta per l'autenticazione del client sul server di autenticazione. Poiché questa condizione è dimostrata finora e correlata ai frame di autenticazione EAP mostrati nelle immagini del pacchetto, la maggior parte dei messaggi EAP vengono rimossi dagli output del debug per semplicità:

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

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

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

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

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

*dot1xMsgTask: Jun 22 00:23:15.103: ec:85:2f:15:39:32
  Sending EAP-Request/Identity to mobile ec:85:2f:15:39:32
  (EAP Id 1)
```

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

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

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_2: Jun 22 00:23:15.284: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
```

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

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

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

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

Con questo metodo, l'access point e il client wireless memorizzano nella cache i PMK delle associazioni protette già stabilite. Pertanto, se il client wireless esegue il roaming in un nuovo access point a cui non è mai stato associato, il client deve eseguire di nuovo un'autenticazione EAP completa, come mostrato in questa immagine, quando il client esegue il roaming in un nuovo access point:

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

Tuttavia, se il client wireless torna a un punto di accesso in cui si è verificata un'associazione/autenticazione precedente, il client invia un frame di richiesta di riassociazione in cui sono elencati più PMKID, che informa l'access point delle PMK memorizzate nella cache da tutti gli access point in cui il client ha eseguito l'autenticazione in precedenza. Pertanto, poiché il client sta eseguendo il roaming a un access point che ha anche una chiave PMK memorizzata nella cache per questo client, non è necessario che il client riesegua l'autenticazione tramite EAP per derivare una nuova chiave PMK. Il client passa semplicemente attraverso l'handshake WPA2 a 4 vie per derivare le nuove chiavi di crittografia transitorie:

No.	Time	Source	Destination	BSS Id	Protocol	Channel frequency	Info
1	0.000000	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Authentication, SN=1506, FN=0, Flags=.....
2	0.002104	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	802.11	2462	Reassociation Request, SN=1134, FN=0, Flags
3	0.007239	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	802.11	2462	Reassociation Response, SN=1507, FN=0, Flag
4	0.014511	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 1 of 4)
5	0.019507	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 2 of 4)
6	0.023478	Cisco_f0:68:d2	Apple_15:39:32	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 3 of 4)
7	0.026743	Apple_15:39:32	Cisco_f0:68:d2	84:78:ac:f0:68:d2	EAPOL	2462	Key (Message 4 of 4)

> **Nota**: questa immagine non mostra il primo frame di autenticazione 802.11 Open System dal client, ma ciò non è dovuto al metodo implementato, in quanto questo frame è sempre richiesto. Il motivo è che questo frame specifico non viene acquisito dalla scheda di rete o dal software per l'immagine del pacchetto wireless utilizzato per annusare i frame over-the-air per questo esempio, ma viene lasciato così sull'esempio per scopi didattici. Tenere presente che questo può accadere quando si eseguono immagini di pacchetti via etere; alcuni frame possono mancare all'immagine, ma in realtà vengono scambiati tra il client e l'access point. In caso contrario, il roaming non inizierà mai da questo esempio.

Di seguito è riportato un riepilogo dei debug WLC per questo metodo di roaming rapido e sicuro:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

FlexConnect con memorizzazione nella cache PMKID/Sticky Key Caching

- Quando si utilizza questo metodo su una configurazione di FlexConnect, potrebbe funzionare e il comportamento potrebbe sembrare simile a quello spiegato in precedenza se si utilizza l'autenticazione centrale per tornare al WLC (con commutazione centrale o locale); tuttavia, questo metodo SKC non è supportato su FlexConnect.
- Questo metodo è supportato ufficialmente solo su CUWN con AP in modalità locale, non su FlexConnect o altre modalità.

Pro con memorizzazione nella cache PMKID/Sticky Key Caching

Questo metodo può essere implementato localmente da access point indipendenti dall'utente, senza la necessità di un dispositivo centralizzato per gestire le chiavi memorizzate nella cache.

Svantaggi con memorizzazione nella cache PMKID/Sticky Key Caching

- Come accennato in precedenza in questo documento, la limitazione principale di questo metodo è che il client può eseguire il roaming veloce e sicuro solo quando torna a un access point a cui era stato associato/autenticato in precedenza. In caso di roaming in un nuovo punto di accesso, il client deve completare di nuovo l'autenticazione EAP completa.
- Il client wireless e gli access point devono ricordare tutti i PMK derivati da ogni nuova autenticazione, quindi questa funzionalità è in genere limitata a una determinata quantità di PMK memorizzati nella cache. Poiché questo limite non è chiaramente definito dallo standard, i fornitori possono definire limiti diversi per le loro implementazioni SKC. Ad esempio, i Cisco WLAN Controller possono attualmente memorizzare nella cache i PMK di un client per un massimo di otto access point. Se un client esegue il roaming a più di otto access point per sessione, gli access point meno recenti vengono rimossi dall'elenco della cache per memorizzare le voci appena memorizzate nella cache.
- Questo metodo è facoltativo e non è ancora supportato da molti dispositivi WPA2, pertanto non è ampiamente adottato e implementato.
- Gli SKC non sono supportati quando si esegue il roaming tra controller, che si verifica quando ci si sposta tra punti di accesso gestiti da WLC diversi, anche se si trovano nello stesso gruppo di mobilità.

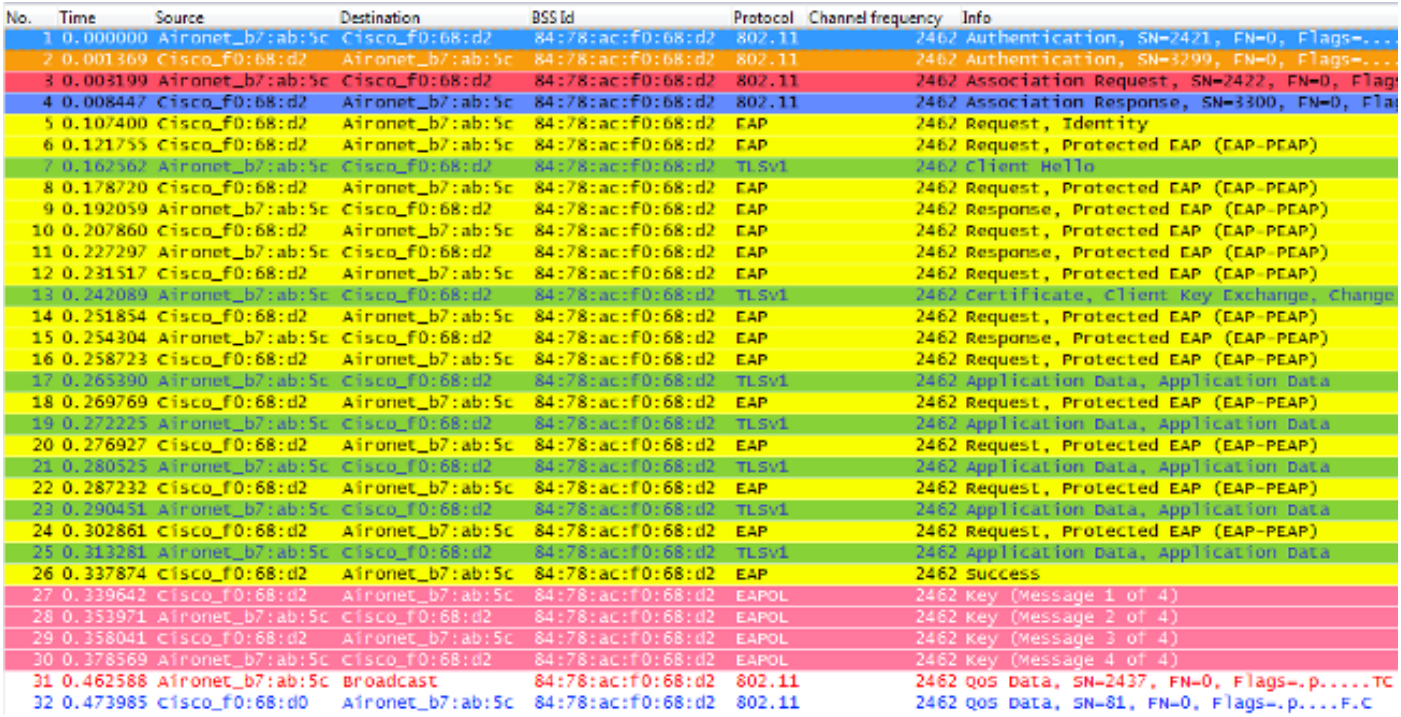
Roaming sicuro e rapido con memorizzazione nella cache delle chiavi opportunistica

La memorizzazione nella cache con chiave opportunistica (OKC, Opportunistic Key Caching), nota anche come memorizzazione nella cache con chiave proattiva (PKC, Proactive Key Caching) (questo termine viene spiegato in dettaglio in una nota successiva), è sostanzialmente un miglioramento del metodo di memorizzazione nella cache PMKID WPA2 descritto in precedenza, motivo per cui viene anche denominata memorizzazione nella cache PMKID proattiva/opportunistica. Pertanto, è importante notare che questo non è un metodo di roaming veloce-sicuro definito dallo standard 802.11 e non è supportato da molti dispositivi, ma proprio come la memorizzazione nella cache PMKID, funziona con WPA2-EAP.

Questa tecnica consente al client wireless e all'infrastruttura WLAN di memorizzare nella cache solo una chiave PMK per tutta la durata dell'associazione del client a questa WLAN (derivata dalla chiave MSK dopo l'autenticazione iniziale 802.1X/EAP con il server di autenticazione), anche in caso di roaming tra più access point, poiché tutti condividono la chiave PMK originale utilizzata come valore di inizializzazione in tutti gli handshake WPA2 a 4 vie. Questa operazione è ancora necessaria, come avviene negli SKC, per generare nuove chiavi di crittografia ogni volta che il client si riassocia agli access point. Affinché gli access point possano condividere questa chiave PMK originale della sessione client, devono essere tutti sottoposti a una sorta di controllo amministrativo, con un dispositivo centralizzato che memorizza nella cache e distribuisce la chiave PMK originale per tutti gli access point. Questa procedura è simile a quella del CUWN, in cui il WLC svolge questo lavoro per tutti i LAP sotto il suo controllo e utilizza i gruppi di mobilità per

gestire questo PMK tra più WLC; pertanto, si tratta di una limitazione per gli ambienti AP autonomi.

Con questo metodo, proprio come nella memorizzazione nella cache PMKID (SKC), l'associazione iniziale a qualsiasi access point è una prima autenticazione regolare alla WLAN, in cui è necessario completare l'intera autenticazione 802.1X/EAP sul server di autenticazione e l'handshake a 4 vie per la generazione di chiavi prima di poter inviare frame di dati. Di seguito è riportata un'immagine della schermata che illustra quanto segue:



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

Gli output del debug mostrano fondamentalmente lo stesso scambio di frame di autenticazione EAP degli altri metodi descritti in questo documento all'autenticazione iniziale sulla WLAN (come mostrato nelle immagini), insieme all'aggiunta di alcuni output che riguardano le tecniche di cache delle chiavi usate dal WLC qui. Anche questo output del comando debug è stato tagliato per visualizzare solo le informazioni importanti:

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

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

*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Received RSN IE with 0 PMKIDs from mobile
  00:40:96:b7:ab:5c
!--- Since this is an initial association, the Association
     Request comes without any PMKID.

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

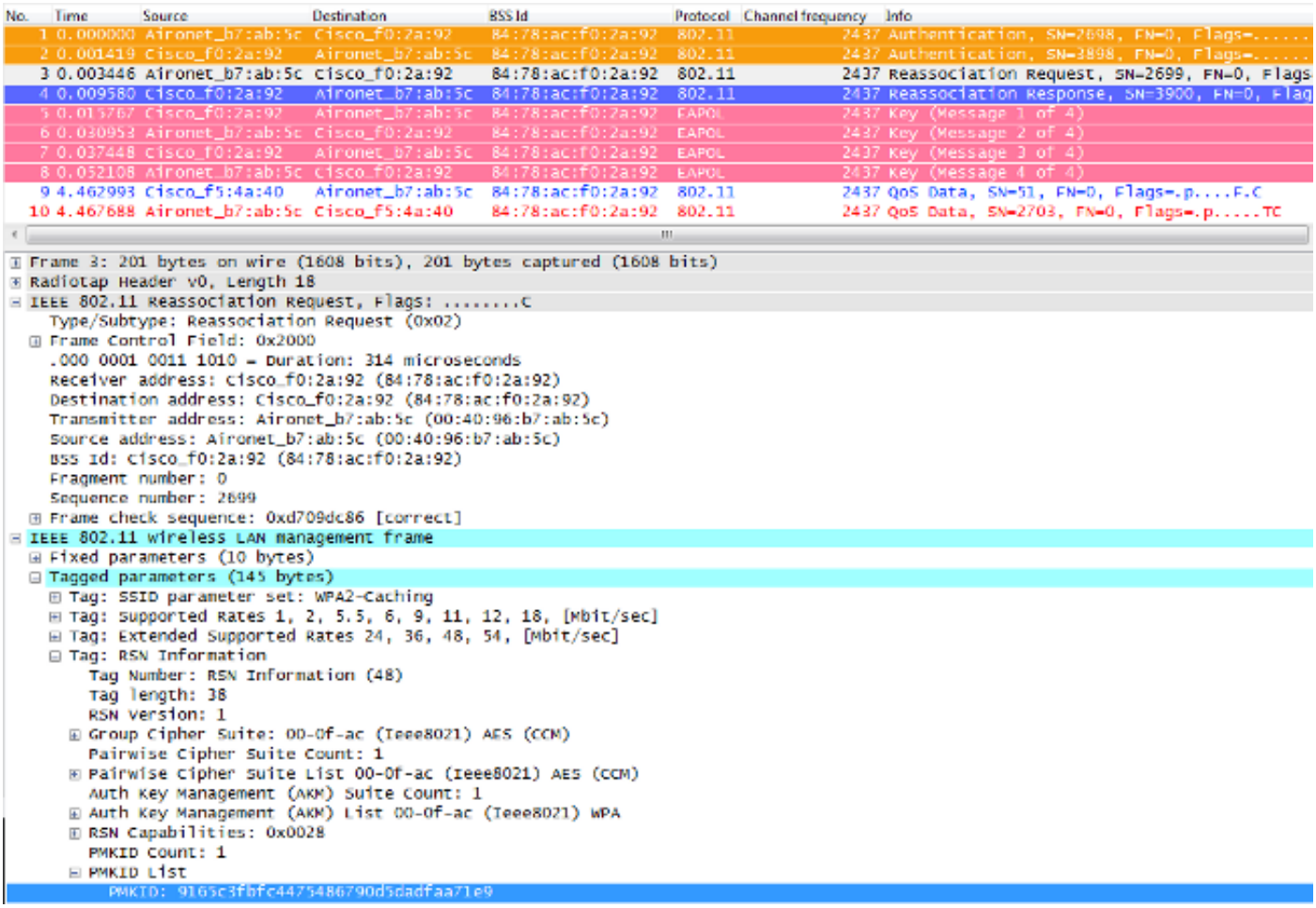
```
*apfMsConnTask_0: Jun 21 21:46:06.516: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:68:d2 (status 0) ApVapId 3 Slot
!--- The Association Response is sent to the client.

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

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_4: Jun 21 21:46:06.844: 00:40:96:b7:ab:5c
  PMK sent to mobility group
!--- The PMK cache entry for this client is shared with the
     WLCs on the mobility group.

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

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

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

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

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

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

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

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

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

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

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

Con questo metodo, il client wireless e il WLC (per tutti gli access point gestiti) memorizzano nella cache l'unica chiave PMK originale dell'associazione sicura stabilita inizialmente. In pratica, ogni volta che il client wireless si connette a un punto di accesso specifico, viene eseguito l'hashing di un PMKID in base all'indirizzo MAC del client, all'indirizzo MAC dell'punto di accesso (BSSID della WLAN) e alla chiave PMK derivata da tale punto di accesso. Pertanto, poiché OKC memorizza nella cache la stessa chiave PMK originale per tutti gli access point e per il client specifico, quando questo client viene associato a un altro access point, l'unico valore che cambia per eseguire l'hashing del nuovo PMKID è il nuovo indirizzo MAC dell'access point.

Quando il client avvia il roaming verso un nuovo access point e invia il frame di richiesta di riassociazione, aggiunge il PMKID sull'elemento di informazioni RSN WPA2 se desidera informare l'access point che una chiave PMK memorizzata nella cache è utilizzata per il roaming a sicurezza rapida. Conosce già l'indirizzo MAC del BSSID (AP) per il punto in cui esegue il roaming, quindi il client esegue semplicemente l'hashing del nuovo PMKID utilizzato in questa richiesta di riassociazione. Quando l'access point riceve questa richiesta dal client, esegue anche l'hashing del PMKID con i valori già disponibili (la chiave PMK memorizzata nella cache, l'indirizzo MAC del client e il relativo indirizzo MAC dell'access point) e risponde con la risposta di riassociazione riuscita che conferma la corrispondenza dei PMKID. La chiave PMK memorizzata nella cache può

essere utilizzata come valore di inizializzazione per l'avvio di un handshake WPA2 a 4 vie al fine di derivare le nuove chiavi di crittografia (e ignorare EAP):



In questa immagine, il frame di richiesta di riassociazione del client viene selezionato ed espanso in modo da visualizzare ulteriori dettagli del frame. Informazioni sull'indirizzo MAC e anche l'elemento di informazioni RSN (Robust Security Network, come da 802.11i - WPA2), in cui vengono visualizzate le informazioni sulle impostazioni WPA2 utilizzate per questa associazione (evidenziato il PMKID ottenuto dalla formula con hash).

Di seguito è riportato un riepilogo dei debug WLC per questo metodo di roaming veloce e sicuro con OKC:

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

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

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

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

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

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

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

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

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

*apfMsConnTask_2: Jun 21 21:48:50.564: 00:40:96:b7:ab:5c
  Sending Assoc Response to station on BSSID 84:78:ac:f0:2a:92
```

```
  (status 0) ApVapId 3 Slot
!--- The Reassociation response is sent to the client, which
     validates the fast-roam with OKC.

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

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

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

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

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

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

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

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

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

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

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

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

Come mostrato all'inizio dei debug, il PMKID deve essere calcolato dopo la richiesta di riassociazione del client. Questa operazione è necessaria per convalidare PMKID e confermare che la chiave PMK memorizzata nella cache viene utilizzata con l'handshake a 4 vie WPA2 per derivare le chiavi di crittografia e completare il roaming a protezione rapida. Non confondere le voci CCKM sui debug; questo non viene usato per eseguire CCKM, ma per OKC, come spiegato in precedenza. CCKM è semplicemente un nome utilizzato dal WLC per tali output, ad esempio il nome di una funzione che gestisce i valori per calcolare il PMKID.

FlexConnect con memorizzazione nella cache delle chiavi opportunistica

- L'autenticazione centrale è supportata. Ciò include la commutazione dei dati locale e centrale. Se l'access point fa parte dello stesso gruppo FlexConnect, il roaming a sicurezza rapida viene effettuato dall'access point, altrimenti il roaming a sicurezza rapida viene effettuato dal controller.
  Nota: questa configurazione può funzionare se gli access point non si trovano nello stesso gruppo FlexConnect, ma non è consigliata né supportata.
- L'autenticazione locale Flex è supportata. In modalità connessa, la cache può essere distribuita dall'access point al controller e quindi agli altri access point nel gruppo FlexConnect.
- Èsupportata la modalità standalone. Se la cache è già presente nell'access point (a causa della distribuzione precedente), il roaming protetto rapido funzionerà. La nuova autenticazione in modalità autonoma non supporta il roaming protetto veloce.

Vantaggi della memorizzazione nella cache della chiave opportunistica

- Il client wireless e l'infrastruttura WLAN non devono ricordare più PMKID, ma è sufficiente memorizzare nella cache la chiave PMK originale dall'autenticazione iniziale alla rete WLAN. Quindi, è necessario rieseguire il reset del PMKID appropriato (utilizzato nella richiesta di riassociazione) richiesto per ogni associazione protetta dell'access point per convalidare il roaming veloce e sicuro.
- In questo caso, il client wireless esegue il roaming sicuro e rapido verso un nuovo access point sulla stessa WLAN/SSID, anche se non è mai stato associato a tale access point (non è il caso degli SKC). Finché il client esegue l'autenticazione 802.1X/EAP iniziale con un access point gestito dalla distribuzione centralizzata che gestisce la cache PMK per tutti gli access point in cui il client effettua il roaming, non sono necessarie ulteriori autenticazioni complete per il resto della durata del client su questa WLAN.

Svantaggi con memorizzazione nella cache delle chiavi opportunistica

- Questo metodo viene implementato solo in un ambiente centralizzato in cui tutti gli access point sono sottoposti a un tipo di controllo amministrativo (ad esempio un controller WLAN) responsabile della memorizzazione nella cache e della condivisione della chiave PMK originale dalla sessione client. Pertanto, si tratta di una limitazione relativa agli ambienti AP autonomi.
- Le tecniche applicate in questo metodo non sono suggerite né descritte nello standard 802.11, quindi il supporto varia notevolmente da un dispositivo all'altro. Tuttavia, questo è ancora il metodo che è stato più adottato durante l'attesa per 802.11r.

Nota sul termine "memorizzazione nella cache attiva dei tasti"

La memorizzazione nella cache con chiave proattiva (o PKC) è nota come OKC (Opportunistic Key Caching) e i due termini vengono utilizzati in modo intercambiabile quando descrivono lo stesso metodo qui descritto. Tuttavia, questo era solo un termine che è stato utilizzato da Airspace nel 2001 per un vecchio metodo di memorizzazione nella cache delle chiavi, che è stato poi utilizzato dallo standard 802.11i come base per la "preautenticazione" (un altro metodo Fast Secure Roaming brevemente spiegato di seguito). PKC non è Preautenticazione o OKC

(Opportunistic Key Caching), ma quando si sente o si legge di PKC, il riferimento è fondamentalmente a OKC e non alla Preautenticazione.

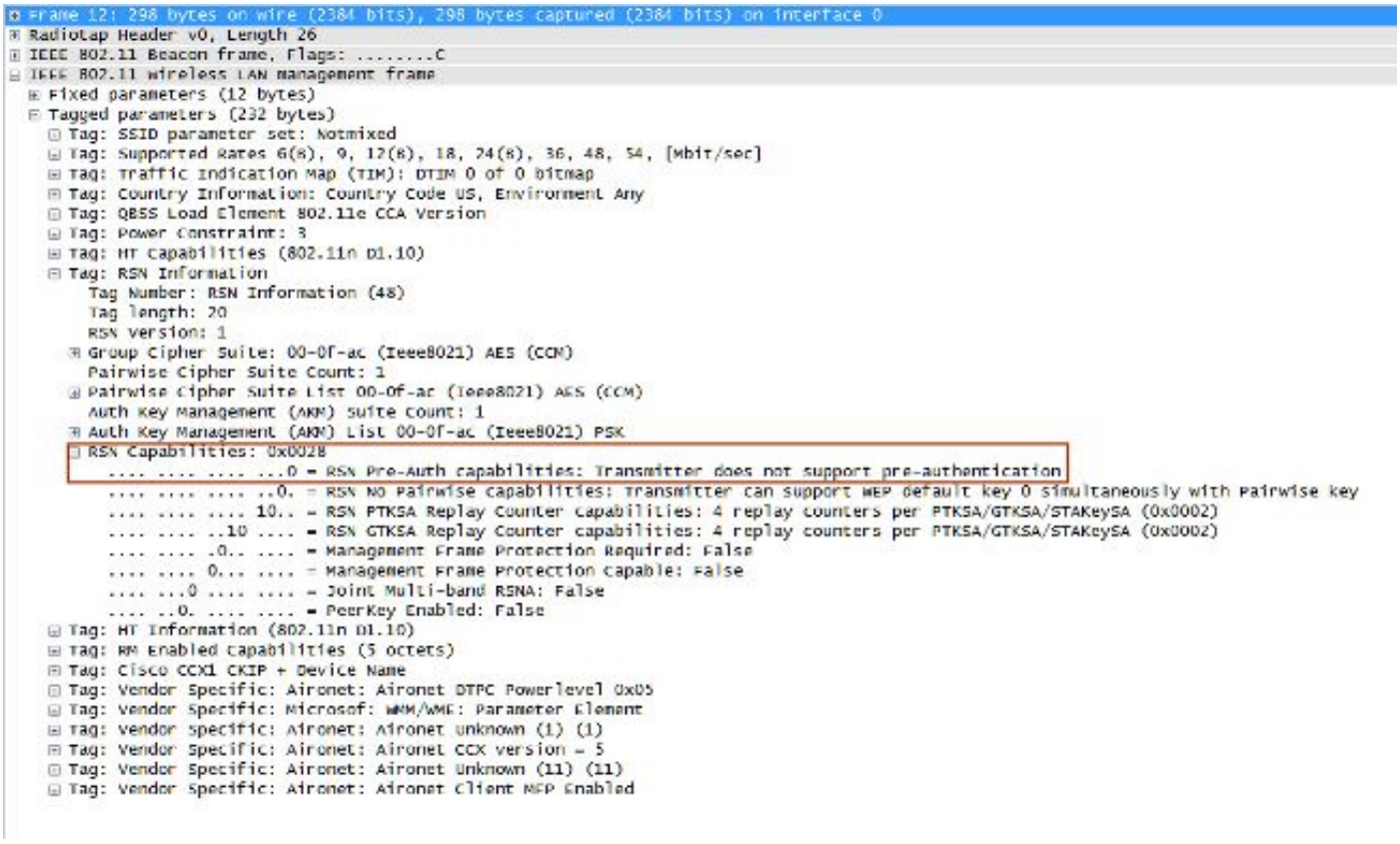
Roaming veloce e sicuro con preautenticazione

Questo metodo è suggerito anche dallo standard IEEE 802.11 nella modifica della sicurezza 802.11i, quindi funziona anche con WPA2, ma è l'unico metodo Fast Secure Roaming non supportato dall'infrastruttura WLAN Cisco. Per questo motivo, viene spiegato solo brevemente qui e senza output.

Con la preautenticazione, i client wireless possono eseguire l'autenticazione con più access point alla volta mentre sono associati all'access point corrente. In questo caso, il client invia i frame di autenticazione EAP all'access point corrente a cui è connesso, ma è destinato agli altri access point a cui il client desidera eseguire la preautenticazione (access point adiacenti che potrebbero essere candidati al roaming). Il punto di accesso corrente invia questi frame ai punti di accesso di destinazione sul sistema di distribuzione. Il nuovo punto di accesso esegue l'autenticazione completa sul server RADIUS per questo client, quindi viene completato un intero nuovo handshake di autenticazione EAP e il nuovo punto di accesso funge da autenticatore.

L'idea è quella di eseguire l'autenticazione e derivare la chiave PMK con gli access point adiacenti prima che il client esegua il roaming verso di essi, in modo che quando è il momento di eseguire il roaming, il client sia già autenticato e con una chiave PMK già memorizzata nella cache per questa nuova associazione sicura da punto di accesso a client, quindi devono solo eseguire l'handshake a 4 vie ed eseguire un roaming veloce dopo che il client ha inviato la richiesta iniziale di riassociazione.

Di seguito è riportata un'immagine di un beacon AP che mostra il campo IE RSN che annuncia il supporto per la preautenticazione (questa immagine è di un Cisco AP, dove è confermato che la preautenticazione non è supportata):



Pro con preautenticazione

Esiste una chiave PMK per ogni associazione protetta tra punti di accesso e client, che può essere considerata un vantaggio in termini di sicurezza nel caso in cui un punto di accesso venga compromesso e le chiavi vengano rubate (non è possibile utilizzarle con altri punti di accesso). Tuttavia, questo vantaggio in termini di sicurezza viene gestito dall'infrastruttura WLAN in modi diversi rispetto ad altri metodi.

Svantaggi con preautenticazione

- Poiché esiste una chiave PMK per punto di accesso, i client hanno un limite sul numero di punti di accesso che possono essere preautenticati.
- Ogni volta che un client esegue la preautenticazione con un nuovo punto di accesso, si verifica uno scambio di autenticazione EAP completo, che comporta un carico maggiore sulla rete e sul server di autenticazione.
- La maggior parte dei client wireless non supporta questo metodo, in quanto non è mai stato adottato in modo completo (OKC è stato più adottato).

Roaming Fast-Secure con 802.11r

La tecnica di roaming a sicurezza rapida basata sulla modifica 802.11r (ufficialmente denominata **Fast BSS Transition** dallo standard 802.11 e nota come **FT**) è il primo metodo ufficialmente ratificato (nel 2008) dall'IEEE per lo standard 802.11 come soluzione per eseguire transizioni veloci tra punti di accesso (Basic Service Set o BSS), che definisce chiaramente la gerarchia di chiavi utilizzata quando si gestiscono e memorizzano le chiavi su una WLAN. Tuttavia, la sua adozione è stata lenta, principalmente a causa delle altre soluzioni già disponibili quando erano effettivamente necessarie transizioni veloci, come con le implementazioni VoWLAN quando utilizzate con uno dei metodi descritti in precedenza in questo documento. Attualmente solo pochi dispositivi supportano alcune delle opzioni FT (entro il 2013).

Questa tecnica è più complessa da spiegare rispetto agli altri metodi, in quanto introduce nuovi concetti e più livelli di PMK che vengono memorizzati in dispositivi diversi (ogni dispositivo con un ruolo diverso) e fornisce ancora più opzioni per il roaming veloce e sicuro. Di conseguenza, viene fornito un breve riepilogo su questo metodo e sul modo in cui viene implementato con ciascuna opzione disponibile.

802.11r è diverso da SKC e OKC, principalmente per i seguenti motivi:

- I messaggi di handshake (ad esempio lo scambio di PMKID, ANonce ed SNonce) si verificano nei frame di autenticazione 802.11 o nei frame di azione anziché nei frame di riassociazione. A differenza dei metodi di memorizzazione nella cache PMKID, viene evitata la fase di handshake a 4 vie separata, che viene eseguita dopo lo scambio di messaggi di (ri)associazione. L'handshake della chiave con il nuovo access point inizia prima che il client esegua il roaming o la riassociazione completa con il nuovo access point.
- Fornisce due metodi per l'handshake con roaming veloce: tramite AIR e tramite il sistema di distribuzione (DS).
- 802.11r ha più livelli di gerarchia di chiavi.
- Poiché questo protocollo evita l'handshake a 4 vie per la gestione delle chiavi quando un client esegue il roaming (genera nuove chiavi di crittografia -PTK e GTK- senza la necessità di

questo handshake), può essere applicato anche alle impostazioni WPA2 con una chiave PSK e non solo quando per l'autenticazione viene utilizzato 802.1X/EAP. Ciò accelera ulteriormente il roaming per queste configurazioni, in cui non si verificano scambi di handshake EAP o a 4 vie.

Con questo metodo, il client wireless esegue solo un'autenticazione iniziale sull'infrastruttura WLAN quando viene stabilita una connessione al primo access point ed esegue il roaming sicuro e rapido durante il roaming tra access point dello stesso dominio di mobilità FT.

Questo è uno dei nuovi concetti, che in pratica si riferisce ai punti di accesso che utilizzano lo stesso SSID (noto come Extended Service Set o ESS) e gestiscono le stesse chiavi FT. Ciò è simile agli altri metodi illustrati finora. Il modo in cui gli access point gestiscono le chiavi di dominio per la mobilità FT si basa generalmente su una configurazione centralizzata, come il WLC o i gruppi di mobilità; tuttavia, questo metodo può essere implementato anche in ambienti AP autonomi.

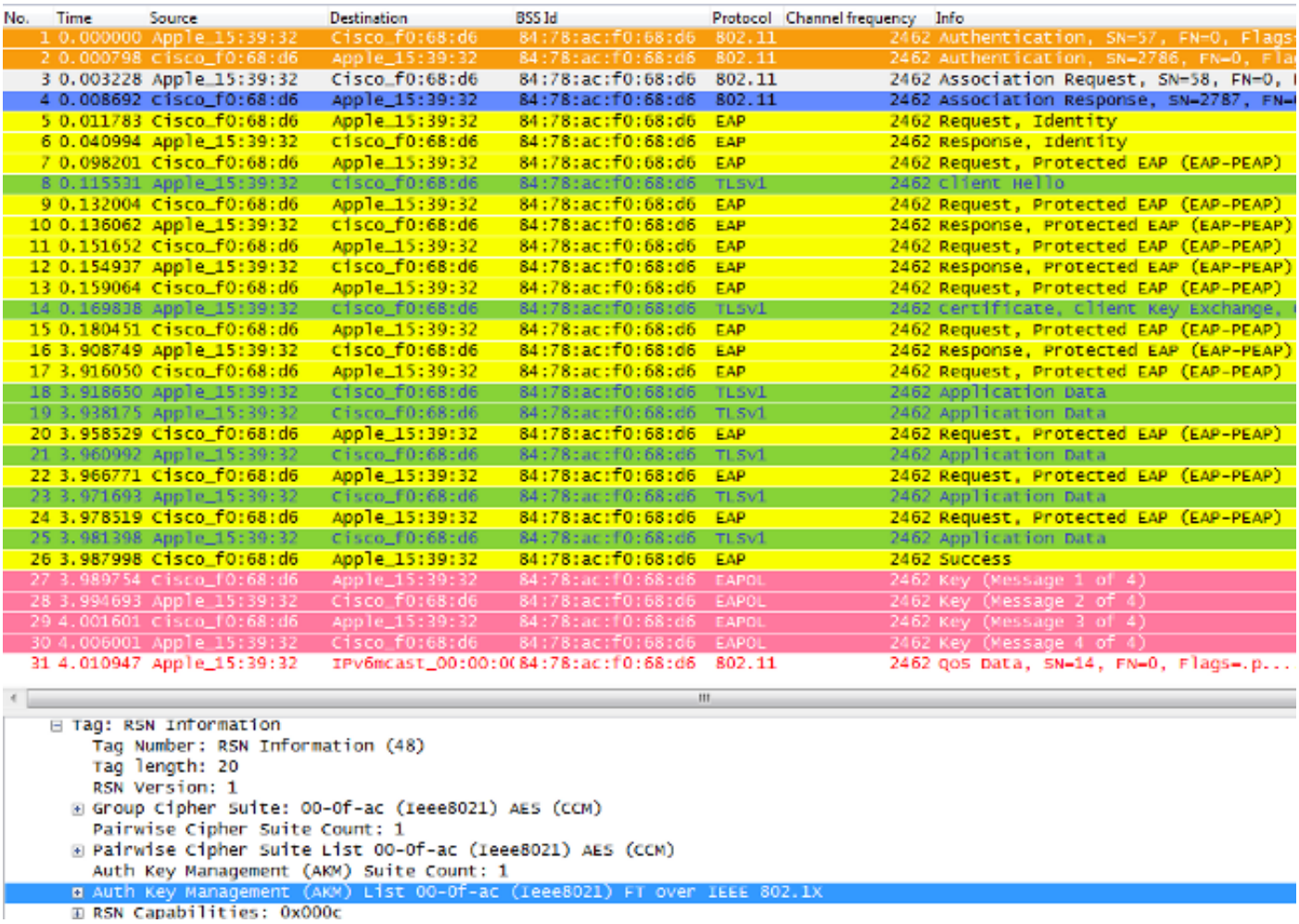
Di seguito è riportato un riepilogo della gerarchia delle chiavi:

- Un MSK è ancora derivato sul supplicant del client e sul server di autenticazione dalla fase iniziale di autenticazione 802.1X/EAP (trasferita dal server di autenticazione all'autenticatore (WLC) una volta completata l'autenticazione). Questo MSK, come negli altri metodi, viene utilizzato come seme per la gerarchia di chiavi FT. Quando si utilizza WPA2-PSK anziché un metodo di autenticazione EAP, PSK è fondamentalmente questo MSK.
- Una chiave master Pairwise R0 (PMK-R0) è derivata dalla chiave MSK, che è la chiave di primo livello della gerarchia di chiavi FT. I portachiavi per il PMK-R0 sono il WLC e il client.
- Una chiave di secondo livello, denominata PMK-R1 (Pairwise Master Key R1), è derivata dal PMK-R0 e i portachiavi sono il client e gli access point gestiti dal WLC che contiene il PMK-R0.
- La chiave di terzo e ultimo livello della gerarchia di chiavi FT è PTK, che è la chiave finale utilizzata per cifrare i frame di dati unicast 802.11 (simile agli altri metodi che usano WPA/TKIP o WPA2/AES). Questo PTK è derivato su FT dal PMK-R1 e i portachiavi sono il client e gli AP gestiti dal WLC.

  Nota: a seconda del fornitore della WLAN e delle impostazioni di implementazione (ad esempio, i punti di accesso autonomi, FlexConnect o Mesh), l'infrastruttura WLAN può trasferire e gestire le chiavi in modo diverso. È anche possibile modificare i ruoli dei detentori di chiavi, ma poiché questo argomento esula dall'ambito del presente documento, gli esempi basati sul riepilogo della gerarchia di chiavi fornito in precedenza costituiscono l'argomento successivo. Le differenze non sono in realtà così rilevanti per comprendere il processo, a meno che non sia effettivamente necessario analizzare in modo approfondito i dispositivi dell'infrastruttura (e il relativo codice) per individuare un problema software.

Transizione rapida BSS over-the-air

Con questo metodo, la prima associazione a qualsiasi access point è una regolare prima autenticazione alla WLAN, in cui l'intera autenticazione 802.1X/EAP sul server di autenticazione e l'handshake a 4 vie per la generazione di chiavi devono verificarsi prima dell'invio dei frame di dati, come mostrato nella seguente immagine della schermata:



Le principali differenze sono le seguenti:

- La negoziazione della gestione delle chiavi di autenticazione è leggermente diversa dalla normale negoziazione WPA/WPA2, pertanto vengono utilizzate alcune informazioni aggiuntive per eseguire questa negoziazione quando si verifica l'associazione a un'infrastruttura WLAN che supporta FT. Come mostrato nell'immagine, viene selezionato il frame di richiesta di associazione dal client e viene evidenziato il campo AKM dell'elemento di informazione RNS per indicare che il client desidera eseguire FT su 802.1X/EAP.
- Viene inoltre mostrato l'elemento di informazioni sul dominio di mobilità (parte di FT), dove il campo **Capacità e criterio FT** indica se la transizione BSS veloce è completata over-the-air o over-the-DS durante il roaming veloce (questo indica Over-the-air in questa immagine).
- Viene inoltre aggiunto un altro elemento di informazione (Fast BSS Transition o FT IE, descritto più avanti in questo documento) con le informazioni necessarie per eseguire la sequenza di autenticazione FT durante il roaming FT.
- La generazione di chiavi è diversa a causa della gerarchia delle chiavi, quindi anche se l'handshake a 4 vie FT è simile all'handshake a 4 vie WPA/WPA2, in realtà è leggermente diverso nel contenuto.

I debug mostrano fondamentalmente lo stesso scambio di frame di autenticazione EAP degli altri metodi dopo l'autenticazione iniziale sulla WLAN (come osservato dalle immagini), ma vengono aggiunti alcuni output che riguardano le tecniche di cache delle chiavi usate dal WLC; di conseguenza, questo output di debug viene tagliato per mostrare solo le informazioni rilevanti:

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

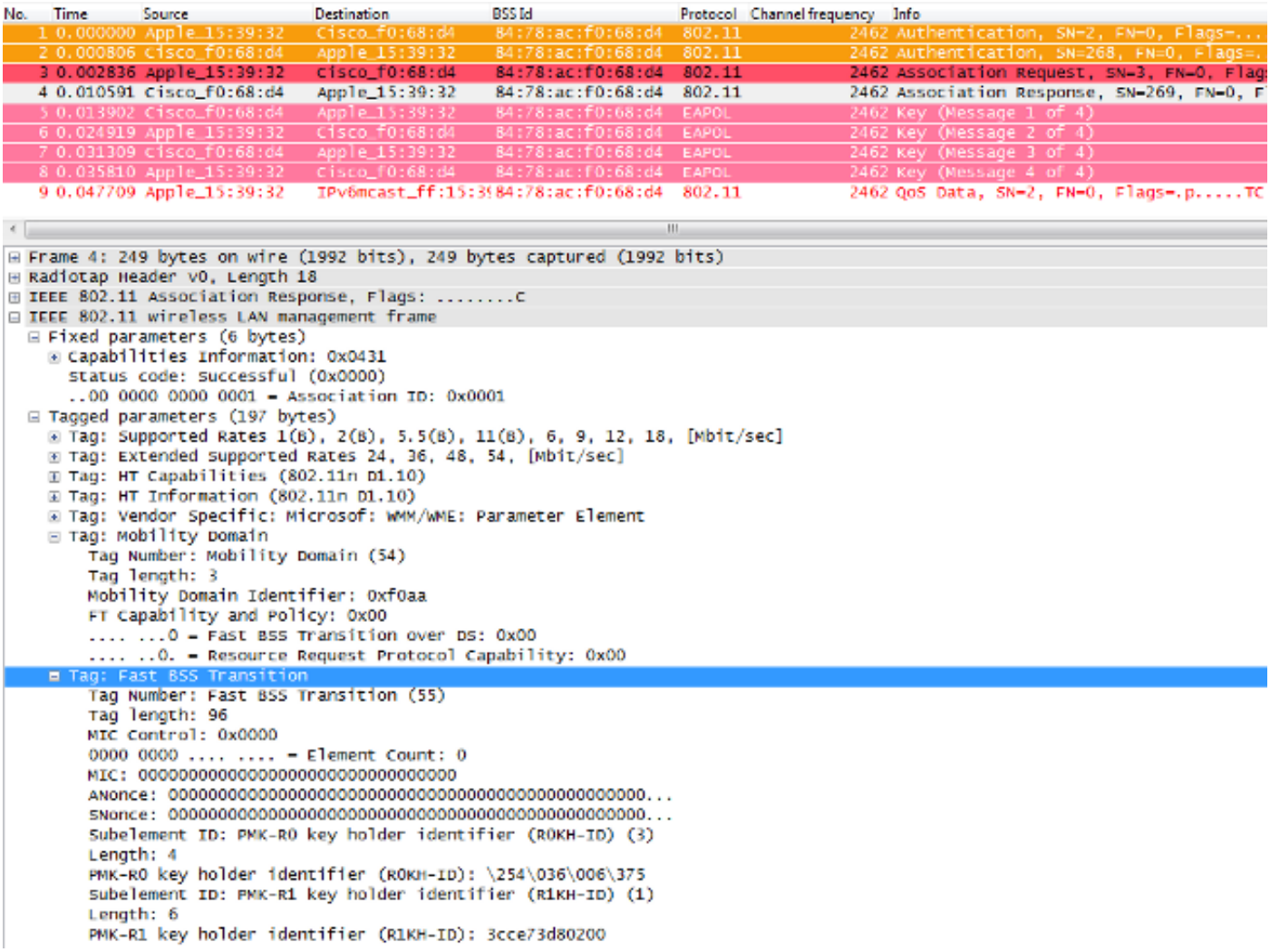
```
  84:78:ac:f0:68:d6
!--- This is the Association request from the client.

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

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

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.479: ec:85:2f:15:39:32
  Received EAP Response from mobile ec:85:2f:15:39:32
  (EAP Id 2, EAP Type 25)

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.627: ec:85:2f:15:39:32
```

```
  Processing Access-Accept for mobile ec:85:2f:15:39:32
!--- The client is validated/authenticated by the RADIUS Server.

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

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

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

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

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

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

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

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

*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
*Dot1x_NW_MsgTask_2: Jun 27 19:25:23.639: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from
  mobile ec:85:2f:15:39:32
!--- Message-2 of the FT 4-Way handshake is received
      successfully from the client.
```

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

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

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

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

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

> **Nota**: per eseguire il debug di questo metodo e raggiungere gli output aggiuntivi 802.11r/FT mostrati qui, viene abilitato un ulteriore debug insieme al **client di debug**, ossia gli **eventi di debug ft abilitati**.

Di seguito sono riportate le immagini e i debug di un'associazione iniziale alla WLAN quando si esegue FT con WPA2-PSK (anziché un metodo 802.1X/EAP), in cui viene selezionato il frame Association Response dall'access point per visualizzare l'elemento Fast BSS Transition Information (evidenziato). Vengono inoltre visualizzate alcune delle informazioni principali necessarie per eseguire l'handshake a 4 vie FT:



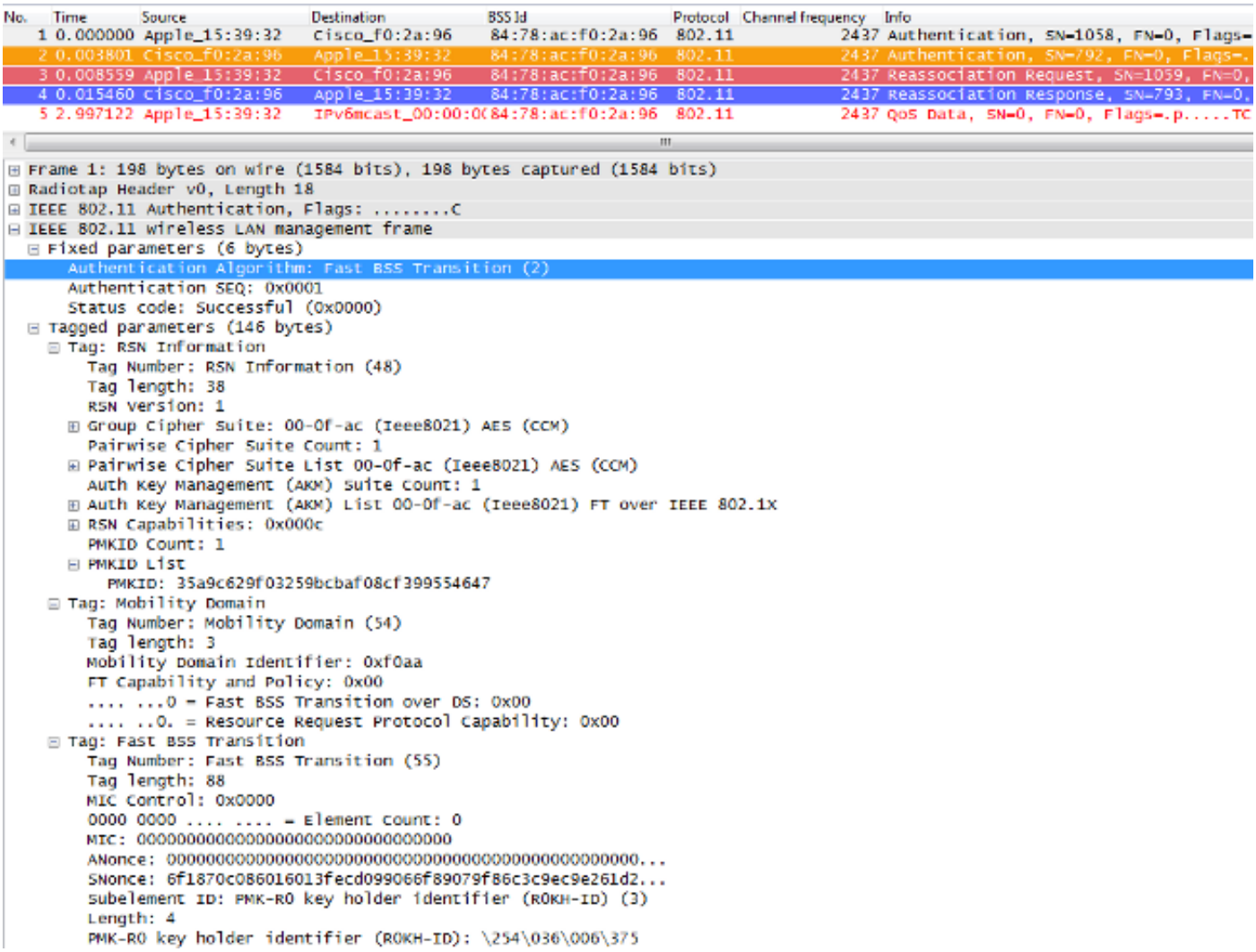

```
*apfMsConnTask_0: Jun 27 19:29:09.136: ec:85:2f:15:39:32
  Association received from mobile on BSSID
  84:78:ac:f0:68:d4

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 20 for mobile
  ec:85:2f:15:39:32

*apfMsConnTask_0: Jun 27 19:29:09.137: Sending assoc-resp
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:68:d0-00
  thread:144be808

*apfMsConnTask_0: Jun 27 19:29:09.137: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in Initial
  assoc Resp to mobile

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Sending R0KH-ID as:-84.30.6.-3

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
  Sending R1KH-ID as 3c:ce:73:d8:02:00

*apfMsConnTask_0: Jun 27 19:29:09.137: ec:85:2f:15:39:32
```

```
  Including FT IE (length 98) in Initial Assoc Resp to mobile

*apfMsConnTask_0: Jun 27 19:29:09.138: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID 84:78:ac:f0:68:d4
  (status 0) ApVapId 5 Slot 0

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Creating a PKC PMKID Cache entry for station
  ec:85:2f:15:39:32 (RSN 2)

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Resetting MSCB PMK Cache Entry 0 for station
  ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 8

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Setting active key cache index 8 ---> 0

*dot1xMsgTask: Jun 27 19:29:09.141: ec:85:2f:15:39:32
  Adding BSSID 84:78:ac:f0:68:d4 to PMKID cache at
  index 0 for station ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: New PMKID: (16)

*dot1xMsgTask: Jun 27 19:29:09.142:
  [0000] 17 4b 17 5c ed 5f c7 1d 66 39 e9 5d 3a 63 69 e7

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Creating global PMK cache for this TGr client

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Created PMK Cache Entry for TGr AKM:PSK
  ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  R0KH-ID:172.30.6.253   R1KH-ID:3c:ce:73:d8:02:00
  MSK Len:48 pmkValidTime:1813

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Initiating RSN PSK to mobile ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: ec:85:2f:15:39:32
  Found an cache entry for BSSID 84:78:ac:f0:68:d4 in
  PMKID cache at index 0 of station ec:85:2f:15:39:32

*dot1xMsgTask: Jun 27 19:29:09.142: Including PMKID
  in M1  (16)

*dot1xMsgTask: Jun 27 19:29:09.142:
  [0000] 17 4b 17 5c ed 5f c7 1d 66 39 e9 5d 3a 63 69 e7

*dot1xMsgTask: Jun 27 19:29:09.143: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32
  state INITPMK (message 1), replay counter
  00.00.00.00.00.00.00.00

*apfMsConnTask_0: Jun 27 19:29:09.144: ec:85:2f:15:39:32
  Got action frame from this client.

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.152: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32
```

```
*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Received EAPOL-key in PTK_START state (message 2) from
  mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Calculating PMKR0Name

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: Adding MDIE,
  ID is:0xaaf0

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Adding TIE for reassociation deadtime:20000 milliseconds

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.153: ec:85:2f:15:39:32
  Adding TIE for R0Key-Data valid time :1813

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.154: ec:85:2f:15:39:32
  Sending EAPOL-Key Message to mobile ec:85:2f:15:39:32 state
  PTKINITNEGOTIATING (message 3), replay counter
  00.00.00.00.00.00.00.01

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.163: ec:85:2f:15:39:32
  Received EAPOL-Key from mobile ec:85:2f:15:39:32

*Dot1x_NW_MsgTask_2: Jun 27 19:29:09.163: ec:85:2f:15:39:32
  Received EAPOL-key in PTKINITNEGOTIATING state (message 4)
  from mobile ec:85:2f:15:39:32
```

Con lo standard 802.11r, l'associazione iniziale alla WLAN è la base usata per derivare le chiavi di base usate da questa tecnica, proprio come negli altri metodi di roaming veloci e sicuri. Le principali differenze si verificano quando il client inizia a effettuare il roaming; FT non solo evita 802.1X/EAP quando questo viene utilizzato, ma in realtà esegue un metodo di roaming più efficiente che combina i frame iniziali 802.11 di autenticazione e riassociazione del sistema aperto (che sono sempre utilizzati e richiesti quando il roaming tra AP) al fine di scambiare le informazioni FT e derivare nuove chiavi di crittografia dinamica al posto dell'handshake a 4 vie.

L'immagine seguente mostra i frame scambiati quando viene eseguita una transizione BSS veloce via etere con sicurezza 802.1X/EAP. Viene selezionato il frame di autenticazione del sistema aperto dal client all'access point per visualizzare gli elementi di informazione del protocollo FTP necessari per avviare la negoziazione della chiave FT. Questa opzione viene utilizzata per derivare il nuovo PTK con il nuovo AP (basato sul PMK-R1). Il campo che mostra l'algoritmo di autenticazione è evidenziato per mostrare che questo client non esegue una semplice autenticazione di sistema aperto, ma una transizione BSS veloce:

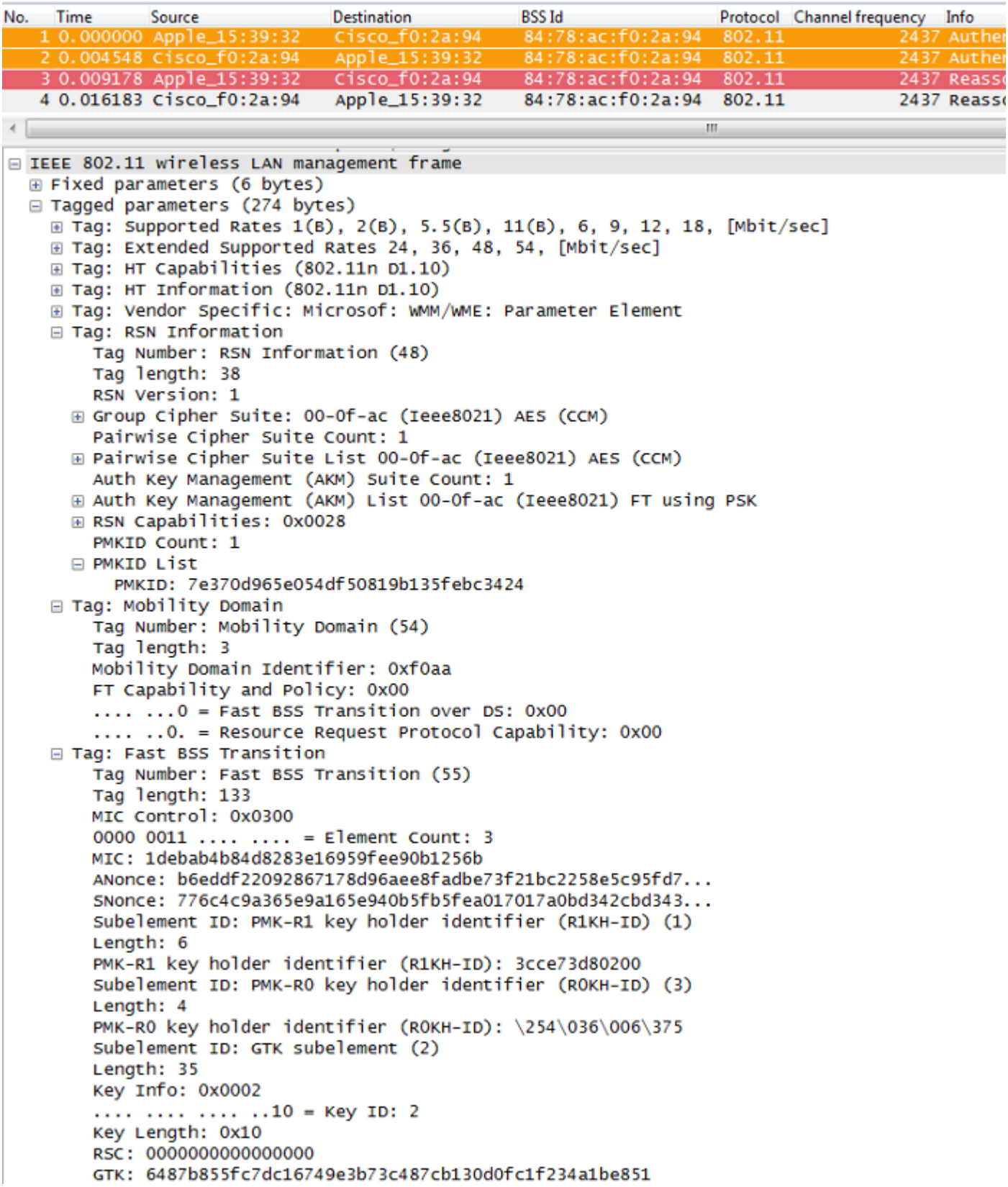


Di seguito sono riportati gli output di debug del WLC quando si verifica questo evento di roaming FTP con 802.1X/EAP:

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

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

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

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

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

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

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

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

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

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

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

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

Di seguito è riportata un'immagine che mostra una transizione BSS rapida via etere con sicurezza WPA2-PSK, in cui viene selezionato il frame di risposta di riassociazione finale dall'access point al client per visualizzare ulteriori dettagli su questo scambio di file FT:



Di seguito sono riportati gli output di debug quando questo evento di roaming FTP si verifica con PSK, simili a quelli quando si utilizza 802.1X/EAP:

```
*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Doing preauth for this client over the Air

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Doing local roaming for destination address
```

```
  84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Got 1 AKMs in RSNIE

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  RSNIE AKM matches with PMK cache entry :0x4

*apfMsConnTask_2: Jun 27 19:29:29.854: ec:85:2f:15:39:32
  Created a new preauth entry for AP:84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.854: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_2: Jun 27 19:29:29.867: Processing assoc-req
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38

*apfMsConnTask_2: Jun 27 19:29:29.867: ec:85:2f:15:39:32
  Reassociation received from mobile on BSSID
  84:78:ac:f0:2a:94

*apfMsConnTask_2: Jun 27 19:29:29.868: ec:85:2f:15:39:32
  Marking this mobile as TGr capable.

*apfMsConnTask_2: Jun 27 19:29:29.868: ec:85:2f:15:39:32
  Processing RSN IE type 48, length 38 for mobile
  ec:85:2f:15:39:32

*apfMsConnTask_2: Jun 27 19:29:29.869: ec:85:2f:15:39:32
  Roaming succeed for this client.

*apfMsConnTask_2: Jun 27 19:29:29.869: Sending assoc-resp
  station:ec:85:2f:15:39:32 AP:84:78:ac:f0:2a:90-00
  thread:144bef38

*apfMsConnTask_2: Jun 27 19:29:29.869: Adding MDIE,
  ID is:0xaaf0

*apfMsConnTask_2: Jun 27 19:29:29.869: ec:85:2f:15:39:32
  Including FT Mobility Domain IE (length 5) in
  reassociation assoc Resp to mobile

*apfMsConnTask_2: Jun 27 19:29:29.870: ec:85:2f:15:39:32
  Sending Assoc Response to station on BSSID
  84:78:ac:f0:2a:94 (status 0) ApVapId 5 Slot 0

*dot1xMsgTask: Jun 27 19:29:29.874: ec:85:2f:15:39:32
  Finishing FT roaming for mobile ec:85:2f:15:39:32
```

Come mostrato nell'immagine, una volta negoziata la transizione Fast BSS all'associazione iniziale alla WLAN, i quattro frame utilizzati e richiesti per il roaming (autenticazione del sistema aperto dal client, autenticazione del sistema aperto dall'access point, richiesta di riassociazione e risposta di riassociazione) vengono fondamentalmente utilizzati come handshake FT a 4 vie per derivare la nuova chiave di crittografia PTK (unicast) e GTK (chiave di crittografia multicast/broadcast).

Questo sostituisce l'handshake a 4 vie che normalmente si verifica dopo lo scambio di questi frame e la negoziazione del contenuto FT e della chiave su questi frame è fondamentalmente la stessa sia che si utilizzi 802.1X/EAP o PSK come metodo di sicurezza. Come mostrato nell'immagine, il campo AKM è la differenza principale, il che conferma se il client esegue FT con PSK o 802.1X. Pertanto, è importante notare che questi quattro frame in genere non dispongono

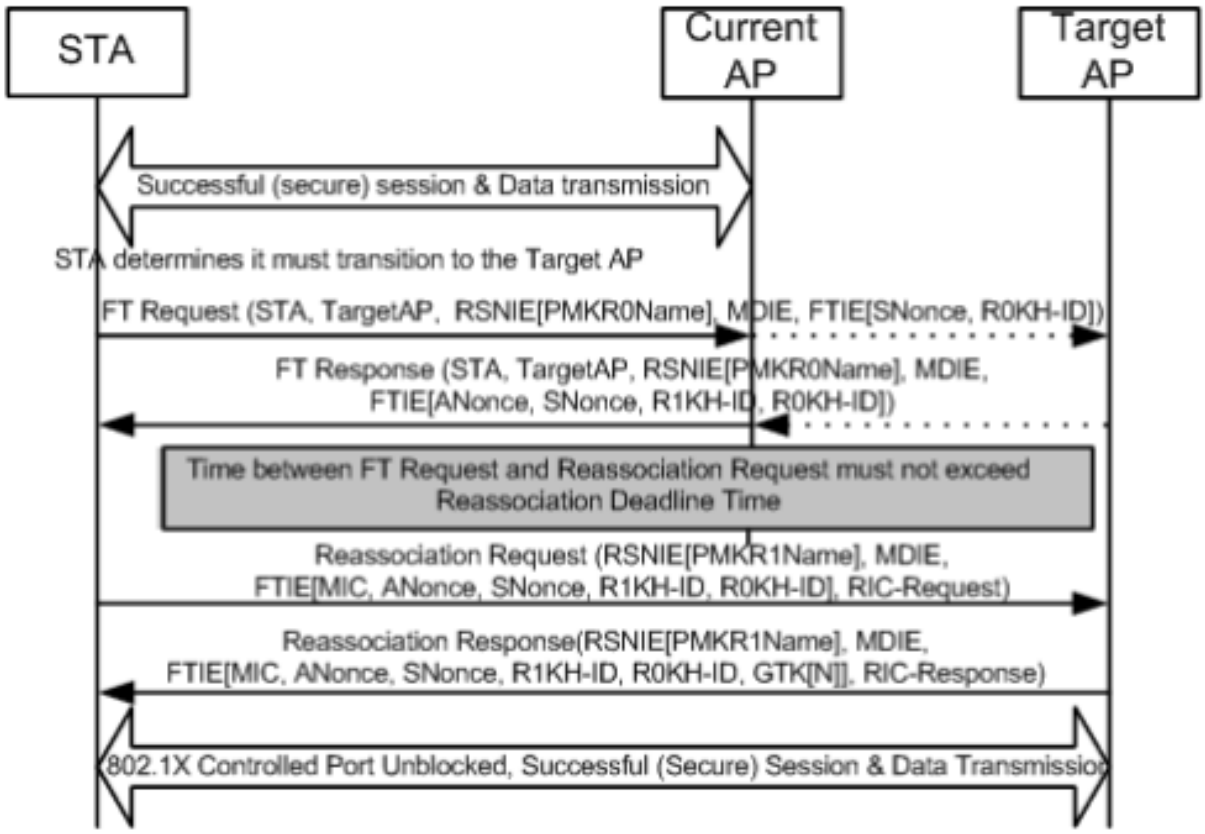
di questo tipo di informazioni di sicurezza per la negoziazione delle chiavi, ma solo quando il client FT esegue il roaming se 802.11r viene implementato e negoziato tra il client e l'infrastruttura WLAN al momento dell'associazione iniziale.

Transizione BSS rapida su DS

802.11r consente un'altra implementazione di Fast BSS Transition, in cui il roaming FT viene avviato dal client con il nuovo access point per il quale il client esegue il roaming Over-the-DS (sistema di distribuzione), e non Over-the-Air. In questo caso, per avviare la negoziazione della chiave vengono utilizzati i frame di azione FTP anziché i frame di autenticazione del sistema aperto.

In pratica, una volta che il client decide di poter eseguire il roaming verso un punto di accesso migliore, invia un frame di richiesta azione FT all'access point originale dove è attualmente connesso prima del roaming. Il client indica il BSSID (indirizzo MAC) del punto di accesso di destinazione in cui desidera eseguire il roaming. L'access point originale inoltra il frame di richiesta azione FT all'access point di destinazione sul sistema di distribuzione (in genere l'infrastruttura cablata) e l'access point di destinazione risponde al client con un frame di risposta azione FT (anche sul DS, in modo da poterlo finalmente inviare via etere al client). Una volta completato lo scambio del frame di azione FT, il client completa il roaming FT; il client invia la richiesta di riassociazione all'access point di destinazione (questa volta via etere) e riceve una risposta di riassociazione dal nuovo access point per confermare la derivazione delle chiavi in roaming e finale.

In sintesi, ci sono quattro frame per negoziare la transizione BSS rapida e derivare nuove chiavi di crittografia, ma in questo caso i frame di autenticazione del sistema aperto vengono sostituiti con i frame di richiesta/risposta azione FT, che vengono scambiati con il punto di accesso di destinazione sul sistema di distribuzione con il punto di accesso corrente. Questo metodo è valido anche per entrambi i metodi di sicurezza 802.1X/EAP e PSK, tutti supportati dai Cisco Wireless LAN Controller. Tuttavia, poiché la transizione Over-the-DS non è supportata e implementata dalla maggior parte dei client wireless nel settore WiFi (e poiché gli output di frame exchange e debug sono fondamentalmente gli stessi), nel presente documento non vengono forniti esempi. Al contrario, questa immagine viene usata per visualizzare la transizione BSS veloce over-the-DS:



FlexConnect con 802.11r

- L'autenticazione centrale è supportata. Ciò include la commutazione dei dati locale e centrale. Gli access point devono appartenere allo stesso gruppo FlexConnect.
- Autenticazione locale non supportata.
- Modalità autonoma non supportata.

Vantaggi di 802.11r

- Questo metodo è il primo che utilizza una gerarchia di chiavi chiaramente definita dall'IEEE sullo standard 802.11 come modifica (802.11r), quindi l'implementazione di queste tecniche FT è più compatibile tra i fornitori e senza interpretazioni diverse.
- Lo standard 802.11r consente di utilizzare diverse tecniche utili, in base alle esigenze (over-the-air e over-the-DS, per la sicurezza 802.1x/EAP e per la sicurezza PSK).
- Il client wireless esegue il roaming sicuro e rapido verso un nuovo access point sulla stessa WLAN/SSID, anche se non è mai stato associato a tale access point e senza la necessità di salvare più PMKID.
- Questo è il primo metodo di roaming veloce e sicuro che consente un roaming più veloce anche con la sicurezza PSK ed evita l'handshake a 4 vie richiesto quando si utilizza il roaming tra punti di accesso con PSK WPA/WPA2. Lo scopo principale dei metodi di roaming a sicurezza rapida è quello di evitare l'handshake 802.1X/EAP quando questo metodo di sicurezza è implementato; tuttavia, per la sicurezza PSK l'evento di roaming è accelerato ancora di più con 802.11r quando si evita l'handshake a 4 vie.

Svantaggi con 802.11r

- Alcuni dispositivi client wireless supportano effettivamente le transizioni Fast BSS e, nella maggior parte dei casi, non supportano tutte le tecniche disponibili in 802.11r.
- A causa del fatto che queste implementazioni sono molto recenti, non vi sono abbastanza risultati di test dagli ambienti di produzione reale o abbastanza risultati di debug per risolvere eventuali problemi che possono presentarsi.
- Quando si configura una WLAN/SSID in modo da utilizzare uno dei metodi FT, solo i client wireless che supportano 802.11r possono connettersi a questa WLAN/SSID. Le impostazioni FT non sono facoltative per i client, quindi i client wireless che non supportano 802.11r devono connettersi con una WLAN/SSID separata in cui FT non è configurato affatto.

Adattivo 802.11r

- Alcuni client legacy non possono essere associati a una WLAN/SSID con 802.11r abilitato anche per la "modalità mista" (che si spera possa avere sugli stessi client SSID che supportano e non supportano 802.11r). In questo caso, il driver del supplicant client responsabile dell'analisi dell'elemento di informazioni sulla rete di sicurezza robusto (RSN IE) è obsoleto e non è a conoscenza delle suite AKM aggiuntive in IE. A causa di questo limite, i client non possono inviare richieste di associazione alle WLAN che annunciano il supporto 802.11r e, di conseguenza, è necessario configurare una WLAN/SSID per i client 802.11r e una WLAN/SSID separata per i client che non supportano 802.11r.
- Per risolvere questo problema, l'infrastruttura LAN wireless di Cisco ha introdotto la funzionalità Adaptive 802.11r. Quando la modalità FTP è impostata su Adaptive a livello di WLAN, la WLAN annuncia l'ID del dominio di mobilità 802.11r su una WLAN abilitata per 802.11i. Alcuni dispositivi client Apple iOS10 identificano la presenza di MDIE su una WLAN 80211i/WPA2 ed eseguono un handshake proprietario per stabilire l'associazione 802.11r. Una volta che il client ha completato con successo l'associazione 802.11r, può eseguire il roaming FTP come in una normale WLAN abilitata 802.11r. FT Adaptive è applicabile solo a determinati dispositivi Apple iOS10 (e versioni successive). Tutti gli altri client possono continuare a disporre dell'associazione 802.11i/WPA2 sulla WLAN ed eseguire il metodo FSR applicabile, se supportato.
- Per ulteriori informazioni su questa nuova funzionalità, introdotta per i dispositivi iOS10 per eseguire 802.11r su una rete WLAN/SSID dove 802.11r non è realmente abilitato (quindi altri client non 802.11r possono connettersi correttamente), vedere Enterprise Best Practices for Cisco IOS Devices on Cisco Wireless LAN.

Conclusioni

- Tenere presente che il client decide sempre di eseguire il roaming verso un determinato access point e che il WLC/AP non può decidere questa scelta per il client. L'evento di roaming viene avviato dal client wireless una volta che lo ritiene necessario.
- Il WLC supporta una combinazione della maggior parte o di tutti i metodi FSR (Fast-Secure Roaming) sullo stesso WLAN/SSID. Tuttavia, questo normalmente non funziona, in quanto dipende fortemente dal comportamento del client (molto diverso tra i diversi dispositivi mobili) al fine di supportare o anche capire quello che il WLC cerca di pubblicizzare come supportato. Anziché raggiungere l'interoperabilità in un solo SSID, in genere i problemi sono più numerosi di quelli che si prevede di risolvere, pertanto questa soluzione non è consigliata. Se necessario, è necessario completare test approfonditi con tutti i possibili client da utilizzare su

questa WLAN.

- Èmolto importante capire che sono stati sviluppati metodi di roaming veloci e sicuri per accelerare il processo di roaming della WLAN quando ci si sposta da un punto di accesso all'altro se la sicurezza della WLAN/SSID è abilitata. Quando non è presente alcuna protezione, non c'è nulla da accelerare, in quanto l'access point client-AP si limita a scambiare i frame di gestione wireless che sono sempre necessari quando si effettua il roaming tra gli access point prima dell'invio dei frame dati (autenticazione di sistema aperto dal client, autenticazione di sistema aperto dall'access point, richiesta di riassociazione e risposta di riassociazione). Pertanto, questa operazione non può essere eseguita più rapidamente. Se si riscontrano problemi di roaming senza sicurezza, non esistono metodi di roaming veloce per migliorare il roaming, ma solo metodi per verificare se la configurazione e la progettazione della WLAN/SSID sono appropriate per consentire alle stazioni client wireless di effettuare il roaming di conseguenza tra le celle di copertura AP.
- 802.11r/FT è implementato con WPA2-PSK per accelerare gli eventi di roaming con questa sicurezza ed evitare l'handshake a 4 vie, come spiegato nella sezione 802.11r.
- Tutti i metodi presentano vantaggi e svantaggi, ma alla fine è sempre necessario verificare se le stazioni client wireless supportano il metodo specifico che si desidera implementare e se l'infrastruttura WLAN di Cisco supporta tutti i metodi disponibili. Pertanto, è necessario selezionare il metodo migliore realmente supportato dai client wireless che si connettono alla specifica WLAN/SSID. Ad esempio, in alcune implementazioni è possibile creare una WLAN/SSID con CCKM per i telefoni IP wireless Cisco (che supportano WPA2/AES con CCKM, ma non 802.11r) e quindi un'altra WLAN/SSID con WPA2/AES tramite 802.11r/FT per i client wireless che supportano questo metodo di roaming Fast Secure (o utilizzare OKC, se supportato).
- Se i client wireless non supportano nessuno dei metodi di roaming a sicurezza rapida disponibili, è necessario accettare il fatto che tali client possano sempre sperimentare i ritardi spiegati in questo documento quando eseguono il roaming tra punti di accesso su una rete WLAN/SSID con sicurezza 802.1X/EAP (che può causare interruzioni alle applicazioni/ai servizi client).
- Tutti i metodi, ad eccezione di SKC (WPA2 PMKID Caching), sono supportati per il roaming sicuro e veloce tra i punti di accesso gestiti da diversi WLC (roaming tra controller), a condizione che si trovino sullo stesso gruppo di mobilità.
- CUWN supporta completamente tutti i diversi metodi di roaming Fast-Secure descritti in questo articolo quando viene utilizzata l'autenticazione 802.1X/EAP per WPA/WPA2. CUWN non supporta il roaming Fast-Secure su metodi che funzionano con WPA2-RSN (CCKM, PMKID Caching/SKC, OKC/PKC) quando si utilizza PSK (WPA2-Personal), in cui la maggior parte dei metodi Fast-Roaming non è necessaria. Tuttavia, CUWN supporta il roaming Fast-Secure nel caso di WPA2-FT (802.11r) con PSK, come spiegato anche in questo articolo.

Informazioni correlate

- Guida alla distribuzione della transizione rapida BSS 802.11r
- Supporto tecnico e download Cisco

Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l’accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).