

# IPWorks 3GPP AAA Server-ePDG SWm and SWm+ Interface

---

## INTERWORK DESCRIPTION

**Copyright**

© Ericsson AB 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Prerequisites	1
1.2	Related Information	1
<b>2</b>	<b>Interface Overview</b>	<b>3</b>
2.1	Interface Role	3
2.2	Services	3
2.3	Encapsulation and Addressing	4
<b>3</b>	<b>Procedures</b>	<b>5</b>
3.1	SIM-based Authentication	5
3.2	Non-SIM based Authentication	10
<b>4</b>	<b>Information Model</b>	<b>15</b>
4.1	ePDG Initiated Full Authentication and Authorization	15
4.2	ePDG Initiated Re-Authentication and Re-Authorization	17
4.3	ePDG Initiated Re-Authorization	17
4.4	ePDG Initiated Session Termination	18
4.5	ePDG Initiated Fast Re-authentication	19
4.6	AAA Server Initiated Session Termination	19
4.7	AAA Server Initiated Re-Authorization	20
<b>5</b>	<b>Information Element</b>	<b>23</b>
5.1	Diameter AVPs	23
5.2	Messages	40
<b>6</b>	<b>Error Handling</b>	<b>49</b>
6.1	Diameter Error Handling	49
6.2	EAP Error Handling	50
<b>7</b>	<b>Formal Syntax</b>	<b>53</b>
<b>8</b>	<b>Related Standards</b>	<b>55</b>
	<b>Reference List</b>	<b>57</b>





# 1 Introduction

This document describes the SWm/SWm+ interface between the IPWorks 3GPP AAA Server and ePDG.

## Scope

The SWm/SWm+ reference point is used to authenticate and authorize the User Equipment (UE), also used to transport GTPv2/PMIPv6 related mobility parameters in a case the UE attaches to the EPC through the S2b and SWn reference points (for example, IP Mobility Mode Selection information).

The document covers the following topics:

- Interface Overview
- Procedure
- Information Model
- Information Elements
- Error Handling
- Formal Syntax
- Related Standards

## Target Groups

This document is intended for personnel needing to understand the logical entity, including interfaces and protocols, of the IPWorks.

## 1.1 Prerequisites

N/A

## 1.2 Related Information

Trademark information, typographic conventions, definition and explanation of acronyms and terminology can be found in the following documents:

- *Trademark Information*, Reference [1]
- *Typographic Conventions*, Reference [2]



- *Glossary of Terms and Acronyms*, Reference [3]



## 2 Interface Overview

This section describes the interface between IPWorks 3GPP AAA Server and ePDG, as shown in Figure 1.

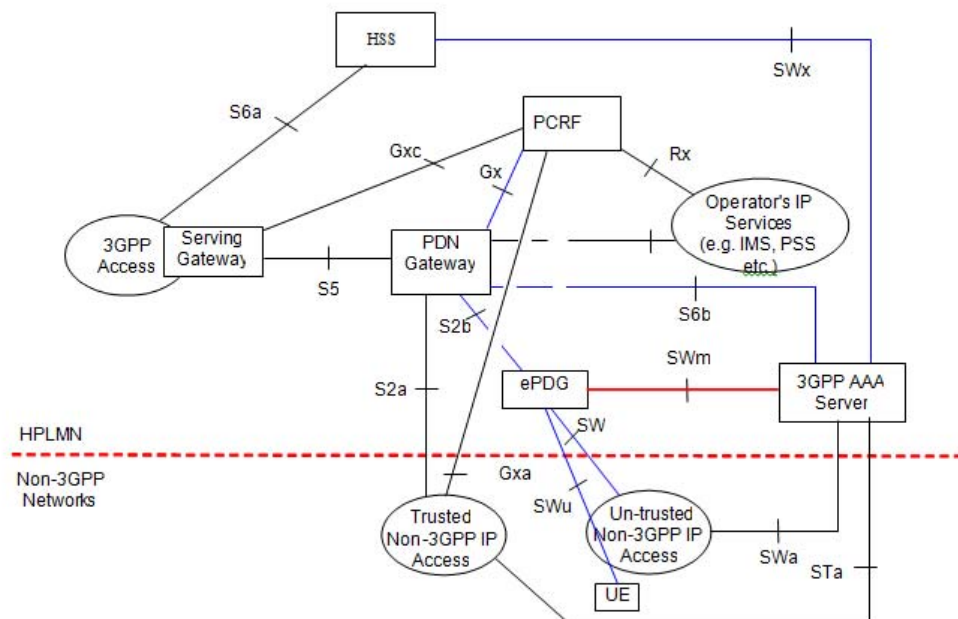


Figure 1 SWm/SWm+ Interface between IPWorks 3GPP AAA Server and ePDG

### 2.1 Interface Role

This section describes the role of the SWm/SWm+ interface in the EPC network.

For the SWm/SWm+ interface, IPWorks AAA server acts as 3GPP AAA server in the EPC network, and also takes the role of authentication and authorization for UE who attaches to the EPC through ePDG.

### 2.2 Services

This section describes the services that the SWm/SWm+ interface offers.

The services offered by the SWm/SWm+ are shown in Table 1.

*Table 1 Offered Services*

Offered Service	Description
Authentication, Authorization	<p>The 3GPP AAA Server is used to authenticate/authorize the UE from Non-3GPP IP Access Network:</p> <ul style="list-style-type: none"><li>• Authenticate/Authorize the UE from Trusted/Untrusted Non-3GPP IP Access Network.</li><li>• Transport GTPv2/PMIPv6 related mobility parameters in a case the UE attaches to the EPC through the S2b and SWn reference points (for example, IP Mobility Mode Selection information).</li></ul>

## 2.3 Encapsulation and Addressing

This section describes what lower level protocol the SWm/SWm+ interface uses:

- SCTP
- TCP
- Diameter Base Protocol





## 3 Procedures

This section describes the procedures used with the offered and used interfaces of IPWorks:

- *SIM-based Authentication*
- *Non-SIM based Authentication*

### 3.1 SIM-based Authentication

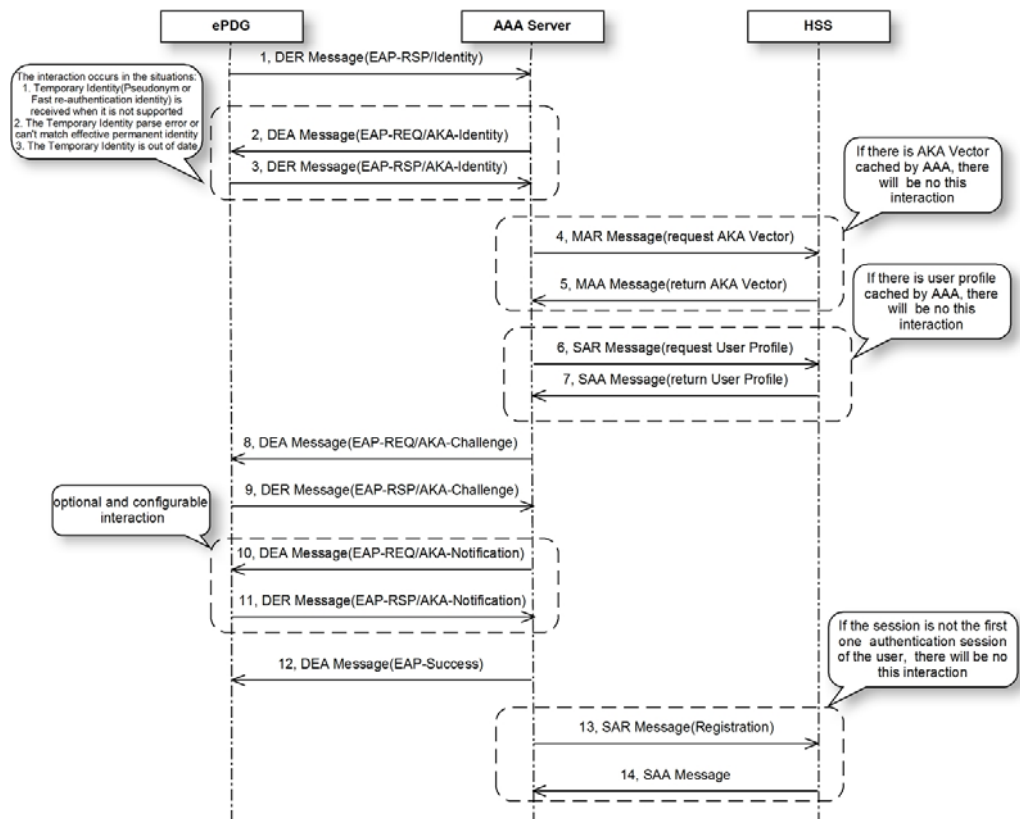
SIM-based authentication includes the following two topics:

- *ePDG Initiated full Authentication and Authorization*
- *ePDG Initiated Re-Authentication and Re-Authorization*
- *ePDG Initiated Re-Authorization*
- *ePDG Initiated Session Termination*
- *ePDG Initiated Fast Re-Authentication*
- *AAA Initiated Session Termination*
- *AAA Server Initiated Re-Authorization*

#### 3.1.1 **ePDG Initiated Full Authentication and Authorization on SWm Interface**

The procedure is triggered when the UE attaches to the EPC using the s2b reference point.

The authentication on SWm interface is based on EAP-AKA. The Diameter messages are DER and DEA (see Section 4.1 on page 15).



**Figure 2 ePDG Initiated Full Authentication and Authorization on SWm Interface**

The procedure is triggered when the UE attaches to the EPC using the S2b reference point with anonymous username.

IPWorks provides the customized feature IMSI Mask Handling which can be enabled by configuration. The process is shown in Figure 3.

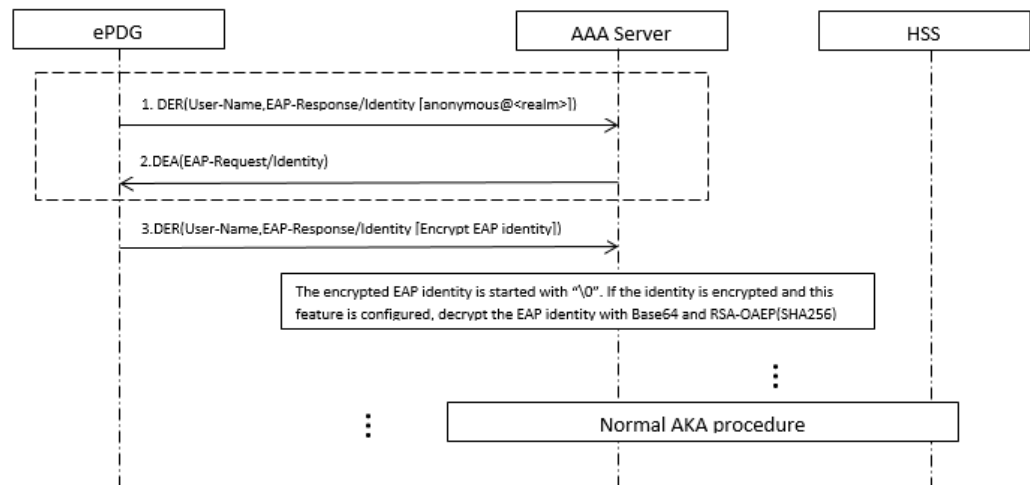


Figure 3 ePDG Initiated Anonymous Authentication and Authorization on SWm Interface

### 3.1.2 ePDG Initiated Re-Authentication and Re-Authorization

For detailed procedure, refer to Section 3.1.1 on page 5.

### 3.1.3 ePDG Initiated Re-Authorization

The procedure is triggered by the ePDG when checking whether the user authorization parameters previously provided by the 3GPP AAA server are modified.

The Diameter messages are AAR and AAA (see Section 4.3 on page 17).

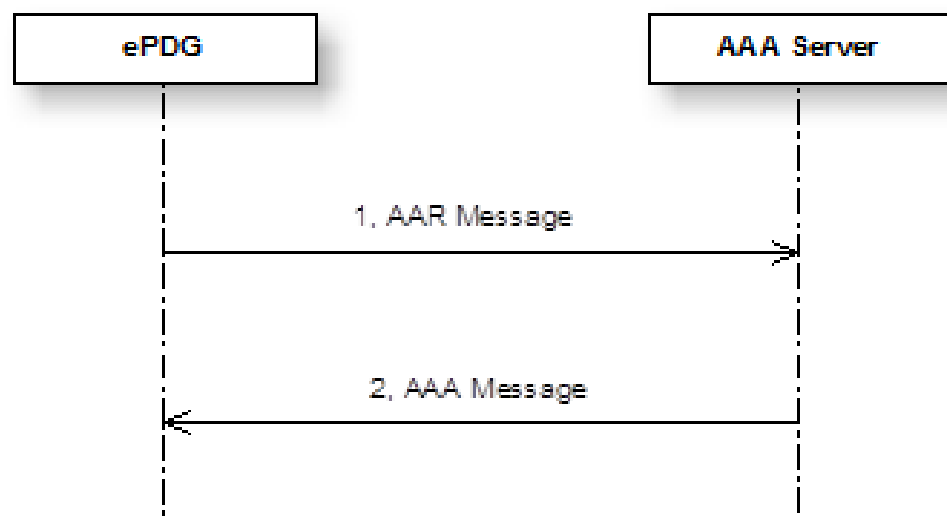


Figure 4 ePDG Initiated Re-Authorization

### 3.1.4 ePDG Initiated Session Termination on SWm Interface

This procedure is triggered when the user connection is to be released, the ePDG informs the 3GPP AAA server to remove the access information.

The Diameter messages are STA and STR (see Section 4.4 on page 18).

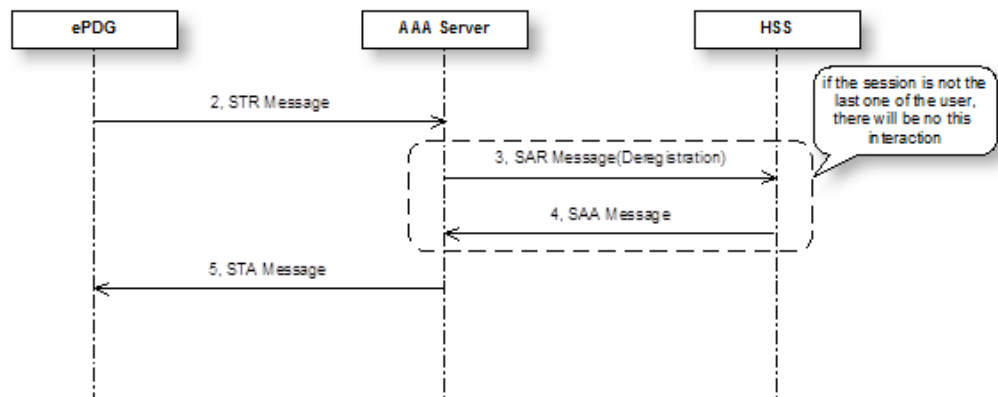


Figure 5 ePDG Initiated Session Termination on SWm Interface

### 3.1.5 ePDG Initiated Fast Re-Authentication

The procedure is triggered when the UE attaches to the EPC using the s2b reference point with the Fast Re-authentication Identity of EAP-AKA.

The Diameter messages are DER and DEA (see Section 4.5 on page 19).

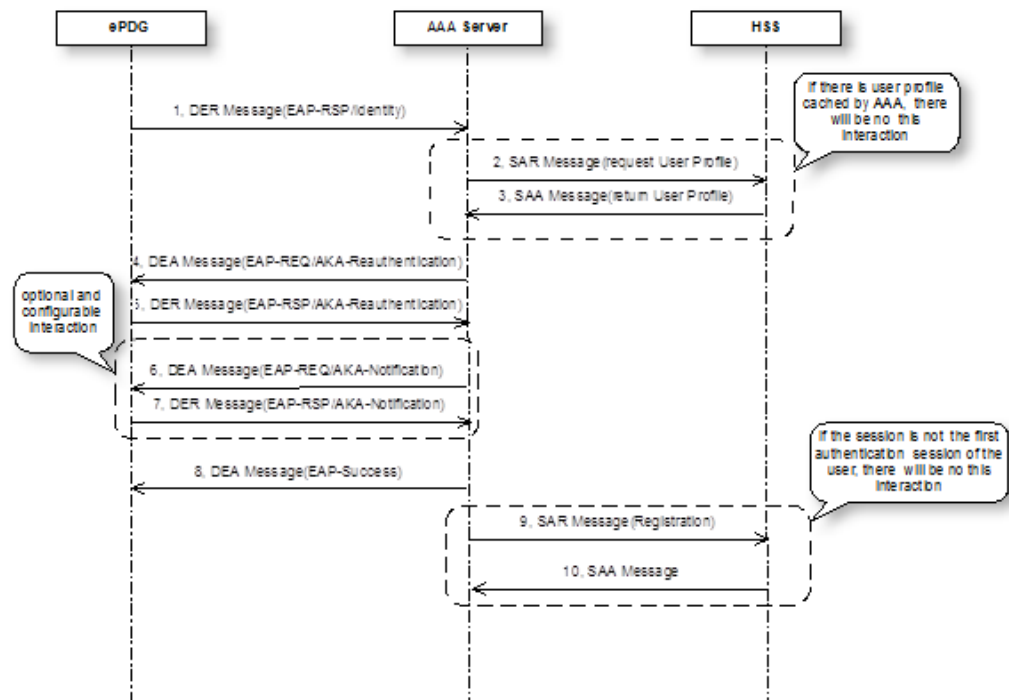


Figure 6 ePDG Initiated Fast Re-Authentication

### 3.1.6

#### AAA Server Initiated Session Termination on SWm Interface

This procedure is triggered when the system administrator wants to detach the user from the 3GPP AAA server.

The procedure is based on Diameter session abort messages. The Diameter messages are Section 4.6 on page 19.

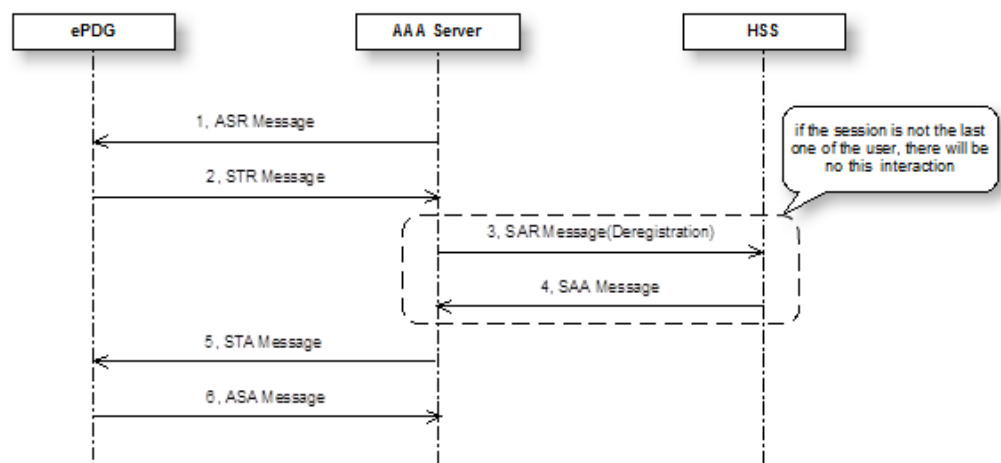


Figure 7 AAA Server Initiated Session Termination

### 3.1.7 AAA Server Initiated Re-Authorization

This procedure is triggered when subscriber profile is modified in HSS.

The procedure is based on Diameter RAR(Section 4.7.1 on page 20), AAR(Section 4.3.1 on page 17), AAA(Section 4.3.2 on page 17) and RAA(Section 4.7.2 on page 21) messages.

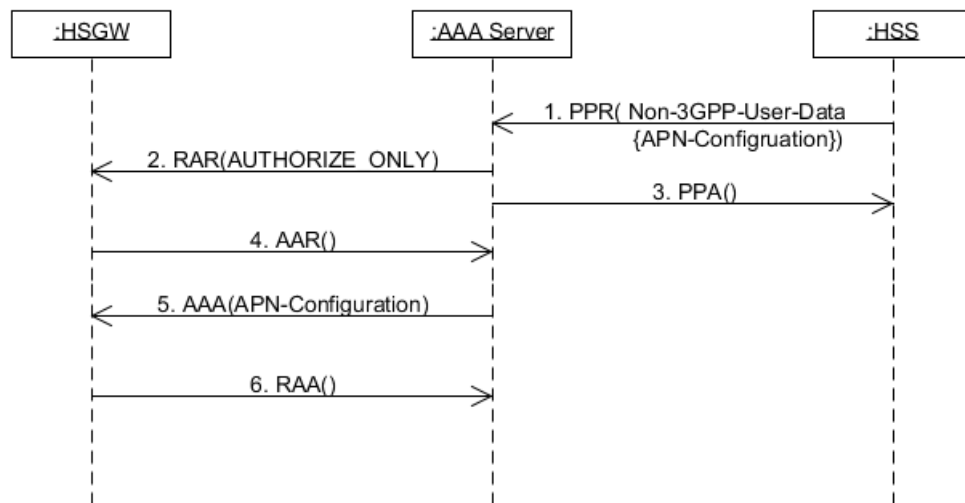


Figure 8 AAA Server Initiated Re-Authorization

## 3.2 Non-SIM based Authentication

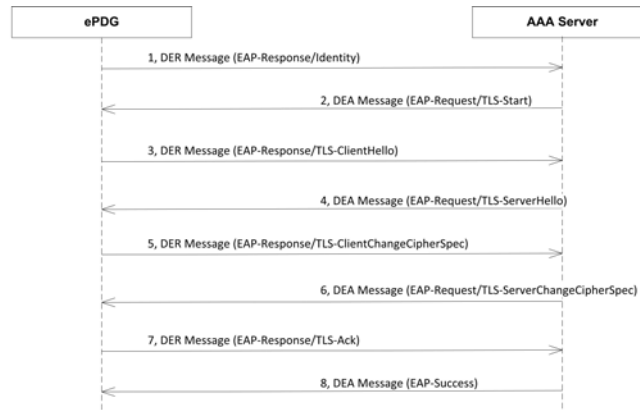
Non-SIM based Authentication includes the following topics:

- *ePDG Initiated full Authentication and Authorization*
- *ePDG Initiated Re-Authentication and Re-Authorization*
- *ePDG Initiated Re-Authorization*
- *ePDG Initiated Session Termination*
- *AAA Initiated Session Termination*

### 3.2.1 ePDG Initiated full Authentication and Authorization on SWm+ Interface

The procedure is triggered when the UE attaches to the EPC using the s2b reference point.

The authentication is based on EAP-TLS. The Diameter messages are DER and DEA (see Section 4.1 on page 15).



*Figure 9 ePDG Initiated full Authentication and Authorization*

The procedure is triggered when the UE attaches to the EPC using the S2b reference point with anonymous user name.

IPWorks provides the customized feature IMSI Mask Handling which can be enabled by configuration. The process is showed in Figure 10.

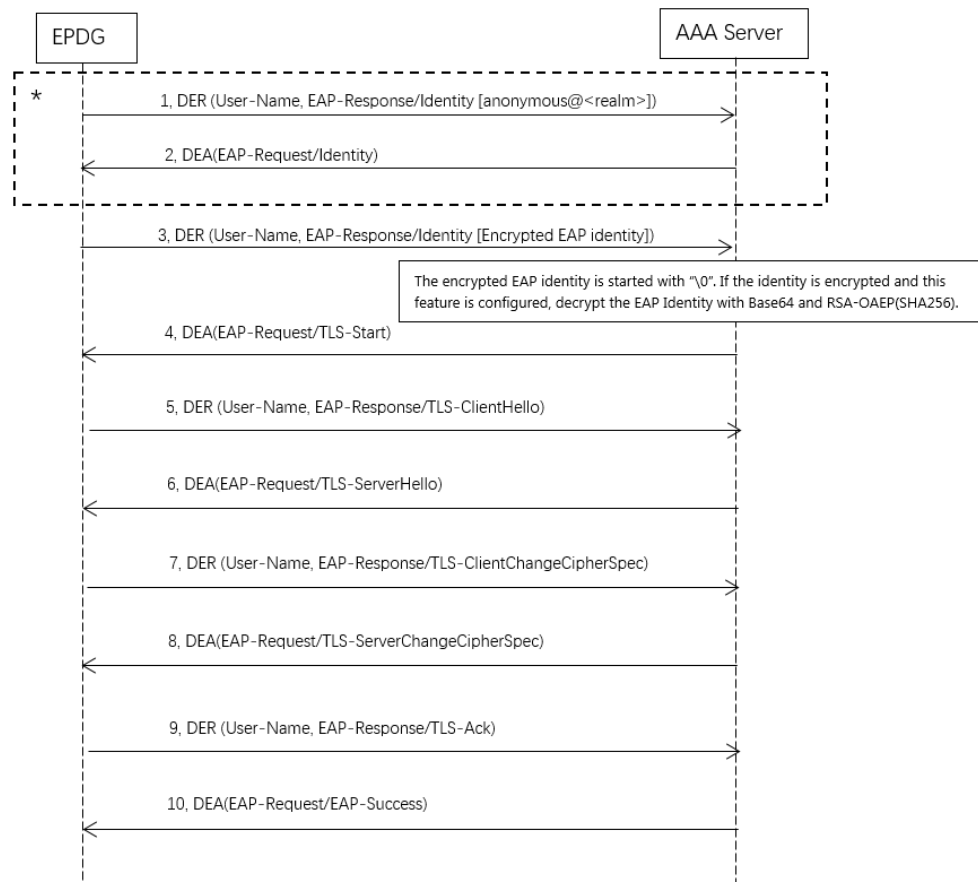


Figure 10 ePDG Initiated Anonymous Authentication and Authorization

### 3.2.2 ePDG Initiated Re-Authentication and Re-Authorization

For detailed procedure, see Section 3.2.1 on page 10.

### 3.2.3 ePDG Initiated Re-Authorization

The procedure is triggered by the ePDG when checking whether the user authorization parameters previously provided by the 3GPP AAA server are modified.

The Diameter messages are AAR and AAA (see Section 4.3 on page 17).



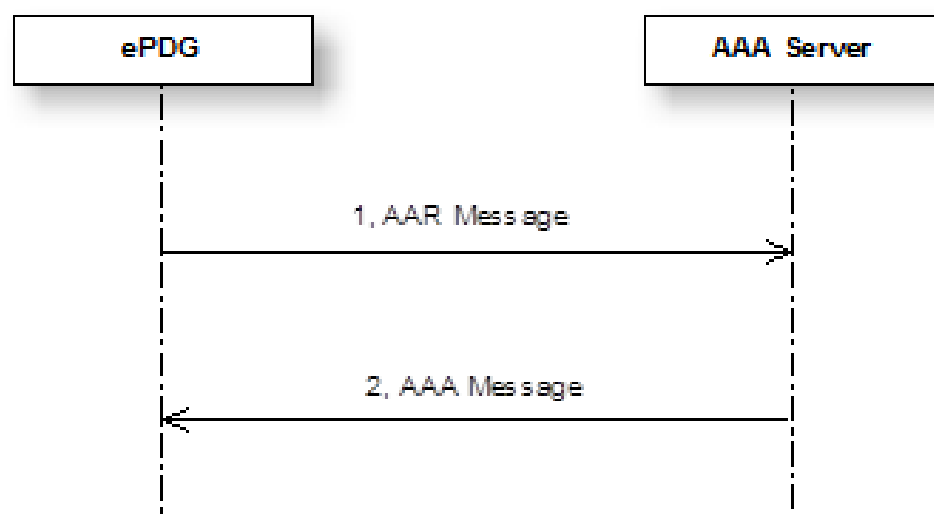


Figure 11 ePDG Initiated Re-Authorization

### 3.2.4 ePDG Initiated Session Termination on SWm+ Interface

This procedure is triggered when the user connection is to be released, the ePDG informs the 3GPP AAA server to remove the access information.

The Diameter messages are STA and STR (see Section 4.4 on page 18).

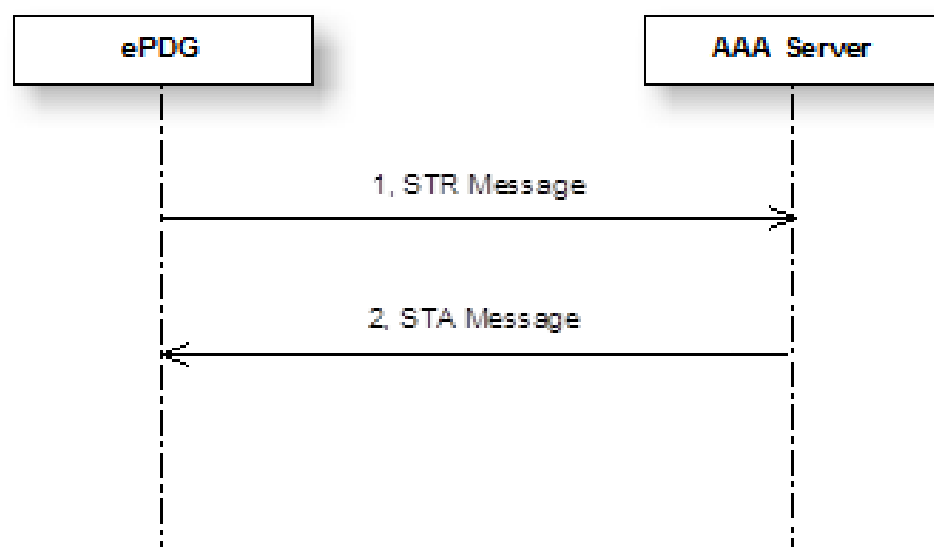


Figure 12 ePDG Initiated Session Termination

### 3.2.5 AAA Initiated Session Termination on SWm+ Interface

This procedure is triggered when the system administrator wants to detach the user from the 3GPP AAA server.

The procedure is based on Diameter session abort messages. The Diameter messages are Section 4.6 on page 19.

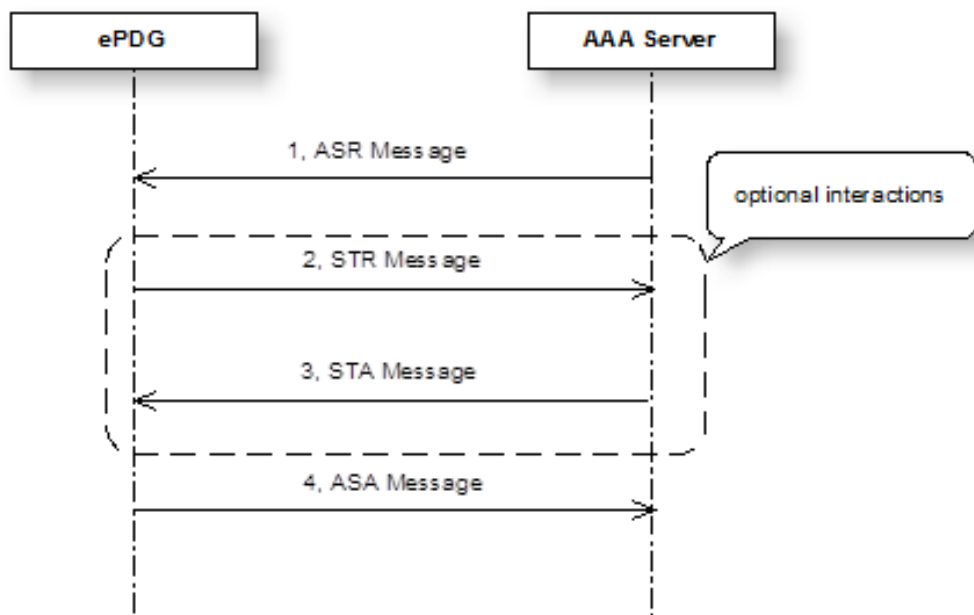


Figure 13 AAA Initiated Session Termination

## 4 Information Model

This section describes the information model including mandatory and optional parameters of each service operation. This document only covers diameter messages and AVPs involved in the application.

**Note:** For the description and format of the Base Protocol, see Reference [4].

Table 2 shows the Naur Form (ABNF) format used in the commands in the subsections.

*Table 2 Naur Form (ABNF) format*

{ }	Mandatory
< >	Mandatory with fixed place
[ ]	Optional
*	Optional
*n	Zero or more Occurrences
n	Occurrences

### 4.1 ePDG Initiated Full Authentication and Authorization

#### 4.1.1 DER Command

The Diameter-EAP-Request (DER) command is sent from ePDG to 3GPP AAA server. The command is set in the Command Code field(set to 268) and the "R" bit is set in the Command Flags field.

The format of this command is listed as below:

```
<Diameter-EAP-Request>::=<Diameter Header: 268, REQ, PXY, 16777264>
<Session-Id>
{Auth-Application-Id}
{Origin-Host}
{Origin-Realm}
{Destination-Realm}
{Auth-Request-Type}
{EAP-Payload}
[User-Name]
[RAT-Type]
[Service-Selection]
[MIP6-Feature-Vector]
[QoS-Capability]
[Visited-Network-Identifier]
[UE-Local-IP-Address]
[AAA-Failure-Indication]
```

```
[Terminal-Information]
[Emergency-Services]
* [Supported-Features] ...
* [Proxy-Info]
* [Route-Record]
* [AVP]
```

#### Note:

- The following AVPs are mandatory when the DER message carries the EAP-Response/Identity package:
  - User-Name
- The following AVPs must be carried in the first DER message that transfers the EAP-RSP/Identity message:
  - AVPs used to fetch authentication vector and user profile from HSS.
- The 3GPP AAA server records these AVPs into the session for later re-authentication and/or re-authorization use.
  - Auth-Grace-Period and Authorization-Lifetime AVPs (If exist).
- Consult Section 5.2 on page 40 for the EAP Messages encapsulated in Diameter EAP-Payload AVP.

### 4.1.2

#### DEA Command

The Diameter-EAP-Answer (DEA) command is sent from 3GPP AAA server to ePDG. The command is set in the Command Code field(set to 268) and the "R" bit is cleared in the Command Flags field.

The format of this command is listed as below:

```
<Diameter-EAP-Request>::=<Diameter Header: 268, REQ, PXY, 16777264>
<Session-Id>
{Auth-Application-Id}
{Auth-Request-Type}
{Result-Code}
{Origin-Host}
{Origin-Realm}
{EAP-Payload}
[EAP-Master-Session-Key]
[APN-OI-Replacement]
[APN-Configuration]
[MIP6-Feature-Vector]
[Mobile-Node-Identifier]
[Trace-Info]
[Subscription-ID]
[Session-Timeout]
[MIP6-Agent-Info]
[3GPP-Charging-Characteristics]
```



```
[Visited-Network-Identifier]
* [Redirect-Host]
[Redirect-Host-Usage]
[Redirect-Max-Cache-Time]
* [Supported-Features]
...
* [Proxy-Info]
* [AVP]
```

## 4.2 ePDG Initiated Re-Authentication and Re-Authorization

Refer to Section 4.1 on page 15.

## 4.3 ePDG Initiated Re-Authorization

### 4.3.1 AAR Command

The AA-Request (AAR) command is sent from ePDG to 3GPP AAA Server/Proxy.

This command is set in the Command-Code field(set to 265) and the "R" bit set in the Command Flags field.

The format of this command is listed as below:

```
<AA-Request>::=<Diameter Header:265, REQ,PXY,16777264>
<Session-Id>
{Auth-Application-Id}
{Origin-Host}
{Origin-Realm}
{Destination-Realm}
{Auth-Request-Type}
[User-Name]
...
* [Proxy-Info]
* [Route-Record]
* [AVP]
```

**Note:** The following AVPs are mandatory for the AAR message:

- User-Name

### 4.3.2 AAA Command

The AA-Answer (AAA) command is sent from 3GPP AAA Server/Proxy to ePDG.

This command is set in the Command-Code field(set to 265) and the "R" bit cleared in the Command Flags field.



The format of this command is listed as below:

```
<AA-Answer>::=<Diameter Header:265,REQ,PXY,16777264>
<Session-Id>
{Auth-Application-Id}
{Auth-Request-Type}
{Result-Code}
{Origin-Host}
{Origin-Realm}
[User-Name]
[APN-OI-Replacement]
[APN-Configuration]
[Trace-Info]
[Subscription-ID]
[3GPP-Charging-Characteristics]
[Session-Timeout]
...
* [Proxy-Info]
* [AVP]
```

## 4.4 ePDG Initiated Session Termination

### 4.4.1 STR Command

The Session-Termination-Request (STR) command is sent from ePDG to 3GPP AAA server.

This command is set in the Command-Code field (set to 275) and the "R" bit set in the Command Flags field.

**Note:** The Command Code value and Augmented Backus-Naur Form (ABNF) follows the format defined in the IETF RFC 3588, see Section 4.4.1 on page 18.

The format of this command is listed as below:

```
<Session-Termination-Request>::<Diameter Header: 275,REQ,PXY,16777264>
<Session-Id>
{Origin-Host}
{Origin-Realm}
{Destination-Realm}
{Auth-Application-Id}
[User-Name]
...
* [Proxy-Info]
* [Route-Record]
* [AVP]
```

**Note:** The following AVPs are mandatory for the STR message:

- User-Name



## 4.4.2 STA Command

The Session-Termination-Answer (STR) command is sent from 3GPP AAA server to ePDG.

This command is set in the Command-Code field (set to 275) and the "R" bit cleared in the Command Flags field.

**Note:** The IETF RFC 3588 Session-Termination-Request command reuses this Command Code value and Augmented Backus-Naur Form (ABNF), see Section 4.4.2 on page 18.

The format of this command is listed as below:

```
<Session-Termination-Answer>::=<Diameter Header: 275, PXY, 16777264>
    <Session-Id>
    {Result-Code}
    {Origin-Host}
    {Origin-Realm}
    [Redirect-Host-Usage]
    [Redirect-Max-Cache-Time]
    ...
    * [Proxy-Info]
    * [AVP]
```

## 4.5 ePDG Initiated Fast Re-authentication

Refer to Section 3.1.1 on page 5

## 4.6 AAA Server Initiated Session Termination

### 4.6.1 ASR Command

The Abort-Session-Request (ASR) command is sent from 3GPP AAA Server/Proxy to ePDG.

This command is set in the Command-Code field (set to 274) to the "R" bit is set in the Command Flags field.

The format of this command is listed as below:

```
<Abort-Session-Request>::=<Diameter Header: 274, REQ, PXY, 16777264>
    <Session-Id>
    {Origin-Host}
    { Origin-Realm }
    { Destination-Realm }
    { Destination-Host }
    { Auth-Application-Id }
    [ User-Name ]
    [ Auth-Session-State ]
```

```

...
* [ AVP ]

```

**Note:** The following AVPs are mandatory for the ASR message:

- User-Name

## 4.6.2 ASA Command

The Abort-Session-Answer (ASA) command is sent from ePDG to 3GPP AAA Server/Proxy.

This command is set in the Command-Code field (set to 274) to the "R" bit is cleared in the Command Flags field.

The format of this command is listed as below:

```

<Abort-Session-Answer>:: <Diameter Header: 274, PXY, 16777264>
                        <Session-Id>
                        { Result-Code }
                        { Origin-Host }
                        { Origin-Realm }
                        ...
                        * [AVP]

```

## 4.6.3 STR Command

See Section 4.4.1 on page 18.

## 4.6.4 STA Command

See Section 4.4.2 on page 18.

# 4.7 AAA Server Initiated Re-Authorization

## 4.7.1 RAR Command

The Re-Auth-Request (RAR) command shall be indicated by the Command-Code field set to 258 and the "R" bit set in the Command Flags field, and shall be sent from a 3GPP AAA Server/Proxy to a ePDG. The ABNF is based on the one in IETF RFC 4005 [4] and is defined as follows:

```

< Re-Auth-Request > ::=
< Diameter Header: 258, REQ, PXY, 16777264 >
    < Session-Id >
        { Origin-Host }
        { Origin-Realm }
        { Destination-Realm }
        { Destination-Host }
        { Auth-Application-Id }

```





```

{ Re-Auth-Request-Type }
[ User-Name ]
...
*[ AVP ]

```

## 4.7.2 RAA Command

The Re-Auth-Answer (RAA) command shall be indicated by the Command-Code field set to 258 and the "R" bit cleared in the Command Flags field, and shall be sent from a ePDG to a 3GPP AAA Server/Proxy. The ABNF is based on the one in IETF RFC 4005 [4] and is defined as follows:

```

< Re-Auth-Answer > ::=
< Diameter Header: 258, PXY, 16777264 >
  < Session-Id >
    { Result-Code }
    { Origin-Host }
    { Origin-Realm }
    [ User-Name ]
    ...
  *[ AVP ]

```





## 5 Information Element

### 5.1 Diameter AVPs

This section describes the Diameter AVPs included in the SWm/SWm+ messages, for more information, refer to Reference [4].

#### 5.1.1 IETF Protocol AVPs

Table 3 IETF Protocol AVPs

Attribute Name	AVP Code	Value Type	Must	May	Should not	Must not	Description
MIP6-Home-Link-Prefix	125	OctetString	M	P		V	See Section 5.1.1.1 on page 2
Re-Auth-Request-Type	285	Enumerated	M	P		V	See Section 5.1.1.2 on page 2
MIP-Home-Agent-Addresses	334	Address	M			V	See Section 5.1.1.3 on page 2
MIP-Home-Agent-Host	348	Grouped	M			V	See Section 5.1.1.4 on page 2
EAP-Payload	462	OctetString	M			V	See Section 5.1.1.5 on page 2
EAP-Master-Session-Key	464	OctetString					See Section 5.1.1.6 on page 2
MIP6-Agent-Info	486	Grouped	M			V	See Section 5.1.1.7 on page 2
Service-Selection	493	UTF8String	M	P		V	See Section 5.1.1.8 on page 2
Mobile-Node-Identifier	506	UTF8String	M	P		V	See Section 5.1.1.9 on page 2
QoS-Profile-Id	573	Unsigned32	M			V	See Section 5.1.1.11 on page
QoS-Profile-Template	574	Grouped	M			V	See Section 5.1.1.10 on page
QoS-Capability	578	Grouped	M			V	See Section 5.1.1.12 on page
Subscription-ID	433	Grouped	M	P		V	See Section 5.1.1.13 on page
Subscription-Id-Type	450	Enumerated	M	P		V	See Section 5.1.1.14 on page
Subscription-Id-Data	444	UTF8String	M	P		V	See Section 5.1.1.15 on page
Proxy-Info	284	Grouped	M			V, P	See Section 5.1.1.16 on page
Route-Record	282	DiamIdent	M			V	See Section 5.1.1.17 on page

##### 5.1.1.1 MIP6-Home-Link-Prefix

The MIP6-Home-Link-Prefix AVP defined in RFC 5447 is of type OctetString and contains the Mobile IPv6 home network prefix information in a network byte order.

### 5.1.1.2 Re-Auth-Request-Type

The Re-Auth-Request-Type AVP (AVP Code 285) is of type Enumerated. And it indicates whether the user is to be authorized only or authenticated and authorized.

AUTHORIZE_ONLY	0
AUTHORIZE_AUTHENTICATE	1

### 5.1.1.3 MIP-Home-Agent-Address

The MIP-Home-Agent-Address AVP is of type Address defined in RFC 4004.

This AVP shall contain either IPv4 or IPv6 address of the PDN-GW and this IP address shall be used as the PDN-GW IP address as indicated defined in 3GPP TS 29.272.

### 5.1.1.4 MIP-Home-Agent-Host

The MIP-Home-Agent-Host AVP is of type Grouped and is defined in RFC 4004.

This AVP shall contain a FQDN of the PDN-GW which shall be used to resolve the PDN-GW IP address using the Domain Name Service function as defined in 3GPP TS 29.272.

The AVP Data field has the following ABNF grammar:

```
MIP-Home-Agent-Host ::= <AVP Header: 348 >
                        { Destination-Realm }
                        { Destination-Host }
                        * [ AVP ]
```

### 5.1.1.5 EAP-Payload

The EAP-Payload AVP is of type OctetString defined in RFC 4072.

This AVP is used to encapsulate the actual EAP packet that is being exchanged between the EAP client and the home Diameter server.

### 5.1.1.6 EAP-Master-Session-Key

The EAP-Master-Session-Key AVP is of type OctetString defined in RFC 4072.

This AVP contains keying material for protecting the communications between the user and the NAS.



#### 5.1.1.7 MIP6-Agent-Info

The MIP6-Agent-Info AVP is of type Grouped as defined in RFC 5447. It contains the identity of the PDNGW as defined in 3GPP TS 29.272.

The identity of PDN GW is either an IP address transported in MIP-Home-Agent-Address or an FQDN transported in MIP-Home-Agent-Host. FQDN shall be used if known.

The AVP Data field has the following ABNF grammar:

```
MIP6-Agent-Info ::= <AVP Header: 486>
                    *2 [ MIP-Home-Agent-Address ]
                    [ MIP-Home-Agent-Host ]
                    [ MIP6-Home-Link-Prefix ]
                    * [ AVP ]
```

Within the MIP6-Agent-Info AVP, if static address allocation is used, there may be either: an IPv4 address or an IPv6 address of the PGW contained in one MIP-Home-Agent-Address AVP; both IPv4 address and IPv6 address of the PGW contained in two MIP-Home-Agent-Address AVPs.

#### 5.1.1.8 Service-Selection

The Service-Selection AVP is of type UTF8String defined in Reference RFC5778.

This AVP shall contain either the APN Network Identifier (i.e. an APN without the Operator Network Identifier), or the wild card value as defined in Reference 3GPP TS 29.272.

The contents of the Service-Selection AVP shall be formatted as a character string composed of one or more labels separated by dots (".") or as the wild card APN, i.e., consisting of only one ASCII label.

#### 5.1.1.9 Mobile-Node-Identifier

The Mobile-Node-Identifier AVP is of type UTF8String defined in RFC 5779 and contains the mobile node identifier (MN-Identifier; see [RFC5213]) in the NAI [RFC4282] format.

This AVP is used on the MAG-to-HAAA interface. The Mobile-Node-Identifier AVP is designed for deployments where the MAG does not have a way to find out such MN identity that could be used in subsequent PBU/PBA exchanges (e.g., due to identity hiding during the network access authentication) or the HAAA wants to assign periodically changing identities to the MN.

The Mobile-Node-Identifier AVP is returned in the answer message that ends a successful authentication (and possibly an authorization) exchange between the MAG and the HAAA, assuming the HAAA is also able to provide the MAG with the MN-Identifier in the first place. The MAG MUST use the received

MN-Identifier, if it has not been able to get the mobile node identifier through other means. If the MAG already has a valid mobile node identifier, then the MAG MUST silently discard the received MN-Identifier.

#### 5.1.1.10 QoS-Profile-Template

The QoS-Profile-Template AVP (AVP Code 574) is of type Grouped defined in RFC 5777 and defines the namespace of the QoS profile (indicated in the Vendor-ID AVP) followed by the specific value for the profile.

The Vendor-Id AVP contains a 32-bit IANA Private Enterprise Number(PEN), and the QoS-Profile-Id AVP contains the template identifier assigned by the vendor. The vendor identifier of zero (0) is used for the IETF.

The AVP Data field has the following ABNF grammar:

```
QoS-Profile-Template ::= < AVP Header: 574 >
                        { Vendor-Id }
                        { QoS-Profile-Id }
                        * [ AVP ]
```

#### 5.1.1.11 QoS-Profile-Id

The QoS-Profile-Id AVP (AVP Code 573) is of type Unsigned32 defined in RFC 5777 and contains a QoS profile template identifier.

An initial QoS profile template is defined with value of 0 and can be found in [RFC5624]. The registry for the QoS profile templates is created with the same document.

#### 5.1.1.12 QoS-Capability

The QoS-Capability AVP contains a list of supported Quality of Service profile templates (and therefore the support of the respective parameter AVPs).

This AVP is defined in IETF RFC 5777.

The AVP Data field has the following ABNF grammar:

```
QoS-Capability ::= < AVP Header: 578 >
                  1*{ QoS-Profile-Template }
                  * [ AVP ]
```

#### 5.1.1.13 Subscription-ID

The Subscription-ID AVP is of type Grouped and indicates the user identity to be used for charging purposes. It is defined in the IETF RFC 4006 [20]. EPC shall make use only of the IMSI and MSISDN values. This grouped AVP shall set the sub-AVP Subscription-Id-Type to value "END\_USER\_E164" and shall set the sub-AVP Subscription-IdData to the MSISDN value.



The AVP Data field has the following ABNF grammar:

```
Subscription-Id ::= < AVP Header: 443 >
                  { Subscription-Id-Type }
                  { Subscription-Id-Data }
```

#### 5.1.1.14 Subscription-Id-Type

The Subscription-Id-Type AVP (AVP Code 450) is of type Enumerated as shown in Table 4.

This AVP determines which type of identifier is carried by the Subscription-Id AVP.

*Table 4 Subscription-Id-Type*

END_USE R_E164	0	The identifier is in international E.164 format (e.g., MSISDN), according to the ITU-T E.164 numbering plan defined in [E164] and [CE164].
END_USE R_IMSI	1	The identifier is in international IMSI format, according to the ITU-T E.212 numbering plan as defined in [E212] and [CE212].
END_USE R_SIP_URI	2	The identifier is in the form of a SIP URI, as defined in [SIP].
END_USE R_NAI	3	The identifier is in the form of a Network Access Identifier as defined in [NAI].
END_USE R_PRIVATE	4	The Identifier is a credit-control server private identifier.

#### 5.1.1.15 Subscription-Id-Data

The Subscription-Id-Data AVP (AVP Code 444) is used to identify the end user and is of type UTF8String.

The Subscription-Id-Type AVP defines which type of identifier is used.

#### 5.1.1.16 Proxy-Info

The Proxy-Info AVP is of type Grouped defined in RFC 6733. This AVP shall contain the identity and local state information of the Diameter node that creates and adds it to a message.

#### 5.1.1.17 Route-Record

The Route-Record AVP is of type DiameterIdentity defined in RFC 6733. The identity added in this AVP must be the same as the one received in the Origin-Host of the Capabilities Exchange message.

## 5.1.2 3GPP AVPs

Table 5 describes the 3GPP AVPs defined in the SWm/SWm+ application, their AVP Code values, types and possible AVP flag values.

The 3GPP AVPs have Vendor-ID= 10415.

**Table 5 3GPP AVPs in SWm/SWm+ Application**

Attribute Name	AVP Code	Value Type	Must	May	Should not	Must not	Description
3GPP-Charging-Characteristics	13	UTF8String	V			M	See Section 5.1.2
Max-Requested-Bandwidth-DL	515	Unsigned32	M,V				See Section 5.1.2
Max-Requested-Bandwidth-UL	516	Unsigned32	M,V				See Section 5.1.2
Visited-Network-Identifier	600	OctetString	M,V				See Section 5.1.2
Supported-Features	628	Grouped	V			M	See Section 5.1.2
Feature-List-ID	629	Unsigned32	V			M	See Section 5.1.2
Feature-List	630	Unsigned32	V			M	See Section 5.1.2
Served-Party-IP-Address	848	Address	M,V				See Section 5.1.2
QoS-Class-Identifier	1028	Enumerated	M,V				See Section 5.1.2
MIP6-Feature-Vector	1032	Enumerated	M,V	P			See Section 5.1.2
Allocation-Retention-Priority	1034	Grouped	V			M	See Section 5.1.2
Priority-Level	1046	Unsigned32	V			M	See Section 5.1.2
Pre-emption-Capability	1047	Enumerated	V			M	See Section 5.1.2
Pre-emption-Vulnerability	1048	Enumerated	V			M	See Section 5.1.2
Terminal-Information	1401	Grouped	M,V				See Section 5.1.2
IMEI	1402	UTF8String	M,V				See Section 5.1.2
Software-Version	1403	UTF8String	M,V				See Section 5.1.2
Context-Identifier	1423	Unsigned32	M,V				See Section 5.1.2
APN-OI-Replacement	1427	UTF8String	M,V				See Section 5.1.2
APN-Configuration	1430	Grouped	M,V				See Section 5.1.2
EPS-Subscribed-QoS-Profile	1431	Grouped	M,V				See Section 5.1.2
VPLMN-Dynamic-Address-Allowed	1432	Enumerated	M,V				See Section 5.1.2
AMBR	1435	Grouped	M,V				See Section 5.1.2
PDN-GW-Allocation-Type	1438	Enumerated	M,V				See Section 5.1.2
PDN-Type	1456	Enumerated	M,V				See Section 5.1.2
3GPP2-MEID	1471	OctetString	M,V				See Section 5.1.2
Specific-APN-Info	1472	Grouped	M,V				See Section 5.1.2
AN-Trusted	1503	Enumerated	M,V	P			See Section 5.1.2
ANID	1504	UTF8String	M,V				See Section 5.1.2
AAA-Failure-Indication	1518	Unsigned32	V			M,P	See Section 5.1.2
Trace-Info	1505	Grouped	V			M,P	See Section 5.1.2





**Table 5** 3GPP AVPs in SWm/SWm+ Application

Attribute Name	AVP Code	Value Type	Must	May	Should not	Must not	Description
Trace-Data	1458	Grouped	M, V				See Section 5
Trace-Reference	1459	OctetString	M, V				See Section 5
Trace-Depth	1462	Enumerated	M, V				See Section 5
Trace-NE-Type-List	1463	OctetString	M, V				See Section 5
Trace-Interface-List	1464	OctetString	M, V				See Section 5
Trace-Event-List	1465	OctetString	M, V				See Section 5
OMC-Id	1466	OctetString	M, V				See Section 5
Trace-Collection-Entity	1452	Address	M, V				See Section 5
UE-Local-IP-Address	2805	Address	V				See Section 5
Emergency-Services	1538	Unsigned32	V			M,P	See Section 5

### 5.1.2.1 3GPP-Charging-Characteristics

The 3GPP-Charging-Characteristics AVP is of type UTF8String. It contains the Charging Characteristics is defined in Reference 3GPP TS 29.061.

The structure of the Charging Characteristics value according to 3GPP TS 32.299 is as follows:

**Table 6** 3GPP-Charging-Characteristics

8	7	6	5	4	3	2	1	
B4	B3	B2	B1	P3	P2	P1	P0	octet 1
B12	B11	B10	B9	B8	B7	B6	B5	octet 2

Bits P0-P3 refer to the Charging Characteristics Profile Index and B1-B12 may be used by the operator for non-standardised behavior.

Each octet of the Charging Characteristics value is represented via 2 UTF-8 encoded characters in the 3GPP-Charging-Characteristics AVP, defining its hexadecimal representation. For example, if P3 and P1 are set to 1, and all the B bits are set to 0, the value of octet 1 is 10, which hexadecimal representation is 0x0A, and in text form is "0A". Octet 2 is set to 0, represented as 0x00 in hexadecimal and "00" in text, so the 3GPP-Charging-Characteristics value in UTF-8 would be "0A00".

### 5.1.2.2 Max-Requested-Bandwidth-DL

The Max-Requested-Bandwidth-DL AVP is of type Unsigned32 defined in 3GPP TS 29.124 and it indicates the maximum requested bandwidth in bits per second for a downlink IP flow.

The bandwidth contains all the overhead coming from the IP-layer and the layers above, for instance: IP, UDP, RTP and RTP payload.

### 5.1.2.3 Max-Requested-Bandwidth-UL

The Max-Requested-Bandwidth-UL AVP is of type Unsigned32 defined in 3GPP TS 29.124, and it indicates the maximum requested bandwidth in bits per second for an uplink IP flow.

The bandwidth contains all the overhead coming from the IP-layer and the layers above, for instance: IP, UDP, RTP and RTP payload.

### 5.1.2.4 Visited-Network-Identifier

The Visited-Network-Identifier AVP defined in Reference 3GPP TS 29.229 is of type OctetString.

This AVP contains an identifier of the visited network when it is received in MAR message. Otherwise it contains the identity of the network where the PDN-GW was allocated, in the case of dynamic PDN-GW assignment.

The AVP shall be encoded as defined in 3GPP TS 29.272:  
`mnc<MNC>.mcc<MCC>.3gppnetwork.org`

### 5.1.2.5 Supported-Features

The Supported-Features AVP is of type Grouped and it is defined in Reference 3GPP TS 29.229. If this AVP is present it may inform the destination host about the features that the origin host supports. The Feature-List AVP contains a list of supported features of the origin host. The Vendor-Id AVP and the Feature-List AVP together identify which feature list is carried in the Supported-Features AVP. Where a Supported-Features AVP is used to identify features that have been defined by 3GPP, the Vendor-Id AVP contains the vendor ID of 3GPP.

Vendors may define proprietary features, but it is strongly recommended that the possibility is used only as the last resort. Where the Supported-Features AVP is used to identify features that have been defined by a vendor other than 3GPP, it contains the vendor ID of the specific vendor in question. If there are multiple feature lists defined by the same vendor, the Feature-List-ID AVP differentiates those lists from one another.

The destination host uses the value of the Feature-List-ID AVP to identify the feature list.

The AVP Data field has the following ABNF grammar:

```
Supported-Features ::= <AVP Header: 628, Vendor-Id: 10415>
                        {Vendor-Id}
                        {Feature-List-ID}
                        {Feature-List}
                        * [AVP]
```



#### 5.1.2.6 Served-Party-IP-Address

The Served-Party-IP-Address AVP defined in 3GPP TS 32.299 is of type Address.

This AVP contains the IPv4 address, the IPv6 address or the IPv6 prefix of the user, if static IP address allocation is used. For the IPv6 prefix, the lower 64 bits of the address shall be set to zero.

#### 5.1.2.7 QoS-Class-Identifier

The QoS-Class-Identifier AVP is of type Enumerated defined in 3GPP TS 29.212, and it identifies a set of IP-CAN specific QoS parameters that define the authorized QoS, excluding the applicable bitrates for the IP-CAN bearer or service flow.

#### 5.1.2.8 MIP6-Feature-Vendor

The MIP6-Feature-Vector AVP contains a 64 bit flags field of supported mobility capabilities of the NAS. This AVP is defined in IETF RFC 5447. The NAS may include this AVP in a request message to indicate the mobility capabilities of the NAS to the 3GPP AAA server. Similarly, the Diameter server may include this AVP in an answer message to inform the NAS about which of the NAS indicated capabilities are supported or authorized by the 3GPP AAA Server.

Following capabilities are supported on S6b reference point in PMIPv6 mode:

- PMIPv6\_SUPPORTED
- IP4\_HOA\_SUPPORTED

#### 5.1.2.9 RAT-Type

The RAT-Type AVP is of type Enumerated defined in 3GPP TS 29.212 and is used to identify the radio access technology that is serving the UE.

#### 5.1.2.10 Allocation-Retention-Priority

The Allocation-Retention-Priority AVP is of type Grouped defined in 3GPP TS 29.212. It indicates Priority of Allocation and Retention for the corresponding Access Point Name (APN) configuration within the Priority-Level AVP.

The AVP Data field has the following ABNF grammar:

```
Allocation-Retention-Priority ::= <AVP Header: 1034 , Vendor-Id: 10
                                { Priority-Level }
                                *[ AVP ]
```

### 5.1.2.11 Priority-Level

The Priority-Level AVP is of type Unsigned 32 defined in 3GPP TS 29.212. The priority level defines the relative importance of a resource request. The AVP is used for deciding whether a bearer establishment or modification request can be accepted or needs to be rejected in case of resource limitations. The AVP can also be used to decide which existing bearers to pre-empt during resource limitations. The priority level defines the relative importance of a resource request. Values 1 to 15 are defined, with value 1 as the highest level of priority.

### 5.1.2.12 Pre-emption-Capability

The Pre-emption-Capability AVP is of type Enumerated defined in 3GPP TS 29.212. The AVP defines whether a service data flow can get resources that were already assigned to another service data flow with a lower priority level.

The following values are defined:

PRE-EMPTION_CAPABILITY_ENABLED (0)	This value indicates that the service data flow is allowed to get resources that were already assigned to another service data flow with a lower priority level.
PRE-EMPTION_CAPABILITY_DISABLED (1)	This value indicates that the service data flow is not allowed to get resources that were already assigned to another service data flow with a lower priority level. This is the default value applicable if this AVP is not supplied.

### 5.1.2.13 Pre-emption-Vulnerability

The Pre-emption Vulnerability AVP is of type Enumerated defined in 3GPP TS 29.212. The AVP defines whether a service data flow can lose the resources assigned to it in order to admit a service data flow with higher priority level.

The following values are defined:

PRE-EMPTION_VULNERABILITY_ENABLED (0)	This value indicates that the resources assigned to the service data flow can be pre-empted and allocated to a service data flow with a higher priority level. This is the default value applicable if this AVP is not supplied.
PRE-EMPTION_VULNERABILITY_DISABLED (1)	This value indicates that the resources assigned to the service data flow shall not be pre-empted and allocated to a service data flow with a higher priority level.

### 5.1.2.14 APN-Configuration

The APN-Configuration AVP is of type Grouped defined in 3GPP TS 29.272. It contains the information related to APN configuration for a single APN.

The AVP Data field has the following ABNF grammar:

```
APN-Configuration ::= <AVP Header: 1430, Vendor-Id: 10415>
{ Context-Identifier }
```



```

{Service-Selection}
{PDN-Type}
*2 [Served-Party-IP-Address]
[MIP6-Agent-Info]
[Visited-Network-Identifier]
[PDN-GW-Allocation-Type]
[EPS-Subscribed-QoS-Profile]
[VPLMN-Dynamic-Address-Allowed]
[3GPP-Charging-Characteristics]
[AMBR]
* [Specific-APN-Info]
* [AVP]

```

#### 5.1.2.15 Context-Identifier

The Context-Identifier AVP is of type Unsigned32 defined in 3GPP TS 29.272 and uniquely identifies an EPS APN configuration within a user subscription.

#### 5.1.2.16 APN-OI-Replacement

The APN-OI-Replacement AVP is of type UTF8String defined in 3GPP TS 29.272. This AVP Indicates the domain name to replace the APN Operator Identifier (OI) when constructing the Packet Data Network (PDN) Gateway (GW) Fully Qualified Domain Name (FQDN) upon which to perform a Domain Name System (DNS) resolution. The contents of the APN-OI-Replacement AVP are formatted as a character string composed of one or more labels separated by dots (".").

#### 5.1.2.17 EPS-Subscribed-QoS-Profile

The EPS-Subscribed-QoS-Profile AVP is of type Grouped defined in 3GPP TS 29.272 and contains the bearer-level QoS parameters (QoS Class Identifier and Allocation Retention Priority) associated to the default bearer for an APN.

The AVP Data field has the following ABNF grammar:

```

EPS-Subscribed-QoS-Profile ::= <AVP Header: 1431, Vendor-Id: 10415>
    { QoS-Class-Identifier }
    { Allocation-Retention-Priority }
    * [ AVP ]

```

#### 5.1.2.18 VPLMN-Dynamic-Address-Allowed

The VPLMN-Dynamic-Address-Allowed AVP is of type Enumerated defined in 3GPP TS 29.272. It indicates whether the UE is allowed to use a dynamic address allocated in the Visited PLMN (VPLMN).

The following values are defined:

0	NOT ALLOWED
1	ALLOWED

#### 5.1.2.19 AMBR

The AMBR AVP is of type Grouped defined in 3GPP TS 29.212, and contains AVPs that indicate the aggregate maximum bitrates requested for the uplink and downlink bandwidth.

The AVP Data field has the following ABNF grammar:

```
AMBR ::= <AVP Header: 1435 , Vendor-Id:10415>
        {Max-Requested-Bandwidth-UL}
        {Max-Requested-Bandwidth-DL}
        * [AVP]
```

#### 5.1.2.20 PDN-GW-Allocation-Type

The PDN-GW-Allocation-Type AVP is of type Enumerated defined in 3GPP TS 29.212 and indicates whether the PDN GW address is statically allocated or dynamically selected by other nodes.

The following values are defined:

0	ALLOWED
1	DYNAMIC

#### 5.1.2.21 PDN-Type

The PDN-Type AVP is of type Enumerated defined in 3GPP TS 29.272 and indicates the address type of PDN.

The following values are defined:

Table 7 PDN-Type

AVP Value	Meaning
0	IPv4: This value shall be used to indicate that the PDN can be accessed only in IPv4 mode.
1	IPv6: This value shall be used to indicate that the PDN can be accessed only in IPv6 mode.



AVP Value	Meaning
2	IPv4v6: This value shall be used to indicate that the PDN can be accessed both in IPv4 mode, in IPv6 mode, and also from UEs supporting dualstack IPv4v6.
3	IPv4_OR_IPv6: This value shall be used to indicate that the PDN can be accessed either in IPv4 mode, or in IPv6 mode, but not from UEs supporting dualstack IPv4v6.

#### 5.1.2.22 Specific-APN-Info

The Specific-APN-Info AVP is of type Grouped defined in 3GPP TS 29.272. It shall only be present in the APN configuration when the APN is a wild card APN. It shall contain the APN which is not present in the subscription context but the UE is authorized to connect to and the identity of the registered PDN-GW and optionally the PLMN Id of the PDN GW.

The AVP Data field has the following ABNF grammar:

```
Specific-APN-Info ::= <AVP Header : 1472, Vendor Id: 10415 >
                        {Service-Selection}
                        [MIP6-Agent-Info]
                        [Visited-Network-Identifier]
                        * [AVP]
```

#### 5.1.2.23 AN-Trusted

The AN-Trusted AVP is of type Enumerated defined in 3GPP TS 29.273.

The AN-Trusted AVP sent from the 3GPP AAA Server to the Non-3GPP access network conveys the decision about the access network being trusted or untrusted by the HPLMN.

The following values are defined:

TRUSTED (0)	This value is used when the non-3GPP IP access network is to be handled as trusted.
UNTRUSTED (1)	This value is used when the non-3GPP IP access network is to be handled as untrusted.

#### 5.1.2.24 ANID

The ANID AVP is of type UTF8String defined in 3GPP TS 29.212 and contains the Access Network Identity. It is used for key derivation function when the authentication method is EAP-AKA'. See 3GPP TS 24.302 for possible values.

#### 5.1.2.25 Terminal-Information

The Terminal-Information AVP is of type Grouped defined in 3GPP TS 29.272. This AVP contains the information about the user's terminal.

The AVP Data field has the following ABNF grammar:

```
Terminal-Information ::= <AVP Header: 1401 , Vendor-Id: 10415>
                        [IMEI]
                        [3GPP2-MEID]
                        [Software-Version]
                        * [AVP]
```

#### 5.1.2.26 IMEI

The IMEI AVP is of type UTF8String defined in 3GPP TS 29.272. This AVP contains the International Mobile Equipment Identity.

#### 5.1.2.27 3GPP2-MEID

The 3GPP2-MEID AVP is of type OctetString defined in 3GPP TS 29.272. This AVP contains the Mobile Equipment Identifier of the user's terminal.

#### 5.1.2.28 Software-Version

The Software-Version AVP is of type UTF8String. This AVP defined in 3GPP TS 29.272, contains the Software Version of the International Mobile Equipment Identity.

#### 5.1.2.29 Feature-List-ID

The Feature-List-ID AVP is of type Unsigned32 defined in 3GPP TS 29.229 and it contains the identity of a feature list.

#### 5.1.2.30 Feature-List

The Feature-List AVP is of type Unsigned32 defined in 3GPP TS 29.229 and it contains a bit mask indicating the supported features of an application. When the bit set, indicates the corresponding feature is supported by the application.

#### 5.1.2.31 AAA-Failure-Indication

The AAA-Failure-Indication AVP is of type Unsigned32 and it shall contain a bitmask.

Table 8 shows the meaning of the bits:

**Note:** Bits not defined in this table must be cleared by the sender and discarded by the receiver.





*Table 8 AAA-Failure-Indication*

Bit	Name	Description
0	AAA Failure	This bit, when set, indicates that a previously assigned 3GPP AAA Server is unavailable.

### 5.1.2.32

#### Trace-Info

The Trace-Info AVP is of type Grouped. This AVP shall contain the information related to subscriber and equipment trace function and the required action, i.e. activation of deactivation.

The AVP Data field has the following ABNF grammar:

```
Trace-Info ::= <AVP header: 1505 10415>
               [Trace-Data]
               [Trace-Reference]
               * [AVP]
```

Either the Trace-Data or the Trace-Reference AVP shall be included. When trace activation is needed, Trace-Data AVP shall be included, while the trace deactivation request shall be signalled by including the Trace-Reference directly under the Trace-Info. The Trace-Reference AVP is of type OctetString. The Diameter AVP is defined in 3GPP TS 29.272.

### 5.1.2.33

#### Trace-Data

The Trace-Data AVP is of type Grouped. The Diameter AVP is defined in 3GPP TS 29.272. This AVP shall contain the information related to trace function.

The AVP Data field has the following ABNF grammar:

```
Trace-Data ::= <AVP header: 1458 10415>
               {Trace-Reference}
               {Trace-Depth}
               {Trace-NE-Type-List}
               [Trace-Interface-List]
               {Trace-Event-List}
               [OMC-Id]
               {Trace-Collection-Entity}
               * [AVP]
```

### 5.1.2.34

#### Trace-Reference

The Trace-Reference AVP is of type OctetString.

This AVP shall contain the concatenation of MCC, MNC and Trace ID, where the Trace ID is a 3 byte Octet String.

Table 9 shows the content of this AVP shall be encoded as octet strings:

**Table 9 Trace-Reference**

8	7	6	5	4	3	2	
MCC digit 2				MCC digit 1			
MNC digit 3				MCC digit 3			
MNC digit 2				MNC digit 1			
Trace ID							

### 5.1.2.35

#### Trace-Depth

The Trace-Depth AVP is of type Enumerated. The possible values are those defined in 3GPP TS 32.422 for Trace Depth.

Trace depth shall be an enumerated parameter with the following possible values:

0	Minimum
1	Medium
2	Maximum
3	MinimumWithoutVendorSpecificExtension
4	MediumWithoutVendorSpecificExtension
5	MaximumWithoutVendorSpecificExtension

### 5.1.2.36

#### Trace-NE-Type-List

The Trace-NE-Type-List AVP is of type OctetString. Octets are coded according to 3GPP TS 32.422.

The Network Element types are listed as follows:

- MSC Server
- MGW
- RNC
- SGSN



- GGSN
- BM-SC
- MME
- SGW
- PDN GW
- eNB

#### **5.1.2.37 Trace-Interface-List**

The Trace-Interface-List AVP is of type OctetString. Octets are coded according to 3GPP TS 32.422.

#### **5.1.2.38 Trace-Event-List**

The Trace-Event-List AVP is of type OctetString. Octets are coded according to 3GPP TS 32.422.

#### **5.1.2.39 OMC-Id**

The OMC-Id AVP is of type OctetString. Octets are coded according to 3GPP TS 29.002

#### **5.1.2.40 Trace-Collection-Entity**

The Trace-collection-Entity AVP is of type Address and contains the IPv4 or IPv6 address of the Trace Collection Entity, as defined in 3GPP TS 32.422.

#### **5.1.2.41 UE-Local-IP-Address**

The UE-Local-IP-Address AVP is of type Address and contains the UE local IP address. The UE local IP address type can be IPv4 or IPv6.

#### **5.1.2.42 Emergency-Services**

The Emergency-Services AVP of type Unsigned32 and it shall contain a bitmask. The PGW shall include this information element, with the Emergency-Indication bit set, during the establishment of an emergency PDN connection.

**Note:** Bits not defined in this table shall be cleared by the sender and discarded by the receiver.

The meaning of the bits is defined in Table 10:

*Table 10 Emergency-Services*

Bit	Name	Description
0	Emergency-Indication	This bit, when set, indicates a request to establish a PDN connection for emergency services.

## 5.2 Messages

### 5.2.1 EAP Messages

#### 5.2.1.1 EAP-Response/Identity Message

The type of EAP Response message is *Identity*, and the user identity is carried in the data value.

**Note:** When the customized feature IMASK Mask Handling is enabled, for an identity encrypted non-SIM user, the EAP-Response/Identity must be encrypted with RSA-OAEP (SHA265) + BASE64.

#### 5.2.1.2 EAP-Request/Identity Message

**Note:** It is for customized feature IMASK Mask Handling. It can be enabled by configuring.

When IPWorks AAA received a DER message with anonymous identity, IPWorks AAA replies with EAP-Request/Identity.

#### 5.2.1.3 EAP-Success

EAP Success message does not have any EAP data carried.

#### 5.2.1.4 EAP-Failure

EAP Failure message does not have any EAP data carried.

### 5.2.2 EAP-AKA Messages and Attributes

Table 11 guides which attributes are in which kinds of messages, and in what quantity.

The messages denoted with numbers in parentheses are listed as follows:

- 1.EAP-Request/AKA-Identity
- 2. EAP-Response/AKA-Identity
- 3. EAP-Request/AKA-Challenge



- 4. EAP-Response/AKA-Challenge
- 5. EAP-Request/AKA-Notification
- 6. EAP-Response/AKA-Notification
- 7. EAP-Response/AKA-Client-Error
- 8. EAP-Request/AKA-Reauthentication
- 9. EAP-Response/AKA-Reauthentication
- 10. EAP-Response/AKA-Authentication-Reject
- 11. EAP-Response/AKA-Synchronization-Failure

The column denoted with "E" indicates whether the attribute is a nested attribute that **MUST** be included within AT\_ENCR\_DATA:

- "0" indicates that the attribute **MUST NOT** be included in the message;
- "1" indicates that the attribute **MUST** be included in the message;
- "0-1" indicates that the attribute is sometimes included in the message
- "0\*" indicates that the attribute is not included in the message in cases specified in this document, but **MAY** be included in the future versions of the protocol.

**Table 11** EAP-AKA Attributes

Attribute	1 <sup>(1)</sup>	2	3	4	5	6	7	8	9	10
AT_PERMANENT_ID_REQ	0-1	0	0	0	0	0	0	0	0	0
AT_ANY_ID_REQ	0-1	0	0	0	0	0	0	0	0	0
AT_FULLAUTH_ID_REQ	0-1	0	0	0	0	0	0	0	0	0
AT_IDENTITY	0	0-1	0	0	0	0	0	0	0	0
AT_RAND	0	0	1	0	0	0	0	0	0	0
AT_AUTN	0	0	1	0	0	0	0	0	0	0
AT_RES	0	0	0	1	0	0	0	0	0	0
AT_AUTS	0	0	0	0	0	0	0	0	0	0
AT_NEXT_PSEUDONYM	0	0	0-1	0	0	0	0	0	0	0
AT_NEXT_REAUTH_ID	0	0	0-1	0	0	0	0	0-1	0	0
AT_IV	0	0	0-1	0*	0-1	0-1	0	0	0	0
AT_ENCR_DATA	0	0	0-1	0*	0-1	0-1	0	0	0	0

**Table 11 EAP-AKA Attributes**

AT_PADDIN G	0	0	0-1	0*	0-1	0-1	0	0	0	0
AT_CHECKC ODE	0	0	0-1	0-1	0	0	0	0	0	0
AT_RESULT _IND	0	0	0-1	0-1	0	0	0	0	0	0
AT_MAC	0	0	1	1	0-1	0-1	0	0	0	0
AT_COUNT ER	0	0	0	0	0-1	0-1	0	1	1	0
AT_COUNT R_TOO_SMA LL	0	0	0	0	0	0	0	0	0-1	0
AT_NONCE_ S	0	0	0	0	0	0	0	1	0	0
AT_NOTIFIC ATION	0	0	0	0	1		0	0	0	0
AT_CLIENT_ ERROR_CO DE	0	0	0	0	0	0	1	0	0	0

(1) Numbers in the first row represent the EAP-AKA messages.

### 5.2.3 EAP-TLS and TLS Messages

The EAP protocol provides a standard mechanism for support of multiple authentication methods. Through the use of EAP, support for a number of authentication schemes may be added. EAP-Transport Layer Security (EAP-TLS) includes support for certificate-based mutual authentication and key derivation, utilizing the protected ciphersuite negotiation, mutual authentication and key management capabilities of the TLS protocol. The TLS Handshake Protocol is one of the defined higher level clients of the TLS Record Protocol. This protocol is used to negotiate the secure attributes of a session. Handshake messages are supplied to the TLS Record Layer, where they are encapsulated within one or more TLSPlaintext structures, which are processed and transmitted as specified by the current active session state. The EAP-TLS authentication protocol details please refer to RFC5216. The TLS protocol details please refer to RFC5246. Here only describe the basic functionalities of involved EAP-TLS and TLS messages.

Table 12 guides which attributes are in which kinds of messages, and in what quantity.

The messages denoted with numbers in parentheses are listed as follows:

- 1. EAP-Response/TLS-ClientHello
- 2. EAP-Request/TLS-ServerHello
- 3. EAP-Response/TLS-ClientChangeCipherSpec
- 4. EAP-Request/TLS-ServerChangeCipherSpec



The column denoted with "E" indicates whether the attribute is a nested attribute that MUST be included within AT\_ENCR\_DATA:

- "0" indicates that the attribute MUST NOT be included in the message;
- "1" indicates that the attribute MUST be included in the message;
- "0-1" indicates that the attribute is sometimes included in the message
- "0\*" indicates that the attribute is not included in the message in cases specified in this document, but MAY be included in the future versions of the protocol.

**Table 12 EAP-TLS Attributes**

Attribute	1 <sup>(1)</sup>	2	3	4
client_hello	1	0	0	0
server_hello	0	1	0	0
server_key_exchange	0	1	0	0
certificate_request	0	1	0	0
server_hello_done	0	1	0	0
certificate	0	1	1	0
client_key_exchange	0	0	1	0
certificate_verify	0	0	1	0
change_cipher_spec	0	0	1	1
finished	0	0	1	1

(1) Numbers in the first row represent the EAP-TLS and TLS messages.

### 5.2.3.1 TLS-Start

Once having received the peer's Identity, the EAP server MUST respond with an EAP-TLS/Start packet, which is an EAP-Request packet with EAP-Type=EAP-TLS, the Start (S) bit set, and no data. The EAP-TLS conversation then begins, with the peer sending an EAP-Response packet with EAP-Type=EAP-TLS.

### 5.2.3.2 Client\_hello

When a client first connects to a server, it is required to send the ClientHello as its first message. The client can also send a ClientHello in response to a HelloRequest or on its own initiative to renegotiate the security parameters in an existing connection. The client hello includes a list of compression algorithms supported by the client, ordered according to the client's preference.

The structure of the message is displayed as follows:

```
struct {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suites<2..216-2>;
    CompressionMethod compression_methods<1..28-1>;
    select (extensions_present) {
        case false:
            struct {};
        case true:
            Extension extensions<0..216-1>;
    };
} ClientHello;
```

### 5.2.3.3 Server\_hello

The server sends this message in response to a `ClientHello` message when it finds an acceptable set of algorithms. If the server does not find such a match, it responds with a handshake failure alert.

The structure of the message is displayed as follows:

```
Structure of this message:
struct {
    ProtocolVersion server_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suite;
    CompressionMethod compression_method;
    select (extensions_present) {
        case false:
            struct {};
        case true:
            Extension extensions<0..216-1>;
    };
} ServerHello;
```

### 5.2.3.4 Server Certificate

The server **MUST** send a `Certificate` message whenever the agreed-upon key exchange method uses certificates for authentication (this includes all key exchange methods defined in this document except `DH_anon`). This message always immediately follows the `ServerHello` message. This message conveys the server's certificate chain to the client. The certificate **MUST** be appropriate for the negotiated cipher suite's key exchange algorithm and any negotiated extensions.

The structure of the message is displayed as follows:





```
opaque ASN.1Cert<1..224-1>;
struct {
  ASN.1Cert certificate_list<0..224-1>;
} Certificate;
```

### 5.2.3.5 Server\_key\_exchange

This message is sent immediately after the server certificate message (or the server hello message, if this is an anonymous negotiation). The server key exchange message is sent by the server only when the server certificate message (if sent) does not contain enough data to allow the client to exchange a premaster secret.

The structure of the message is displayed as follows:

```
struct {
  select (KeyExchangeAlgorithm) {
    case dh_anon:
      ServerDHParams params;
    case dhe_dss:
    case dhe_rsa:
      ServerDHParams params;
      digitally-signed struct {
        opaque client_random[32];
        opaque server_random[32];
        ServerDHParams params;
      } signed_params;
    case rsa:
    case dh_dss:
    case dh_rsa:
      struct {} ;
  };
} ServerKeyExchange;
```

### 5.2.3.6 Certificate\_request

A non-anonymous server optionally requests a certificate from the client, if appropriate for the selected cipher suite. This message, if sent, immediately follows the Server Key Exchange message (if it is sent; otherwise, the Server Certificate message).

The structure of the message is displayed as follows:

```
enum {
  rsa_sign(1), dss_sign(2), rsa_fixed_dh(3), dss_fixed_dh(4),
  rsa_ephemeral_dh_RESERVED(5), dss_ephemeral_dh_RESERVED(6),
  fortaleza_dms_RESERVED(20), (255)
} ClientCertificateType;
opaque DistinguishedName<1..216-1>;
struct {
```

```
ClientCertificateType certificate_types<1..2^8-1>;
SignatureAndHashAlgorithm
    supported_signature_algorithms<2^16-1>;
DistinguishedName certificate_authorities<0..2^16-1>;
```

### 5.2.3.7 Server\_hello\_done

The server hello done message is sent by the server to indicate the end of the server hello and associated messages. After sending this message the server waits for a client response.

This message means that the server is done sending messages to support the key exchange, and the client can proceed with its phase of the key exchange. Upon receipt of the server hello done message the client should verify that the server provided a valid certificate if required and check that the server hello parameters are acceptable.

The structure of the message is displayed as follows:

```
struct {
    } ServerHelloDone;
```

### 5.2.3.8 Client Certificate

This is the first message the client can send after receiving a ServerHelloDone message. This message is only sent if the server requests a certificate. If no suitable certificate is available, the client MUST send a certificate message containing no certificates. That is, the certificate\_list structure has a length of zero. If the client does not send any certificates, the server MAY at its discretion either continue the handshake without client authentication, or respond with a fatal handshake\_failure alert. Also, if some aspect of the certificate chain was unacceptable (e.g., it was not signed by a known, trusted CA), the server MAY at its discretion either continue the handshake (considering the client unauthenticated) or send a fatal alert. Client certificates are sent using the Certificate structure defined in Section 5.2.3.4

### 5.2.3.9 Client\_key\_exchange

This message is always sent by the client. It will immediately follow the client certificate message, if it is sent. Otherwise it will be the first message sent by the client after it receives the server hello done message.

The structure of the message is displayed as follows:

```
struct {
    select (KeyExchangeAlgorithm)
    {
        case rsa:
            EncryptedPreMasterSecret;
        case dhe_dss:
```



```

        case dhe_rsa:
        case dh_dss:
        case dh_rsa:
        case dh_anon:
            ClientDiffieHellmanPublic;
    } exchange_keys;
} ClientKeyExchange;

```

### 5.2.3.10 Certificate\_verify

This message is used to provide explicit verification of a client certificate. This message is only sent following a client certificate that has signing capability (i.e., all certificates except those containing fixed Diffie-Hellman parameters). When sent, it MUST immediately follow the client key exchange message.

The structure of the message is displayed as follows:

```

Struct {
    digitally-signed struct {
        opaque handshake_messages[handshake_messages_length]
    }
} CertificateVerify;

```

### 5.2.3.11 Change\_cipher\_spec

The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed under the current (not the pending) connection state. The message consists of a single byte of value 1.

The ChangeCipherSpec message is sent by both the client and the server to notify the receiving party that subsequent records will be protected under the newly negotiated CipherSpec and keys

The structure of the message is displayed as follows:

```

struct {
    enum { change_cipher_spec(1), (255) } type;
} ChangeCipherSpec;

```

### 5.2.3.12 Finish

Finished message is always sent immediately after a change cipher spec message to verify that the key exchange and authentication processes were successful. It is essential that a change cipher spec message be received between the other handshake messages and the Finished message.

The structure of the message is displayed as follows:

```

struct {

```



```
        opaque verify_data[verify_data_length];  
    } Finished;
```



## 6 Error Handling

### 6.1 Diameter Error Handling

Table 13 Diameter Error Handling

Error Scenario	Return Code
Checking Diameter messages according to dictionary	DIAMETER_AVP_UNSUPPORTED(5001) + Failed-AVP
	DIAMETER_INVALID_AVP_VALUE(5004) + Failed-AVP
	DIAMETER_MISSING_AVP(5005) + Failed-AVP
	DIAMETER_AVP_NOT_ALLOWED(5008)
	DIAMETER_AVP_OCCURS_TOO_MANY_TIMES(5009)
	DIAMETER_UNSUPPORTED_VERSION(5011)
Missing mandatory AVPs required by the 3GPP TS, for example, the User-Name AVP in the DER message	Result-Code = "DIAMETER_MISSING_AVP"
	Failed-AVP should be set with missing AVP
Checking the RAT-Type in the initial authentication DER message failed	Result-Code = "DIAMETER_UNABLE_TO_COMPLY"
HSS returns DIAMETER_ERROR_USER_UNKNOWN	Experimental-Result-Code = "DIAMETER_ERROR_USER_UNKNOWN"
HSS returns DIAMETER_ERROR_USER_NO_NON_3GPP_SUBSCRIPTION	Experimental-Result-Code = "DIAMETER_ERROR_USER_NO_NON_3GPP_SUBSCRIPTION"
HSS returns DIAMETER_ERROR_ROAMING_NOT_ALLOWED	Experimental-Result-Code = "DIAMETER_ERROR_ROAMING_NOT_ALLOWED"
HSS returns DIAMETER_ERROR_IDENTITY_ALREADY_REGISTERED	Result-Code = "DIAMETER_REDIRECT_INDICATION"
HSS returns other errors	Result-Code = "DIAMETER_UNABLE_TO_COMPLY"
For the re-authentication and re-authorization, User-Name or Identity does not belong to the same session	Experimental-Result-Code = "DIAMETER_ERROR_USER_UNKNOWN"
For the re-authentication and re-authorization, RAT-Type does not belong to the same session	Result-Code = "DIAMETER_UNABLE_TO_COMPLY"
For the re-authorization, session is not found in the AAA server	Result-Code = DIAMETER_UNKNOWN_SESSION_ID
For the re-authorization, user does not belong to the same session	Experimental-Result-Code = DIAMETER_ERROR_USER_UNKNOWN
For the re-authorization, user profile is not found in AAA server	Result-Code = "DIAMETER_AUTHORIZATION_REJECTED"
For the session termination, session is not found in AAA server	Result-Code = "DIAMETER_UNKNOWN_SESSION_ID"
For the session termination, user does not belong to the same session	Experimental-Result-Code = DIAMETER_ERROR_USER_UNKNOWN
The de-registration from HSS fails	Result-Code = "DIAMETER_UNABLE_TO_COMPLY"
No requested APN in UserProfile of SIM-based user	Experimental-Result-Code = "DIAMETER_ERROR_USER_NO_APN_SUBSCRIPTION"
No requested APN in provisioned data of non-SIM based user	Experimental-Result-Code = "DIAMETER_ERROR_USER_NO_APN_SUBSCRIPTION"
User not found or user was disabled for no-SIM user authorization	Result-Code = "DIAMETER_AUTHORIZATION_REJECTED"
No matched APN-Configuration info in the configurable APN parameter files for no-SIM based user authorization	Result-Code = "DIAMETER_UNABLE_TO_COMPLY"



Table 13 Diameter Error Handling

Error Scenario	Return Code
EAP-TLS authentication failed for no-SIM based user authentication	Result-Code = "DIAMETER_AUTHENTICATION_REJECTED"
IMEI Check Failure	Experimental-Result-Code= "DIAMETER_ERROR_ILLEGAL_EQUIPMENT"

## 6.2 EAP Error Handling

Table 14 EAP Error Handling

Error Scenario	Return Code
EAP message length, code and type error	Result-Code = "DIAMETER_AUTHENTICATION_REJECTED", EAP-Payload should be set with the value from the request DER
Unknown subtype of EAP AKA message	Discard the message
EAP AKA message attribute error	AAA server MUST issue the EAP-Request/AKA-Notification packet with an AT_NOTIFICATION code that implies failure (16384 or 0). The Result-Code should be set to DIAMETER_MULTI_ROUND_AUTH. If the EAP-Response/ AKA-Notification is received from the Trusted Non-3GPP Access Network, AAA server should send EAP-Failure message with code equals 4, the Result-Code of DEA message should be set to DIAMETER_AUTHENTICATION_REJECTED.
EAP-Response/Identity started with unknown digit (not '0', '2', '4,' \0') or with the unspecified prefix string	Result-Code = "DIAMETER_AUTHENTICATION_REJECTED"
Fail to validate certificate attribute in EAP-Response/TLS-ClientChangeCipherSpec	Result-Code = "DIAMETER_AUTHENTICATION_REJECTED"
Validating AT_MAC or AT_RES in EAP-Response/AKA-Challenge message fails	AAA server MUST issue the EAP-Request/AKA-Notification packet with an AT_NOTIFICATION code that implies failure (0). The Result-Code should be set to DIAMETER_MULTI_ROUND_AUTH. If the EAP-Response/ AKA-Notification is received from the Trusted Non-3GPP Access Network, AAA server should send EAP-Failure message with code equals 4, the Result-Code of DEA message should be set to DIAMETER_AUTHENTICATION_REJECTED.
AT_CHECKCODE error	AAA server MUST issue the EAP-Request/AKA-Notification packet with an AT_NOTIFICATION code that implies failure (16384 or 0). The Result-Code should be set to DIAMETER_MULTI_ROUND_AUTH. If the EAP-Response/ AKA-Notification is received from the Trusted Non-3GPP Access Network, AAA server should send EAP-Failure message with code equals 4, the Result-Code of DEA message should be set to DIAMETER_AUTHENTICATION_REJECTED.
Receiving EAP-Response/AKA-Client-Error and EAP-Response/AKA-Authentication-Reject from Access Network	AAA server sends EAP-Failure to the peer and the DEA Result-Code is DIAMETER_AUTHENTICATION_REJECTED.
	For re-authentication, AAA server should send SAR message to HSS with the Server-Assignment-Type set to AUTHENTICATION_FAILURE
Receiving EAP-Response/AKA-Synchronization-Failure from Access Network	The 3GPP AAA server should delete the old authentication vector and send MAR message to fetch new authentication vectors from HSS
The re-authentication fails (with the EAP-Failure returned to ePDG)	AAA server should send SAR message to HSS with the Server-Assignment-Type set to AUTHENTICATION_FAILURE
The authentication using pseudonym identity is not supported and AAA server receives the pseudonym identity	The AAA server should send EAP-Request/AKA-Identity to request permanent identity.
The pseudonym identity is supported and AAA server can't parse or recognize it.	
The pseudonym identity is supported and the received pseudonym identity is out of date	

*Table 14 EAP Error Handling*

Error Scenario	Return Code
The authentication using fast re-authentication identity is not supported and AAA server receives the fast re-authentication identity	The AAA server should send EAP-Request/AKA-Identity to request permanent identity.
The fast re-authentication identity is supported and AAA server can't parse or recognize it	
The fast re-authentication identity is supported and the received fast re-authentication identity is out of date	
Unknown type of EAP response message(data type not 'identity', 'aka', 'aka_prime','tls')	Discard the message







## 7 Formal Syntax

This section refers to the specification where the formal syntax notation is defined. It also describes any deviations from the specification, if applicable.

Not Applicable.





## 8 Related Standards

- 3GPP EPS AAA interfaces 3GPP TS 29.273 version 12.5.0
- Diameter Base Protocol RFC 3588
- Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA) RFC 4187
- Extensible Authentication Protocol (EAP) RFC 3748 Diameter Extensible Authentication Protocol (EAP) Application RFC 4072
- EAP-TLS Authentication Protocol RFC 5216





## Reference List

### IPWorks Library Documents

- [1] *Trademark Information*
- [2] *Typographic Conventions*
- [3] *Glossary of Terms and Acronyms*

### Standards

- [4] [Diameter Base Protocol RFC 3588](#)
- [5] [Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement \(EAP-AKA\) RFC 4187](#)
- [6] [Extensible Authentication Protocol \(EAP\) RFC 3748](#)
- [7] [Diameter Extensible Authentication Protocol \(EAP\) Application RFC 4072](#)
- [8] [Universal Mobile Telecommunications System \(UMTS\); LTE3GPP EPS AAA interfaces; Evolved Packet System \(EPS\); \(3GPP TS 29.273 version 12.5.0 Release 12\)](#)
- [9] [EAP-TLS Authentication Protocol RFC 5216](#)