

3GPP2 S.R0138-0

Version 1.0

Version Date: May 14, 2009



3RD GENERATION
PARTNERSHIP
PROJECT 2
"3GPP2"

eHRPD Security Framework

© 3GPP2 2009

COPYRIGHT NOTICE

3GPP2 and its Organizational Partners claim copyright in this document and individual Organizational Partners may copyright and issue documents or standards publications in individual Organizational Partner's name based on this document. Requests for reproduction of this document should be directed to the 3GPP2 Secretariat at secretariat@3gpp2.org. Requests to reproduce individual Organizational Partner's documents should be directed to that Organizational Partner. See www.3gpp2.org for more information.

EDITOR

1 Violeta Cakulev
2 Alcatel Lucent
3 Tel: (+1) 908-582-3207
4 Email: cakulev@alcatel-lucent.com
5

6 **REVISION HISTORY**

7

REVISION HISTORY		
1.0	<i>Initial publication</i>	<i>May 2009</i>

8

Table of Contents

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34

eHRPD Security Framework.....	i
1 Introduction.....	1
1.1 Scope	1
2 References.....	1
2.1 Normative References	1
2.2 Informative References.....	2
3 Definitions, Abbreviations and Acronyms.....	2
3.1 Definitions	2
3.1.1 Abbreviations and Acronyms	2
4 Architecture	3
5 Access Authentication and Authorization.....	5
5.1 EAP Protocol Negotiation	5
5.2 UE Behavior	6
5.2.1 UE Identity Management	6
5.2.2 UE Network Access Authentication.....	6
5.3 HSGW Behavior.....	7
5.4 3GPP AAA Server Behavior	7
5.5 HSS Behavior	8
5.6 Call Flows.....	8
5.6.1 Initial Authentication.....	8
5.6.2 Fast Re-Authentication.....	13
6 Key Generation	15
6.1 Pairwise Master Key (PMK) Generation.....	16
6.2 Access Network Key Generation.....	17
7 Key Distribution	18
7.1 Master Session Key and Inter-HSGW Handoff.....	18
7.2 HSGW – eAN Key Distribution	19
7.3 Multi-Key Key Exchange Protocol and Intra-HSGW inter-eAN Handoff.....	19
7.4 Call Flows.....	20
7.4.1 HSGW – eAN Key Distribution.....	20
7.4.2 Multi-Key Key Exchange Protocol	21

1 Introduction

This document defines eHRPD (evolved HRPD) Security framework. This document describes only normal operation. Handling of error cases and unsuccessful scenarios resulting from protocol failures is described in other relevant standards.

This is an informative document. If there is a misalignment between this document and procedures defined in normative reference documents, the normative reference documents take precedence.

1.1 Scope

This document describes security mechanisms in eHRPD.

2 References

2.1 Normative References

- [1] **3GPP2:** X.S0057-0 v1.0, “E-UTRAN – eHRPD Connectivity and Interworking: Core Network Aspects”, January 2009.
- [2] **3GPP2:** C.S0067-A v1.0, “Key Exchange Protocols for cdma2000 High Rate Packet Data Air Interface”, February 2009.
- [3] **3GPP2:** S.S0078-B v1.0, “Common Security Algorithms”
- [4] **IETF:** RFC1661: “The Point-to-Point Protocol (PPP)”, July 1994.
- [5] **IETF:** RFC3588: “Diameter Base Protocol”, September 2003.
- [6] **IETF:** RFC3748: “Extensible Authentication Protocol (EAP)”, June 2004.
- [7] **IETF:** RFC4005: “Diameter Network Access Server Application”, August 2005.
- [8] **IETF:** RFC4072: “Diameter Extensible Authentication Protocol (EAP) Application”, August 2005.
- [9] **IETF:** RFC4187: “Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)”, January 2006.
- [10] **IETF:** RFC4282: “The Network Access Identifier”, December 2005.
- [11] **IETF:** draft arko-eap-aka-kdf, “Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)”.

[Editor’s Note: The above document is a work in progress and should not be referenced unless and until it is approved and published. Until such time as this Editor’s Note is removed, the inclusion of the above document is for informational purposes only.]
- [12] **3GPP:** TS 33.402: “Security aspects of non-3GPP accesses (Release 8)”
- [13] **3GPP:** TS 23.003: “Numbering, addressing and identification (Release 8)”

- 1 [14] **3GPP:** TS 24.302: “Access to the 3GPP Evolved Packet Core (EPC) via
2 non-3GPP access networks; Stage 3; (Release 8)”
- 3 [15] **3GPP2:** A.S0022-0 v1.0, “E-UTRAN – eHRPD Connectivity and
4 Interworking: Access Network Aspects (E-UTRAN – HRPD IOS)”, January
5 2009.
6

7 **2.2 Informative References**

8 This section provides references to other documents that may be useful for the reader of this
9 document.

10

11 **3 Definitions, Abbreviations and Acronyms**

12 This section contains definitions, symbols and abbreviations that are used throughout the
13 document.

14 **3.1 Definitions**

15

16 **3.1.1 Abbreviations and Acronyms**

17 The following list provides abbreviations and acronyms used throughout this document

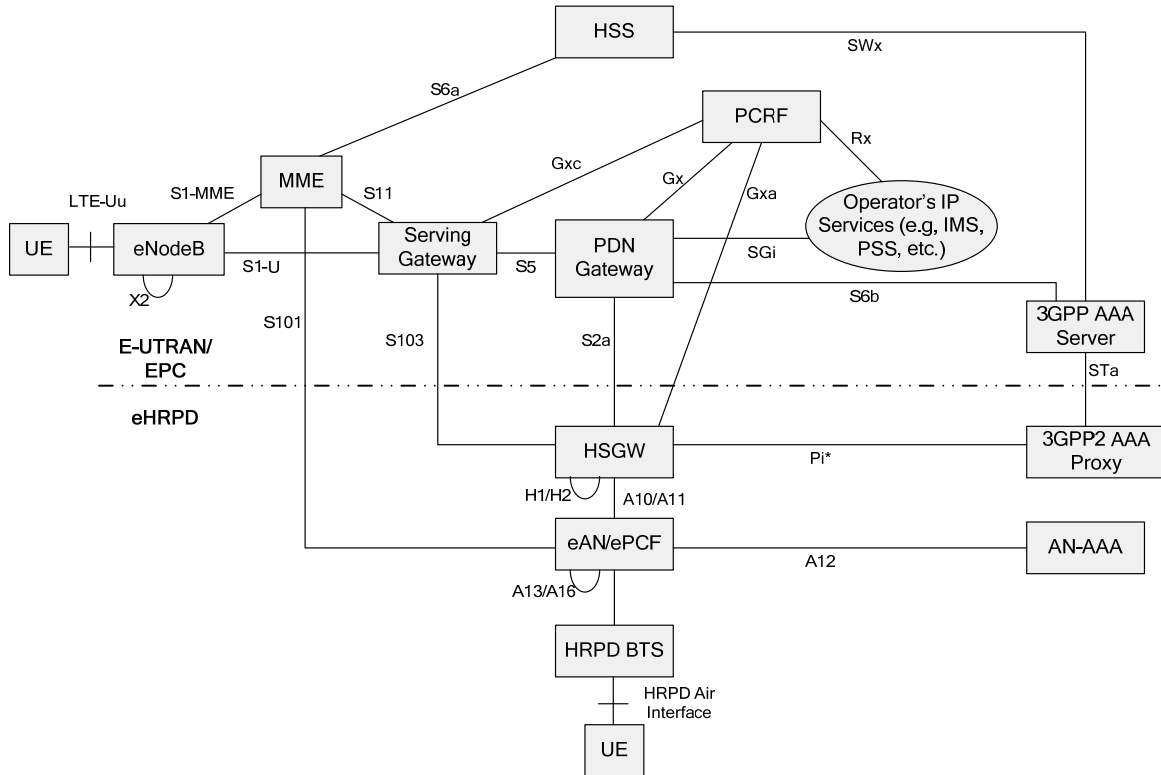
18	3GPP	3rd Generation Partnership Project
19	3GPP2	3rd Generation Partnership Project 2
20	AAA	Authentication, Authorization, Accounting
21	AKA	Authentication and Key Agreement
22	eAN	Evolved Access Network
23	EAP	Extensible Authentication Protocol
24	EPC	Evolved Packet Core
25	ePCF	Evolved Packet Control Function
26	EPS	Evolved Packet System
27	E-UTRAN	Evolved Universal Terrestrial Radio Access Network
28	HRPD	High Rate Packet Data
29	HSGW	HRPD Serving Gateway
30	HSS	Home Subscriber Service
31	IMSI	International Mobile Subscriber Identity
32	IP	Internet Protocol
33	IP-CAN	IP Connectivity Access Network
34	MSK	Master Session Key
35	NAI	Network Access Identifier
36	P-GW	Packet Data Network Gateway (specified by 3GPP)
37	PDN	Packet Data Network
38	RAN	Radio Access Network

1
2
3
4
5
6
7

UE User Equipment

4 Architecture

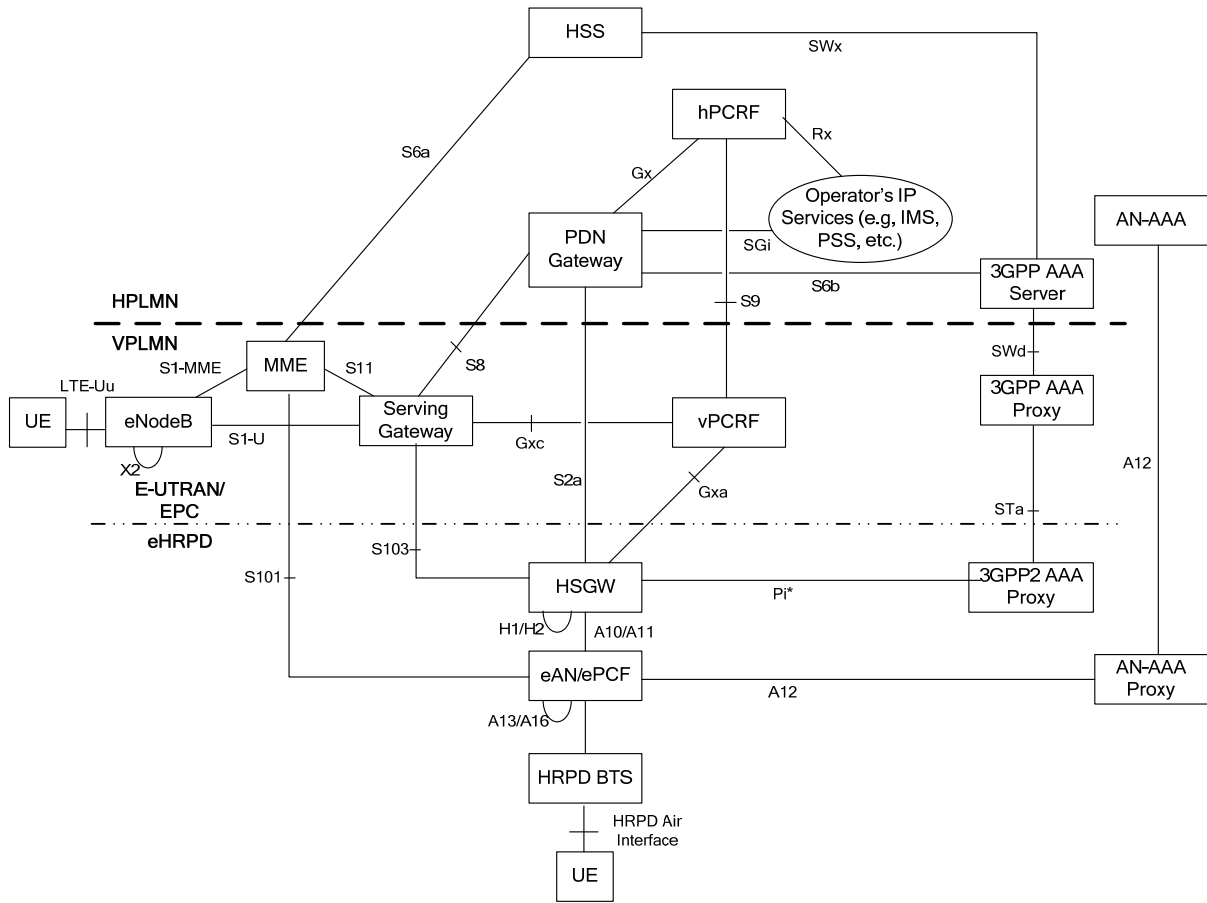
Figure 1 below shows the architecture for interworking between the 3GPP Evolved Universal Terrestrial Radio Access Network (E-UTRAN) and the 3GPP2 evolved High Rate Packet Data (eHRPD) network.



8
9
10
11
12
13
14

Figure 1 E-UTRAN – eHRPD Interworking: Non-Roaming Architecture

Figure 2 below illustrates the E-UTRAN – eHRPD interworking architecture for home-routed traffic. In this case the anchor point (i.e., the P GW) is located in the home network.



1
2

3
4

Figure 2 E-UTRAN – eHRPD Interworking: Security Roaming Architecture (Home-Routed Traffic)

5
6

Figure 3 below illustrates the E-UTRAN – eHRPD interworking architecture for local breakout traffic. In this case the anchor point (i.e., the P-GW) is located in the visited network.

1 Once the UE receives LCP Configure-Request message from the HSGW that contains
2 Authentication-Protocol option set to C227, the UE responds with LCP Configure-Ack,
3 indicating to the HSGW the acceptance of the EAP based authentication for PPP session
4 establishment, as described in [6] and [4] .

5 Once the HSGW receives Configure-Ack from the UE indicating acceptance of the EAP based
6 authentication, the HSGW selects EAP as the PPP authentication protocol and proceeds to play
7 the role of EAP authenticator.

8 **5.2 UE Behavior**

9 The UE supports the EAP-AKA' protocol defined in [11] for Network Access Authentication.
10 Therefore, upon receiving initial EAP Request indicating EAP-AKA' as EAP method Type,
11 the UE does not respond with EAP Nak indicating that the authentication Type is
12 unacceptable.

13 **5.2.1 UE Identity Management**

14 The UE has a permanent identity formatted as the Network Access identifier NAI [10]. This
15 NAI is based on IMSI as defined in [13]. The UE supports temporary identities (pseudonym
16 and fast-reauthentication identity) as specified in [14] and [12]. Temporary identities are
17 defined in [13] and are one time identities.

18 Upon receiving the EAP Request / Identity, the UE responds with the EAP Response / Identity
19 carrying its Network Access Identifier (NAI) format specified in [13]. See [14] for further
20 information on UE identity management.

21 If upon successful EAP-AKA' access authentication (see [11]), a protected pseudonym and/or
22 re-authentication identity were received, the UE stores the temporary identity(s) for the
23 subsequent authentication transaction.

24 **5.2.2 UE Network Access Authentication**

25 Upon receiving the EAP Request / AKA' Challenge, the UE checks whether the AMF
26 separation bit is set. This separation bit distinguishes the authentication vector created for the
27 eHRPD access. If the separation bit is not set to '1', the UE rejects the authentication.
28 Otherwise, the UE executes the AKA algorithms.

29 Upon executing AKA algorithms, the UE verifies the AUTN. If the AUTN is incorrect per
30 [12], the UE rejects the authentication. If the sequence number SQN of the AUTN is outside of
31 valid dynamic range, the UE initiates a synchronization procedure, c.f. [11].

32 If AUTN is correct, the UE computes the RES. The UE uses the value received in
33 AT_KDF_INPUT of the EAP Request/AKA' Challenge to post-process the IK, CK and derive
34 IK' and CK' as specified in [12]:

35 The UE then computes additional new keying material including MSK as specified in [11].
36 The UE checks the integrity of the EAP payload received in AT_MAC with this new keying
37 material.

1 The UE sends the EAP Response / AKA' Challenge, as specified in [11]. The UE includes
2 AT_RES attribute with the computed RES, and AT_MAC attribute to integrity protect the EAP
3 payload.

4 **5.3 HSGW Behavior**

5 The HSGW supports [6], Extensible Authentication Protocol (EAP).

6 The HSGW is the EAP authenticator in eHRPD network.

7 To satisfy the requirements of the EAP-AKA' protocol, the HSGW provides the access
8 network identity to the 3GPP AAA, which acts as the EAP Authentication server. The format
9 and value of the serving network identity is outside the scope of this document. It can consist
10 of the access type (HRPD), NAS-ID = {the FQDN of the HSGW}, the MCC and MNC
11 associated with the HSGW, etc. For details refer to [14].

12 As an EAP authenticator, the HSGW is the entity that receives the MSK from the 3GPP AAA
13 after EAP authentication.

- 14 ▪ If the Diameter protocol is used, the HSGW supports [5], [7], and [8].

15 **5.4 3GPP AAA Server Behavior**

16 3GPP AAA Server requirements are defined in [12].

17 Specifically, the 3GPP AAA Server acts as the EAP Authentication Server.

18 The 3GPP AAA Server receives the EAP Response/Identity message that contains the
19 subscriber identity and the access type over the STa. The 3GPP AAA Server checks whether it
20 has an unused authentication vector with AMF separation bit = 1 and the matching access
21 network identifier available for that UE.

22 If not, the 3GPP AAA Server requests the Authentication Vector from the HSS, indicating that
23 the Vector is destined for the trusted non-3GPP access authentication, e.g., EAP-AKA'
24 authentication. In addition, the 3GPP AAA forwards to the HSS the access network identity
25 received from the HSGW.

26 The Authentication Vector received from the HSS contains the AMF with the separation bit set
27 to '1'.

28 Once the Authentication Vector is received, the 3GPP AAA invokes the EAP-AKA' protocol.
29 The 3GPP AAA Server generates new keying material from the received CK' and IK' and
30 sends AT_RAND, AT_AUTN, AT_MAC and (if generated) protected pseudonym and/or
31 protected re-authentication id, to the HSGW in EAP Request/AKA'-Challenge message. The
32 3GPP AAA Server also includes the access network identity in this message. The access
33 network identity is defined in [14]. The sending of the re-authentication id depends on 3GPP
34 operator's policies on whether to allow fast re-authentication process or not.

35 The 3GPP AAA Server can also include AT_RESULT_IND attribute in the same EAP
36 Request/AKA'-Challenge message, in order to indicate that it wishes to protect the success

1 result message at the end of the process (if the outcome is successful). The protection of result
2 messages depends on home operator's policies.

3 Upon receiving EAP Response/ AKA' Challenge, the 3GPP AAA Server checks the received
4 AT_MAC and compares XRES to the received RES. If the comparison is successful, the 3GPP
5 AAA Server sends the EAP Success message to the HSGW. If protected success indication
6 was negotiated, the 3GPP AAA server first sends an EAP Notification, as specified in [9]. The
7 3GPP AAA Server also includes the key MSK in the message sent to the HSGW in the serving
8 system. The EMSK is retained at the 3GPP AAA server either until the expiration of the
9 lifetime of current session, or until the next EAP authentication, whichever comes first. For the
10 purpose of this document, the use of the EMSK is not defined.

11 **5.5 HSS Behavior**

12 Upon receiving from the 3GPP AAA server an indication that the authentication vector is for
13 EAP-AKA' as defined in [11], the HSS generates an authentication vector with AMF
14 separation bit = 1. The HSS then transforms this authentication vector into a new
15 authentication vector as specified in [12], and derives CK' and IK'.

16 The HSS then sends this transformed authentication vector to the 3GPP AAA server.

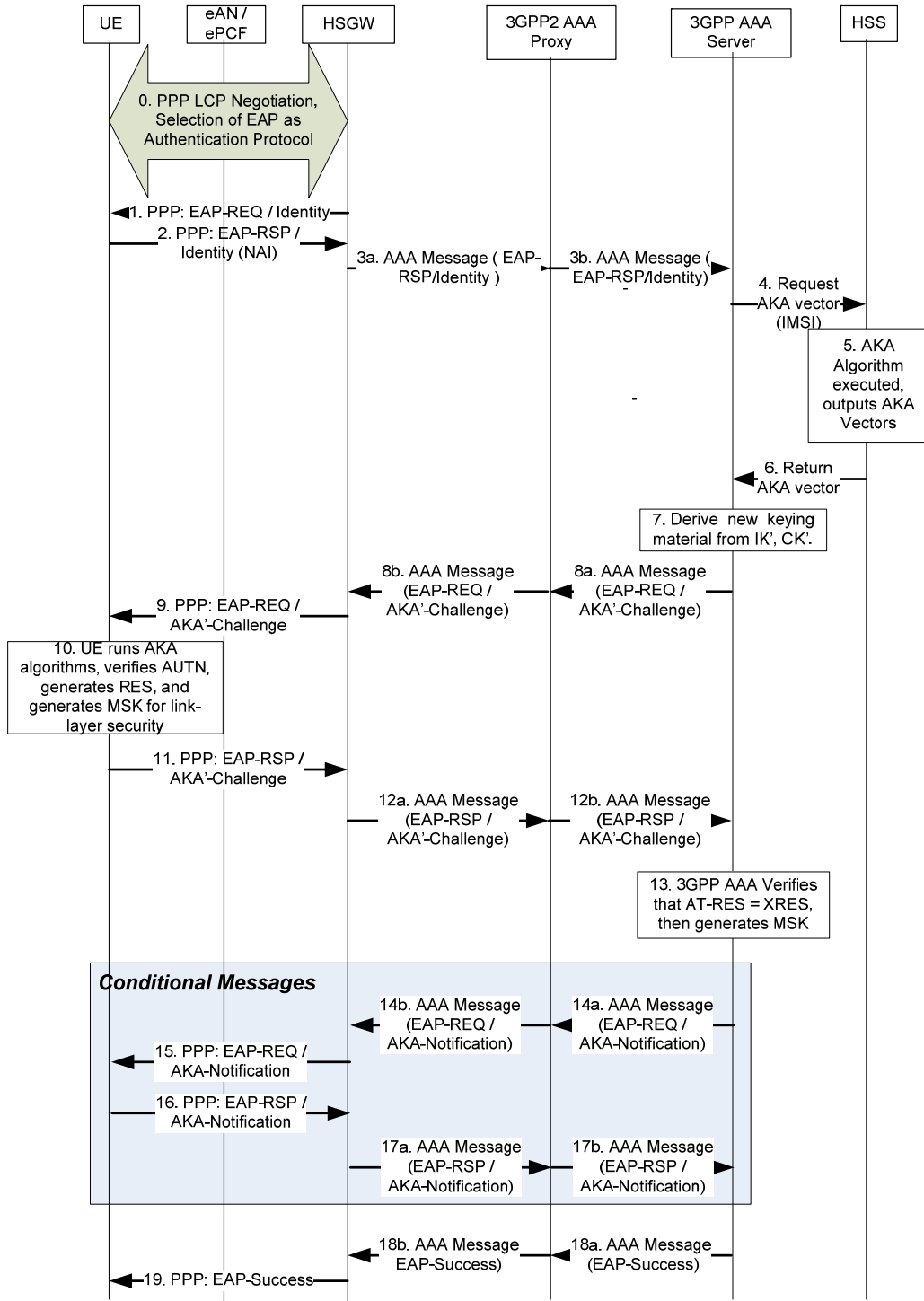
17 The HSS and/or 3GPP AAA server need to ensure, based on local policy, that the non-3GPP
18 access requesting the authentication, is authorized to use the access network identity used to
19 calculate CK' and IK'. The exact details of how to achieve this are outside the scope of this
20 document.

21 **5.6 Call Flows**

22

23 **5.6.1 Initial Authentication**

24 The following figure shows authentication of the UE with the 3GPP AAA Server using EAP-
25 AKA' via the 3GPP2 AAA Proxy and the authenticator in the HSGW, or directly from the
26 3GPP AAA Server via the authenticator in the HSGW. Note that all EAP-AKA' procedures in
27 this message flow follow the rules of [11].



1

2

Figure 4 Initial Authentication using EAP-AKA'

- 1 The steps in Figure 4 are described below.
- 2 0. PPP LCP negotiation occurs. EAP is negotiated as the authentication protocol.
- 3 1. The HSGW sends an EAP-Request / Identity message to the UE over the A10 main
4 signaling connection.
- 5 2. The UE responds with EAP-Response / Identity (NAI). If UE uses its permanent NAI, it
6 uses the IMSI-based Network Access Identifier (NAI) format specified in [13].
- 7 3. The HSGW forwards the unmodified NAI received in the EAP-Response/Identity message
8 to the EAP Server in the 3GPP AAA.
- 9 3a: The HSGW, as the authenticator, encapsulates the EAP payload in an AAA message and
10 forwards it to the 3GPP2 AAA Proxy. The HSGW also includes the access type and the
11 access network identity to assist the 3GPP AAA server in determining the access network
12 name (See Step 4). The format of the network identity is outside the scope of this
13 document. It can consist of the access type (HRPD), NAS-ID = {the FQDN of the
14 HSGW}, the MCC and MNC associated with the HSGW, etc.
- 15 3b.The 3GPP2 AAA Proxy forwards the unmodified contents to the 3GPP AAA Server.
- 16 4. The 3GPP AAA Server terminates the EAP protocol. If the UE identified itself with the
17 pseudonym, the 3GPP AAA server determines the real identity of the MS and derives the
18 IMSI from it. The 3GPP AAA Server checks that it has an unused authentication vector
19 with AMF separation bit = 1 and the matching access network identifier available for that
20 subscriber. If not, a set of new authentication vector is retrieved from HSS using IMSI.
21 The 3GPP AAA server ensures that the access network is authorized to use the claimed
22 name received in step 3. The 3GPP AAA server includes an indication that the
23 authentication vector is required for EAP-AKA', as defined in [11], and the identity of the
24 access network in which the authenticator resides in a message sent to the HSS.
- 25 5. The HSS calculates the AKA vector(s).
- 26 6. The HSS returns the authentication vector(s), including RAND, AUTN, XRES, IK' and
27 CK', to the 3GPP AAA. The 3GPP AAA Server stores the authentication vector(s).
- 28 7. New keying material is derived from IK' and CK' according to the EAP-AKA'[11]. A
29 new pseudonym and/or re-authentication ID may be chosen, and if chosen, are protected
30 (i.e. encrypted and integrity protected) using keying material generated from EAP-AKA'.
- 31 8. The 3GPP AAA Server sends AT_RANDOM, AT_AUTN, a message authentication code for
32 EAP payload in AT_MAC attribute, AT_KDF, AT_KDF_INPUT, and two temporary user
33 identities (if they are generated): protected pseudonym and/or protected re-authentication
34 id) in EAP Request/AKA'-Challenge message. Creation and use of the re-authentication
35 ID depends on the operator's policies on whether to allow fast re-authentication processes
36 or not. It implies that, at any time, the 3GPP AAA Server decides (based on policies set by
37 the operator) whether to include the re-authentication id or not, thus allowing or
38 disallowing the triggering of the fast re-authentication process. The 3GPP AAA Server
39 may use a protected success indication by including the AT_RESULT_IND attribute in
40 the EAP Request/AKA'-Challenge message, in order to indicate to the UE that it would
41 like result indications in both successful and unsuccessful cases. The inclusion of the

- 1 result indications for the protection of the result messages depends on home operator's
2 policies.
- 3 8a: The 3GPP AAA Server sends the EAP-Request / AKA'-Challenge and other
4 parameters to the 3GPP2 AAA Proxy.
- 5 8b. The 3GPP2 AAA Proxy forwards the EAP-Request / AKA'-Challenge and other
6 parameters to the HSGW.
- 7 9. The HSGW sends the EAP-Request / AKA'-Challenge and other parameters to the UE.
- 8 10. The UE first checks whether the AMF separation bit is set to 1. If this is not the case the
9 UE rejects the authentication. If the AMF separation bit is set to 1, the UE runs the AKA
10 algorithms on the UE. The UE verifies that AUTN is correct and thereby authenticates the
11 network. If AUTN is incorrect, the terminal rejects the authentication (not shown in this
12 example). If the sequence number SQN is out of synch or outside the valid dynamic range,
13 the terminal initiates a synchronization procedure (not shown in this example), c.f. [11]. If
14 AUTN is correct, the UE computes RES. The UE uses access network identity received in
15 the EAP-request / AKA'-Challenge message from 3GPP AAA server, in addition to other
16 input parameters as defined in [11], in order to derive the CK', IK'. Using CK' and IK',
17 and UE identity, the UE derives the MSK and EMSK keys, according to EAP-AKA' [11]
18 and [14]. Using computed key material, the UE validates integrity of the received EAP
19 payload in the AT_MAC attribute of the EAP message. If a protected pseudonym and/or
20 re-authentication identity were received, then the UE stores the temporary identity(s) for
21 future authentications.
- 22 11. The UE calculates a new AT_MAC value covering the EAP payload with the new keying
23 material. UE sends EAP Response/AKA'-Challenge to the HSGW containing calculated
24 RES in the AT_RES attribute and the new calculated AT_MAC attribute. The UE
25 includes in this message the result indication in the AT_RESULT_IND attribute if it
26 supports this attribute and if it received the same indication from the 3GPP AAA Server.
27 Otherwise, the UE omits this indication.
- 28 12. The HSGW forwards the authentication response to the EAP server.
- 29 12a: The HSGW encapsulates the EAP-Response / AKA'-Challenge, AT_RES, and
30 AT_MAC in a AAA message and sends it to the 3GPP2 AAA Proxy.
- 31 12b. The 3GPP2 AAA Proxy forwards the message to the 3GPP AAA Server.
- 32 13. The 3GPP AAA Server checks the received AT_MAC and verifies that the received
33 AT_RES value is the same as the XRES value received in step 6 above. The remainder of
34 this flow assumes that the comparison succeeds. If the 3GPP AAA Server sent a
35 pseudonym and/or fast re-authentication identity to the UE in the step 8, it now associates
36 these identities with the permanent identity of the UE.
- 37 Steps 14 through 17 are conditional based on the EAP Server and the UE having indicated
38 the use of protected successful result indications.
- 39 14. If the 3GPP AAA Server requested previously to use protected result indications (e.g., by
40 including the AT_RESULT_IND attribute in the EAP Request message) and received the

- 1 same result indications from the UE, the 3GPP AAA Server sends the message EAP
2 Request/AKA'-Notification.
- 3 14a: The 3GPP AAA Server sends EAP Request/AKA'-Notification to the 3GPP2 AAA
4 Proxy.
- 5 14b. The 3GPP2 AAA Proxy forwards it to the HSGW.
- 6 15. The HSGW sends EAP Request/AKA'-Notification to the UE.
- 7 16. The UE sends EAP Response/AKA'-Notification to the HSGW.
- 8 17. The HSGW forwards the AKA' Notification Response to the 3GPP AAA server.
- 9 17a: The HSGW encapsulates the EAP-Response / AKA'-Notification in a AAA
10 message and sends it to the 3GPP2 AAA Proxy.
- 11 17b. The 3GPP2 AAA Proxy forwards the message to the 3GPP AAA Server. The
12 3GPP AAA Server ignores the contents of this message if the AT-NOTIFICATION
13 code in the EAP-AKA Notification was "success".
- 14 18. The 3GPP AAA Server creates an EAP-Success message that also includes the MSK and
15 the subscription profile that has been retrieved from the HSS, (See [11]), in the underlying
16 AAA protocol message (i.e. not at the EAP level).
- 17 18a: The 3GPP AAA Server sends The EAP-Success message and other parameters in a
18 AAA message to the 3GPP2 AAA Proxy.
- 19 18b. The 3GPP2 AAA Proxy forwards the information on to the HSGW.
- 20 If the peer indicated that it wants to use protected success indications with
21 AT_RESULT_IND, then the peer does not accept EAP-Success after a successful
22 EAP/AKA'-Reauthentication round. In this case, the peer only accepts EAP-Success
23 after receiving an EAP-AKA' Notification with the AT_NOTIFICATION code
24 "Success".
- 25 If the peer receives an EAP-AKA' notification that indicates failure, then the peer can no
26 longer accept the EAP-Success packet, even if the server authentication was successfully
27 completed.
- 28 19. The HSGW stores the MSK Key to be used for generating the keying material for
29 protecting communication with the authenticated UE. The HSGW also stores the other
30 parameters sent in the AAA message. The HSGW signals EAP-Success to the UE. If the
31 UE received the pseudonym and/or fast reauthentication identity in step 9, it now accepts
32 these identities as valid for next authentication attempt.
- 33 At this point, the EAP-AKA' exchange is successfully completed, and the UE and the access
34 network share keying material derived during that exchange.

1 The authentication process may fail at any moment, for example because of unsuccessful
2 checking of AT_MAC protecting EAP payload, or no response from the UE after a network
3 request. In that case, the EAP-AKA' process will be terminated as specified in [11] and an
4 indication shall be sent to the HSS.

5 **5.6.2 Fast Re-Authentication**

6 Fast re-authentication for EAP-AKA' is specified in [11]. Fast re-authentication re-uses keys
7 derived on the previous full authentication. Fast re-authentication does not involve the HSS,
8 and does not involve the handling of AKA authentication vectors, which makes the procedure
9 faster.

10 UE and 3GPP AAA server implement fast re-authentication for EAP-AKA'. Its use is optional
11 and depends on operator policy. If fast re-authentication for EAP-AKA' is used the 3GPP AAA
12 server indicates this to the UE by means of sending the re-authentication identity to the UE in
13 the EAP Request message.

14 Fast re-authentications for EAP-AKA' generates new MSK key, which may be used for
15 renewing session key used for protection in the non-3GPP access network.

16 The access network identity does not change when going from the full to the fast re-
17 authentication process. If this happens, the re-authentication process is terminated as defined in
18 [11].

19

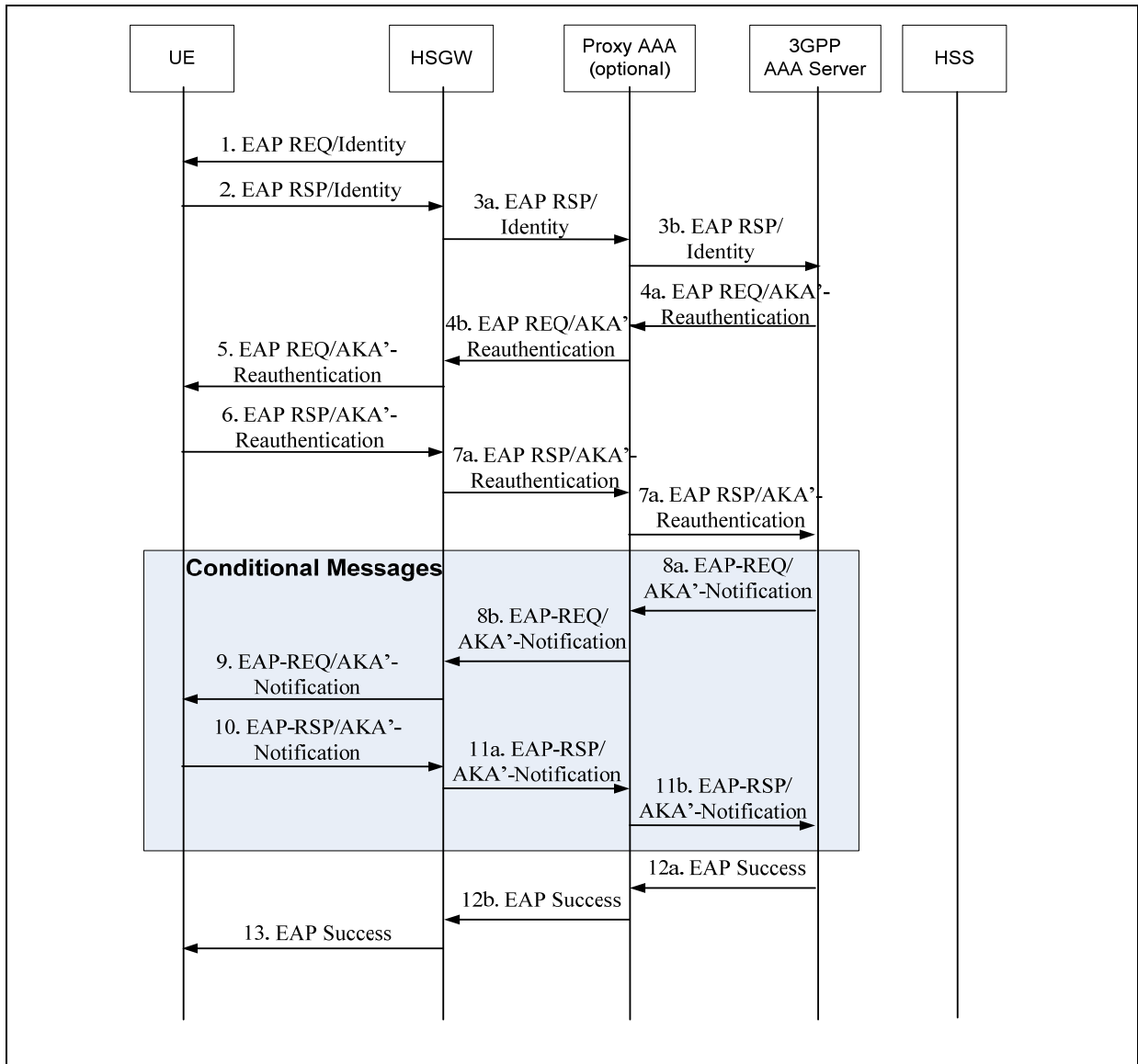


Figure 5 Fast-Reauthentication using EAP-AKA'

1. HSGW sends an EAP Request/Identity to the UE.
2. UE replies with an EAP Response/Identity containing a re-authentication identity (this identity was previously delivered by AAA server in a full authentication procedure).
3. The HSGW forwards the EAP Response/Identity to the AAA server. Intermediate Proxy AAA's may perform routing and forwarding functions.
4. The AAA server initiates the Counter (which was initialized to one in the full authentication process) and sends it in the EAP Request message, together with the NONCE, the AT_MAC (calculated over the NONCE) and a protected re-authentication ID for a next fast re-authentication. If the AAA server is not able to deliver a re-authentication identity, next time the UE forces a full-authentication (to avoid the use of the re-authentication identity more than once).

The 3GPP AAA Server can include AT_RESULT_IND attribute in the message sent to the UE, in order to indicate that the success result message at the end of the process

- 1 shall be protected (if the outcome is successful). The protection of result messages
2 depends on home operator's policies.
- 3 The 3GPP AAA server can fail to recognize the identity as it may have been altered
4 by proxies. In this case, the 3GPP AAA server can, as in the case of a full
5 authentication, instead perform an EAP-AKA' method specific identity request; i.e.
6 "EAP-Request/AKA' identity [Any identity]" in order to obtain a more reliable
7 identity. This is however only used in case the server fails to recognize the identity, as
8 otherwise the purpose of fast re-authentication is defeated.
- 9 5. The HSGW forwards the EAP Request message to the UE.
 - 10 6. The UE verifies that the Counter value is fresh and the AT_MAC is correct, and it
11 sends the EAP Response message with the same Counter value (it is up to the AAA
12 server to step it up) and a calculated AT_MAC.
- 13 If the UE supports the protected result indication, then the UE includes in this
14 message the AT_RESULT_IND attribute if it received the same indication from the
15 3GPP AAA. Otherwise, the UE omits this indication.
- 16 7. The HSGW forwards the response toward the AAA server.
 - 17 8. The AAA server verifies that the Counter value is the same as it sent, and the
18 AT_MAC is correct, and sends the message EAP Request/AKA'-Notification,
19 previous to the EAP Success message, if the 3GPP AAA Server requested previously
20 to use protected success result indications. The message EAP Request/AKA'-
21 Notification is integrity protected with the AT_MAC, and includes an encrypted copy
22 of the Counter used in the present re-authentication process.
 - 23 9. The HSGW forwards the EAP Request/AKA'-Notification message to the UE.
 - 24 10. The UE sends the EAP Response/AKA'-Notification.
 - 25 11. The HSGW forwards the EAP Response/AKA'-Notification message toward the
26 3GPP AAA server. The 3GPP AAA Server ignores the contents of this message.
 - 27 12. The AAA server sends an EAP Success message.
 - 28 13. The EAP Success message is forwarded to the UE.

29 6 Key Generation

30 This section provides methods and procedures for generating the Pairwise Master Key (PMK)
31 from MSK, which in turn is used for generation of Session Key(s) (SKey). The UE and eAN
32 then use Session Key to generate keys for over-the-air MessageIntegrityCode, authentication
33 and encryption.

34 A pictorial representation of key generation is provided in Figure 6 , while detailed description
35 is provided in following sub-sections.

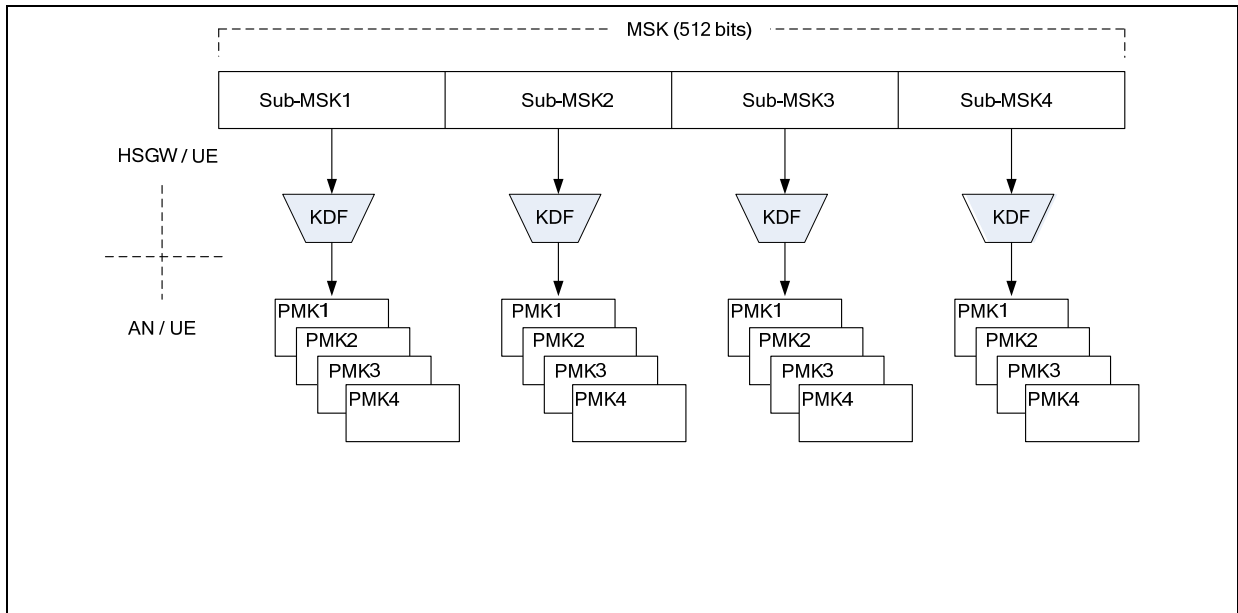


Figure 6 eHRPD Key Generation

6.1 Pairwise Master Key (PMK) Generation

As a result of successful access authentication based on EAP-AKA' [11], both UE and HSGW obtain the MSK. The UE and HSGW separate the 512 bits of the MSK into four equal portions of 128 bits each, i.e., four Sub-MSKs. The UE and HSGW use each Sub-MSK to generate four PMKs as follows:

$$\text{PMK1} = \text{HMAC-SHA-256}(\text{Sub-MSK}, \text{"pmk@hrpd.3gpp2"}, 0x01), [0:127]$$

$$\text{PMK2} = \text{HMAC-SHA-256}(\text{Sub-MSK}, \text{"pmk@hrpd.3gpp2"}, 0x01) [128:255],$$

$$\text{PMK3} = \text{HMAC-SHA-256}(\text{Sub-MSK}, \text{"pmk@hrpd.3gpp2"}, 0x02) [0:127],$$

$$\text{PMK4} = \text{HMAC-SHA-256}(\text{Sub-MSK}, \text{"pmk@hrpd.3gpp2"}, 0x02) [128:255],$$

where the key label "pmk@hrpd.3gpp2" is set to ASCII strings without NULL termination.

The UE and HSGW can pre-compute the PairwiseMasterKeyID associated with each PMK as specified in [2] as follows:

$$\text{PairwiseMasterKeyID} = 128 \text{ most significant bits of } \text{ehmacsha256}(\text{key}=\text{PairwiseMasterKey}, \text{key_length}=\text{length of PairwiseMasterKey in units of octets}, \text{message} = \text{"PairwiseMasterKeyID"}, \text{message_length} = \text{length of message in units of bits}, \text{message_offset}=0, \text{MAC_length}=16)$$

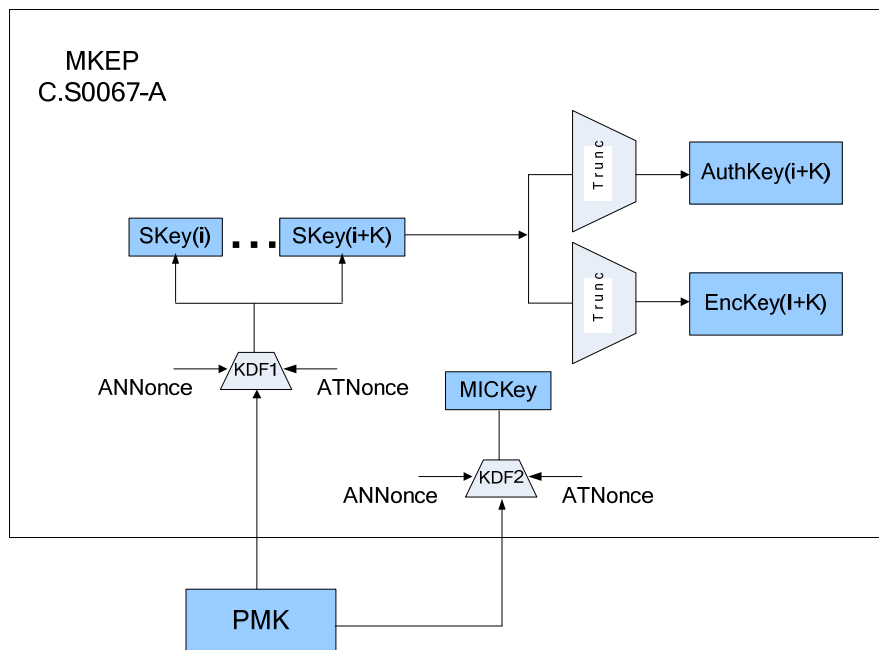
In addition, the UE and HSGW can also pre-compute the PMKs and PairwiseMasterKeyIDs associated with the other Sub-MSKs. This pre-computation of PMKs and

1 PairwiseMasterKeyIDs enables the UE to identify the PMK it needs to use upon receiving
 2 request from the access network to derive session keys for access security.

3 6.2 Access Network Key Generation

4 As a result of successful Multi-Key Key Exchange Protocol (MKEP) message exchange, both
 5 the UE and the eAN generate Session Key(s) (SKey(S)) as specified in [C.S0067-A] and as
 6 summarized here.

7 A pictorial representation of Session Key and over-the-air key generation is provided in Figure
 8 7.



9

10

Figure 7 SKey(s) and over-the-air key generation

11

12

13

14

The eAN and the UE use PMK, ANNonce and ATNonce as an input to generate SKey(s) and MICKey. ANNonce and ATNonce are corresponding fields of Multi-Key Key Exchange Protocol (MKEP) messages KeyRequest and KeyReponse, respectively. Each SKey is then truncated into authorization key and encryption key.

15

16

The UE and the eAN derive SKey[i] as follows, where i is SessionKeyIndex field of the corresponding MKEP KeyRequest message:

17

- Set k and m to an 8-bit number with value zero.

18

19

- while $k < (\text{NumSessionKeys} + 1)$, where NumSessionKeys is field of the corresponding KeyRequest message

20

21

22

23

Set SKeyTemp[i+k] to $N_{\text{MKEPSessionKeyLen}}$ least significant bits of { SKeyTemp[i+k] | 128 most significant bits of $\text{ehmacsha256}(\text{key}=\text{PairwiseMasterKey}, \text{key_length}=\text{length of PairwiseMasterKey in units of octets}, \text{message}=\text{ATNonce}|\text{ANNonce}|m, \text{message_length}=\text{length of message in units of bits}, \text{message_offset}=0, \text{MAC_length}=16)$ }, where the ehmacsha256 function is specified in [3], ANNonce and ANNonce are

1 corresponding fields of KeyRequest and KeyReponse messages respectively, and m is represented as an 8-bit
2 field.

3 Set m to m+1

4 Set k to k+1.

5 The UE and the eAN derive the MICKey[i] as follows, where i is the SessionKeyIndex field of
6 the corresponding KeyRequest message:

- 7 ▪ Set MICKey[i] to the 128 most significant bits of {
8 ehmacsha256(key=PairwiseMasterKey, key_length=length of PairwiseMasterKey in
9 units of octets, message=ATNonce|ANNonce, message_length=length of message in
10 units of bits, message_offset=0, MAC_length=16) }, where the ehmacsha256 function
11 is specified in [3]

12 The keys used for authentication and encryption are generated from the session key as follows.
13 The keys derived from SKey[i] are referred to by the subscript i.

14 The AN and the UE set FACAAuthKey[i], FPCAAuthKey[i], RACAAuthKey[i], and
15 RPCAAuthKey[i] to SKey[i][127:0], where i is the session key index.

16 The AN and the UE set FACEncKey[i], FPCencKey[i], RACencKey[i], and RPCencKey[i] to
17 SKey[i][255:128], where i is the session key index.

18 The UE and the eAN compute and store a MICKey, Authentication Key, and Encryption Key.
19 The keys derived from SKey[i] are referred to by the subscript i. The eAN and the UE use the
20 Authentication Key and Encryption Key derived from the SKey with index i, where i is the
21 value of the InUseSessionKeyIndex attribute.

22

23 7 Key Distribution

24 This section describes key distribution details in eHRPD.

25 7.1 Master Session Key and Inter-HSGW Handoff

26 The HSGW sets its MSK to either the value of the MSK received from the AAA or to the
27 value of the MSK received from another HSGW in the MSK Info field during the inter-HSGW
28 handoff.

29 The HSGW uses the 128 most significant bits of the MSK (Sub-MSK) as the Master Session
30 Key for the derivation of PMKs. The HSGW declares the remaining portion of the received
31 MSK as the unused MSK information. The HSGW sets the value of the MSK Lifetime to the
32 remaining lifetime of the authorized EAP session.

33 During Inter-HSGW handoff, the Source-HSGW sends the unused portion of the MSK to the
34 Target-HSGW in the MSK Info field, only if the unused portion of the MSK information is \geq
35 128 bits, the Target-HSGW is trusted, and the link between the HSGWs is secure (e.g. IPsec is
36 used). The Target-HSGW sets its MSK to the value of the received MSK context and acts as
37 described above.

1 If the lifetime of the received MSK is close to expiry, or, during the inter-HSGW handoff, if
2 the length of the received MSK Info is equal to 128 bits, or if the MSK is not received, the
3 Target-HSGW initiates the authentication as soon as possible to continue with the session.
4 When the new MSK AVP is received from the AAA, the Target-HSGW deprecates the current
5 MSK value and replaces it with the value received in the MSK AVP. The Target-HSGW
6 derives the new PMK from the new MSK as described in section 6.1.

7 7.2 HSGW – eAN Key Distribution

8 If the HSGW receives an indication in A11-Registration Request message that the PMK is
9 needed for this session, the HSGW returns the unused PMK. If the HSGW determines that it
10 has no unused PMKs, the HSGW sets Sub-MSK as the 128-bit portion (Sub-MSK) occupying
11 the highest order bit positions of the unused MSK information. The HSGW uses the Sub-MSK
12 for the computation of PMKs using the procedures described in section 6.1. Once the HSGW
13 generates the PMK or determines that the new PMK needs to be sent to the eAN/ePCF, the
14 HSGW sends a PMK and its lifetime in seconds to the eAN using A11-Registration Response
15 or A11-Session Update message [15]. The lifetime of the PMK is set to not more than the
16 remaining value of the MSK lifetime. If the HSGW runs out of PMKs, the HSGW can use the
17 unused MSK information to generate new PMKs as described above.

18 7.3 Multi-Key Key Exchange Protocol and Intra-HSGW inter-eAN 19 Handoff

20 If the MKEP is negotiated for a session, an eAN includes PMK Information IE in a A11-
21 Registration Request message sent to HSGW, indicating to the HSGW that the PMK is needed
22 for this session.

23 Once the eAN receives PMK(s) from the HSGW, the eAN can trigger MKEP. Example call
24 flow for MKEP can be found in Section 7.4.2, while the details can be found in [2]. When the
25 MKEP is triggered, the eAN indicates to the UE which PMK to use by including the PMK_ID
26 in the KeyRequest Message. The UE selects appropriate PMK that corresponds to the indicated
27 PMK_ID, either by computing the PMK and PMK_ID values in a real time, or from a buffer of
28 precomputed values. The KeyRequest message also indicates to the UE in the
29 NumSessionKeys parameter how many Session Key sets needs to be computed in one
30 execution of the MKEP.

31 Upon successful MKEP message exchange, the eAN can indicate to the UE which
32 Authentication key and Encryption key to use by including the InUseSessionKeyIndex
33 attribute in an AttributeUpdateRequest message sent on the Control Channel. The eAN and the
34 UE use the Authentication Key and Encryption Key derived from the SKey with index *i*, where
35 *i* is the value of the InUseSessionKeyIndex attribute received in AttributeUpdateRequest
36 message.

37 If the eAN wants to use Authentication Key and Encryption Key derived from another SKey
38 (different from one in use) to preserve the cryptographic separation, the eAN sends the
39 AttributeUpdateRequest with another InUseSessionKeyIndex. Upon this, the eAN and the UE
40 use the Authentication Key and Encryption Key derived from this new SKey.

41 If the eAN determines that new set of SKeys needs to be obtained, the eAN can trigger a new
42 MKEP message exchange.

1 During, inter-eAN handoff (A13 or A16 session transfer) the Source-eAN sends unused SKeys
 2 included in SKey Parameter of Session State Information Record (SSIR) and existing PMKs in
 3 the PMK Parameter of SSIR to the Target-eAN.

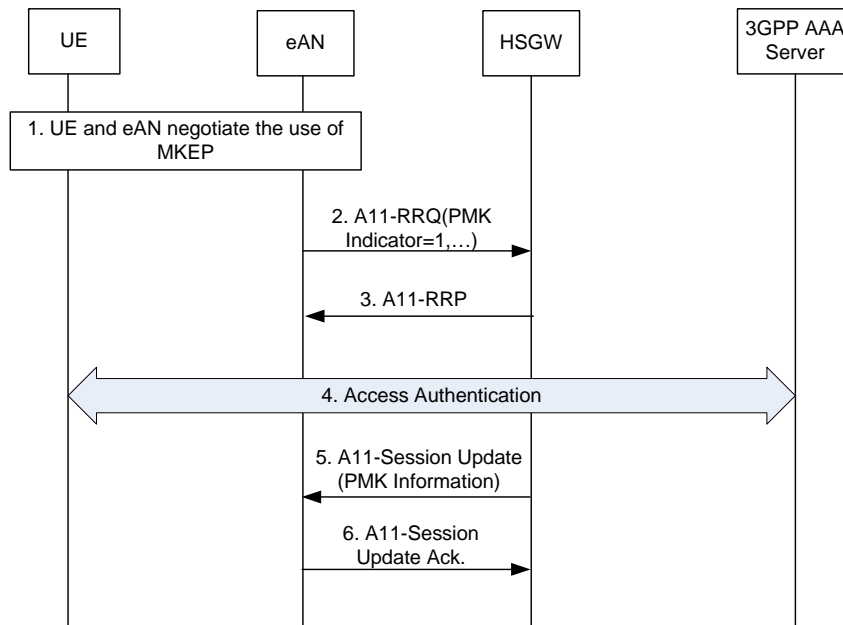
4 7.4 Call Flows

5

6 7.4.1 HSGW – eAN Key Distribution

7

Figure 8 below depicts example call flow for PMK exchange between HSGW and eAN.



8

9 **Figure 8 Key Distribution between HSGW and eAN**

10

Figure 8 Key Distribution between HSGW and eAN

11

The steps in Figure 8 are described below.

12

1. During session negotiation UE and eAN negotiate to use Multi-Key Key Exchange Protocol.

13

14

2. The eAN sends A11-Registration Request to the HSGW to set up A10 connection(s). In this message the eAN includes PMK Indicator indicating to the HSGW that once available, the PMK needs to be sent to the eAN.

15

16

17

3. The HSGW send the A11-Registration Response to acknowledge A11- Registration Request received in the previous step.

18

19

4. The UE and 3GPP AAA server perform mutual authentication. As a result of successful authentication, MSK is delivered to the HSGW. The HSGW uses MSK to derive PMK(s) as described in Section 6.1.

20

21

- 1 5. The HSGW sends A11-Session Update message to the eAN and includes PMK
2 Information IE with the derived PMK(s).
- 3 6. The eAN acknowledges A11-Session Update message by sending A11-Session
4 Update Ack. Message.
- 5 At this point eAN received PMK(s) and can trigger MKEP protocol to generate session keys at
6 any time.

7.4.2 Multi-Key Key Exchange Protocol

Figure 9 depicts example call flow for successful MKEP message exchange.

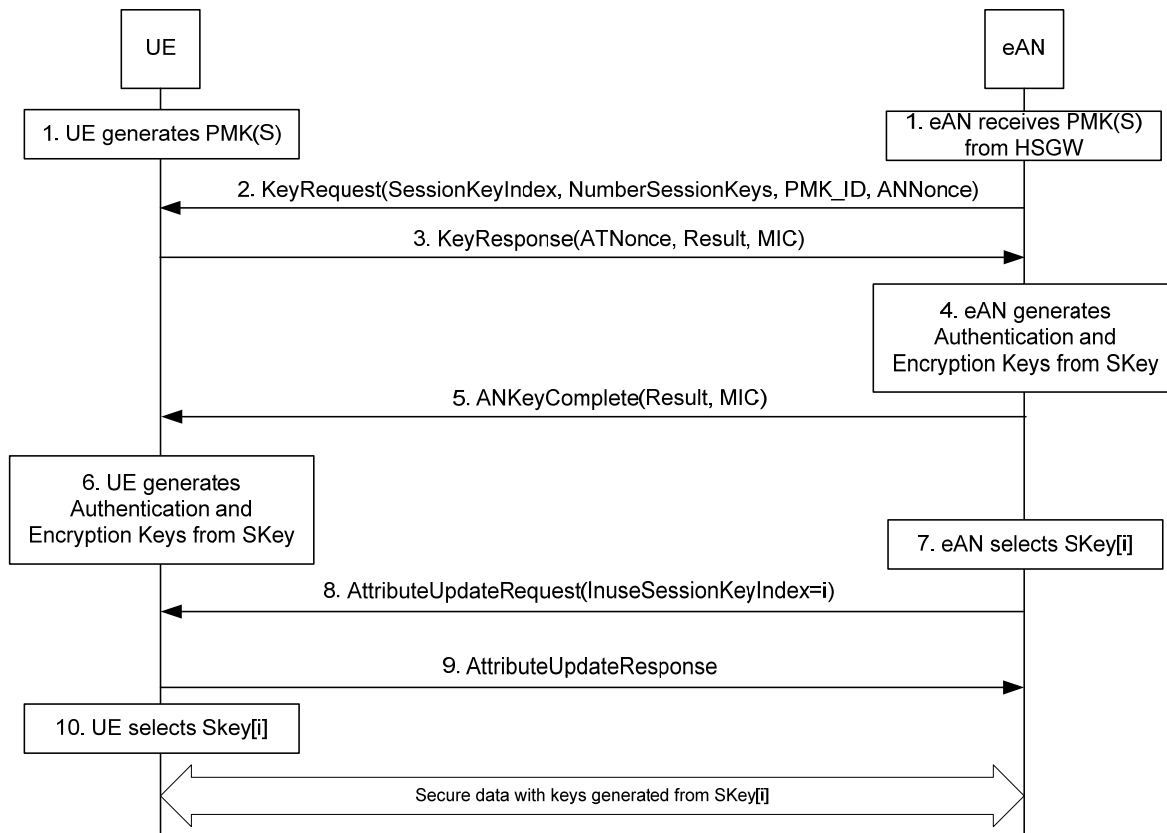


Figure 9 Multi-Key Key Exchange Protocol

The steps in Figure 9 are described below.

1. Upon successful authentication, the UE generates PMK(s), while eAN receives PMK(s) from the HSGW.
2. The eAN send KeyRequest message to the UE. In this message the eAN includes SessionKeyIndex, NumberSessionKeys indicating the number of session key that need to be generated in a single run of the MKEP protocol, PMK_ID indicating the

- 1 PMK to be used for session key generation and ANNonce. eAN sets ANNonce to a
2 128-bit random number.
- 3 3. Upon receiving KeyRequest message from the eAN, the UE first determines which
4 PMK to use based on the received PMK_ID. The UE sets ATNonce to a 128-bit
5 random number and generates temporary session key and MICKey using the identified
6 PMK, ATNonce and received ANNonce as input. The UE sends KeyResponse
7 message to the eAN and includes MIC (generated using MICKey), ATNonce and the
8 Result code. The UE may include other parameters as specified in [2].
- 9 4. Upon receiving KeyResponse, the eAN generates MICKey and temporary session
10 keys. The eAN verifies received MIC using generated MICKey. If the verification is
11 successful, the eAN sets the session keys to the temporary session keys and generates
12 authentication keys and encryption keys as described in section 6.2.
- 13 5. The eAN sends ANKeyComplete message to the UE, including MIC and the Result
14 code.
- 15 6. Upon receiving ANKeyComplete message the UE verifies received MIC and if the
16 verification is successful, the UE sets the session keys to the temporary session keys
17 and generates authentication keys and encryption keys as described in section 6.2.
- 18 7. The eAN selects SessionKeyIndex i and the session key SKKey[i] associated with that
19 selected i .
- 20 8. The eAN sends AttributeUpdateRequest message to the UE including
21 InUseSessionKeyIndex set to value i , indicating to the UE which session key to use.
- 22 9. The UE sends AttributeUpdateResponse message to the eAN.
- 23 10. The UE selects SKKey[i] to generate authentication and encryption key for securing the
24 data.
25