Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**PKMv2 Security Framework Corrections**
Date Submitted	**2005-01-27**
Source(s)	Yigal Eliaspur, Avishay Shraga, Sanjay Bakshi, David Ayoun Intel Corporation Jeff Mandin Streetwaves Networking Ren Jing, Li Rui, Tian Feng ZTE Corporation YongChang Samsung Chulsik Yoon ETRI Shujun Dang, Jianjun Wu Huawei Technologies Co.,Ltd yigal.eliaspur@intel.com Voice +972-54-7884877 sanjay.bakshi@intel.com avishay.shraga@intel.com david.ayoun@intel.com
Re:	IEEE P802.16e/D5a
Abstract	There are still areas in the PKMv2 security framework that requires major corrections This contribution proposed a resolution for those major issues.
Purpose	Adoption of proposed changes into P802.16e /D5a-2004
Notice	
Release	

Table of Contents

PKMv2 Security Framework Corrections

Yigal Eliaspur, Avishay Shraga, Sanjay Bakshi, David Ayoun,
Jesse Walker, David Johnston, Rosner Gedon
Intel Corporation

1 Motivation

There are still areas in the PKMv2 security framework that requires major corrections This contribution proposed a resolution for those major issues.

2 Overview

2.1 Remedy 1 – Key Hierarchy corrections

Proposal C802.16e-04/217r1 was accepted in session 34 but was not incorporated to the text of draft 5a.

This remedy rephrases the previous contribution and creates a well defined set of security contexts and a clear key hierarchy for use in PKMv2. The approach adapts the PMK1 key material and key deriving methodologies to achieve backward compatibility.

Some modifications were made to the original contribution.
The changes added to this contribution are:

- a group message authentication (H/OMAC) was created. The purpose is to ensure the authenticity of multicast management messages (GKEK transmission for instance).
- ~~EIK and EEK were removed. EIK and EEK are used to encrypt the EAP exchange. 1) the EAP exchange is by definition secure and encrypted by standard, thus these keys are redundant. 2) these keys were not defined in case RSA authorization was not used (no PAK available for the derivation).~~
- ~~EAK is introduced. EAK is an intermediate key created for mobility purposes mainly. This key may be common to several BSs (key-zone) and may be generated in a separate network entity. So that the whole time-consuming EAP exchange may not be necessary at each hand-over.~~
- Authorization Association (AA) is replaced by the AK context. The AA was defined without support for EAP only authorization. The AAID was defined by using mutual authorization products that is not available when this process is not executed. Therefore a more general AK context was defined and some variables were added for replay attack protection.
- G~~E~~KEKEK was removed. GKEK is the key encrypting the GTEK multicast transmission. GKEK is randomly generated in the BS, encrypted with MSS's KEK and transmitted to each MSS in the group. Since GKEK is the same for all MSSs in the group, each of them receive this way the cipher for each MSS and

the plaintext which is the key itself. However it is virtually impossible to find back the key knowing the plaintext and the AES encrypted cipher. So that GKEKEK is not necessary.

Modification ~~Details~~Summary

In PKMv2, certificated RSA mutual authorization and EAP authentication can be used independently. That is, at capabilities negotiation, the MSS notify to the BS which mode is supported, and the BS decides which authorization model will be adopted and can choose to use only RSA authorization, EAP authentication or both. In each of these 3 options, the key material is derived differently.

2.2 Remedy 2 – OMAC calculation correction

There are two TLV types by which OMAC is used in the standard: one is the OMAC tuple used in management messages and the other is a bare OMAC digest used only in PKM messages. The OMAC tuple and digest as defined in version 16e_D5a do not provide ample protection against replay attacks of the messages they are meant to protect. The purpose of this contribution is to add a 4byte packet counter that is part of the OMAC key context and computed in the OMAC digest to protect against replay attacks of management message. If an MSS or BS receive a MAC message with an OMAC tuple or digest with a packet counter number that is not greater than the previous message received on the same CID they will drop that message.

In the OMAC TLVs add a 4 byte Packet Number (PN) field to be added to both the digest and the tuple. This field will be sequentially updated on every MAC management packet sent with OMAC digest or tuple. The context of the field will be updated in the UL and the DL by the MSS and the BS respectively, each in it s own context (i.e. – the MSS will maintain a separate context than the BS and the PN in UL will increment independently of the PN in the DL). Multicast CID messages will also maintain their own OMAC context with a separate PN.
To ensure the uniqueness of the PN, the MSS or the BS must perform re-authentication and acquire a new AK context before the PN rolls over.
The PN will be added to the OMAC digest calculation and the OMAC_TLV attributes will be removed from the calculation. Removing the OMAC_TLV attributes is due to the explicit inclusion of the OMAC sequence id and the OMAC PN in the digest description. The frame number will also be removed from the OMAC calculation

2.3 Remedy 3 – 3 Way handshake and TEK exchange corrections

During initial network entry or upon handoff when shared AK has been established, the BS and SS must: -

- Establish the liveliness of Authorization Key (AK). The AK is used:
    - to derive a hash key that's used for link management messages
    - to derive a Key Encryption Key for transporting the actual traffic encryption keys
- Negotiate cryptographic capabilities and distribute an SAID list to the SS

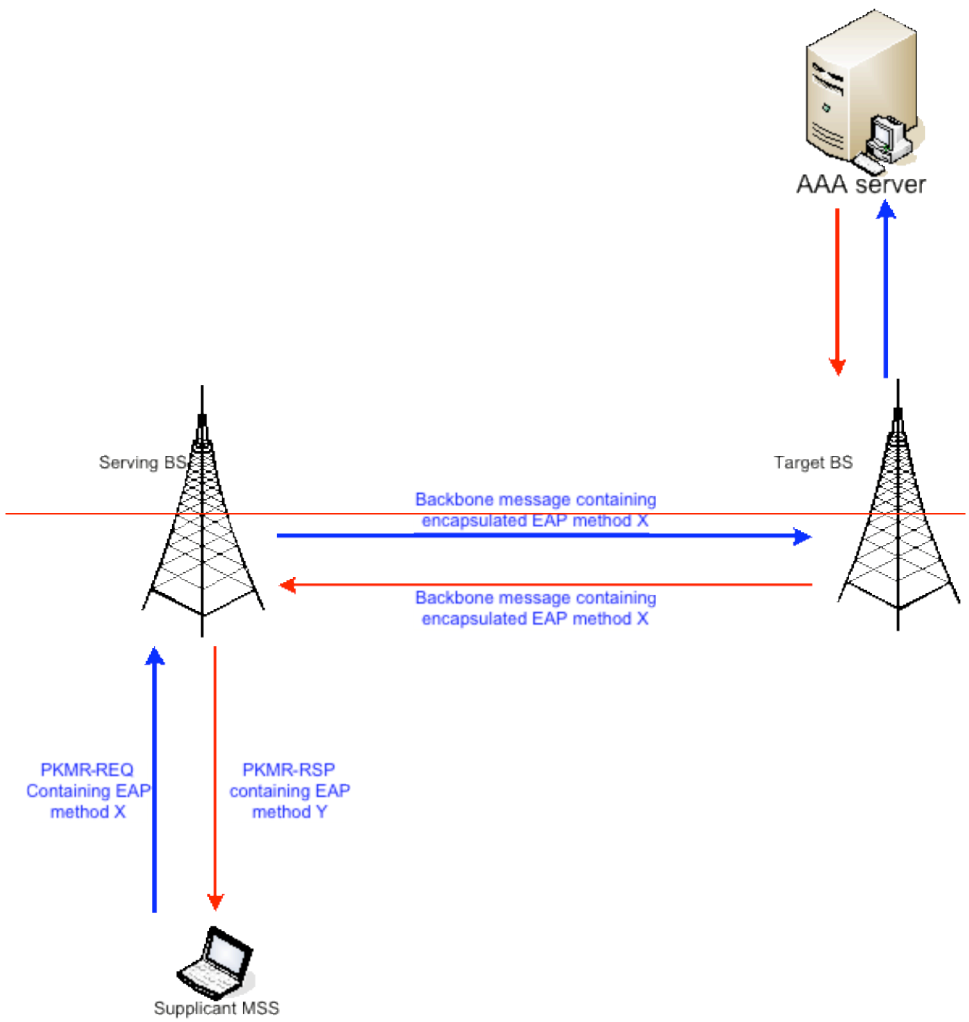
- In case of ~~network entry and~~ HO, ~~generate~~ exchange ~~fresh~~ TEKs for the SAs that were active on the previous serving BS or incase of network entry exchange TEKs for the SAs associated with optional static services,

The number of roundtrips between BS and MSS to perform all the above exchanges ~~greatly~~ impacts the over all time taken to complete the handoff and re-start the subscriber's application level flows. This contribution proposes to perform all the above within a modified 3-way handshake transaction. Initial network entry and re-authentication use a 3-message exchange, while in the handover/reentry case, the first message is merged into the preceding RNG-RSP, so that fast reauthorization is accomplished with a single roundtrip. The 3-way handshake guarantees the identity and "liveliness" of both BS and SS, and is proven secure under the Bellare-Rogaway model,~~First two messages of the proposed 3-way handshake protocol are mandatory while the third message is optional. An implementation thus has the choice to just use the first two messages i.e. a 2-way handshake to shorten the HO time or go for the complete 3-way handshake which is proven secure under the Bellare-Rogaway model.~~

~~2.4Remedy 4 – Pre-authentication support~~

~~The authentication process of an SS with the authentication server (e.g. AAA) may take a prolonged amount of time. In 802.16D, this is not a real problem since this task is performed infrequently. In 802.16E, though, the mobility of the MSS will make it work with many BS over a short period. It will have to authenticate with as many BS's and may therefore lose some data and service continuity in the process. The pre-authentication feature is supposed to fix this problem. However, this feature as it is defined currently (P80216e_D5a) relies on an EAP protocol not yet approved and is not compatible with previous versions of the EAP protocol. This contribution proposes a way to send an EAP method to a BS (authenticator) through another BS (Serving BS) from the MSS (supplicant) and back, and effectively tunnel the EAP method through the serving BS. All this can be done at a non-critical time with all the neighbor BS's while the link with the serving BS is still good. The purpose is to shorten the hand-over procedure and thus minimize the lost data and delay at this moment.~~
~~Additionally, there is a need for the MSS especially in pre-authentication, to initiate the EAP authentication and make the BS poll it.~~

~~1.**Details**~~
~~Two MAC management messages are added: PKMR-REQ and PKMR-RSP for Privacy Key Management Remote. These messages are similar to PKM-REQ and PKM-RSP respectively but the serving BS when receiving them will only verify the signature (OMAC or HMAC) and, if valid, forward the message to the target BS.~~
~~The general model is illustrated in figure 1.~~



~~Figure 1~~

~~A new type is added to both the PKM-REQ and PKMR-REQ for the MSS to start the EAP authentication~~

~~2.5~~2.4 Remedy 5 – HO optimization support

Draft 802.16D5a defines some optimizations to the HO process using an Optimization TLV.
Among these optimizations, there is a bit that can indicate the MSS it can skip PKM stage in the network-entry after HO.
There is no definition how security is established in this case but the hidden assumption is that the entire security context (keys and other parameters) is transferred from source BS to target BS in the back-bone.
This security context sharing is considered not secure enough.
This contribution supply mechanisms to establish and maintain a unique security context between the MSS and the target BS while still connected to the source BS and by this allowing skipping of PKM steps in network re-entry while maintaining high level of security.

In order to achive this goal, this contribution defines the following:

~~a)A key-zone is as set of one or more BSs that are populated, by the same authenticator, with a per-BS, unique AK derived itself from the PMK, product of the EAP authentication with the MSS.~~

~~b)A capability bit for the MSS to determine if a target BS supports pre-authentication.~~

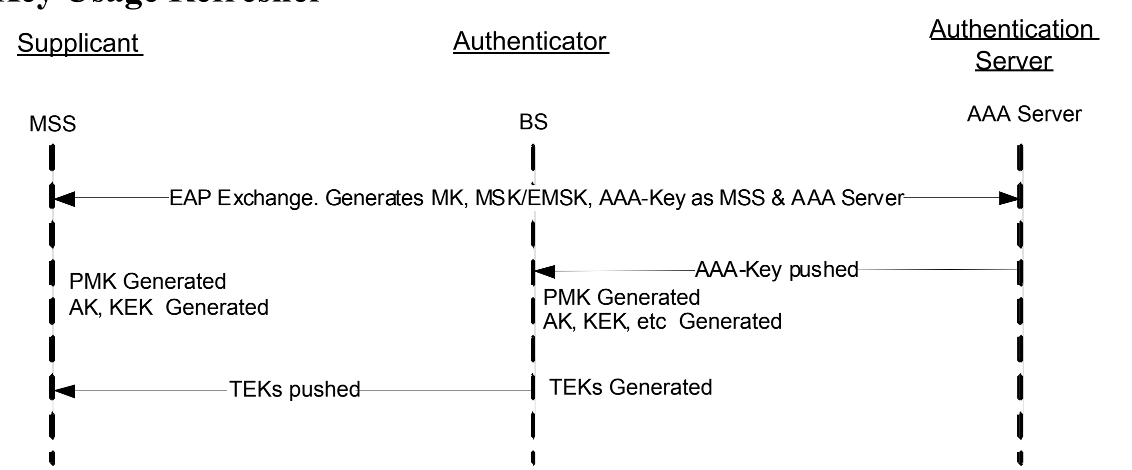
~~c)~~a) A configuration bit that signals to the MSS if it can use the current PMK to generate authentication material with target BS ~~which belongs to same key-zone as the current serving BS~~.~~.~~

~~d)A configuration bit that signals to the MSS if a target BS which belongs to another key-zone still maintained an authentication context with it~~

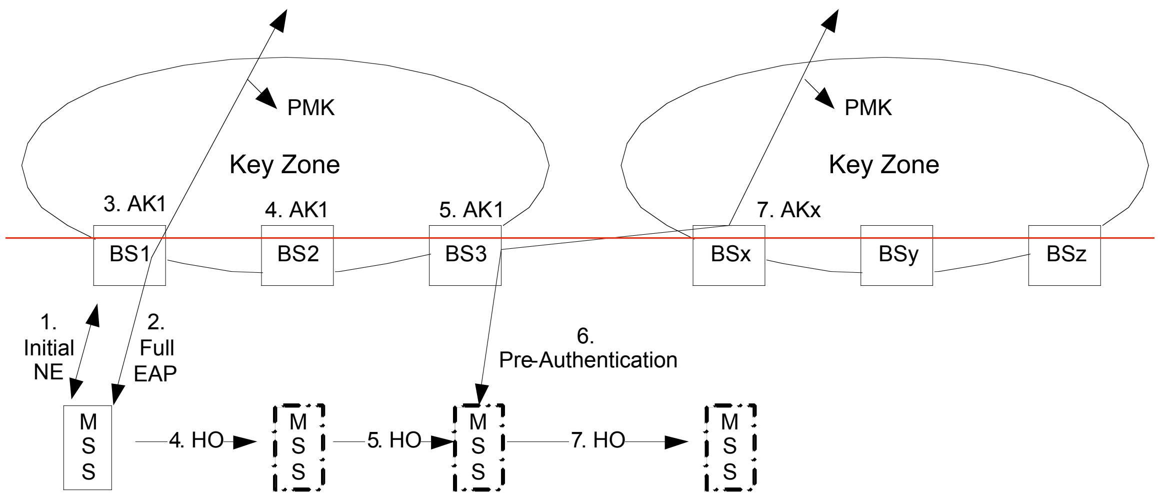
~~e)~~b) A configuration bit that signals to the MSS if it can use its existing TEKs with the potential target BS(s) after handoff.

Given the above knowledge, an MSS can get information about its authentication status with the target BS and determine if it can obtain authentication with target BS in advance or if full EAP authentication is needed upon HO. By being able to skip full EAP authentication upon HO, the overall HO duration is greatly reduced.

Key Usage Refresher



~~Key Zone usage for Fast HO~~



~~2.6~~2.5 Remedy 6 – Mutual Authorization support

Mutual Authorization as defined in 802.16e_D5 has to be updated from two reasons: the first is some mistakes/incoherencies with other parts of the security suite and the other is that although the contribution 229r1 was accepted in session 32, it was not fully integrated into the standard.
This contribution defines the text that should be inserted into 802.16e in order to align the standard with the original changes needed in the original contribution and include the updates here.
In general, the mutual certificate exchange in PKMv2 is defined as follows:

auth_req: SS -> BS: SS-Random | Cert(SS) | Security suites Primary SAID(equals Basic CID)
auth_reply: BS -> SS: SS-Random | BS-Random | RSA-OAEP-Encrypt(PubKey(SS), PAK | Id(SS)) | Lifetime | PAKSeqNo | SAIDList | Cert(BS) | Sig(BS)
auth_ack: SS -> BS: BS-Random | SS_MAC_Address | OMAC (PAK[s1], BS_Random | SS_MAC_Address)

The PAK (Primary Authorization Key) is generated as a cryptographically strong random number in the BS and transmitted to the SS, encrypted with RSA during the above PKMv2 mutual authorization exchange.
The following are defined in this proposal:
- The auth-req/rsp/ack messages
- A BS certificate

~~2.7~~2.6 Remedy 7 – Authorization Policy Support Negotiation

The Authorization Policy Support field indicates some parameters about the authorization authentication protocol supported by the station. It is a bitfield where each field indicates a feature supported by the station. Since it is a bitfield any feature can be supported or not. There is also a PKM version support field that is independent from the previous field. However many combinations of this bitfield are not possible and therefore superfluous. This contribution proposes a way to make the number of redundant combinations lower. There is however a condition to be backward compatible to 802.16d.

1. **Details**

There are only a few number of combinations supported in the protocol:
PKMv1 RSA 1-way
PKMv1 EAP
PKMv2 RSA 2-way
PKMv2 EAP
PKMv2 EAP+RSA 2-way

The Mutual Auth./Unidirectional Auth. Bit is therefore not necessary since it is determined by the PKM version. We propose to remove it.

Furthermore, the OMAC/HMAC bit is a not feature absent/present bit. To correct this we make it an OMAC feature bit. Implicitly, since HMAC is the default, it means that if the OMAC bit is off, then HMAC is supported.

We leave the RSA and EAP bits although not all the combinations can exist, but we add some explicit instructions in the standard.

3 Text Change

3.1 Remedy 1 – Key Hierarchy corrections

[Insert after 7.2.2.1 the following text]

7.2.2.2 Key Derivation

The PKMv2 key hierarchy defines what keys are present in the system and how the keys ~~they~~ are generated.
Since there are two authentication schemes, one based on RSA and one based on EAP, there are two primary sources of keying material.

The keys used to protect unicast and multicast traffic are derived from source key material generated by the authentication and authorization processes. The authorization process yields the pre-Primary AK (pre-PAK) and the EAP based authentication process yields the ~~EAP AK (EAK).~~AAA-Key. Keys used to protect MBS traffic are derived from the MBSAK, which is supplied by means outside the scope of this specification. These keys form the roots of the key hierarchy.

All PKMv2 key derivations are based on the Dot16KDF algorithm as defined in 7.x.x.x Dot16KDF.

7.2.2.2.1 Certificated RSA authorization:

The pre-PAK (Primary Authorization Key) is sent by the BS to the MSS encrypted with the public key from the certificate.
Pre-PAK is mainly used to generate the PAK. The optional EIK ~~and EEK~~ for EAP exchange (see 7.2.2.2.2) are also generated from pre-PAK:

~~EEK~~ | EIK | PAK = Dot16KDF(pre-PAK, SSID | ~~BSID |~~ "~~EPK+~~EIK+PAK", ~~416~~288)

~~PAK is 160 bits long.~~
PAK will be used to generate the AK (see below) if RSA authorization was used.
PAK is 160 bits long.

7.2.2.2.2 EAP authentication

If a mutual authorization took place before the EAP exchange, the EAP messages may be protected using ~~two keys (EEK – EAP Encryption Key and~~ EIK – EAP Integrity Key~~)~~ derived from pre-PAK (see 7.2.2.2.1).
EIK and EEK are 128 bits long.

The product of the EAP exchange which is transferred to 802.16 layer is the AAA-key. This key is derived (or may be equivalent to the 512-bits MSK). This key is known to the AAA server, to the ~~BS authenticator~~ Authenticator*[s2] (transferred from AAA server) and to the MSS. The MSS and the authenticator derive a PMK (Pairwise Master Key) by truncating the AAA-key after 160 bits.

The PMK derivation from the AAA-key :

PMK = truncate (AAA-key,160)

If more keying material is needed for future link ciphers, the key_length of the PMK may be increased.

~~The next derivation step creates a key which is unique between the BS and MSS called AK (Authentication Key).EAK (EAP authentication key).~~
~~This key is created by the SS and the authenticator and transferred from the authenticator to the BS.~~
~~The purpose of this key is to allow an authenticator which is not collocated with the BS and serves more than one BS using a single PMK.~~

~~The EAK will be used to generate the AK (see below) in case EAP authentication was used.~~
~~The EAK will be derived from PMK,BSID and SSID:~~

~~EAK <= Dot16KDF (PMK, SSID | BSID | "EAK", 160)~~

7.2.2.2.3 Authorization Key (AK) derivation

The AK will be derived by the ~~BS~~ authenticator and the MSS from the ~~the EAK (~~PMK (from EAP exchange) and the PAK (from RSA exchange). ~~If one of these keys is not available, the AK will be equal to the other key.~~
Note that PAK can be used only in initial network entry. In cases of HO and re-authentication: Only EAP keys are applicable.

```
If (PAK and ~~EAK~~PMK)
        AK <= Dot16KDF (~~EAK~~PMK, SSID | BSID | PAK | "AK", 160)
Else
        If (PAK)
                AK <= Dot16KDF (0, SSID | BSID | PAK | "AK", 160)~~PAK~~
        Else                    // PMK only ~~// EAK only~~
                AK = Dot16KDF(PMK, SSID | BSID | "AK", 160);
                ~~AK <= EAK~~
        Endif
Endif
```

7.2.2.2.4 Key Encryption Key (KEK) derivation

KEK is derived directly from the AK
The Key Encryption Key (KEK) is defined in 7.2.2.2.9 with the OMAC/HMAC definition

KEK is used to encrypt the TEKs, GKEK and all other keys sent by the BS to MSS in unicast message.

7.2.2.2.5 Group Key Encryption Key (GKEK) derivation

GKEK is randomly generated at the BS and transmitted to the MSS encrypted with the KEK. There is one GKEK per Group Security Association.
GKEK is used to encrypt the GTEKs sent in multicast messages by the BS to the MSSs in the same multicast group.

7.2.2.2.6 Traffic Encryption Key (TEK)

The TEK is generated as a random number ~~is~~ in the BS and is encrypted using the corresponding TEK encryption algorithm (e.g. AES_KEY_WRAP for SAs with TEK encryption algorithm identifier in the cryptographic suite is equal to 0x04), keyed with the KEK and transferred between BS and SS in the TEK exchange.

7.2.2.2.7 Group Traffic Encryption Key (GTEK)

The GTEK is used to encrypt multicast data packets and it is shared between all MSSs that belong to the multicast group. There are 2 GTEKs per GSA.

The GTEK is randomly generated at the BS and is encrypted using AES_KEY_WRAP and transmitted to the MSS in multicast or unicast messages. In multicast the message will be encrypted by the GKEK. In unicast, it will be encrypted by the KEK.

.

7.2.2.2.8 MBS Transport Key (MTK)

The generation and transport of the MAK (MBS AK) is outside the scope of the 802.16 standard. It is provided through means defined at higher layers. However the keying is used in the link cipher, therefore its existence needs to be defined in layer 2.

The MTK is used to protect transport data. It is defined as follows:
MTK <= Dot16KDF(MAK, MGTEK, 128)

7.2.2.2.9 Message authentication keys (OMAC/HMAC) and KEK derivation

MAC (message authentication code) keys are used to sign management messages in order to validate the authenticity of these messages. The MAC to be used is negotiated at SS Basic Capabilities negotiation.
There is a different key for UL and DL messages and also a OMAC key for each multicast group (this is DL direction only).

The keys used for OMAC calculation and for KEK are as follows:

OMAC_KEY_U | OMAC_KEY_D | KEK <= Dot16KDF(AK, SSID | BSID | "OMAC_KEYS+KEK", 384)
OMAC_KEY_GD <= Dot16KDF(GKEK, "GROUP OMAC KEY",128) (used for group management messages MAC)

The keys used for HMAC calculation and for KEK are as follows:

HMAC_KEY_U | HMAC_KEY_D | KEK <= Dot16KDF(AK, SSID | BSID | "HMAC_KEYS+KEK", 448)
HMAC_KEY_GD <= Dot16KDF(GKEK, "GROUP HMAC KEY", 160) (used for group management messages MAC)

7.2.2.2.10 Key hierarchy

Figure xx1 outlines the process to calculate the AK when the authorization process has taken place, but where the EAP based authentication process hasn't taken place, or the EAP method used has not yielded an AAA-key:

Figure xx1: AK with RSA only authorization process

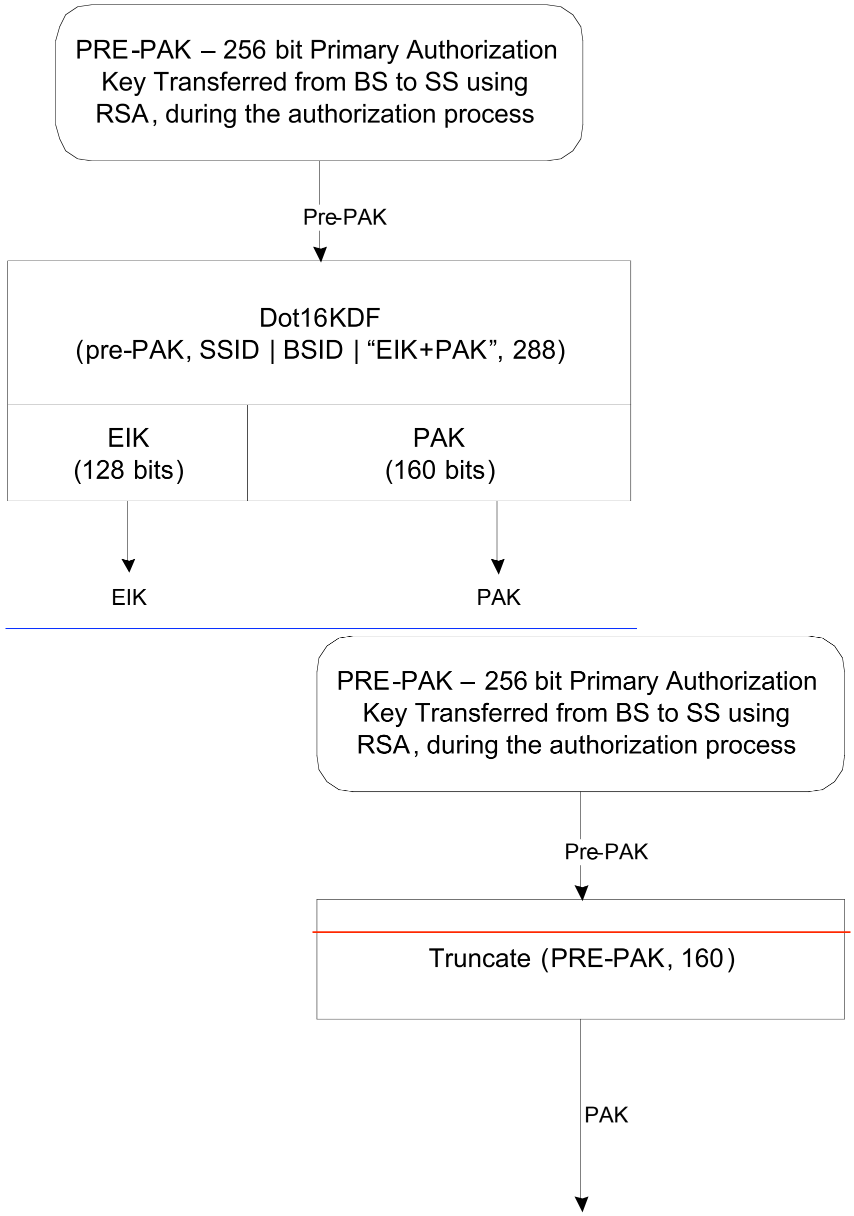
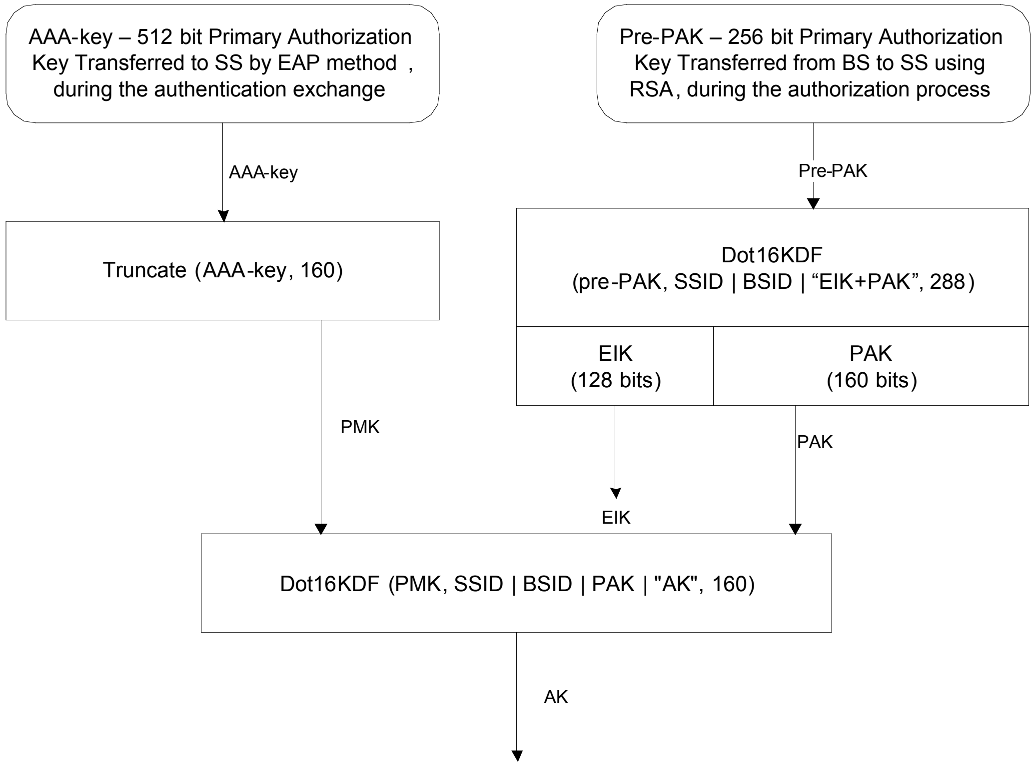


Figure xx2 outlines the process to calculate the AK when both the authorization exchange has taken place, yielding a PAK and the EAP based authentication exchange has taken place, yielding an AAA-key:

Figure xx2: AK with RSA+EAP authorization process


AAA-key – 512 bit Primary Authorization Key Transferred to SS by EAP method , during the authentication exchange
Pre-PAK – 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
AAA-key
Pre-PAK
Truncate (AAA-key, 160)
Dot16KDF (pre-PAK, SSID | BSID | "EIK+PAK", 288)
EIK (128 bits)
PAK (160 bits)
PMK
PAK
EIK
Dot16KDF (PMK, SSID | BSID | PAK | "AK", 160)
AK

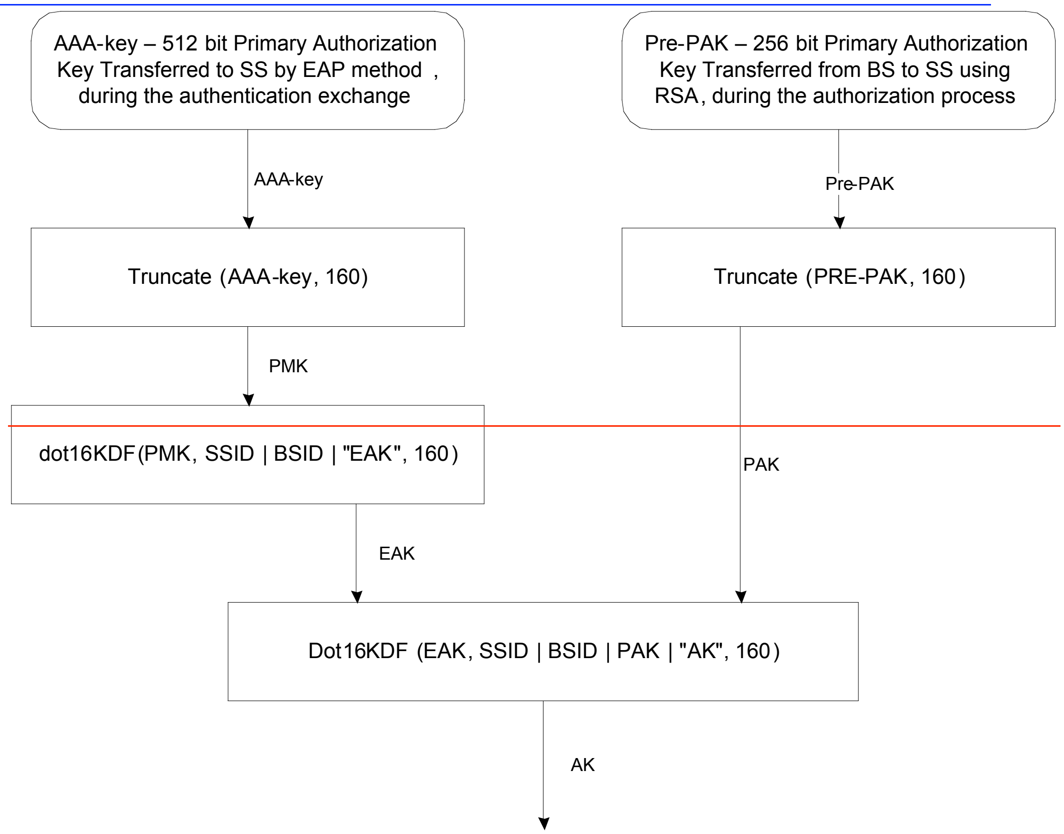

AAA-key – 512 bit Primary Authorization Key Transferred to SS by EAP method , during the authentication exchange
Pre-PAK – 256 bit Primary Authorization Key Transferred from BS to SS using RSA, during the authorization process
AAA-key
Pre-PAK
Truncate (AAA-key, 160)
Truncate (PRE-PAK, 160)
PMK
dot16KDF(PMK, SSID | BSID | "EAK", 160)
PAK
EAK
Dot16KDF (EAK, SSID | BSID | PAK | "AK", 160)
AK

Figure xx3 outlines the process to calculate the AK when only the EAP based authentication exchange has taken place, yielding an AAA-key:

Figure xx3: AK with EAP only authentication

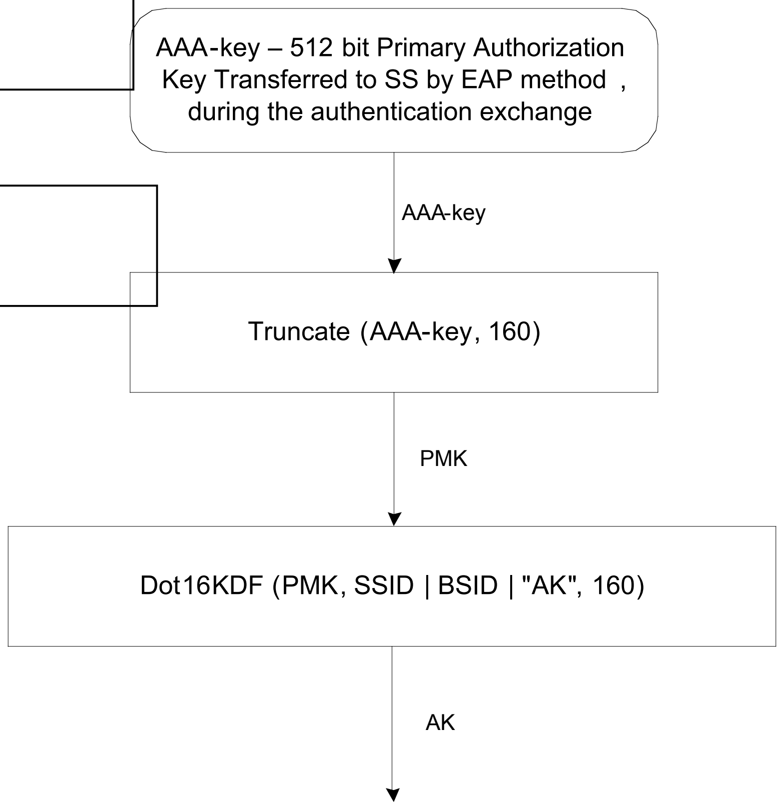


Figure xx4 outlines the unicast key hierarchy starting from the AK:

Figure xx4: ~~AK~~ HMAC/OMAC/KEK ~~key~~ derivation from AK

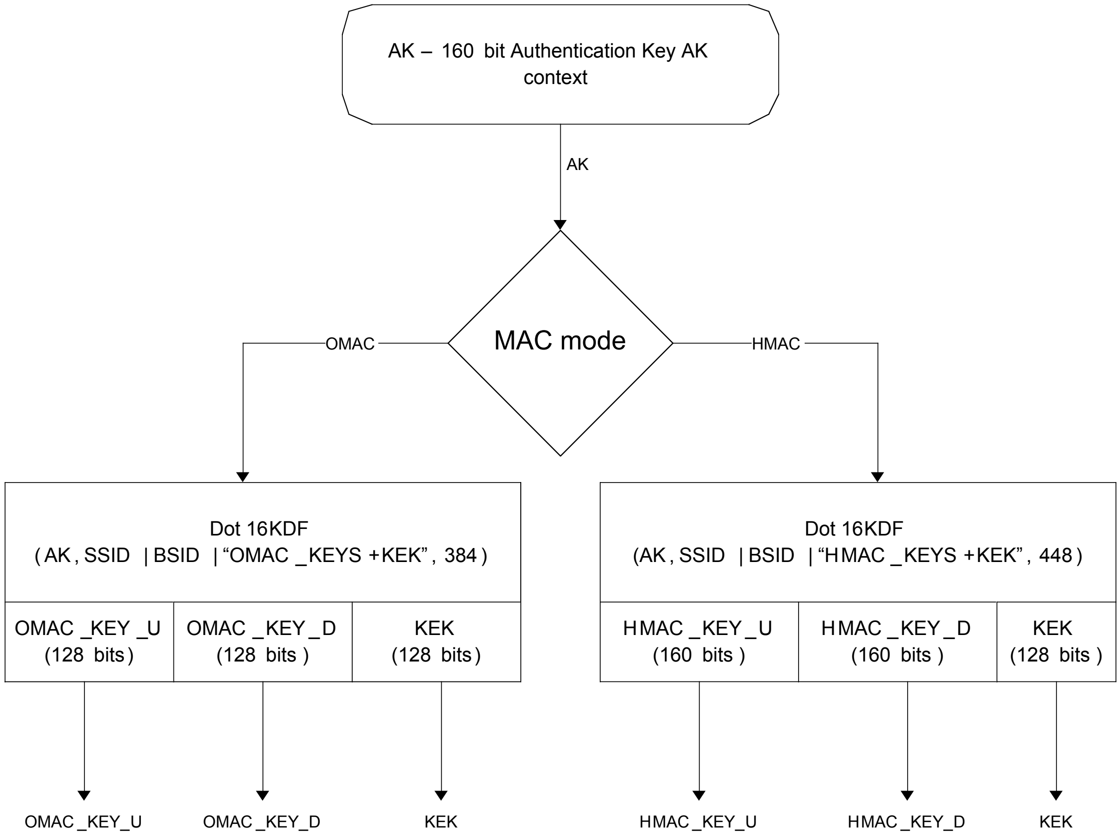
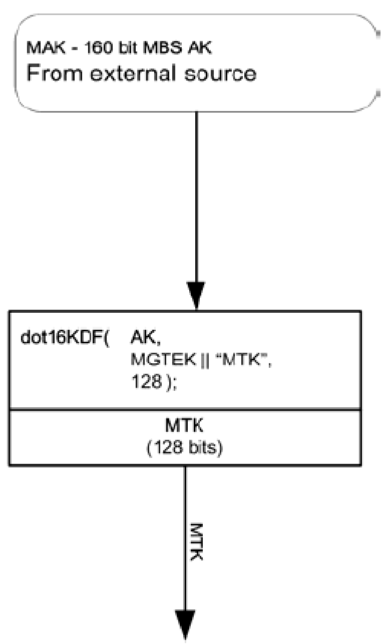


Figure xx5 outlines the MBS key hierarchies starting from the MAK:

Figure xx5: ~~AK~~ MTK key derivation from MAK



7.2.2.3 Associations

Keying material is held within associations. There are three types of association: The security associations (SA) that maintain keying material for unicast connections, group security associations (GSA) that hold keying material for multicast groups and MBSGSAs which hold keying material for MBS services.

7.2.2.3.1 Security Associations

A security association contains keying material that is used to protect unicast connections. The contents of an SA are:

The SAID, a 16 bit identifier for the SA. The SAID shall be unique within a BS.
The KEK, a 128 bit key encryption key, derived from the AK.
TEK0 and TEK1, 128 bit traffic encryption keys, generated within the BS and transferred from the BS to the SS using a secure key exchange.
The TEK Lifetimes TEK0 and TEK1, a key aging lifetime value.
PN0 and PN1, 32 bit packet numbers for use by the link cipher
RxPN0 and RxPN1, 32 bit receive sequence counter, for use by the link cipher.

7.2.2.3.2 Group Security Association

The Group Security Association (GSA) contains keying material used to secure multicast groups. These are defined separately from SAs since GSA offer a lower security bound that unicast security associations, since keying material is shared between all members of the group, allowing any member of the group to forge traffic as if it came from any other member of the group.

The contents of a GSA are:

The Group Key Encryption Key (GKEK). Serves the same function as an SA KEK but for a GSA
The Group Traffic Encryption Key (GTEK). Served the same function as an SA TEK but for a GSA.

7.2.2.3.3 MBS Group Security Association

The primary keying material in the MBS Group Security Association is the MAK. This serves the same function as the AK in the Authorized Association, however the MAK is provisioned by an external entity, such as an MBS server. The MAK may be common between members of an MBS group.

The contents of an MBSGSA are:

The MAK, a 160 bit MBS AK, serves the same function as the AK but local to the MBSGSA.

The MGTEK, a 128 bit MBS Group Traffic Encryption Key, used indirectly to protect MBS traffic. It is updated more frequently that the MAK.
The MTK, MBS Traffic Key, a 128 bit key used to protect MBS traffic, derived from the MAK and MGTEK.

7.2.2.4 Security context

The security context is a set of parameters linked to a key in each hierarchy that defines the scope while the key usage is considered to be secure.

Examples of these parameters are key lifetime and counters ensuring the same encryption will not be used more than once. When the context of the key expires, a new key should be obtained to continue working.

The purpose of this section is to define the context that belongs to each key, how it is obtained and the scope of its usage.

7.2.2.4.1 AK context

The context of AK includes all the parameters connected to AK and keys derived directly from it.
When one parameter from this context expires, a new AK should be obtained in order to start a new context.
Obtaining of new AK means re-authentication – doing the whole EAP and/or PAK due to the authorization policies negotiated between the MSS and BS until obtaining a new PMK and/or PAK which AK may be derived from.

Derivation of AK after HO is done separately in the MSS and network from a common PMK, PAK, SSID and BSID. The PMK and/or PAK may be used to derive keys to several BSs sharing the same PMK and/or PAK.

~~Obtaining of new AK PMK means re-authentication – doing the whole EAP until obtaining a new PMK from which (E)AK may be derived. from. The PMK may be used to derive keys to several BSs belonging to the same key-zone.~~
~~Derivation of EAK key after HO is done separately independently in the SS and network from a common PMK , SSID and BSID. The PMK may be used to derive keys to several BSs belonging to the same key-zone.~~
In HO scenario, if the SS was previously ~~already~~ connected to the TBS, the derived AK will be identical to the last one, as long as the PMK stays the same.
In order to maintain security in this scenario: the context of the AK must be cached by both sides and to be used from the point it stopped, if context lost by one side, re-authentication is needed to establish new PMK and new AK context
The AK context includes:

Parameter	**Size**	**Usage**
Primary AK (PAK)	160 bit	A key yielded from the mutual authorization exchange. Only present at initial network entry and only if the certificated RSA exchange took place,

		as a result of the mutual authorization policy negotiation.
PAKID	64 bits	De~~Ar~~rived from the mutual authorization, present when PAK is present.
PAK lifetime		Der~~Arr~~ived from the mutual authorization, present when PAK is present.
~~EAP AK (EAK)~~	~~160 bit~~	~~A key yielded from the EAP authentication. Always present~~
~~EAK lifetime~~		~~The lifetime of EAK, arrived from EAP.~~
PMK	160 bits	A key yielded from the EAP authentication.
PMK lifetime		The lifetime of PMK derived from EAP
PMKID	64 bits	hash 64(EAP session-id)
AK	160 bit	The authentication key, calculated as f(PAK,~~EAK~~PMK), if only EAP, AK=f(~~EAK~~PMK).
AKID	64bits	Calculated according to the keys that contributed to AK: - If AK=f(~~EAK~~PMK,PAK) then AKID=hash 64(EAP session-id | PAKID | BSID) - If AK=f(~~EAK~~ PMK) then AKID=hash 64(EAP session-id | BSID) - If AK=PAK then AKID = PAKID
AK lifetime		This is the time this key is valid, it is calculated AK lifetime=MIN(PAK lifetime, ~~EAK~~ PMK lifetime) – when this expires re-authentication is needed
H/OMAC_KEY_U	160/128 bit	The key which is used for signing UL management messages
H/OMAC_PN_U	32 bit	Used to avoid UL replay attack on management – when this expires re-authentication is needed
H/OMAC_KEY_D	160/128 bit	The key which is used for signing DL management messages
H/OMAC_PN_D	32 bit	Used to avoid rDL eply attack on management – when this expires re-authentication is needed
KEK	160 bit	Used to encrypt transport keys from the BS to the SS

7.2.2.4.2 GKEK context

The GKEK is the head of the group key hierarchy. There is a separate GKEK for each group (each GSA).
This key is randomly generated by the BS and transferred to the SS encrypted with KEK. It is used to encrypt group TEKs (GTEK) when broadcasting them to all SSs.
The GKEK context includes:

Parameter	Size	usage
GKEK	128 bit	Randomly generated by BS and transmitted to SS under KEK
GKEKID	64 bits	Arrives from BS with GKEK

GKEK lifetime		Arrives from BS with GKEK – when this expires a new GKEK should be obtained
H/OMAC_KEY_G	160/128 bit	The key which is used for signing group DL GTEK update messages, calculated by KDF(omac_PAD,GKEK)
H/OMAC_PN_G	32 bit	Used to avoid DL replay attack on management – – when this expires a new GKEK should be obtained

[Insert under the cryptographic algorithms section]

7.x.x.x.x Dot16KDF

The Dot16KDF algorithm is a CTR mode construction that may be used to derive an arbitrary amount of keying material from source keying material.

In the case that the HMAC/OMAC setting in the authentication policy bits is set to OMAC, the algorithm is defined as:

```
Dot16KDF(key, astring, keylength) is
{
	result = null;
	Kin = Truncate (key, 128);
	for (i = 0;i <= int((keylength-1)/128); i++)
	{
		result <= result | ~~Truncate~~ [s3]~~(~~OMAC(Kin, i | astring | keylength), 128);
	}
	return Truncate (result, keylength);
}
```

In the case that the HMAC/OMAC setting in the authentication policy bits is set to HMAC, the algorithm is defined as:

```
Dot16KDF(key, astring, keylength) is
{
	Kin = Truncate (key, 160);
	return Truncate (SHA-1(astring | Kin), keylength);
}
```

The key is a cryptographic key that is used by the underlying digest algorithm (SHA-1 or OMAC-AES). ‘astring’ is an octet string used to alter the output of the algorithm. ‘keylength’ is used to determine the length of key material to generate and is used in the digest input data to prevent extension attacks. Truncate(x,y) is the rightmost y bits of a value x only if y <= x.

[Change 11.9.3 as below to add in support for the AES Key Wrap algorithm]

11.9.3 TEK

Description: This attribute contains a quantity that is a TEK key, encrypted with a KEK derived from the AK.

Type	Length	Value (String)
8	8, 16 or 24	Encrypted TEK

When the TEK encryption algorithm identifier in the SA is 0x01, the length shall be 8 and the TEK shall be encrypted with 3DES in EDE mode according to the procedure defined in 7.5.2.1.

When the TEK encryption algorithm identifier in the SA is 0x03, the length shall be 16 and the TEK shall be encrypted with AES in ECB mode according to the procedure in 7.5.2.3

When the TEK encryption algorithm identifier in the SA is 0x04, the length shall be 24 and the TEK shall be encrypted with the AES Key Wrap algorithm according to the procedure in 7.5.2.4

[Insert an entry for AES Key Wrap into table 375a]

Table 375a—TEK encryption algorithm identifiers

Value	Description
0	Reserved
1	3-DES EDE with 128-bit key
2	RSA with 1024-bit key
3	ECB mode AES with 128-bit key
4	AES Key Wrap with 128 bit key
~~3~~5-255	Reserved

3.2 Remedy 2 – OMAC calculation correction

6.3.2.3.23 SS Basic Capability Request (SBC-REQ) message

[Replace the following section]

7.5.4 Calculation of OMAC-Digests

The calculation of the keyed hash in the OMAC-Digest attribute and the OMAC Tuple shall use the OMAC Algorithm with AES. The downlink authentication key OMAC_KEY_D shall be used for authenticating messages in the downlink direction. The uplink authentication key OMAC_KEY_U shall be used for authenticating messages in the uplink direction. Uplink and downlink message authentication keys are derived from the AK (see 7.5.4 below for details).

For authentication multicast messages (in the DL only) a OMAC_KEY_GD~~G~~ sha~~e~~ll be used (one for each group), group authentication key is derived from GKEK
In the PKM version 2 protocol, The OMAC sequence number in the OMAC tuple shall be equal to the ~~48~~64 bit AKID of the AK from which the OMAC_KEY_x was derived. In the PKM version 1 protocol, the 4 least significant bits of the OMAC sequence number in the OMAC tuple shall be equal to the 4 bit AK sequence number and the 44 most significant bits shall be equal to 0.
The OMAC Packet Number (OMAC_PN_*) is a 4 byte sequential counter that is incremented in the context of UL messages by the MSS, and in the context of DL messages by the BS,. The BS will also maintain a separate OMAC_PN_* for multicast packets per each GSA and increment that counter in the context of each multicast packet from the group. For MAC messages that have no CID e.g. RNG-REG message, the OMAC_PN_* context will be the same as used on the basic CID.
The OMAC_PN counters are part of the OMAC security context and must be unique for each MAC management message with the OMAC tuple or digest.
In the receiving side, the PN comparison will be made on CID basis meaning – a packet is considered valid it it's PN is higher than the PN of last message in the same CID (or any other mechanism defined for H-ARQ OOO problem) – in order to avoid replay attack between different CIDs, the CID is part of the calculation of the OMAC.
The digest shall be calculated over a field consisting of the OMAC key sequence number followed by the OMAC_PN_* ~~the frame number~~ , expressed as an unsigned 32 bit number, followed by the 16 bit Connection ID on which the message is sent, followed by 16 bit of zero padding (for the header to be aligned with AES block size) and followed by the entire MAC management message with the exception of the OMAC-TLV~~Digest but including the OMAC Tuple attributes~~.

The least significant bits of the digest shall be truncated to yield a 64 bit length digest.
The OMAC key sequence number is identical to the KEYid it was derived from i.e AKID if derived from AK.

I.e.:
OMAC digest <= Truncate64(OMAC(OMAC_KEY_*, OMAC sequence number | OMAC_PN_*~~| Frame number~~ | CID |16 bit zero padding | MAC_Management_Message~~| OMAC_TLV_Attributes~~))

If the message is included in an MPDU that has no CID, e.g. A RNG-REQ message, the CID used shall take the value ~~0~~of the basic CID.

~~The frame number in which a message containing an OMAC tuple may be fragmented and so be transmitted in more than one frame number. In this case, the frame number used in the OMAC calculation shall take the value of the frame number of the frame in which the first fragment is transmitted.~~

[Replace the following section]

11.1.2.2 OMAC Tuple

This parameter contains the OMAC_PN_* and ~~the OMAC Key Sequence Number~~ concatenated with an OMAC-Digest itself used for message authentication. The OMAC_PN_* is stored in the 32 least significant bits of ~~the tuple, followed with OMAC Key Sequence Number~~ which ~~is stored in the~~ next ~~48~~64 ~~bits of~~ the OMAC Tuple. The OMAC-Tuple attribute format is shown in Table 347 and Table 348.
When included in a MAC management message, the OMAC tuple shall always be the final tuple in the message.
A message received, that contains an OMAC tuple, shall not be considered authentic if the length field of the tuple is not ~~1720~~12, or if the locally computed value of the digest does not match the digest in the message.
Non authentic messages shall be discarded
Informative note: It would be appropriate for a MIB to increment an error count on receipt of a non authentic message, so that management can detect an active attack.

Table 345a—OMAC Tuple definition

Type	Length	Value	Scope
??	~~1420~~12	See Table 346a	DSx-REQ, DSx-RSP, DSx-ACK, REG-REQ, REG-RSP, RES-CMD, DREG-CMD, TFTP-CPLT

Table 346a—OMAC Tuple definition

Field	Length	Note
OMAC Packet Number counter, OMAC_PN_*	32 bits	This context is different er UL, DL
~~OMAC Key sequence number~~	~~48~~64 ~~bits~~	
OMAC Digest	64 bits	OMAC with AES 128

[Replace the following section]

11.9.32 OMAC Digest

Description: This attribute contains a packet number counter OMAC_PN_* incremented per packet on each direction and the Message Authentication Code used for message authentication. The OMAC algorithm is defined in draft SP 800-38B.

The OMAC digest includes

Type	Length	Value
40	~~8~~12	~~A 64-bit (8 byte) keyed OMAC~~See Table XXXX

Table XXXX—OMAC digest definition

Field	Length	Note
OMAC Packet Number counter, OMAC_PN_*	32 bits	This context is different in UL, DL
OMAC Digest	64 bits	OMAC with AES 128

3.3 Remedy 3 – 3 Way handshake and TEK exchange corrections

3.3.1 Summary of the solution:

Notation:

AK (Authentication Key)

This is the authentication key and can be created in three ways:

1) Just from PAK if only MSS authorization is done

2) Just from EAK if just EAP based user authentication is done

3) From PAK and EAK combined if both MSS authorization and user authentication are done.

AKID

This identifier is used to identify the shared AK, so depending upon how AK has been constructed it is also constructed based on three sources identified above.

RandomBS
A random value chosen by the BS (once per protocol run)

~~RandomSS~~NonceSS
A number ~~random value~~ chosen by the SS (once per protocol run). It can be counter or a random number.

KEK (Key Encryption Key)
This is a 128 bit key encryption key derived from the AK and is used to encrypt all TEK and GKEK exchanges between the MSS and the BS.

HMAC/OMAC
These keys are used for signing some of the management messages and are also derived from AK. OMAC keys are 128 bits long while HMAC keys are 160 bits long

TEK
These keys are randomly generated by the BS and transferred to the MSS encrypted by KEK and MACed (protected) by OMAC/HMAC. These are used to encrypt all the unicast traffic between MSS and BS.

GKEK
This key is randomly generated by the BS and transferred to the MSS encrypted by KEK and signed by OMAC/HMAC. This key is used to encrypt GTEK when it is sent from the BS as a multicast to MSSs that are members of a single multicast group.

GTEK
This key is randomly generated by the BS and transferred to all MSSs that are members of a single multicast group after encrypting it with GKEK. It can also be sent as a unicast to a single MSS, in which case it is encrypted with KEK. This key is used to encrypt multicast and broadcast traffic and is shared by all members of the group.

Protocol

1. BS → MSS Challenge:
    a. RandomBS
       This is an random number generated by the BS
    b. AKID
       This identifies the AK that is used for protecting this session.
    c. OMAC/HMAC Digest
       Message integrity tuple of this message.
2. MSS → BS Request:
    a. ~~RandomSS~~NonceSS
       A number chosen by the SS (once per protocol run). It can be counter or a random number. This is ~~a freshly generated random number generated by MSS that is~~ used by MSS to ensure the freshness of the corresponding reply from BS.

b. AKID
This identifies the AK to the BS that is used for protecting this message.
c. RandomBS
This is the random number that is returned by SS to BS in the Response..
d. Security Capabilities
Describe requesting MSS's security capabilities. This includes the data encryption and data authentication algorithms the MSS supports.
e. OMAC/HMAC Digest
Message integrity code for this message.

3. BS → MSS Response:
a. ~~RandomSS~~NonceSS
This is ~~a random number that was~~ passed to the BS in the request by MSS and is returned by BS to MSS in the response.
b. RandomBS
This is ~~a freshly generated r~~the random number generated by BS that is used by BS to ensure the freshness of the corresponding *optional* confirm message from MSS.
c. AKID
This identifies the AK to the BS that was used for protecting this message.
d. SA_TEK_Update
A TLV list each of which identifies the primary and static SAs, their SA identifiers (SAID) and additional properties of the SA (e.g., type, cryptographic suite) that the MSS is authorized to access. In case of HO, the details of any Dynamic SAs that the requesting MSS was authorized in the previous serving BS are also included.
Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included. Thus, SA_TEK_Update provides a shorthand method for renewing active SAs used by the MSS in its previous serving BS. The TLVs specify SAID in the target BS that shall replace active SAID used in the previous serving BS and also "older" TEK-Parameters and "newer" TEK-Parameters relevant to the active SAIDs. The update may also include multicast/broadcast Group SAIDs (GSAIDs), associated GTEK-Parameters pairs and GKEK for the each GSA.
In case of unicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number and the cipher block chaining (CBC) initialization vector. The TEKs are encrypted with KEK.
In case of broadcast or group SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of a GSAID's GTEK. This would include the newer GTEK, the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The type and length of the GTEK is equal to ones of the TEK. The GKEK should be identically shared within the same multicast group or the broadcast group. Contrary Key-Update Command, the GTEKs and GKEKs are encrypted with KEK because they are transmitted as a unicast here.
Multiple iterations of these TLVs may occur suitable to re-creating and re-assigning all active SAs and their (G)TEK pairs for the MSS from its previous serving BS. If any of the Security Associations parameters change, then those Security Associations parameters encoding TLVs that have changed will be added. When SA-TEK-Update is present in this message the SAID Update in REG-RSP shall not be present. When

SA-TEK-Update is absent in this message then the SAID Update shall be present in REG-RSP.

e. OMAC/HMAC Digest
   Message integrity code of this message.

~~*Optional* MSS → BS Confirm:~~

~~*This message is sent by MSS only if BS has included RandomBS in its Response.*~~

~~RandomBS~~

~~This is a random number that was passed to the MSS in the response by BS and is returned by MSS to BS in the confirm.~~

~~OMAC/HMAC Digest~~

~~Message integrity code of this message.~~

3.3.2 Changes to 802.16e D5a text:

[6.3.2.3.9 Change Table 26a – PKM Message Codes]

13	EAP Transfer	PKM-REQ
~~14~~	~~EAP Establish-Key Request~~	~~PKM-RSP~~
~~15~~	~~EAP Establish-Key Reply~~	~~PKM-REQ~~
~~16~~	~~EAP Establish-Key Reject~~	~~PKM-REQ~~
~~17~~	~~EAP Establish-Key Confirm~~	~~PKM-RSP~~
18	Pre-Auth-Request	PKM-REQ
19	Pre-Auth-Reply	PKM-RSP
20	Pre-Auth-Reject	PKM-RSP
21	PKMv2 Auth Request	PKM-REQ
22	PKMv2 Auth Reply	PKM-RSP
XX	**SA-TEK-Challenge**	**PKM-RSP**
X~~X~~Y	**SA-TEK-Request**	**PKM-REQ**
X~~Y~~Z	**SA-TEK-Response**	**PKM-RSP**
~~XZ~~	~~SA-TEK-Confirm~~	~~PKM-REQ~~

Remove EAP-Establish-Key-Request, EAP-Establish-Key-Reply, EAP-Establish-Key-Reject and EAP-Establish-Key-Confirm

[Remove 6.3.2.3.9.12, 6.3.2.3.9.13, 6.3.2.3.9.14 and 6.3.2.3.9.15 and replace it with following: -]

6.3.2.3.9.12 SA-Challenge message

The BS transmits the SA-Challenge message as a first step in the 3-way handshake at initial network entry and at reauthorization. It identifies an AK to be used for the Secure Association, and includes a random number challenge to be included by the MSS in its SA-TEK-Request.

Attribute	Contents
RandomBS	A freshly generated random number of 64 bits
AKID	This identifies the AK to the BS that was used for protecting this message.
OMAC/HMAC	Message integrity tuple for this message

6.3.2.3.9.13 SA-TEK-Request message

The MSS transmits the SA-TEK-Request message after receipt and successful HMAC/OMAC verification of an SA-Challenge from the BS. The SA-TEK_Request proves liveliness of the SS and its possession of the AK . If this message is being generated during initial network entry, then it constitutes a request for SA-Descriptors identifying the primary and static SAs and GSAs the requesting SS is authorized to access and their particular properties (e.g., type, cryptographic suite).

If this message is being generated upon HO, then it constitutes a request for establishment (in the target BS) of TEKs, GTEKs and GKEKs at the MSS and renewal of active primary, static and dynamic SAs and associated SAIDs used by the MSS in its previous serving BS.
~~The MSS transmits the SA-TEK-Request message as a first step in the 3-way handshake. If this message is being generated during initial network entry, then this is a request for SA-Descriptors identifying the primary and static SAs and GSAs the requesting SS is authorized to access and their particular properties (e.g., type, cryptographic suite).~~
~~If this message is being generated upon HO, then this is a request for establishment of TEKs, GTEKs and GKEKs at the MSS and renewal of active primary, static and dynamic SAs and associated SAIDs used by the MSS in its previous serving BS in the target BS.~~

Attribute	Contents
~~RandomSS~~NonceSS	A 64bit number chosen by the SS (once per protocol run). It can be counter or a random number.~~A freshly generated random number of 64 bits~~
RandomBS	The 64bit Random number from the SA Challenge
AKID	This identifies the AK to the BS that was used for protecting this message.
Security-Capabilities	Describes requesting MSS's security capabilities
OMAC/HMAC	Message integrity code of this message

The Security-Capabilities attribute is a compound attribute describing the requesting MSS's security capabilities. This includes the data encryption and data authentication algorithms the MSS supports.

6.3.2.3.9.14 SA-TEK-Response message

The BS transmits the SA-TEK-Response message as a second step in the 3-way handshake.

Attribute	Contents
~~RandomSS~~NonceSS	The ~~random~~ number received from MSS.
RandomBS	A freshly generated random number of 64 bits. This is optional.
AKID	This identifies the AK to the SS that was used for protecting this message.
SA_TEK_Update	A compound TLV list each of which specifies an SA identifier (SAID) and additional properties of the SA that the MSS is authorized to access. Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included.
OMAC/HMAC	Message integrity ~~code of~~ tuple for this message

SA_TEK_Update
A compound TLV list each of which identifies the primary and static SAs, their SA identifiers (SAID) and additional properties of the SA (e.g., type, cryptographic suite) that the MSS is authorized to access. In case of HO, the details of any Dynamic SAs that the requesting MSS was authorized in the previous serving BS are also included.
Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included. Thus, SA_TEK_Update provides a shorthand method for renewing active SAs used by the MSS in its previous serving BS. The TLVs specify SAID in the target BS that shall replace active SAID used in the previous serving BS and also "older" TEK-Parameters and "newer" TEK-Parameters relevant to the active SAIDs. The update

may also include multicast/broadcast Group SAIDs (GSAIDs) and associated GTEK-Paramters pairs.
In case of unicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number and the cipher block chaining (CBC) initialization vector. The TEKs are encrypted with KEK.
In case of group or multicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of a GSAID's GTEK. This would include the ~~old and~~ newer GTEK parameter pairs, ~~the GKEK,~~ the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The type and length of the GTEK is equal to ones of the TEK. The GKEK should be identically shared within the same multicast group or the broadcast group. The GTEKs and GKEKs are encrypted with KEK because they are transmitted as a unicast here.
Multiple iterations of these TLVs may occur suitable to re-creating and re-assigning all active SAs and their (G)TEK pairs for the MSS from its previous serving BS. If any of the Security Associations parameters change, then those Security Associations parameters encoding TLVs that have changed will be added.
This TLV may be sent in a single frame along with unsolicited REG-RSP.

~~6.3.2.3.9.12 SA-TEK-Confirm message~~

~~The MSS optionally transmits the SA-TEK-Confirm message in response to SA-TEK-Response message only if the SA-TEK-Response message contains RandomBS challenge.~~

~~Attribute~~	~~Contents~~
~~RandomBS~~	~~The random number received from BS~~
~~OMAC/HMAC~~	~~Message integrity code of this message~~

[Add 11.7.X]

11.7.X SA_TEK_Update

This field provides a translation table that allows an MSS to update its security associations and TEK pairs so that it may continue security service after a hand-over to a new serving BS.

Name	Type	Length (1 byte)	Value
SA_TEK_Update	?	Variable	Compound

The following TLV values shall appear in each SA_TEK_Update TLV.

Name	Type	Length (1 byte)	Value
SA_TEK_Update Type	??	1	1 : TEK parameters for a SA 2 : GTEK parametes for a GSA 3 to 255: Reserved, Not used
New SAID	20.1	2	New SAID after hand-over to new BS
Old SAID	20.1	2	Old SAID before hand-over from old BS. In case of initial network entry, old SAID is same as new SAID
Old TEK Parameters	13/GTEK	variable	"Older" generation of key

	Type?		parameters relevant to SAID. The Compound field contains the sub-attributes as defined in Table 370.
New TEK/GTEK –Parameters	13/GTEK Type?	variable	"Newer" generation of key parameters relevant to (G)SAID. The Compound field contains the sub-attributes as defined in Table 370.
GKEK Parameters	GKEK Type?	Variable	GKEK and its lifetime for the corresponding GTEK pair if this TLV is for a GSA

[Add section 7.8.X]

7.8.X SA-TEK 3-way handshake

Depending on mutual authorization/EAP, AK can be derived in three different ways as documented in section XXX. Before the 3-way handshake begins, the BS and MSS shall both derive a shared AK, KEK and HMAC/OMAC as per section XXX.

The SA-TEK 3-way handshake sequence proceeds as follows:

1. During initial network entry or reauthorization, the BS shall send *SA-Challenge to the MSS after* protecting it with the OMAC/HMAC tuple. If the BS does not receive *SA-TEK-Request* from the BS within SAChallenge*Timer*, it shall send another challenge. The BS may send *SA-Challenge* up to SAChallenge*MaxResends* times. If the BS reaches its maximum number of resends, it shall discard the AK and may initiate full re-authentication or drop the MSS.

2. During network re-entry or handover, the BS begins the 3-way-handshake by appending the SaChallenge TLV to the RNG-RSP. If the BS does not receive *SA-TEK-Request* from the BS within SaChallenge*Timer*, it shall discard the AK and may initiate full re-authentication or drop the MSS. If the BS receives *RNG-REQ* during the period that *SA-TEK-Request* is expected, it shall send a new RNG-RSP with another SaChallenge TLV.

3. The MSS shall send *SA-TEK-Request* to the BS after protecting it with the OMAC/HMAC. If the MSS does not receive *SA-TEK-Response* from the BS within SATEK*Timer*, it shall resend the request. The MSS may resend the *SA-TEK-Request* up to SATEK*RequestMaxResends* times. If the MSS reaches its maximum number of resends, it shall discard the AK and may do full re-authentication or decide to connect to another BS or take some other action. The message shall include RandomBS, ~~RandomSS~~NonceSS, AKID, SS's Security Capabilities and OMAC/HMAC.

4. Upon receipt of *SA-TEK-Request*, a BS shall confirm that the supplied AKID refers to an AK that it has available. If the AKID is unrecognized, the BS shall ignore the message. The BS shall verify the OMAC/HMAC. If the OMAC/HMAC is invalid, the BS shall ignore the message.

5. Upon successful validation of the *SA-TEK-Request*, the BS shall send *SA-TEK-Response* back to the MSS. The message shall include a compound TLV list each of which identifies the Primary and static SAs, their SA identifiers (SAID) and additional properties of the SA (e.g., type, cryptographic suite) that the MSS is authorized to access. In case of HO, the details of any Dynamic SAs that the requesting MSS was authorized in the previous serving BS are also included.
   Additionally, in case of HO, for each active SA in previous serving BS, corresponding TEK, GTEK and GKEK parameters are also included. Thus, SA_TEK_Update provides a shorthand method for renewing active SAs used by the MSS in its previous serving BS. The TLVs specify SAID in the target BS that shall replace active SAID used in the previous serving BS and also "older" TEK-Parameters and "newer" TEK-Parameters relevant to the active SAIDs. The update may also include multicast/broadcast Group SAIDs (GSAIDs) and associated GTEK-Paramters pairs.
   In case of unicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of an SAID's TEK. This would include the TEK, the TEK's remaining key lifetime, its key sequence number and the cipher block chaining (CBC) initialization vector. The TEKs are encrypted with KEK.
   In case of group or multicast SAs, the TEK-Parameters attribute contains all of the keying material corresponding to a particular generation of a GSAID's GTEK. This would include the GTEK, the GKEK, the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The type and length of the GTEK is equal to ones of the TEK. The GKEK should be identically shared within the same multicast group or the broadcast group. Contrary Key-Update Command, the GTEKs and GKEKs are encrypted with KEK because they are transmitted as a unicast here.
   Multiple iterations of these TLVs may occur suitable to re-creating and re-assigning all active SAs and their (G)TEK pairs for the MSS from its previous serving BS. If any of the Security Associations parameters change, then those Security Associations parameters encoding TLVs that have changed will be added.
   The OMAC/HMAC shall be the final attribute in the message's attribute list.

6. Upon receipt of *SA-TEK-Response*, an MSS shall verify the OMAC and ensure the presence of correct ~~RandomSS~~NonceSS. If the OMAC or ~~RandomSS~~NonceSS is invalid, the MSS shall ignore the message. Upon successful validation of the received *SA-TEK-Response*, the MSS shall install the received TEKs and associated parameters appropriately. Verification of OMAC is done as per section XXX. *If RandomBS was present in SA-TEK-Response, the MSS shall send SA-TEK-Confirm to the BS and an OMAC/HMAC digest.*

[Add following to 6.3.2.3.6 Ranging Response (RNG-RSP) message]

The following two fields will be present in RNG-RSP only when the corresponding RNG-REQ indicates the MSS is trying to do a re-entry or a handoff to this BS.

AKID

This identifies the AK that is used for protecting this message.

RandomBS

This is a random number generated by the BS that shall be later returned by the MSS in SA-TEK-Request message.

[Update Table 31-Key Request in section 6.3.2.3.9.5 as follows]

Attribute	Contents
~~Key-Sequence-Number~~	~~AK sequence number~~
AKID	This identifies the AK to the BS that was used for protecting this message
~~RandomSS~~NonceSS	A number chosen by the SS (once per protocol run). It can be counter or a random number. ~~This is a random number that was passed to the BS in the request by MSS and is returned by BS to MSS in the response.~~
SAID	Security Association ID
OMAC/HMAC-Digest	Message integrity code of this message

[Update Table 33-Key Reply in section 6.3.2.3.9.5 as follows]

Attribute	Contents
~~Key-Sequence-Number~~	~~AK sequence number~~
AKID	This identifies the AK to the BS that was used for protecting this message
~~RandomSS~~NonceSS	A number chosen by the SS (once per protocol run). It can be counter or a random number. This ~~is a random number that was passed to the BS in the key request by MSS and~~ is returned by BS to MSS
SAID	Security Association ID
TEK-Parameters	"Older" generation of key parameters relevant to SAID
TEK-Parameters	"Newer" generation of key parameters relevant to SAID
OMAC/HMAC-Digest	Message integrity code of this message

[Add following two new MAC management messages in section 6.3.2.3]

6.3.2.3.XX GSA Key Request Message

This message is sent by MSS to query the GTEK parameters from BS for a GSA.

Attribute	Contents
~~RandomSS~~NonceSS	A number chosen by the SS (once per protocol run). It can be counter or a random number.~~This is a random number that was passed to the BS in the request by MSS and is returned by BS to MSS in the response.~~
GSAID	Global Security Association ID
OMAC/HMAC-Digest	Message integrity code of this message

6.3.2.3.XX GSA Key Reply Message

This message is sent by BS to send the GTEK parameters in response to a query from MSS.

Attribute	Contents
~~RandomSS~~NonceSS	A number chosen by the SS (once per protocol run). It can be counter or a random number. This is returned by BS to MSS~~This is a random number that was passed to the BS in the request by MSS and is returned by BS to MSS in the response.~~

GSAID	Security Association ID
GTEK-Parameters	"Newer" generation of key parameters relevant to GSAID
GKEK-Parameters	Group Key Encryption Key protected by KEK derived from shared AK and other GKEK parameter e.g. Key lifetime.
OMAC/HMAC-Digest	Message integrity code of this message

[Replace 6.3.2.3.9.21 Key Update Command messages with follows]

6.3.2.3.9.21 Group Key Update Command messages

This message is sent by BS to push the GTEK and/or GKEK parameters.

Attribute	Contents
GSAID	Security Association ID
Key Push Modes	Usage code of Key Update Command message
Key Push Counter	Counter one greater than that of older generation
GTEK-Parameters	"Newer" generation of key parameters relevant to GSAID
GKEK-Parameters	Group Key Encryption Key protected by KEK derived from shared AK and other GKEK parameter e.g. Key lifetime.
OMAC/HMAC-Digest	Message integrity code of this message

GSAID is SAID for the multicast group or the broadcast group. The type and length of the GSAID is equal to ones of the SAID.

There are two types in the Group Key Update Command message, GKEK update mode and GTEK update mode. The former is used to update GKEK and the latter is used to update GTEK for the multicast service or the broadcast service. Key Push Modes indicates this usage code of the Group Key Update Command message. The Group Key Update Command message for the GKEK update mode is carried on the Primary Management connection, but one for the GTEK update mode is carried on the Broadcast connection. A few attributes in the Group Key Update Command message shall not be used according this Key Push Modes attribute's value. See 11.9.33 for details.

Key Push Counter is used to protect for replay attack. This value is one greater than that of older generation.

The Group Key Update Command message contains only newer generation of key parameters, because this message inform an MSS next traffic key material. The GTEK-Parameters attribute is a compound attribute containing all of the keying material corresponding to a newer generation of a GSAID's GTEK. This would include the GTEK, the GTEK's remaining key lifetime, the GTEK's key sequence number, and the cipher block chaining (CBC) initialization vector. The GTEK is TEK for the multicast group or the broadcast group. The type and length of the GTEK is equal to ones of the TEK. The GKEK (Group Key Encryption Key) can be randomly generated from a BS or an ASA server. The GKEK should be identically shared within the same multicast group or the broadcast group. The GTEK is encrypted with GKEK for the multicast service or the broadcast service. GKEK parameters contain the GKEK encrypted by the KEK and GKEK lifetime. See 7.5.4.4 for details.

The OMAC/HMAC-Digest attribute shall be the final attribute in the message's attribute list. Inclusion of the keyed digest allows the receiving client to authenticate the Group Key Update Command message. The OMAC/HMAC-Digest's authentication key is derived from the AK for the GKEK update mode and GKEK for the GTEK update mode. See 7.5.4.3 for details.

11.7.X GKEK Parameters

Description: This attribute is a compound attribute, consisting of a collection of sub-attributes. These sub-attributes represent all security parameters relevant to a particular generation of an GSAID's GKEK. A summary of the KEK-Parameters attribute format is shown below.

Name	Type	Length (1 byte)	Value
GKEK Parameters	?	Variable	Compound

Attribute	Contents
GKEK	GKEK, encrypted with KEK
Key-Lifetime	GKEK remaining lifetime

11.7.X SAChallengeTuple

This compound TLV enables the BS to abbreviate the 3-way handshake during handover by appending the initial challenge to the RNG-RSP message

Name	Type	Length (1 byte)	Value
SAChallenge	?	X	Compound

The following TLV values shall appear in each SaChallenge TLV.

Name	Type	Length (1 byte)	Value
RandomBS	??	8 bytes	
AKId	??	8 bytes	

~~3.4Remedy 4 – Pre-authentication support~~

~~[In IEEE P80216e_D5a modify table 26 – PKM message codes]~~

~~Table 26 – PKM message codes~~

~~Code~~	~~PKM message type~~	~~MAC Management message name~~
~~0-2~~	*~~Reserved~~*	~~—~~
~~3~~	~~SA Add~~	~~PKM-RSP~~
~~4~~	~~Auth Request~~	~~PKM-REQ~~
~~5~~	~~Auth Reply~~	~~PKM-RSP~~
~~6~~	~~Auth Reject~~	~~PKM-RSP~~
~~7~~	~~Key Request~~	~~PKM-REQ~~
~~8~~	~~Key Reply~~	~~PKM-RSP~~
~~9~~	~~Key Reject~~	~~PKM-RSP~~
~~10~~	~~Auth Invalid~~	~~PKM-RSP~~
~~11~~	~~TEK Invalid~~	~~PKM-RSP~~
~~12~~	~~Auth Info~~	~~PKM-REQ~~
~~13~~	~~EAP Transfer~~	~~PKM-REQ/PKM-RSP~~
~~14~~	~~EAP-Establish-Key-Request~~	~~PKM-RSP~~

~~15~~	~~EAP-Establish-Key Reply~~	~~PKM-REQ~~
~~16~~	~~EAP-Establish-Key Reject~~	~~PKM-REQ~~
~~17~~	~~EAP-Establish-Key Confirm~~	~~PKM-RSP~~
~~18~~	~~Pre-Auth-Request~~	~~PKM-REQ~~
~~19~~	~~Pre-Auth-Reply~~	~~PKM-RSP~~
~~20~~	~~Pre-Auth-Reject~~	~~PKM-RSP~~
~~18-20~~	~~*reserved*~~	~~—~~
~~21~~	~~PKMv2-Auth-Request~~	~~PKM-REQ~~
~~22~~	~~PKMv2-Auth-Reply~~	~~PKM-RSP~~
~~23~~	~~Key Update Command~~	~~PKM-RSP~~
~~24~~	~~EAP Start~~	~~PKM-REQ~~
~~25~~	~~EAP probe~~	~~PKM-REQ~~
~~26~~	~~EAP dest unreachable~~	~~PKM-RSP~~
~~27~~	~~EAP not needed~~	~~PKM-RSP~~
~~2425-255~~	~~*reserved*~~	~~—~~

~~*[Delete sections ‘6.3.2.3.9.16 Pre-Auth-Request message’, ‘6.3.2.3.9.17 Pre-Auth-Reply message’ and ‘6.3.2.3.9.18 Pre-Auth-Reject message’]*~~

~~*[Insert section 6.3.2.3.9.22]*~~

~~**6.3.2.3.9.22 EAP start**~~

~~When an MSS has to initiate an authentication process with a BS, it sends an EAP start message.~~

~~Code: 24~~

~~This message has no attribute.~~

~~*[Insert section 6.3.2.3.9.23]*~~

~~**6.3.2.3.9.23 EAP probe**~~

~~This message is a conditional start message, it is uses for pre-authentication only in case the SS does not know whether it has a PMK for this BS or not.~~
~~If the BS does not have a shared PMK of the SS, it will act as if he received EAP start message i.e. start EAP session.~~
~~If the BS does have a PMK of this SS he will send an “ EAP not needed” message which tell the SS the PMK ID it shares with it.~~
~~In case the SS does not have this PMK ID, it will use EAP_start to force the BS do authentication with it.~~
~~.~~

~~Code: 25~~

~~This message has no attribute.~~

~~*[Insert section 6.3.2.3.9.22]*~~

~~**6.3.2.3.9.24 EAP dest unreachable**~~

~~This message will be the response of the source BS to the SS in the case that target BS is nor reachable for pre-authentication.~~

~~Code: 26~~

~~This message has no attribute.~~

~~*[Insert section 6.3.2.3.9.22]*~~

~~**6.3.2.3.9.25 EAP not needed**~~
~~This message is uses to inform the SS that target BS already share a PMK with it, because there may be several PMK stored in the device, the BS notify the SS the PMKid of the shared PMK.~~

~~Code: 27~~

~~**Attribute**~~	~~**Contents**~~
~~PMKid~~	~~Id of already established PMK for this BS~~

~~*[insert text and tables as follows]*~~

~~**6.3.2.3.59 Privacy key management – remote (PKMR) messages (PKMR-REQ/PKMR-RSP)**~~

~~PKMR employs two MAC message types: PKMR Request (PKMR-REQ) and PKMR Response (PKMR-RSP), as described in Table xx1.~~

~~**Table xx1—PKMR MAC messages**~~

~~**Type Value**~~	~~**Message Name**~~	~~**Message Description**~~
~~??~~	~~PKMR-REQ~~	~~Privacy Key Management – Remote Request [MSS→ BS]~~
~~??~~	~~PKMR-RSP~~	~~Privacy Key Management – Remote Response [BS→ MSS]~~

~~These MAC management message types distinguish between PKMR requests (SS to BS) and PKMR responses (BS to SS). Each message encapsulates one EAP message in the Management Message Payload.~~

~~PKMR protocol messages transmitted from the SS to the BS (PKMR-REQ) shall use the form shown in Table xx2. They are transmitted on the MSSs Primary Management Connection.~~

~~**Table xx2—PKMR request (PKMR-REQ) message format**~~

~~**Syntax**~~	~~**Size**~~	~~**Notes**~~
~~PKMR-REQ message format {~~		
~~**Management Message Type = ??**~~	~~8 bits~~	
~~**Target BSID**~~	~~24 bits~~	~~Least significant 24 bits of the target BS ID~~
~~**Code**~~	~~8 bits~~	
~~**PKMR identifier**~~	~~8 bits~~	
~~**TLV Encoded Attributes**~~	~~*variable*~~	~~TLV specific~~
~~**OMAC/HMAC Tuple**~~	~~23/16~~	~~According to agreement in capabilities phase. This signature is calculated from keys used with the serving BS and will be verified by the serving BS~~
~~}~~		

~~PKMR protocol messages transmitted from the BS to the SS (PKMR-RSP) shall use the form shown in Table xx3. They are transmitted on the SSs Primary Management Connection.~~

~~Table xx3—PKMR response (PKMR-RSP) message format~~

~~**Syntax**~~	~~**Size**~~	~~**Notes**~~
~~PKMR-REQ message format {~~		
~~**Management Message Type = ??**~~	~~8 bits~~	
~~**Source BSID**~~	~~24 bits~~	~~Least significant 24 bits of the source BS ID~~
~~**Code**~~	~~8 bits~~	
~~**PKMR identifier**~~	~~8 bits~~	
~~**TLV Encoded Attributes**~~	~~*variable*~~	~~TLV specific~~
~~**OMAC/HMAC Tuple**~~	~~128 or 184~~	~~According to agreement in capabilities phase.~~ ~~**This signature is for the serving BS and not the target BS**~~
~~}~~		

~~The parameters shall be as follows:~~

~~**Code**~~

~~The Code is one byte and identifies the type of PKMR packet. When a packet is received with an invalid Code, it shall be silently discarded. The code values are defined in Table xx4.~~

~~**PKM Identifier**~~

~~The Identifier field is one byte. An SS uses the identifier to match a BS response to the SS's requests.~~

~~The SS shall increment (modulo 256) the Identifier field whenever it issues a new PKMR message. A "new" message is an PKMR-REQ that is not a retransmission being sent in response to a Timeout event. For retransmissions, the Identifier field shall remain unchanged.~~

~~The Identifier field in a BS's PKMR-RSP message shall match the Identifier field of the PKMR-REQ message the BS is responding to.~~

~~On reception of a PKMR-RSP message, the SS associates the message with a particular state machine (EAP stack for EAP messages).~~

~~**Attributes**~~

~~PKMR attributes carry the specific authentication, authorization, and key management data exchanged between client and server. Each PKMR packet type has its own set of required and optional attributes. Unless explicitly stated, there are no requirements on the ordering of attributes within a PKMR message. The end of the list of attributes is indicated by the LEN field of the MAC PDU header.~~

~~Table xx4—PKMR message codes~~

~~**Code**~~	~~**PKMR message types**~~	~~**MAC management message name**~~
~~13~~	~~EAP transfer~~	~~PKMR-REQ/PKMR-RSP~~
~~24~~	~~EAP start~~	~~PKMR-REQ~~

~~25~~	~~EAP probe~~	~~PKMR-REQ~~
~~26~~	~~EAP dest unreachable~~	~~PKMR-RSP~~
~~27~~	~~EAP not needed~~	~~PKMR-RSP~~

~~Formats for each of the PKMR messages are described in the following subclauses. The descriptions list the PKMR attributes contained within each PKMR message type. The attributes themselves are described in 11.9. Unknown attributes shall be ignored on receipt and skipped over while scanning for recognized attributes.~~
~~The BS shall silently discard all requests that do not contain ALL required attributes.~~

~~6.3.2.3.59.1 EAP Transfer message~~

~~When an MSS has an EAP message received from an EAP method for transmission to a remote BS or when a remote BS has an EAP message received from an EAP method for transmission to the MSS, it encapsulates it in an EAP Transfer message.~~

~~Code: 13~~

~~Attributes are shown in Table xx5.~~

~~Table xx5 – EAP transfer attributes~~

~~Attribute~~	~~Contents~~
~~EAP protocol~~	~~Contains the EAP authentication data, not interpreted in the MAC~~

~~The EAP Payload field carries data in the format described in section 4 of RFC2284bis~~

~~6.3.2.3.59.2 EAP Start message~~

~~When an MSS has to initiate an authentication process with a BS, it sends an EAP start message.~~

~~Code: 24~~

~~This message has no attribute.~~*~~[Insert section 11.9.?]~~*
~~11.9.? PMKid~~
~~This field contains the id of the PMK the BS has for a specific SS~~
~~Since PMK is unique per EAP session, the id is calculated from EAP session id~~

~~Type~~	~~Length~~	~~Value~~
~~XX~~	~~8~~	~~Hash64(EAP session id)~~

~~3.5~~3.4 Remedy 5 – HO optimization support

~~3.5.1~~3.4.1 Summary of the solution:

This contribution defines ~~one capability bit, three~~two configuration bits~~, and a key zone information field~~that notify the SS on what stages he can skip after after HO.~~.~~ The

serving BS sends these pieces of information to the MSS providing information regarding a potential target BS to which MSS may handoff.
Also defined here are some messages and decision flows that allow the MSS, based on the information it received, to determine how to obtain authentication with target BS, in advance, while maintaining high level of security.

Authentication
Obtaining authentication between an MSS and a BS is equivalent to sharing of a unique AK (and its context) between them. According to the key-hierarchy as defined in Remedy 1, the source for AK after HO is only EAP authentication which means AK=~~EAK~~ PMK~~which is derived from PMK~~. ~~For the purpose of this contribution, we will ignore EAK and consider that AK is derived directly from PMK.~~

Authentication Context caching
Authentication Context consists of AK and associated keying material derived from it and other context such as lifetimes, identifiers, etc.
In order to prevent certain replay attacks, the deriving of AK from PMK for a BS-SS pair is allowed only once during PMK lifetime. This implies that the lifetime of AK should be the same as that of PMK and that the Authentication Context needs to be cached for the PMK lifetime by both MSS and BS. This is because when AK is derived for the first time, a fresh context is created to support the usage of AK.
This context contains the key material such as key IDs, counters and key lifetimes.
If AK will be derived again from the same PMK and a fresh context will be created again, the system will not be secure because it will be vulnerable to replay attacks.
Thus the same AK and associated context must to be used in situations where MSS hands over to a BS it was previously connected to, when the corresponding PMK is still valid.
In case either if MSS or BS lose the AK and its associated context for some reason, it must invalidate the related PMK to prevent reuse.

~~Key-Zone~~
~~A key-zone is as set of one or more BSs that are populated, by the same authenticator, with a per-BS, unique AK derived itself from the PMK which is the product of the EAP authentication with the MSS. One way to have a common PMK and maintain security is by putting it in a central server that populates the BSs with their unique AK. The network architecture to support key-zone is out of scope here.~~
~~An important property of key-zone is that an MSS authenticated with one BS in a key-zone is automatically pre-authenticated with all the other BSs in the key-zone. A network can have multiple key-zones. A BS always belongs to a single key-zone which implies that for an MSS, a BS always has a single PMK source for AK.~~

~~Capability Bit: Pre-Authentication Support (1-bit)~~
~~This bit informs the MSS if the target BS has the pre-authentication capability. By default this bit is set unless a BS informs other member BSs in the key-zone that it cannot support this capability.~~

Configuration Bit: Omit ~~PMK~~PKM ~~Valid~~
This bit informs the MSS if ~~the target BS and current serving BS share a valid PMK~~it can OMIT PKM authentication phase when HO to the target BS. If this bit is set then ~~serving~~ SS and target BS share the same PMK ~~for the MSS and that~~thus MSS ~~MSS~~ may derive AK for the target BS from its current PMK or use an already derived AK+context if exists.

~~This bit can also serve as a notification to the MSS that a BS in the same key-zone lost the AK context for the MSS and that the MSS will need to re-authenticate and generate a fresh PMK.~~

~~Configuration Bit: Authentication Context Valid~~
~~This bit informs the MSS if a particular BS in a different key-zone has a valid context for the MSS. If the MSS had previously been connected to that BS then this bit implies that the BS has a valid Authentication context otherwise the MSS is just pre-authenticated with the BS and has a valid PMK if PMK Valid bit is set.~~
~~It is used by the MSS before HO to a target BS in a different key-zone to determine if it can use the context it holds for the BS or if it has to perform pre-authentication.~~

Configuration Bit: Skip TEK Exchange
This bit informs the MSS that it may reuse the existing TEKs from the current serving BS for traffic exchange with the target BS for the duration TEKs are valid. This bit is only valid if full authentication was skipped, otherwise new TEK exchange should be done

Information relevancy
The new defined configuration bits will be send in Broadcast and unicast messages which are relevant to the HO process.
The meaning of these bits may be different between messages because in Broadcast it will be only information relevant to all BSs and in unicast it may change according to the specific BS.
The messages are:

- NBR-ADV

~~The information regarding HO as defined above can be separated into 2 groups according to its relevancy:~~
~~The key-zone and the pre-authentication support are relevant to all MSSs connected to the BS and therefore will be transmitted in broadcast messages:~~

- ~~The key-zone will be transmitted in the DCD so it will be also included in the NBR-ADV.~~
- ~~The key-zone and per-authentication support will be transmitted in the NBR-ADV~~

~~The configuration has different values for each MSS and therefore will be sent in unicast messages to the MSS. The key-zone and pre-authentication capability will also be transmitted in the same unicast messages in order to cover the situations where the MSS needs the information but didn't have the chance to receive the broadcast yet.~~
~~The messages are:~~

- MOB-BSHO-REQ
- MOB_BSHO-RSP
- ~~MOB-BSHO-INF~~
- RNG_RSP (in target BS and only the configuration bits).

Using the information by the MSS
~~The MSS should maintain a table of key-zone to PMK mapping for each key-zone it has be through while the corresponding PMK is still valid.~~
When the MSS determines it needs to prepare for HO to a target BS (the way it does it is out of scope for this contribution) it ~~compares the target BS and its serving BS key-zones~~looks at the OMIT_PMK bit in the NBR-ADV:

- If the bit is set, it means the target BS has the same PMK as a source for it's AK. And therefore it should use the cached AK+context for this BS or derive AK and a fresh context from the PMK.
- If the bit is not set, it means the PMK of the target BS is different from the one the SS is currently using and therefore AK can't be derived from current PMK.

-
- ~~If the two BSs are in different key-zones the MSS should check if it has an Authentication Context cached for the target BS or if it has valid PMK for this key-zone (in that order).~~
  - ~~If MSS has a valid Authentication Context cached and the Authentication Context Valid bit of the BS is also set, the MSS can safely reuse the existing AK in the Authentication Context.~~
  - ~~If MSS does not have a valid Authentication Context cached or Authentication Context Valid bit of the BS is not set, the MSS shall use the PMK to derive new AK for the target BS after verifying that the corresponding "PMK Valid" bit is set too for the key-zone of the target BS.~~
  - ~~If MSS neither has a valid Authentication Context cached for the target BS nor a valid PMK for the key-zone of the target BS, it shall do pre-authentication with the target BS to generate a PMK if the pre-authentication capability bit is set for the target BS in the received configuration bits.~~

After actual HO, the MSS will get RNG-RSP signed with OMAC/HMAC using the AK the MSS derived before the HO.
If MSS can't verify the O/HMAC, it should discard this message and perform full network-entry.
If the O/HMAC is verified, the MSS shall check the Skip TEK Exchange configuration bit. If the bit is set MSS should continue using its old TEK keys and context. If it is unset MSS should ask for new TEK as described in 3-way-handsahke remedy of this contribution.

Besides RNG-RSP, the Skip TEK Exchange bit also appears in ~~MOB-BSHO-RSP and MOB-BSHO-REQ messages an~~HO related messages when connected to source BS. ~~d t~~The MSS can use it to assume what it will need to do after HO but the final decision is as defined in the RNG-RSP and the MSS must follow it.

Authentication Context loss behavior
As described above, the caching of Authentication Context is crucial for security and is valid as long as the corresponding PMK is valid.
In the case the Authentication Context is lost by one entity, that entity must invalidate PMK so there will be no possibility that the AK will be re-derived along with the fresh context.
There are ~~four~~ two options for Authentication Context loss and following are the ways to handle them:

- ~~The BS and MSS belong to the same key-zone and the~~ MSS loses the Authentication Context – the MSS shall erase ~~the entry of the corresponding PMK from its key-zone to PMK mapping and~~ all AKs derived from ~~it~~the PMK excepting the AK for the current serving BS and the MSS shall re-authenticate, thus renewing PMK for itself and all other BSs ~~in the key-zone~~whom has the same source PMK.
- ~~The BS and MSS belong to the same key-zone and the~~ BS loses the Authentication Context – the BS shall invalidate PMK of the MSS, the serving BS will notice that PMK is invalid and initiate a re-authentication with the MSS.

- ~~The BS and MSS belong to the different key-zone and the MSS loses the Authentication Context – the MSS shall erase the entry of the corresponding PMK from its key-zone to PMK mapping (and all AKs derived from it). Thus upon need to do HO to this key-zone the MSS shall notice it has no PMK and pre-authenticate.~~
- ~~The BS and SS belong to the different key-zone and the BS loses the Authentication Context – the BS shall invalidate the PMK of the MSS. When the MSS tries to HO to this key-zone, the Authentication Context Valid bit shall not be set and the MSS shall re-authenticate.~~

~~PKM optimization bit in HO optimization~~
~~This contribution changes the meaning of this bit. It will indicate if the HO optimization as described here is supported:~~

- ~~If this bit is set – it means all configuration bits appear and the mechanism should be used.~~
- ~~If this bit is not set – it means no optimization of security supported. The MSS shall perform full PKM phase at every HO.~~

~~3.5.2~~3.4.2 Changes to 802.16e D5a text ~~headlines~~:

[Add following new section to chapter 7]
[7.10 Key management upon HO with Optimization]
This section define the steps at network re-entry related to authentication and key management that an MSS can skip when the bit#1 i.e. Omit PKM ~~REQ/RSP~~ is set and/or bit #7 Skip TEK exchange~~.in the HO Process Optimization structure. Specifically, when this bit is set, it informs the MSS that Fast_HO Support and Fast_HO_Security_State bits are present and contains valid information.~~

Given the mechanisms presented in this chapter an MSS can do fast handoff and derive keying material for use at the target BS in a secure fashion. This section supplies mechanisms to establish and maintain a unique security context between the MSS and the target BS while still connected to the serving BS and thus allowing skipping of PKM steps in network re-entry while maintaining high level of security.

In order to achieve this goal, this section defines Fast_HO_Support which consists of ~~four~~two bits: Omit PKM authentication and skip TEK exchange.
~~key-zone information field and capability bit and Fast_HO_Security_State which consists of three configuration bits.~~ The serving BS sends these bits to the MSS regarding a potential target BS to which MSS may handoff. Given this knowledge, an MSS can determine its authentication status with the target BS and determine if it can obtain authentication with target BS in advance or if full EAP authentication is needed upon HO. By being able to skip full EAP authentication upon HO, the overall HO duration is greatly reduced.

~~7.10.1 Fast_HO_Support~~

~~7.10.1.1 Key-Zone~~
~~A key-zone is as set of one or more BSs that are populated, by the same authenticator, with a per-BS, unique AK derived itself from the PMK which is the product of the EAP authentication with the MSS. One way to have a common PMK and maintain security is by putting it in a central server that populates the BSs with their unique AK. The network architecture to support key-zone is not defined here. An important property of key-zone is that when an MSS authenticates with one BS in a key-zone it is automatically pre-authenticated with all remaining BS in the key-zone. A network can have multiple key-zones. A BS always belongs to a single key-zone which implies that for an MSS, a BS always has a single PMK source for AK.~~

~~7.10.1.2 Capability Bit: Pre-Authentication Support~~
~~This bit informs the MSS if the target BS has the pre-authentication capability. By default this bit is set unless a BS informs other member BSs in the key-zone that it cannot support this capability.~~

7.10.~~2~~1 ~~Fast_~~HO_~~Security_State~~Optimization bits

7.10.~~2~~1.1 ~~Configuration Bit: PMK Valid~~Bit #1 Omit PKM
This bit informs the MSS if the target BS and current serving BS share a valid PMK. If this bit is Set then serving and target BS share same PMK for the MSS and that MSS may

derive AK for the target BS from its current PMK or use an already derived AK+context if exists.
~~This bit can also serve as a notification to the MSS that a BS in the same key-zone lost the AK context for the MSS and that the MSS will need to re-authenticate and generate a fresh PMK.~~

~~7.10.2.2 Configuration Bit: Authentication Context Valid~~

~~This bit informs the MSS if a particular BS in a different key-zone has a valid context for the MSS. If the MSS had previously been connected to that BS then this bit implies that the BS has a valid Authentication context otherwise the MSS is just pre-authenticated with the BS and has a valid PMK if PMK Valid bit is set.~~
~~It is used by the MSS before handoff to a target BS in a different key-zone to determine if it can use the context it has for the BS or if it needs to do pre-authentication.~~

7.10.~~2.3~~1.2 ~~Configuration Bit:~~B Skip TEK Exchange

This bit informs the MSS that it may reuse the existing TEKs from the current serving BS for traffic exchange with the target BS for the duration TEKs are valid.

7.10.3 Key Zone to PMK mapping table

Each MSS maintains a key-zone to PMK mapping table that tell it which key-zones, and thereby its BSs, it is pre-authenticated with and can hence perform optimized HO. Key-zone zero is a special key-zone and means that the corresponding BSs don't belong to any key-zone.

7.10.4 Authentication Context

Authentication Context consists of AK and associated keying material derived from it and related context such as lifetimes, identifiers, etc.

7.10.5 Fast Handoff

Once an MSS determines it needs to handoff from the current serving BS, it transmits MOB-MSSHO-REQ message to the serving BS, who replies with MOB-BSHO-RSP that identifies a list of one or more potential target BSs for HO. Similarly, when the serving BS decides that MSS should handoff due to some reasons, the serving BS send MOB-BSHO-REQ to MSS that identifies a list of one or more potential target BSs for HO.

7.10.5.1 Fast Handoff Preparation

When the MSS determines it needs to prepare for HO to a target BS (the way it does it is out of scope for this contribution) it looks at the OMIT_PMK bit in the NBR-ADV:

- If the bit is set, it means the target BS has the same PMK as a source for it's AK. And therefore it should use the cached AK+context for this BS or derive AK and a fresh context from the PMK.
- If the bit is not set, it means the PMK of the target BS is different from the one the SS is currently using and therefore AK can't be derived from current PMK.

~~Upon receiving, the target BS list, the MSS compares the target BS key-zone with its own key-zone and then follows following steps:~~

- ~~If both serving and target BSs belong to the to the same key-zone and PMK Valid is set for the target BS in the MOB-BSHO-RSP or MOB-MSSHO-REQ messages, the MSS can safely derive the associated AK or re-use an already derived AK if it is valid. Optionally, the MSS can send HO-BS-QRY to the serving BS to verify that the PMK Valid is set for the target BS by examining the HO-BS-INF returned by the serving BS. If however, PMK Valid is not set in one of messages mentioned above the MSS will need to do full authentication after handoff. Alternatively,~~

~~the MSS may choose to try re-authenticating with the current serving BS in case the problem with the target BS is temporal and renewing PMK for the entire key-zone solves it.~~

- ~~If serving and target BSs belong to different key-zones the MSS shall check if it still has a valid Authentication Context cached for the target BS or if it just has valid PMK for this key-zone (in that order).~~
  ~~1a. If MSS has a valid Authentication Context cached and the Authentication Context Valid bit of the target BS is also set, the MSS can safely reuse the existing AK in the Authentication Context.~~
  ~~1b. If MSS does not have a valid Authentication Context cached or Authentication Context Valid bit of the BS is not set, the MSS shall use the PMK to derive new AK for the target BS after verifying that the corresponding "PMK Valid" bit is set too for the key-zone of the target BS.~~
  ~~1c. If MSS neither has a valid Authentication Context cached for the target BS nor a valid PMK for the key-zone of the target BS, it may perform pre-authentication with the target BS if the pre-authentication capability bit is set for the target BS in the received configuration bits.~~
  ~~2. Generate the corresponding AK at target BS and MSS.~~

7.10.5.2 Upon Fast Handoff

Upon actual handoff, the MSS shall follow following steps: -

1. The MSS shall receive a RNG-RSP signed with OMAC/HMAC using the AK that was derived before the HO. This implies that both MSS and BS have a valid AK before ranging is started.
2. If MSS can't verify the O/HMAC, it discards this message and performs full network-entry.
3. If the O/HMAC is verified, the MSS checks the Skip TEK Exchange configuration bit.
4. If Skip TEK Exchange bit is set the MSS may continue using its old TEKs and context. If it is not set the MSS shall ask for new TEKs.

7.10.6 Key Management upon Authentication Context loss from the cache

In order to prevent certain replay attacks, deriving AK from PMK for a BS-SS pair is allowed only once during PMK lifetime. This implies that the lifetime of AK shall be same as that of PMK and that the Authentication Context needs to be cached for the PMK lifetime by both MSS and BS. This is because when AK is derived for the first time, a fresh context is created to support the usage of AK. This context contains the key material such as key IDs, counters and key lifetimes. If AK is derived again from the same PMK and a fresh context will be created again, the system will not be secure because it will be vulnerable to replay attacks. Thus, the same AK and associated context needs to be re-used in situations where MSS handoffs to a previously connected serving BS when the corresponding PMK is still valid. In case either the MSS or BS loses the AK and associated context from some reason, PMK must be invalidated to prevent reuse.

The Authentication Context can be in ~~four~~two situations. Following bullets document the actions that need to be taken in each situation.

- MSS loses the Authentication Context:
  - The MSS shall erase all AKs derived from the PMK excepting the AK for the current serving BS and the MSS shall re-authenticate, thus renewing PMK for itself and all other BSs whom has the same source PMK.
- BS loses the Authentication Context:

- The BS shall invalidate PMK of the MSS, the serving BS will notice that PMK is invalid and initiate a re-authentication with the MSS.

- ~~The BS and MSS belong to the same key zone and the MSS loses the Authentication Context~~
  - ~~The SS shall erase the entry of the corresponding PMK from his key-zone to PMK mapping table and the MSS shall re-authenticate thus renewing PMK for itself and all the BSs in the key-zone.~~
- ~~The BS and MSS belong to the same key-zone and the BS loses the Authentication Context~~
  - ~~The BS shall invalidate PMK and the serving BS upon noticing that PMK is invalid shall initiate a re-authentication with the MSS.~~
- ~~The BS and MSS belong to the different key-zone and the MS loses the Authentication Context~~
  - ~~The MSS shall erase the entry of the corresponding PMK from his key-zone to PMK mapping table (and all AKs derived from it). Thus upon need to conduct handoff to a BS in the key-zone MSS will notice it does not have a PMK and will pre-authenticate.~~
- ~~The BS and SS belong to the different key-zone and the BS loses the Authentication Context~~
  - ~~The BS shall invalidate the PMK of the MSS. When the MSS tries to handoff to a BS in the key-zone, the Authentication Context Valid bit will not bet set and the MSS will pre-authenticate.~~

[Update 6.3.2.3.46]
[6.3.2.3.46 Neighbor Advertisement (MOB_NBR-ADV) message]
[In Table 106d add HO_Security_Context as shown below: -]

if (Skip-Optional-Fields[2]=0) {		
HO Process Optimization	8 bits	HO Process Optimization is provided as part of this message is indicative only. HO process requirements may change at time of actual HO. For each Bit location, a value of '0' indicates the associated reentry management messages shall be required, a value of '1' indicates the reentry management message may be omitted. Regardless of the HO Process Optimization TLV settings, the target BS may send unsolicited SBC-RSP and/ or REG-RSP management messages Bit #0: Omit SBC-REQ/RSP management Bit #2: Omit REG-REQ/ messages during current re-entry processing Bit #1: Omit PKM Authentication RSP management during current re-entry processing Bit #3: Omit Network Address Acquisition management messages during current reentry processing Bit #4: Omit Time of Day Acquisition management messages during current reentry processing Bit #5: Omit TFTP management messages

during current re-entry processing
Bit #6: Full service and operational state transfer or sharing between serving BS and target BS (ARQ, timers, counters, MAC state machines, etc…)
Bit #7: Omit TEK Exchange

}

[Update 6.3.2.3.51]
[6.3.2.3.51 BS HO Request (MOB_BSHO-REQ) message]
[In Table 106j add HO_Security_Context as shown below: -]

….		
If (Mode == 000) {		
N_Recommended	8 bits	
for (j=0 ; j<N_Recommended ; j++) {		N_Recommended can be derived from the known length of the message
Neighbor BSID	48 bits	
Service level prediction	8 bits	
Bit 1 : Omit PKM Authentication		
Bit 2 : Skip TEK Exchange		
}		
}		
else if (Mode == 001) {		
….		

[Update 6.3.2.3.53]
[6.3.2.3.53 BS HO Response (MOB-_BSHO-RSP) message]
[In Table 106l add HO_Security_Context as shown below: -]

….		
If (Mode == 0b000) {		
N_Recommended 8 bits		
For (j=0 ; j<N_Recommended ; j++) {		Neighbor base stations shall be presented in an order such that the first presented is the one most recommended and the last presented is the least recommended.
Neighbor BSID	48 bits	
Preamble index/ Preamble Present & Subchannel Index	8 bits	For the SCa and OFDMA PHY this parameter defines the PHY specific preamble for the neighbor BS. For the OFDM PHY the 5 LSB contain the active DL subchannel index for the neighbor BS. The 3 MSB shall be Reserved and set to '0b000'.
Service level prediction	8 bits	
HO Process Optimization	8 bits	HO Process Optimization is provided as part of this message is indicative only. HO process requirements may change at time of actual HO. For each Bit location, a value of '0' indicates the associated reentry management messages shall be required, a value of '1' indicates the reentry management message may be omitted. Regardless of the HO Process Optimization TLV settings, the target BS may send unsolicited SBC-RSP and/ or REG-RSP management messages Bit #0: Omit SBC-REQ/RSP management

		Bit #2: Omit REG-REQ/ messages during current re-entry processing Bit #1: Omit PKM Authentication RSP management during current re-entry processing Bit #3: Omit Network Address Acquisition management messages during current reentry processing Bit #4: Omit Time of Day Acquisition management messages during current reentry processing Bit #5: Omit TFTP management messages during current re-entry processing Bit #6: Full service and operational state transfer or sharing between serving BS and target BS (ARQ, timers, counters, MAC state machines, etc…) Bit #7: Omit TEK Exchange
}		
HO_ID_included_indicator	1 bit	Indicates if the field HO_IND is included
If (HO_ID_included_indicator == 1) {		
HO_ID	8 bits	ID assigned for use in initial ranging to the target BS once this BS is selected as the target BS
}		
}		
}		
else if (Mode == 0b001) {		
….		

~~**[6.3.2.3.6 Ranging Response (RNG-RSP) message]**~~
~~**[In section 6.3.2.3.6, Add following text after "HO_ID" text]**~~

~~The following TLV parameter shall be included in the RNG-RSP message when the MSS is attempting to perform initial network entry or handoff. When sent at the time of initial network entry only key-zone bits are valid and indicate the key-zone of the BS. Capability and configuration bits shall be zero and shall be ignored at initial network entry. But these bits shall be valid upon handoff if RNG-RSP is O/HMACed and successfully verified.~~

~~**Fast_HO_Support bits**~~

~~The bits in this field identify the key-zone and capability bits for a potential target BS.~~
~~Bits 0 thru 12 identify the key-zone for the target BS.~~
~~Bit 13 is the Pre-Authentication Support that informs the MSS if the target BS is capable of doing pre-authentication.~~

~~**Fast_HO_Security_State bits**~~

~~The bits in this field identify the configuration bits for a potential target BS.~~
~~Bit 1 is the PMK Valid bit. When this bit is set it informs the MSS that there is no need to do authentication and that AK may be derived from the existing PMK that the MSS and BS share. If this bit is not set, then MSS must do full authentication even if a valid AK exists. This bit is relevant only to BSs in the same key-zone.~~
~~Bit 2 is the Authentication Context bit. This bit is redundant in this message since if the MSS and BS are already pre-authenticated then this message would have been signed by the corresponding OMAC/HMAC. This bit is relevant only to BSs in a different key-zone than the serving BS.~~

~~Bit 3 is the Skip TEK Exchange bit. If this is not set, the MSS shall ask for new SAs and TEKs, otherwise it can continue to use the old state.~~

~~[11.6 RNG-RSP TLVs for re-establishment of service flows]~~
~~[Insert following entries in the RNG-RSP TLVs in table 365a as shown below:]~~

~~Table 365a—RNG-RSP message encodings~~

~~Name~~	~~Type (1 byte)~~	~~Length~~	~~Value (Vairable Length)~~
~~…~~	~~…~~	~~…~~	~~…~~
~~HO ID~~	~~22~~	~~1~~	~~…~~
~~Fast_HO_Support~~	**~~??~~**	**~~13 bits~~**	**~~These bits specify the key-zone and capability bits for a potential target BS.~~** **~~Bits 0 to Bits 12 : Key Zone Identifier for the BS~~** **~~Bit 13 : Pre-Auth Support~~**
~~Fast_HO_Security_State~~	**~~??~~**	**~~3 bits~~**	**~~These bits specify the configuration bits for a potential target BS.~~** **~~Bit 1 : PMK Valid~~** **~~Bit 2 : Authentication Context Valid~~** **~~Bit 3 : Skip TEK Exchange~~**
~~Location Update Response~~	~~23~~	~~1~~	~~…~~
~~…~~	~~…~~	~~…~~	~~…~~

~~[6.3.2.3.46 Neighbor Advertisement (MOB_NBR-ADV) message]~~
~~[In Table 106d add HO_Security_Context as shown below: -]~~

~~if (Skip-Optional-Fields[2]=0) {~~		
~~HO Process Optimization~~	~~8 bits~~	~~HO Process Optimization is provided as part of this message is indicative only. HO process requirements may change at time of actual HO. For each Bit location, a value of '0' indicates the associated reentry management messages shall be required, a value of '1' indicates the reentry management message may be omitted. Regardless of the HO Process Optimization TLV settings, the target BS may send unsolicited SBC-RSP and/ or REG-RSP management messages~~ ~~Bit #0: Omit SBC-REQ/RSP management messages during current re-entry processing~~ ~~Bit #1: Omit PKM-REQ/RSP management message during current re-entry processing~~ ~~Bit #2: Omit REG-REQ/RSP management during current re-entry processing~~ ~~Bit #3: Omit Network Address Acquisition management messages during current reentry processing~~ ~~Bit #4: Omit Time of Day Acquisition management messages during current reentry processing~~ ~~Bit #5: Omit TFTP management messages during current re-entry processing~~ ~~Bit #6: Full service and operational state transfer or sharing between serving BS and target BS (ARQ, timers, counters, MAC state machines, etc…)~~ ~~Bit #7: Reserved~~
~~if (Bit #1 is Set) {~~		
~~Fast_HO_Support~~	**~~13 bits~~**	~~Bits 0 to 12 : Key Zone Identifier for the BS~~ ~~Bit 13 : Pre-Auth Support~~
~~Reserved~~	**~~3 bits~~**	~~Shall be set to zero~~
~~}~~		
~~}~~		

~~[In section 6.3.2.3.47, Add following text after "Available Radio Resource" text]~~

~~Fast_HO_Support~~
~~Bits 0 to 12 of this bit-field specify the key-zone to which the neighbor BS belongs. The key-zone of the neighbor may be same as the serving BS transmitting this message.~~
~~Bit 13 of this field specifies whether neighbor BS support pre-authentication capability. If this bit is set, then the MSS can do pre-authentication with the neighbor BS and generate a PMK for the key-zone.~~

~~Since this message is broadcast to all MSS connected to the BS, this message does not contain configuration bits in Fast_HO_Security_State, namely, PMK Valid, Authentication Context Valid & Skip TEK Exchange bits since these bits convey MSS specific information in the target BS.~~

~~[6.3.2.3.51 BS HO Request (MOB_BSHO-REQ) message]~~
~~[In Table 106j add HO_Security_Context as shown below: -]~~

~~….~~
~~If (Mode == 000) {~~
~~N_Recommended 8 bits~~
~~for (j=0 ; j<N_Recommended ; j++) { N_Recommended can be derived from the known length of the message~~
~~Neighbor BSID 48 bits~~
~~Service level prediction 8 bits~~
~~**Fast_HO_Support 13 bits** These bits specify the key zone and capability bits for a potential target BS.~~
~~Bits 0 to 12 : Key Zone Identifier for the BS~~
~~Bit 13 : Pre-Auth Support~~
~~**Fast_HO_ Security_State 3 bits** These bits specify the configuration for a potential target BS.~~
~~Bit 1 : PMK Valid~~
~~Bit 2 : Authentication Context Valid~~
~~Bit 3 : Skip TEK Exchange~~
~~}~~
~~}~~
~~else if (Mode == 001) {~~
~~….~~

~~**[In section 6.3.2.3.47, Add following text after "Service level prediction" text]**~~

~~**Fast_HO_Support**~~
~~Bits 0 to 12 of this bit field specify the key-zone to which the neighbor BS belongs. The key-zone of the neighbor may be the same as the serving BS transmitting this message. Bit 13 of this field specifies whether neighbor BS support pre-authentication capability. If this bit is set, then the MSS can do pre-authentication with the neighbor BS and generate a PMK for the key-zone.~~

~~**Fast_HO_Security_State**~~
~~Bit 1 is the PMK bit Valid which when set informs the MSS that there is no need to authentication and that AK can be derived from the existing PMK that the MSS and BS share upon handoff. If this bit is not set, then MSS shall do full authentication even if a valid AK exists.~~
~~Bit 2 is the Authentication Context Valid bit and when set it informs the MSS that BS in the different key-zone holds the same keying material e.g. PMK and AK context as the MSS.~~
~~Bit 3 is the Skip TEK Exchange bit which when set informs MSS that it can reuse current TEKs at the target BS upon handoff. If this bit is not set, the MSS shall ask for new SAs and TEKs at the target BS upon handoff.~~

~~[6.3.2.3.53 BS HO Response (MOB- BSHO-RSP) message]~~
~~[In Table 106l add HO_Security_Context as shown below: -]~~

~~….~~
~~If (Mode == 0b000) {~~
~~N_Recommended 8 bits~~
~~For (j=0 ; j<N_Recommended ; j++) {~~ ~~Neighbor base stations shall be presented in an order such that the first presented is the one most recommended and the last presented is the least recommended.~~
~~Neighbor BSID 48 bits~~
~~Preamble index/ Preamble Present & Subchannel Index 8 bits~~ ~~For the SCa and OFDMA PHY this parameter defines the PHY specific preamble for the neighbor BS. For the OFDM PHY the 5 LSB contain the active DL subchannel index for the neighbor BS. The 3 MSB shall be Reserved and set to '0b000'.~~
~~Service level prediction 8 bits~~
~~HO process optimization 8 bits~~
~~**if (HO process optimization Bit #1 is Set) {**~~
~~**Fast_HO_Support 13 bits**~~ ~~These bits specify the key zone and capability bits for a potential target BS.~~
~~Bits 0 to 12 : Key Zone Identifier for the BS~~
~~Bit 13 : Pre-Auth Support~~
~~**Fast_HO_Security_State 3 bits**~~ ~~These bits specify the configuration for a potential target BS.~~
~~Bit 1 : PMK Valid~~
~~Bit 2 : Authentication Context Valid~~
~~Bit 3 : Skip TEK Exchange~~
~~}~~
~~HO_ID_included_indicator 1 bit~~ ~~Indicates if the field HO_IND is included~~
~~If (HO_ID_included_indicator == 1) {~~
~~HO_ID 8 bits~~ ~~ID assigned for use in initial ranging to the target BS once this BS is selected as the target BS~~
~~}~~
~~}~~
~~}~~
~~else if (Mode == 0b001) {~~
~~….~~

~~[In section 6.3.2.3.47, Add following text after "HO process optimization" text]~~

~~**Fast_HO_Support**~~
~~Bits 0 to 12 of this bit field specify the key-zone to which the neighbor BS belongs. The key-zone of the neighbor may be the same as the serving BS transmitting this message. Bit 13 of this field specifies whether neighbor BS support pre-authentication capability. If this bit is set, then the MSS can do pre-authentication with the neighbor BS and generate a PMK for the key-zone.~~

~~**Fast_HO_Security_State**~~
~~Bit 1 is the PMK bit Valid which when set informs the MSS that there is no need to authentication and that AK can be derived from the existing PMK that the MSS and BS share upon handoff. If this bit is not set, then MSS shall do full authentication even if a valid AK exists.~~

~~Bit 2 is the Authentication Context Valid bit and when set it informs the MSS that BS in the different key-zone holds the same keying material e.g. PMK and AK context as the MSS.~~
~~Bit 3 is the Skip TEK Exchange bit which when set informs MSS that it can reuse current TEKs at the target BS upon handoff. If this bit is not set, the MSS shall ask for new SAs and TEKs at the target BS upon handoff.~~

~~**[Add following two new MAC management messages in section 6.3.2.3]**~~

~~6.3.2.3.59 Query HO context HO-BS-QRY~~

~~HO-BS-QRY is sent from MSS to BS before initiating HO in order to determine the current HO_Security_Context for the neighbor BS that is maintained by the serving BS.~~

~~Syntax~~	~~Size~~	~~Notes~~
~~HO-BS-QRY{~~		
~~Management Message Type = ?~~	~~8 bits~~	
~~Neighbor BSID~~	~~48 bits~~	
~~OMAC Digest~~		~~Message Digest calculated using OMAC_KEY~~
~~}~~		

~~Neighbor BSID : Identifies the BS for which HO_Security_Context information is being queried.~~

~~6.3.2.3.60 Query HO context HO-BS-INF~~

~~HO-BS-INF is sent from BS to MSS in response to a HO-BS-QRY and contains the current HO_Security_Context for the neighbor BS that is maintained by the serving BS.~~

~~Syntax~~	~~Size~~	~~Notes~~
~~HO-BS-INF{~~		
~~Management Message Type = ?~~	~~8 bits~~	
~~Neighbor BSID~~	~~48 bits~~	
~~Fast_HO_Support~~	~~13 bits~~	~~These bits specify the key zone and capability bits for a potential target BS.~~ ~~Bits 0 to Bits 12 : Key Zone Identifier for the BS~~ ~~Bit 13 : Pre-Auth Support~~
~~Fast_HO_Security_State~~	~~3 bits~~	~~These bits specify the configuration bits for a potential target BS.~~ ~~Bit 1 : PMK Valid~~ ~~Bit 2 : Authentication Context Valid~~ ~~Bit 3 : Skip TEK Exchange~~
~~OMAC Digest~~		~~Message Digest calculated using OMAC_KEY~~
~~}~~		

~~**Neighbor BSID**~~

~~Identifies the BS to which the information in HO_Security_Context belongs.~~

~~**Fast_HO_Support**~~

~~Bits 0 to 12 of this bit-field specify the key-zone to which the neighbor BS belongs. The key-zone of the neighbor may be the same as the serving BS transmitting this message.~~
~~Bit 13 of this field specifies whether neighbor BS support pre-authentication capability. If this bit is set, then the MSS can do pre-authentication with the neighbor BS and generate a PMK for the key-zone.~~

~~**Fast_HO_Security_State**~~

~~Bit 1 is the PMK bit Valid which when set informs the MSS that there is no need to authentication and that AK can be derived from the existing PMK that the MSS and BS share upon handoff. If this bit is not set, then MSS shall do full authentication even if a valid AK exists.~~
~~Bit 2 is the Authentication Context Valid bit and when set it informs the MSS that BS in the different key-zone holds the same keying material e.g. PMK and AK context as the MSS.~~

~~Bit 3 is the Skip TEK Exchange bit which when set informs MSS that it can reuse current TEKs at the target BS upon handoff. If this bit is not set, the MSS shall ask for new SAs and TEKs at the target BS upon handoff.~~

~~3.5.3~~3.4.3 Remedy 6 – Mutual Authorization support

Update 6.3.2.3.9.19 PKMv2 Authorization Request (Auth Request) message

Code: 21

Sent by the SS to the BS as the first frame to ask for authorization.

Attributes are shown in Table 37i

Table xx

Attribute	Contents
MSS_RANDOM	A 64 bit random number generated in the SS
MSS Certificate	Contains the SS's X.509 user certificates
Security ~~Capabilities~~ suites	Describes requesting SS's security capabilities (Optional, only if there is no EAP phase afterwards)
~~AAID~~ SAID	SS's primary SAID equal to the Basic CID

The SS-certificate attribute contains an X.509 SS certificate (See 7.6) issued by the SS's manufacturer. The SS's X.509 certificate and Security Capabilities attribute is as defined 6.3.2.3.9.2

6.3.2.3.9.20 Auth-Reply message

Sent by the BS to a client MSS in response to an Authorization Request, the Authorization Reply message contains a pre-PAK, the key's lifetime, the key's sequence number, and a list of SA-Descriptors identifying the Primary and Static SAs the requesting MSS is authorized to access and their particular properties (e.g., type, cryptographic suite). The pre-PAK shall be encrypted with the MSS's public key. The SA-Descriptor list shall include a descriptor for the ~~Basic CID~~ primary SA reported to the BS in the corresponding Auth-Request. The SS_Random number is returned from the auth-req message, along with a random number supplied by the BS, thus enabling assurance of key liveness.

Code: 22

Attributes are shown in Table 37j.

Table 37j

Attribute	Contents
MSS_RANDOM	A 64 bit random number generated in the SS

BS_RANDOM	A 64 bit random number generated in the BS
~~MSS_Certificate~~	~~Contains the MSS's X.509 user certificate~~
Encrypted pre-PAK	RSA-OAEP-Encrypt(PubKey(MSS), pre-PAK | Id(MSS))
PAK Lifetime	PAK aging timer
PAK Sequence Number	64bit PAK Sequence number
~~AAID/SAID~~ (one or more) SA-Descriptor(s)	The primary SA and zero or more static SAs. Each compound SA-Descriptor attribute specifies an SAID and additional properties of the SA (optional, only if there is no EAP phase afterwards)
BS Certificate	Contains the BS's X.509 certificate
SigBS	An RSA signature over all other attributes in the message

6.3.2.3.9.XX PKMv2 Authorization Acknowledgement (Auth Ack) message

Code: X+2

Sent by the SS to BS as an acknowledgement of successful BS Authorization.

Table xx+2

Attribute	Contents
BS_RANDOM	A 64 bit random number generated in the BS
SS_MAC_Address	Contains the SS's MAC address
OMAC Tuple	OMAC calculated using OMAC key derived from PAK.

7.8.2.2 MSS and BS mutual authorization and AK exchange overview

The BS mutual authorization can take place in 2 cases: The first case is if this is the only mechanism used for authorization and in this case it will be performed upon any network (re)entry.

The second case is when it followed by EAP authentication: in this case the mutual authorization is done only for initial NW-E and only EAP is done in case authentication is needed in re-entry.

MSS mutual authorization, controlled by the PKMv2 Authorization state machine, is the process of

a) The BS authenticating a client MSS's identity

b) The MSS authenticating the BS's identity

c) The BS providing the ~~authenticated~~ Authorized MSS with a~~n~~ pre-PAK, from which a key encryption key (KEK) and message authentication keys ~~are derived~~ can be derived (If EAP is also done the AK and sub-keys is derived from both key sources)

d) The BS providing the authenticated MSS with the identities (i.e., the SAIDs) and properties of primary and static SAs the MSS is authorized to obtain keying information for.

After achieving initial authorization and in case this is the only authorization method used, the MSS periodically seeks reauthorization with the BS; reauthorization

is also managed by the MSS's PKMv2 Authorization state machine. An MSS must maintain its authorization status with the BS in order to be able to refresh aging TEKs and GTEKs. TEK state machines manage the refreshing of TEKs.
The MSS sends an Authorization Request message to its BS immediately after sending the Authentication Information message. This is a request for a~~n~~ pre-AK, as well as for the SAIDs identifying any Static Security SAs the MSS is authorized to participate in. The Authorization Request includes
a) a manufacturer-issued X.509 certificate
b) a description of the cryptographic algorithms the requesting MSS supports; an MSS's cryptographic ~~capabilities~~ suites are presented to the BS as a list of cryptographic suite identifiers, each indicating a particular pairing of packet data encryption and packet data authentication algorithms the MSS supports
c) the MSS's primary Said (equals Basic CID). The Basic CID is the first static CID the BS assigns to an MSS during initial ranging~~—the primary SAID is equal to the Basic CID.~~
In response to an Authorization Request message, a BS validates the requesting MSS's identity, determines the encryption algorithm and protocol support it shares with the MSS, activates a~~n~~ pre-PAK for the MSS, encrypts it with the MSS's public key, and sends it back to the MSS in an Authorization Reply message. Randomnumbers are included in the exchange to ensure liveness. The pre-PAK will be used to derive the PAK and optionally the EEK and EIK for EAP exchange protection (see EAP authentication section).
In response to an PKMv2 Authorization reply message, a SS shall validates the replying BS's identity by X.509 digital certificate, and authenticating the message by running hash function defined in RSA hash function with BS's private key. SS may acknowledge Authorization Reply by sending Authorization Acknowledgement or Authorization Reject
The PKMv2 authorization acknowledge includes:
a) a 64 bits random number received in auth reply
b) SS MAC address
c) OMAC Digest (used PAK for OMAC key derivation and 0 as PN)
The BS shall determine successful mutual authorization upon receiving PKMv2 Authorization acknowledgement message. If the PKMv2 Auth Ack, Auth Reject, or further PKM message such as EAP or KEY Request message is not received during certain time skew, BS may remove authorization state according to the operator policy

An~~,~~ MSS for which this is its only authorization method ,shall periodically refresh its PAK by reissuing an Authorization Request to the BS. Reauthorization is identical to authorization. To avoid service interruptions during reauthorization, successive generations of the MSS's AKs have overlapping lifetimes. Both MSS and BS shall be able to support up to two simultaneously active AKs during these transition periods. The operation of the Authorization state machine's
Authorization Request scheduling algorithm, combined with the BS's regimen for updating and using a client MSS's AKs (see 7.4), ensures that the MSS can refresh TEK keying information without interruption
over the course of the MSS's reauthorization periods.

[Modify Table 368a as follows:]

Table 368a—PKM attribute types

Type	PKM Attribute
~~31~~	~~AA Descriptor~~

~~32~~	~~AA-Type~~

7.8.2.2 MSS and BS mutual authorization and AK exchange overview

MSS mutual authorization, controlled by the PKMv2 Authorization state machine, is the process of

a) The BS authenticating a client MSS's identity
b) The MSS authenticating the BS's identity
c) The BS providing the authenticated MSS with an AK, from which a key encryption key (KEK) and message authentication keys are derived
d) The BS providing the authenticated MSS with the identities (i.e., the SAIDs) and properties of primary and static SAs the MSS is authorized to obtain keying information for.

After achieving initial authorization, an MSS periodically seeks reauthorization with the BS; reauthorization is also managed by the MSS's PKMv2 Authorization state machine. An MSS must maintain its authorization status with the BS in order to be able to refresh aging TEKs and GTEKs. TEK state machines manage the refreshing of TEKs.
The MSS sends an Authorization Request message to its BS immediately after sending the Authentication
Information message. This is a request for an AK, as well as for the SAIDs identifying any Static Security SAs the MSS is authorized to participate in. The Authorization Request includes (see 6.3.2.3.9.19)

a) a manufacturer-issued X.509 certificate
b) a description of the cryptographic algorithms the requesting MSS supports; an MSS's cryptographic capabilities are presented to the BS as a list of cryptographic suite identifiers, each indicating a particular pairing of packet data encryption and packet data authentication algorithms the MSS supports
c) the MSS's Basic CID. The Basic CID is the first static CID the BS assigns to an MSS during initial ranging—the primary SAID is equal to the Basic CID
d) A 64-bit random number generated in the MSS

In response to an Authorization Request message, a BS validates the requesting MSS's identity, determines the encryption algorithm and protocol support it shares with the MSS, activates an AK for the MSS, encrypts it with the MSS's public key, and sends it back to the MSS in an Authorization Reply message. Random numbers are included in the exchange to ensure liveness. The Authorization Reply includes (see 6.3.2.3.9.20)

a) the BS's X.509 certificate, used to verify the BS's identity
b) a pre-PAK encrypted with the MSS's public key
c) a 64-bit PAK sequence number, used to distinguish between successive generations of AKs
d) a PAK lifetime
e) the identities (i.e., the SAIDs) and properties of the single primary and zero or more static SAs the MSS is authorized to obtain keying information for
f) the 64-bit random number generated in the MSS

g) a 64-bit random number generated in the BS, used to ensure key of liveness along with the random number of MSS

h) the RSA signature over all the other attributes in the auth-reply message by BS, used to assure the reality of two PKMv2 authorization messages.

An MSS shall periodically refresh its AK by reissuing an Authorization Request to the BS. Reauthorization is identical to authorization. To avoid service interruptions during reauthorization, successive generations of the MSS's AKs have overlapping lifetimes. Both MSS and BS shall be able to support up to two simultaneously active AKs during these transition periods. The operation of the Authorization state machine's Authorization Request scheduling algorithm, combined with the BS's regimen for updating and using a client MSS's AKs (see 7.4), ensures that the MSS can refresh TEK keying information without interruption over the course of the MSS's reauthorization periods.

~~3.6~~3.5 Remedy 7 – Authorization Policy Support Negotiation

[In IEEE P80216e_D5a p411, line 1, change text and table as follows]

11.8.4 Authorization policy support

This field indicates authorization policy used by the MSS and BS to negotiate and synchronize. A bit value of 0 indicates "not supported" while 1 indicates "supported."

Type	Length	Value	Scope
5.25	1	Bit #0: RSA (if PKM version 1, unidirectional authorization, in PKM version 2, mutual authorization) Bit #1: EAP (if PKM version 1 can not be set with RSA bit) ~~Bit #2: PHY frame number in authentication tuple~~ ~~Bit #3: Mutual Auth/Unidirectional Auth~~ Bit #2: OMAC supported (if set to 0, HMAC is the default) Bit #3-7: R*eserved.* Set to 0	SBC-REQ SBC-RSP