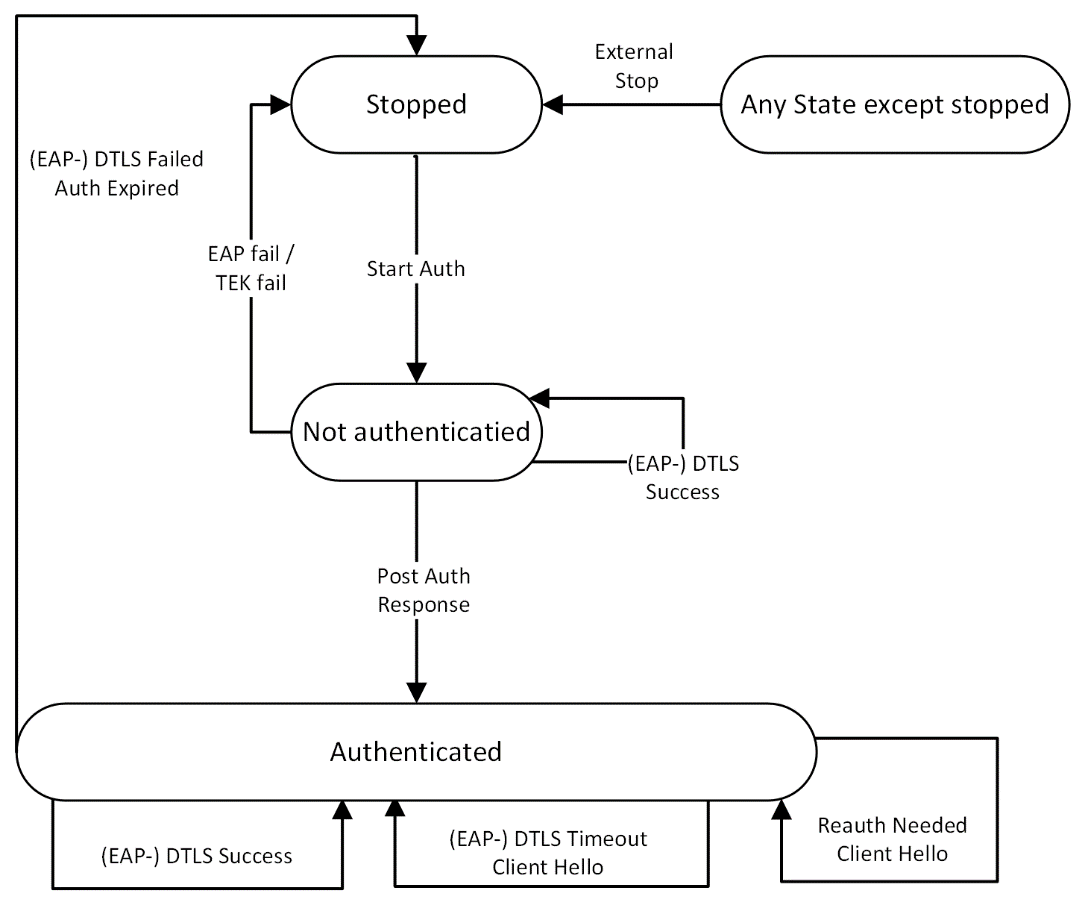
Pg 33-34 change in Auth FSM



**Figure 7-26—Authentication State Machine**

**7.8A.3.1.1 Stopped State**

This is the initial state of the FSM. Nothing is done in this state.

**7.8A.3.1.2 Not Authenticated State**

The TLS/EAP-TLS authentication is in process. Transition from another state into this state triggers the initialization of the authentication process. In this state the SS can only send management messages. No data can be transmitted at this state. A failure in the authentication process shall change the Auth-FSM state to Stopped.

Upon completing the authentication process, both BS and SS derives the PMK, AK and KEK keys and the HMAC keys.

deriving the key, the BS shall send a PKMv2 SA-TEK-Response with the following attributes: AKID, BS\_Random, Frame Number and SA-Descriptors (one or more) . All attributes except the AKID shall be encrypted using the KEK key. AKID must be the first attribute in the message.

Receiving a PKMv2 SA-TEK-Response, the SS shall use its KEK key to decrypt the attributes and shall trigger the SS to switch to Authenticated state.

Note that in PKMv3, the PKMv2 SA-TEK-Challenge and PKMv2 SA-TEK-Request messages are not used and PKMv2 SA-TEK-Response message is sent on initial network only and not on reauthentication or rekeying.

Page 35

**7.8A.3.2.1 Start State**

This is the initial state of the FSM. No resources are assigned to or used by the FSM in this state—e.g., all timers are off, and no processing is scheduled.

**7.8A.3.2.2 Op (operational) Wait State**

The TEK state machine has sent its initial request (PKMv2 Key-Request) for SAID generation and is

waiting for a reply from the BS. The PKMv2 Key-Request message is sent with the following attributes:

current AK Sequence Number, SAID (or GSAID for multicast) and Nonce generated by the SS. Both BS and SS shall calculate the HMAC digest with the HMAC key that was derived from current AK over the decrypted PKMv2 Key Request message to derive the new TEK as described in 19.4.5. The BS shall send a PKMv2 Key-Reply message to the SS with Key Sequence and SAID attributes encrypted by the KEK key derived from current AK. . In the case of multicast service key request (GKEK or GTEK) the BS shall act as in PKMv2.

Note: unlike PKMv2, unicast TEK key materials are not sent over the air.

**7.8A.3.2.3 Op Reauth (reauthorize) Wait State**

The wait state the TEK state machine is placed in if it does not have valid keying material while the

Authorization state machine is in the middle of a reauthorization cycle.

**7.8A.3.2.4 Operational State**

Upon receiving a PKMv2 Key-Reply from the BS, the SS shall start using the new TEK key.

The TEK FSM shall start a TEK Refresh Timer for the TEK lifetime period.

**7.8A.3.2.5 Rekey Wait State**

Upon TEK Refresh Timer expiration, the SS shall send a PKMv2 Key-Request for this SAID. The PKMv2 Key-Request message is sent with the following attributes: current AK Sequence Number, SAID (or GSAID for multicast)and Nonce generated by the SS. Both BS and SS shall calculate the HMAC digest with the HMAC key that was derived from current AK over the Key Request message and shall use it to derive the new TEK as described in 19.4.5. The BS shall send a PKMv2 Key-Reply message to the SS with Key Sequence and SAID attributes encrypted by the KEK key derived from current AK. In the case of multicast service key request (GKEK or GTEK) the BS shall act as in PKMv2.

Note that the newer of the SS two TEKs has not expired and can still be used for both encrypting and

decrypting data traffic.

**7.8A.4 Key derivation and usage**

**7.8A.4.1.3.1 AK usage**

The BS and SS have two active AKs during an AK transition period; the two active keys have overlapping lifetimes.

Once the AK is derived the BS and SS shall use the new AK matching the new PMK context for encrypting following PKM messages. Other messages shall continue to use the old AK until the handshake completes successfully. Upon successful completion of the handshake, all messages shall use the new AK.

Successful completion of the TEK derivation after authentication or reauthentication causes the activation of every AK associated with the new PMK. The BS and SS shall maintain the AK context as long as they retain the AK.

The old AK matching the old PMK context can be used for receiving packets before the “frame number” attribute specified in PKMv2 SA-TEK-response message.

If an SS fails to reauthorize before the expiration of its current AK, the BS shall hold no active AKs for the SS and shall consider the SS unauthorized. A BS shall remove from its keying tables all TEKs associated with an unauthorized SS’s SA.

**7.8A.4.2 Key Encryption Key (KEK) derivation**

The KEK is derived directly from the AK. The KEK is used to encrypt the GKEK that is sent by the BS to SS in multicast messages and to encrypt PKM messages.

The keys used for KEK key material are as follows:

KEK.Key <== HKDF-expand (AK, SS MAC Address | BSID | “KEK\_Key”, kek.keyLength)

KEK.IV <== HKDF-expand (AK, SS MAC Address | BSID | “KEK\_IV”, kek.keyLength)

**7.8A.4.3 GKEK derivation and usage**

GKEK (Group Key Encryption Key) is randomly generated at the BS or a network entity (for example, an ASA server) and transmitted to the SS encrypted with the KEK. There is one GKEK per Group Security Association. GKEK is used to encrypt the GTEKs sent by the BS to the SSs in the same multicast group.

A BS transmits the PKMv2 Group-Key-Update-Command message (6.3.2.3.9.25) for the GKEK update mode to each SS served with the specific multicast / broadcast service before the current GKEK expires and the last GTEK Grace Time of the corresponding current GKEK starts. The purpose of the PKMv2 Group-Key-Update-Command message for the GKEK update mode is to distribute the GKEK. The PKMv2 Group-Key-Update-Command message for the GKEK update mode is carried on the Primary Management connection. A BS intermittently transmits the PKMv2 Group-Key-Update-Command message for the GKEK update mode to each SS in order to reduce the BS’s load in refreshing traffic key material. The GKEK is needed to encrypt the new GTEK.

The GKEK lifetime corresponds to the n (integer being bigger than 1) times of the GTEK lifetime. That is, the GKEK shall be updated once while the GTEK is updated n times.

The multicast and broadcast rekeying overflow is shown in Figure 7-24.

**7.8A.4.4 Traffic encryption key (TEK) Derivation**

Both BS and SS shall derive the TEK directly from the AK. The first traffic key material derived after

successful authentication or reauthentication are derived as follows:

TEK.Key <== HKDF.Expand(AK, SAID|HTEK|“key”, Aes.KeyLength)

TEK.Iv <== HKDF.Expand(AK, SAID |“iv”, Aes.IvLength)

The next traffic keys are derived as follows:

TEK\_N.Key <== HKDF.Expand(TEK\_N-1, SAID|HTEK|“key”, Aes.KeyLength)

TEK\_N.Iv <== HKDF.Expand(TEK\_N-1, SAID|“iv”, Aes.IvLength)

The HTEKis HMAC digest calculated over the decrypted attributes of the KeyRequest message.

TEK\_N-1 is the current active TEK.

The multicast and broadcast rekeying management overflow is shown in Figure 7-24.