3GPP TSG SA WG3 Security — S3#25 **S3-020511**
8 - 11 October 2002
Munich, Germany

Source: Ericsson and Nokia

Title: IETF and WLAN Authentication Methods

Document for: Information

Agenda item:: 7.9

1 Introduction

3GPP-WLAN interworking is expected to be based on sharing a common authentication infrastructure through EAP (Extensible Authentication Protocol) methods. The purpose of this report is to describe the status of the various documents in IETF that are needed to support this.

2 IETF Documents

EAP is a new working group in the IETF, mainly tasked with the publication of a revised version of the base EAP RFC. In addition, there are a number of individual submissions that are currently outside the official scope of the WG (but still discussed in the WG meetings).

The IETF has the following documents that are related to EAP and WLAN interworking:

- Base EAP specification, RFC 2284. This document is in Proposed Standard status and widely supported in products. However, the specification is also currently being updated and improved in “RFC 2284 bis” in draft-ietf-eap-rfc2284bis-06.txt. This new version will clarify the following aspects of RFC 2284:
    - IANA considerations. There were no rules on how to allow the allocation of new EAP type numbers, making the process free and risking running out of the 256 number space in a few years. It is expected that the new IANA rules will require IETF expert review and public specification before allowing new numbers to be allocated.
    - State machine. The original protocol lacked a state machine description. With the use of EAP in multiple environments, and the introduction of more complicated methods (such as tunneling) the need for better rules about the behaviour has become apparent. A design team is currently working on a new state machine, and making it aligned with the new revision of the 802.1X state machine which is being worked on by the IEEE. An early state machine definition can be found from draft-payne-eap-sm-00.txt.
    - Clarifications on error behaviour.
    - Other potential improvements.
- General framework for providing session keys from EAP is a work item of the EAP, but does not have an associated document yet.
- The definition of the GSM authentication method, draft-haverinen-pppext-eap-sim-06.txt.
- The definition of the UMTS authentication method, draft-arkko-pppext-eap-aka-05.txt.

The official plans for completing the RFC 2284 bis work call for submission of the finished document by December 2002. However, it is likely that the IANA considerations part will be submitted before that in order to give guidance for the IANA on allocating new EAP method type numbers. Also, it is likely that the full bis RFC, including state machine descriptions, will slip from the December schedule. This implies that 3GPP should base its work on RFC 2284. 3GPP should also use methods that can provide keys without reliance on the general purpose keying framework (EAP AKA and EAP SIM can do this).

The EAP SIM and EAP AKA methods have recently been updated to address some open issues from IETF mailing list discussions, as described in Section 3. The revised documents are included in the end of this submission. Once there is consensus that all open issues have been resolved, the drafts will be submitted directly to the RFC Editor as Informational submissions outside any Working Group. The IETF process allows this type of submissions as long as they do not collide with work within any existing Working Group, which we believe the case to be here.

3 Recent Changes in EAP AKA and EAP SIM drafts

The following changes have being made to the new versions of EAP AKA and EAP SIM drafts. The changes result from IETF mailing list discussions and comments received on the drafts.

- An optional lightweight re-authentication procedure has been included in the drafts. Such re-authentication can be used network models where EAP authentication is performed frequently. Re-authentication is based on session keys derived on preceding full authentication.
- Annexes for key derivation specification for different wireless LAN technologies, e.g. IEEE 802.11i.
- Optional MAC codes in EAP SIM and EAP AKA changed to mandatory, because you cannot detect if an optional MAC code is removed.
- Refinements on MAC calculation details of EAP SIM (discussion on EAP mailing list). MAC codes now cover the whole EAP packet, similarly to some other EAP methods.
- The user identity (Network Access Identifier) is protected by including it in key derivation in EAP SIM and EAP AKA. (Some other EAP methods also protect the NAI.)
- EAP SIM includes version negotiation to facilitate backward-compatible revisioning of the protocol.
- Editorial improvements

4 Conclusions

3GPP SA3 members are encouraged to actively participate the IETF mailing list discussion about the open issues in the IETF drafts related to WLAN interworking.

Annex A EAP SIM

Point-to-Point Extensions Working Group H. Haverinen
Internet Draft Nokia
J. Salowey
Cisco
October 2002

EAP SIM Authentication
draft-haverinen-pppext-eap-sim-06.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at:
http://www.ietf.org/shadow.html.

This document is an individual submission for the Point-to-Point Extensions Working Group of the Internet Engineering Task Force (IETF). Comments should be submitted to the ietf-ppp@merit.edu mailing list.

Distribution of this memo is unlimited.

Abstract

This document specifies an Extensible Authentication Protocol (EAP) mechanism for authentication and session key distribution using the GSM Subscriber Identity Module (SIM). The mechanism specifies enhancements to GSM authentication and key agreement whereby multiple authentication triplets can be combined to create authentication responses and encryption keys of greater strength than the individual GSM triplets. The mechanism also includes network authentication, user anonymity support and a re-authentication procedure.

Table of Contents

1. Introduction

This document specifies an Extensible Authentication Protocol (EAP) [1] mechanism for authentication and session key distribution using the GSM Subscriber Identity Module (SIM).

GSM authentication is based on a challenge-response mechanism. The A3/A8 authentication algorithms that run on the SIM can be given a 128-bit random number (RAND) as a challenge. The SIM runs an operator-specific algorithm, which takes the RAND and a secret key Ki stored on the SIM as input, and produces a 32-bit response (SRES) and a 64-bit long key Kc as output. The Kc key is originally intended to be used as an encryption key over the air interface. Please find more information about GSM authentication in [2].

In EAP/SIM, several RAND challenges are used for generating several 64-bit Kc keys, which are combined to constitute a longer session key. EAP/SIM also enhances the basic GSM authentication mechanism by accompanying the RAND challenges with a message authentication code in order to provide mutual authentication.

EAP/SIM specifies optional support for protecting the privacy of subscriber identity and an optional re-authentication procedure.

2. Terms

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [3].

This document frequently uses the following terms and abbreviations:

AAA protocol

Authentication, Authorization and Accounting protocol

AAA server

In this document, AAA server refers to the network element that resides on the border of Internet AAA network and GSM network. Cf. EAP server

AuC

Authentication Centre. The GSM network element that can authenticate the subscriber.

Client

The entity that processes the EAP protocol on the supplicant. Typically, it is the end that needs to be authenticated by the authenticator. The Client includes a SIM that executes sensible cryptographic calculations.

EAP

Extensible Authentication Protocol.

EAP Server

The network element that terminates the EAP protocol. Typically, the EAP server functionality is implemented in a AAA server.

GSM

Global System for Mobile communications.

IMSI

International Mobile Subscriber Identifier, used in GSM to identify subscribers.

NAI

Network Access Identifier

SIM

Subscriber Identity Module. The SIM is an application traditionally resident on smart cards distributed by GSM operators.

3. Overview

Figure 1 shows an overview of the EAP/SIM full authentication procedure. This version of EAP/SIM exchange uses three roundtrips to authenticate the user and generate session keys. In this document, the term EAP Server refers to the network element that terminates the EAP protocol. The Authenticator typically communicates with the user's EAP server using an AAA protocol. The AAA communications is not shown in the figure.

The first EAP Request issued by the Authenticator is EAP-Request/Identity. The client's response includes either the user's International Mobile Subscriber Identity (IMSI) or a temporary identity (pseudonym), as specified in Section 1.

Following the client's EAP-Response/Identity packet, the client receives EAP Requests of type 18 (SIM) from the Authenticator and sends the corresponding EAP Responses. The EAP packets that are of the Type SIM also have a Subtype field. On full authentication, the first EAP-Request/SIM packet is of the Subtype 10 (Start). EAP SIM packets encapsulate parameters in attributes, encoded in a Type, Length, Value format. The packet format and the use of attributes are specified in Section 9.

The EAP-Request/SIM/Start packet contains the list of EAP/SIM version supported by the Authenticator in the AT_VERSION_LIST attribute. This packet may also include attributes for requesting the subscriber identity, as specified in Section 7.

The client responds to EAP-Request/SIM/Start with the EAP-Response/SIM/Start packet, which includes the AT_NONCE_MT attribute that contains a random number NONCE_MT, chosen by the client, and the AT_SELECTED_VERSION attribute that contains the version number selected by the client. The version negotiation is protected by including the version list and the selected version in the calculation of session keys (Section 19). The client MUST NOT reuse the NONCE_MT value from previous sessions but the client MUST choose it freshly for each EAP/SIM authentication exchange. The client SHOULD use a good source of randomness to generate NONCE_MT.

In this document, we assume that the EAP server has an interface to the GSM network and it operates as a gateway between the Internet AAA network and the GSM authentication infrastructure. After receiving the EAP Response/SIM/Start, the EAP server obtains n GSM triplets from the user’s home operator’s Authentication Centre (AuC) on the GSM network, where n = 2 or n = 3. From the triplets, the EAP server derives the keying material, as specified in Section 19.

The next EAP Request the Authenticator issues is of the type SIM and subtype Challenge (11). It contains the RAND challenges and a message authentication code attribute AT_MAC to cover the challenges. On receipt of this message, the client runs the GSM authentication algorithm and calculates a copy of the message authentication code. The client then verifies that the calculated MAC equals the received MAC. If the MAC’s do not match, then the client silently ignores the EAP packet and does not send any authentication values to the network. Eventually, if another EAP-Request/SIM/Challenge packet with a valid AT_MAC is not received, the connection establishment will time out.

Since the RAND’s given to a client are accompanied with the message authentication code AT_MAC, and since the client’s NONCE_MT value contributes to AT_MAC, the client is able to verify that the RAND’s are fresh and they have been generated by the GSM network.

If all checks out, the client responds with the EAP-Response/SIM/Challenge, containing the AT_MAC attribute that covers the client’s SRES response values (Section 19). The EAP server verifies that the MAC is correct and sends the EAP-Success packet, indicating that the authentication was successful. The EAP server may also include derived keying material in the message it sends to the Authenticator.

```
     Client                                                Authenticator
        |                                                          |
        |                           EAP-Request/Identity           |
        |<---------------------------------------------------------|
        |                                                          |
        | EAP-Response/Identity                                    |
        |--------------------------------------------------------->|
        |                                                          |
        |                        EAP-Request/SIM/Start             |
        |                        (AT_VERSION_LIST)                 |
        |<---------------------------------------------------------|
        |                                                          |
        | EAP-Response/SIM/Start                                   |
        | (AT_NONCE_MT, AT_SELECTED_VERSION)                       |
        |--------------------------------------------------------->|
        |                                                          |
        |               EAP-Request/SIM/Challenge                  |
        |               (AT_RAND, AT_MAC)                          |
        |<---------------------------------------------------------|
        |                                                          |
   +-------------------------------------+                         |
   | Client runs GSM algorithms,         |                         |
   | verifies AT_MAC and derives         |                         |
   | session keys                        |                         |
   +-------------------------------------+                         |
        |                                                          |
        | EAP-Response/SIM/Challenge                               |
        | (AT_MAC)                                                 |
        |--------------------------------------------------------->|
        |                                                          |
        |                                                          |
        |                                             EAP-Success  |
        |<---------------------------------------------------------|
        |                                                          |
```

Figure 1 EAP/SIM full authentication procedure

EAP SIM also includes a separate re-authentication procedure, which does not make use of the A3/A8 algorithms or the GSM infrastructure. Re-authentication is based on keys derived on full authentication.

4. Version Negotiation

EAP/SIM includes version negotiation so as to allow future developments in the protocol. The version negotiation is performed on full authentication and it uses two attributes, AT_VERSION_LIST (Section 11), which the server includes in EAP-Request/SIM/Start, and AT_SELECTED_VERSION (Section 12), which the client includes in EAP-Response/SIM/Start.

AT_VERSION_LIST includes the EAP/SIM versions supported by the server. The server MUST only include versions that it implements and that are allowed in its security policy. The versions are listed in the order of preference, most preferred versions first. At least one version number MUST be included. The version number for the protocol described in this document is one (0x0001).

If AT_VERSION_LIST does not include a version that is implemented by the client and allowed in the client’s security policy, then the client MUST silently ignore the EAP-Response/SIM/Start packet. If a suitable version is included, then the client includes the AT_SELECTED_VERSION attribute, containing the selected version, in the EAP-Response/SIM/Start packet. The client MUST only indicate a version that is included in AT_VERSION_LIST. If several versions are acceptable, then the client SHOULD choose the version that occurs first in the version list.

The version number list of AT_VERSION_LIST and the selected version of AT_SELECTED_VERSION are included in the key derivation procedure (Section 19). If an attacker modifies either one of these attributes, then the client and the server will derive different keying material. Because K_aut keys are different, the server and client will calculate different AT_MAC values. Hence, the client will detect that AT_MAC is incorrect and discard the EAP-Request/SIM/Challenge packet. The authentication procedure will time out.

5. User identity in EAP-Response/Identity

In the beginning of EAP authentication, the Authenticator issues the EAP-Request/Identity packet to the client. The client responds with EAP-Response/Identity, which contains the user's identity. The formats of these packets are specified in [1].

GSM subscribers are identified with the International Mobile Subscriber Identity (IMSI) [4]. The IMSI is composed of a three digit Mobile Country Code (MCC), a two or three digit Mobile Network Code (MNC) and a not more than 10 digit Mobile Subscriber Identification Number (MSIN). In other words, the IMSI is a string of not more than 15 digits. MCC and MNC uniquely identify the GSM operator.

Internet AAA protocols identify users with the Network Access Identifier (NAI) [5]. When used in a roaming environment, the NAI is composed of a username and a realm, separated with "@" (username@realm). The username portion identifies the subscriber within the realm. The AAA nodes use the realm portion of the NAI to route AAA requests to the correct AAA server. The realm name used in this protocol MAY be chosen by the operator and it MAY a configurable parameter in the EAP/SIM client implementation. In this case, the client is typically configured with the NAI realm of the home operator. Operators MAY reserve a specific realm name for EAP/SIM users. This convention makes it easy to recognize that the NAI identifies a GSM subscriber. Such reserved NAI realm may be useful as a hint as to the first authentication method to use during method negotiation.

There are three types of NAI username portions in EAP/SIM: non-pseudonym permanent usernames that are based on the IMSI, pseudonym usernames and re-authentication usernames. The first two are only used on full authentication and the last one only on re-authentication. When the optional IMSI privacy support is not used, the non-pseudonym permanent username is used. The non-pseudonym permanent username is of the format "1imsi". In other words, the first character of the username is the digit one (ASCII value 0x31), followed by the IMSI. The IMSI is an ASCII string that consists of not more than 15 decimal digits (ASCII values between 0x30 and 0x39) as specified in [9].

The EAP server MAY use the leading "1" as a hint to try EAP/SIM as the first authentication method during method negotiation, rather than for example EAP/AKA. The EAP/SIM server MAY propose EAP/SIM even if the leading character was not "1".

When the optional identity privacy support is used on full authentication, the client MAY use the pseudonym received as part of the previous full authentication sequence as the username portion of the NAI, as specified in Section 7. The client MUST NOT modify the pseudonym received in AT_NEXT_PSEUDONYM. For example, the client MUST NOT append any leading characters in the pseudonym.

On re-authentication, the client uses the re-authentication identity received as part of the previous authentication sequence as the NAI. A new re-authentication identity may be delivered as part of both full authentication and re-authentication. The client MUST NOT modify the re-authentication identity received in AT_NEXT_REAUTH_ID. For example, the client MUST NOT append any leading characters in the re-authentication identity.

If no configured realm name is available, the client MAY derive the realm name from the MCC and MNC portions of the IMSI. In this case, the realm name is obtained by concatenating "mnc", the MNC digits of IMSI, ".mcc", the MCC digits of IMSI and ".owlan.org". For example, if the IMSI is 123456789098765, and the MNC is three digits long, then the derived realm name is "mnc456.mcc123.owlan.org".

If the client is not able to determine whether the MNC is two or three digits long, the client MAY use a 3-digit MNC. If the correct length of the MNC is two, then the MNC used in the realm name will include the first digit of MSIN. Hence, when configuring AAA networks for operators that have 2-digit MNC’s, the network SHOULD also be prepared for realm names with incorrect 3-digit MNC’s.

6. Obtaining Subscriber Identity via EAP/SIM Messages

It may be useful to obtain the identity of the subscriber through means other than EAP Request/Identity. This can eliminate the need for an identity request when using EAP method negotiation. If this was not possible then it might not be possible to negotiate EAP/SIM as the second method since it is not specified how to deal with a new EAP Request/Identity.

If the EAP server does not have any identity (IMSI, pseudonym or re-authentication username) available when sending the first EAP/SIM request, then the EAP server may issue the EAP-Request/SIM/Start packet and includes the AT_ANY_ID_REQ attribute (specified in Section 11). This attribute does not contain any data.

The AT_ANY_ID_REQ attribute requests the client to include the AT_IDENTITY attribute (specified in Section 12) in the EAP-Response/SIM/Start packet. The identity format in the AT_IDENTITY attribute is the same as in the EAP-Response/Identity packet. The AT_IDENTITY attribute contains an IMSI-based permanent identity, a pseudonym identity or a re-authentication identity. If the server does not support re-authentication, it uses the AT_FULLAUTH_ID_REQ attribute instead of the AT_ANY_ID_REQ attribute to directly request for a full authentication identity (either the permanent identity or a pseudonym identity). If the server uses the AT_FULLAUTH_ID_REQ attribute, the client MUST NOT use a re-authentication identity in the AT_IDENTITY attribute.

The use of pseudonyms for anonymity is specified in Section 7. The use of re-authentication identities is specified in Section 8.

This case for full authentication is illustrated in the figure below. In this case, AT_IDENTITY contains either the permanent identity or a pseudonym identity. The same sequence is also used in case the server uses the AT_FULLAUTH_ID_REQ in EAP-Request/SIM/Start.

```
Client                                            Authenticator
       |                                                       |
       |                            +-------------------------------+
       |                            | Server does not have any      |
       |                            | Subscriber identity available|
       |                            | When starting EAP/SIM         |
       |                            +-------------------------------+
       |                                                       |
       |          EAP-Request/SIM/Start                        |
       |          (AT_ANY_ID_REQ, AT_VERSION_LIST)             |
       |<------------------------------------------------------|
       |                                                       |
       |                                                       |
       | EAP-Response/SIM/Start                                |
       | (AT_IDENTITY, AT_NONCE_MT,                            |
       |  AT_SELECTED_VERSION)                                 |
       |------------------------------------------------------>|
       |                                                       |
```

If the client wants to perform full authentication, it includes the permanent identity or a pseudonym identity in the AT_IDENTITY attribute. The client may use these identities in response to either AT_ANY_ID_REQ or AT_FULLAUTH_ID_REQ. In this case, the client MUST include AT_NONCE_MT and AT_SELECTED_VERSION attributes in EAP-Response/SIM/Start message, as required on full authentication.

If the server uses the AT_ANY_ID_REQ and the client wants to perform re-authentication, then the client includes a re-authentication identity in the AT_IDENTITY attribute. On re-authentication, the client MUST NOT include AT_NONCE_MT or AT_SELECTED_VERSION attributes. This case is illustrated below.

```
Client                                                Authenticator
       |                                                        |
       |                            +-------------------------------+
       |                            | Server does not have any      |
       |                            | Subscriber identity available|
       |                            | When starting EAP/SIM         |
       |                            +-------------------------------+
       |                                                        |
       |        EAP-Request/SIM/Start                           |
       |        (AT_ANY_ID_REQ, AT_VERSION_LIST)                |
       |<-------------------------------------------------------|
       |                                                        |
       |                                                        |
       | EAP-Response/SIM/Start                                 |
       | (AT_IDENTITY containing a re-authentication identity) |
       |------------------------------------------------------->|
       |                                                        |
```

If the client uses its full authentication identity and the AT_IDENTITY attribute contains a valid permanent identity or a valid pseudonym identity that the EAP server is able to decode to the permanent identity, then the full authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/SIM/Challenge message.

On re-authentication, if the AT_IDENTITY attribute contains a valid re-authentication identity and the server agrees on using re-authentication, then the server proceeds with the re-authentication sequence and issues the EAP-Request/SIM/Re-authentication packet, as specified in Section 8. If the server does not recognize the re-authentication identity, then the server issues a second EAP-Request/SIM/Start message and includes the AT_FULLAUTH_ID_REQ attribute. In this case, a second EAP/SIM/Start round trip is required. The messages used on the first roundtrip are ignored. This is illustrated below.

```
   Client                                              Authenticator
          |                                                        |
          |                             +------------------------------+
          |                             | Server does not have any     |
          |                             | Subscriber identity available|
          |                             | When starting EAP/SIM        |
          |                             +------------------------------+
          |                                                        |
          |        EAP-Request/SIM/Start                           |
          |        (AT_ANY_ID_REQ, AT_VERSION_LIST)                |
          |<-------------------------------------------------------|
          |                                                        |
          |                                                        |
          | EAP-Response/SIM/Start                                 |
          | (AT_IDENTITY containing a re-authentication identity) |
          |------------------------------------------------------->|
          |                                                        |
          |                             +------------------------------+
          |                             | Server does not recognize    |
          |                             | The re-authentication        |
          |                             | Identity                     |
          |                             +------------------------------+
          |                                                        |
          |     EAP-Request/SIM/Start                              |
          |     (AT_FULLAUTH_ID_REQ, AT_VERSION_LIST)              |
          |<-------------------------------------------------------|
          |                                                        |
          |                                                        |
          | EAP-Response/SIM/Start                                 |
          | (AT_IDENTITY with a full-auth. identity, AT_NONCE_MT, |
          |  AT_SELECTED_VERSION)                                  |
          |------------------------------------------------------->|
          |                                                        |
```

If the server recognizes the re-authentication identity, but still wants to fall back on full authentication, the server may issue the EAP-Request/SIM/Start packet without any identity request attributes (AT_FULLAUTH_ID_REQ or AT_PERMANENT_ID_REQ). In this case, the server only includes the AT_VERSION_LIST attribute, and full authentication proceeds as usual. The client does not include any identity attributes in the EAP-Response/SIM/Start packet.

An extra EAP/SIM/Start round trip is also required in cases when the AT_IDENTITY attribute contains a pseudonym identity that the EAP server fails to decode. The operation in this case is specified in Section 7.

7. Identity Privacy Support

EAP/SIM includes optional identity privacy (anonymity) support that can be used to hide the cleartext IMSI and to make the subscriber’s connections unlinkable to eavesdroppers. Identity privacy is based on temporary identities, or pseudonyms, which are equivalent to but separate from the Temporary Mobile Subscriber Identities (TMSI) that are used on cellular networks.

If identity privacy is not used or if the client does not have any pseudonyms or re-authentication identities are available, the client transmits the permanent identity (based on IMSI) in the EAP-Response/Identity packet or in the AT_IDENTITY attribute.

The EAP-Request/SIM/Challenge message MAY include an encrypted pseudonym in the value field of the AT_ENCR_DATA attribute. The AT_IV and AT_MAC attributes are also used to transport the pseudonym to the client, as described in Section 13. Because the identity privacy support is optional to implement, the client MAY ignore the AT_IV and AT_ENCR_DATA attributes and always transmit the IMSI-based permanent identity in the EAP-Response/Identity packet and in the AT_IDENTITY attribute.

On receipt of the EAP-Request/SIM/Challenge, the client verifies the AT_MAC attribute before looking at the AT_ENCR_DATA attribute. If the AT_MAC is invalid, then the client MUST silently discard the EAP packet. If the AT_MAC attribute is valid, then the client MAY decrypt the encrypted data in AT_ENCR_DATA and use the obtained pseudonym on the next full authentication.

If the client does not receive a new pseudonym in the EAP-Request/SIM/Challenge message, the client MAY use an old pseudonym instead of the permanent identity on next full authentication.

The EAP server produces pseudonyms in an implementation-dependent manner. Please see [6] for examples on how to produce pseudonyms. Only the EAP server needs to be able to map the pseudonym to the permanent identity. Regardless of construction method, the pseudonym MUST conform to the grammar specified for the username portion of an NAI. The EAP SIM server MAY produce pseudonyms that begin with a leading "1" character in order to be able to use the leading character as a hint in EAP method negotiation during next authentication.

On the next full authentication with the EAP server, the client MAY transmit the received pseudonym in the first EAP-Response/Identity packet. The client concatenates the received pseudonym with the "@" character and the NAI realm portion. The client selects the realm name portion similarly as it select the realm name portion when using the permanent identity. If the EAP server successfully decodes the pseudonym received in the EAP-Response/Identity packet to a known client identity (IMSI), the authentication proceeds with the EAP-Request/SIM/Start message as usual.

Because the client may fail to save a pseudonym sent to in an EAP-Request/SIM/Challenge, for example due to malfunction, the EAP server SHOULD maintain at least one old pseudonym in addition to the most recent pseudonym.

If the EAP server requests the client to include its identity in the EAP-Response/SIM/Start packet, as specified in Section 6, the client MAY transmit the received pseudonym in the AT_IDENTITY attribute. If the EAP server successfully decodes the pseudonym to a known identity, then the authentication proceeds with the EAP-Request/SIM/Challenge packet as usual.

If the EAP server fails to decode the pseudonym to a known identity, then the EAP server requests the permanent identity (non-pseudonym identity) by including the AT_PERMANENT_ID_REQ attribute (Section 11) in the EAP-Request/SIM/Start message.

The EAP server issues the EAP-Request/SIM/Start message also in the case when it received the undecodable pseudonym in AT_IDENTITY included the EAP-Response/SIM/Start packet. In this case, an extra EAP/SIM/Start round trip is required.

A received AT_PERMANENT_ID_REQ does not necessarily originate from the valid network, but an active attacker may transmit an EAP-Request/SIM/Start packet with an AT_PERMANENT_ID_REQ attribute to the client, in an effort to find out the true identity of the user. On receipt of EAP-Request/SIM/Start that includes AT_PERMANENT_ID_REQ, the client MAY delay the processing of the message for a while in order to wait for another EAP-Request/SIM/Start without AT_PERMANENT_ID_REQ.

Basically, there are two different policies that the client can employ with regard to AT_PERMANENT_ID_REQ. A "conservative" client assumes that the network is able to maintain pseudonyms robustly. Therefore, if a conservative client has a pseudonym, the client silently ignores the EAP packet with AT_PERMANENT_ID_REQ, because the client believes that the valid network is able to decode the pseudonym. (Alternatively, the conservative client may respond to AT_PERMANENT_ID_REQ in certain circumstances, for example if the pseudonym was received a long time ago.) The benefit of this policy is that it protects the client against active attacks on anonymity. On the other hand, a "liberal" client always accepts the AT_PERMANENT_ID_REQ and responds with the IMSI-based permanent identity. The benefit of this policy is that it works even if the valid network sometimes loses pseudonyms and is not able to decode them to the permanent identity.

Regardless how the identity is communicated to the server, the full authentication sequence is performed similarly in all cases. For example, AT_NONCE_MT and AT_SELECTED_VERSION are always included in the EAP-Response/SIM/Start packet on full authentication, even if they were already transmitted in the previous EAP-Response/SIM/Start. AT_VERSION_LIST is also included in every EAP-Request/SIM/Start message. The values used on the last EAP/SIM/round trip are used and the previous EAP/SIM/Start round trips is ignored. The NONCE_MT value and the version negotiation attributes included in the last EAP-Response/SIM/Start packet are used in all calculations. The EAP/SIM client MAY use the same NONCE_MT value in both EAP-Response/SIM/Start packets.

The value field of the AT_PERMANENT_ID_REQ does not contain any data but the attribute is included to request the client to include the AT_IDENTITY attribute (Section 12) with the permanent authentication identity in the EAP-Response/SIM/Start message. In this case, the AT_IDENTITY attribute contains the client’s permanent identity in the clear.

Please note that the EAP/SIM client and the EAP/SIM server only process the AT_IDENTITY attribute and entities that only pass through EAP packets do not process this attribute. Hence, if the EAP server is not co-located in the authenticator, then the authenticator and other intermediate AAA elements (such as possible AAA proxy servers) will continue to refer to the client with the original identity from the EAP-Response/Identity packet regardless if the decoding fails in the EAP server.

The figure below illustrates the case when the EAP server fails to decode the pseudonym included in the EAP-Response/Identity packet.

```
Client                                                Authenticator
       |                                                        |
       |                              EAP-Request/Identity      |
       |<-------------------------------------------------------|
       |                                                        |
       | EAP-Response/Identity                                  |
       | (Includes a pseudonym)                                 |
       |------------------------------------------------------->|
       |                                                        |
       |                             +-----------------------------+
       |                             | Server fails to decode the  |
       |                             | Pseudonym.                  |
       |                             +-----------------------------+
       |                                                        |
       |  EAP-Request/SIM/Start                                 |
       |  (AT_PERMANENT_ID_REQ, AT_VERSION_LIST)                |
       |<-------------------------------------------------------|
       |                                                        |
       |                                                        |
       | EAP-Response/SIM/Start                                 |
       | (AT_IDENTITY with permanent identity, AT_NONCE_MT,     |
       |  AT_SELECTED_VERSION)                                  |
       |------------------------------------------------------->|
       |                                                        |
```

After the EAP-Response/SIM/Start message, the authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/SIM/Challenge message.

The figure below illustrates the case when the EAP server fails to decode the pseudonym included in the AT_IDENTITY attribute.

```
   Client                                              Authenticator
          |                                                      |
          |                            +-----------------------------+
          |                            | Server does not have any    |
          |                            | Subscriber identity available|
          |                            | When starting EAP/SIM       |
          |                            +-----------------------------+
          |                                                      |
          |        EAP-Request/SIM/Start                         |
          |        (AT_ANY_ID_REQ, AT_VERSION_LIST)              |
          |<-----------------------------------------------------|
          |                                                      |
          |                                                      |
          |EAP-Response/SIM/Start                                |
          |(AT_IDENTITY with a pseudonym identity, AT_NONCE_MT,  |
          | AT_SELECTED_VERSION)                                 |
          |----------------------------------------------------->|
          |                                                      |
          |                                                      |
          |                            +-----------------------------+
          |                            | Server fails to decode the  |
          |                            | Pseudonym in AT_IDENTITY    |
          |                            +-----------------------------+
          |                                                      |
          |                EAP-Request/SIM/Start                 |
          |                (AT_PERMANENT_ID_REQ, AT_VERSION_LIST) |
          |<-----------------------------------------------------|
          |                                                      |
          |                                                      |
          | EAP-Response/SIM/Start                               |
          | (AT_IDENTITY with permanent identity,                |
          |  AT_NONCE_MT, AT_SELECTED_VERSION)                   |
          |----------------------------------------------------->|
          |                                                      |
```

In the worst case, there are three EAP/SIM/Start round trips before the server has obtained an acceptable identity: on the first round, the client sends its re-authentication identity in AT_IDENTITY. The server fails to accept it and request a full authentication identity with a second EAP-Request/SIM/Start. The client responds with a pseudonym identity in AT_IDENTITY. The server fails to decode the pseudonym and has to issue a third EAP-Request/SIM/Start, including AT_PERMANENT_ID_REQ. Finally, the server accepts the client’s EAP-Response/SIM/Start with the AT_IDENTITY attribute and proceeds with full authentication. This is illustrated in the figure below.

```
      Client                                            Authenticator
         |                                                       |
         |                          +-------------------------------+
         |                          | Server does not have any      |
         |                          | Subscriber identity available|
         |                          | When starting EAP/SIM         |
         |                          +-------------------------------+
         |                                                       |
         |        EAP-Request/SIM/Start                          |
         |        (Includes AT_ANY_ID_REQ, AT_VERSION_LIST)      |
         |<------------------------------------------------------|
         |                                                       |
         | EAP-Response/SIM/Start                                |
         | (AT_IDENTITY with re-authentication identity)         |
         |------------------------------------------------------>|
         |                                                       |
         |                          +-------------------------------+
         |                          | Server does not accept        |
         |                          | The re-authentication         |
         |                          | Identity                      |
         |                          +-------------------------------+
         |                                                       |
         |     EAP-Request/SIM/Start                             |
         |     (AT_FULLAUTH_ID_REQ, AT_VERSION_LIST)             |
         |<------------------------------------------------------|
         |                                                       |
         |EAP-Response/SIM/Start                                 |
         |(AT_IDENTITY with a pseudonym identity, AT_NONCE_MT,   |
         | AT_SELECTED_VERSION)                                  |
         |------------------------------------------------------>|
         |                                                       |
         |                          +-------------------------------+
         |                          | Server fails to decode the    |
         |                          | Pseudonym in AT_IDENTITY      |
         |                          +-------------------------------+
         |                                                       |
         |           EAP-Request/SIM/Start                       |
         |           (AT_PERMANENT_ID_REQ, AT_VERSION_LIST)      |
         |<------------------------------------------------------|
         |                                                       |
         |                                                       |
         | EAP-Response/SIM/Start                                |
         | (AT_IDENTITY with permanent identity, AT_NONCE_MT,    |
         |  AT_SELECTED_VERSION)                                 |
         |------------------------------------------------------>|
         |                                                       |
```

After the last EAP-Response/SIM/Start message, the full authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/SIM/Challenge message.

8. Re-Authentication

In some environments, EAP authentication may be performed frequently. Because the EAP SIM full authentication procedure makes use of the GSM SIM A3/A8 algorithms, and it therefore requires 2 or 3 fresh triplets from the Authentication Centre, the full authentication procedure is not very well suitable for frequent use. Therefore, EAP SIM includes a more inexpensive re-authentication procedure that does not make use of the SIM A3/A8 algorithms and does not need new triplets from the Authentication Centre. Re-authentication can be performed in fewer roundtrips than the full authentication.

Re-authentication is optional to implement for both the EAP SIM server and client. On each EAP authentication, either one of the entities may also fall back on full authentication if they do not want to use re-authentication.

Re-authentication is based on the keys derived on the preceding full authentication. The same K_aut and K_encr keys as in full authentication are used to protect EAP SIM packets and attributes, and the original XKEY seed value from full authentication is used to generate fresh application specific keys, as specified in Section 19.

On re-authentication, the client protects against replays with an unsigned 16-bit counter, included in the AT_COUNTER attribute. On full authentication, both the server and the client initialize the counter to one. The counter value of at least one is used on the first re-authentication. On subsequent re-authentications, the counter MUST be greater than on any of the previous re-authentications. For example, on the second re-authentication, counter value is two or greater etc. The AT_COUNTER attribute is encrypted.

The server includes an encrypted server nonce (AT_NONCE_S) in the re-authentication request. The AT_MAC attribute in the client’s response is calculated over NONCE_S to provide a challenge/response authentication scheme. The NONCE_S also contributes to the new application specific keys.

As discussed in Section 7, in some environments the client may assume that the network can reliably store pseudonyms and therefore the client may fail to respond to the AT_PERMANENT_ID_REQ attribute. The network SHOULD store pseudonyms on a reliable database. Because one of the objectives of the re-authentication procedure is to reduce load on the network, the re-authentication procedure does not require the EAP server to contact a reliable database. Therefore, the re-authentication procedure makes use of separate re-authentication user identities. Pseudonyms and the permanent IMSI-based identity are reserved for full authentication only. The network does not need to store re-authentication identities as carefully as pseudonyms. If a re-authentication identity is lost and the network does not recognize it, the EAP server can fall back on full authentication.

If the EAP server supports re-authentication, it MAY include the skippable AT_NEXT_REAUTH_ID attribute in the encrypted data of EAP-Request/SIM/Challenge message (Section 13). This attribute contains a new re-authentication identity for the next re-authentication. The client MAY ignore this attribute, in which case it will use full authentication next time. If the client wants to use re-authentication, it uses this re-authentication identity on next authentication. Even if the client has a re-authentication identity, the client MAY discard the re-authentication identity and use a pseudonym or the IMSI-based permanent identity instead, in which case full authentication will be performed.

The re-authentication identity received in AT_NEXT_REAUTH_ID contains both the username portion and the realm portion of the Network Access Identifier. The EAP Server can choose an appropriate realm part in order to have the AAA infrastructure route subsequent re-authentication related requests to the same AAA server. For example, the realm part MAY include a portion that is specific to the AAA server. Hence, it is sufficient to store the context required for re-authentication in the AAA server that performed the full authentication.

The client MAY use the re-authentication identity in the EAP-Response/Identity packet or, in response to server’s AT_ANY_ID_REQ attribute, the client MAY use the re-authentication identity in the AT_IDENTITY attribute of the EAP-Response/SIM/Start packet.

Even if the client uses a re-authentication identity, the server may want to fall back on full authentication, for example because the server does not recognize the re-authentication identity or does not want to use re-authentication. In this case, the server starts the full authentication procedure by issuing an EAP-Request/SIM/Start packet. This packet always starts a full authentication sequence if it does not include the AT_ANY_ID_REQ attribute. If the server was not able to recover the client’s identity from the re-authentication identity, the server includes either the AT_FULLAUTH_ID_REQ or the AT_PERMANENT_ID_REQ attribute in this EAP request. (As specified in Sections 6 and 7, the server MAY use AT_ANY_ID_REQ, AT_FULLAUTH_ID_REQ or AT_PERMANENT_ID_REQ attributes if it does not know the client’s identity.)

Both the client and the server SHOULD have an upper limit for the number of subsequent re-authentications allowed before a full authentication needs to be performed. Because a 16-bit counter is used in re-authentication, the theoretical maximum number of re-authentications is reached when the counter value reaches 0xFFFF.

In order to use re-authentication, the client and the server need to store the following values: original XKEY, K_aut, K_encr, latest counter value and the next re-authentication identity.

The following figure illustrates the re-authentication procedure. Encrypted attributes are denoted with '*'. The client uses its re-authentication identity in the EAP-Response/Identity packet. As discussed above, an alternative way to communicate the re-authentication identity to the server is for the client to use the AT_IDENTITY attribute in the EAP-Response/SIM/Start message. This latter case is not illustrated in the figure below, and it is only possible when the server requests the client to send its identity by including the AT_ANY_ID_REQ attribute in the EAP-Request/SIM/Start packet.

If the server recognizes the re-authentication identity and agrees on using re-authentication, then the server sends the EAP-Request/SIM/Re-authentication packet to the client. This packet MUST include the encrypted AT_COUNTER attribute, with a fresh counter value, the encrypted AT_NONCE_S attribute that contains a random number chosen by the server, the AT_ENCR_DATA and the AT_IV attributes used for encryption, and the AT_MAC attribute that contains a message authentication code over the packet. The packet MAY also include an encrypted AT_NEXT_REAUTH_ID attribute that contains the next re-authentication identity.

Re-authentication identities are one-time identities. If the client does not receive a new re-authentication identity, it MUST use either the permanent identity or a pseudonym identity on the next authentication to initiate full authentication.

The client verifies that the counter value is fresh (greater than any previously used value). The client also verifies that AT_MAC is correct. The client MAY save the next re-authentication identity from the encrypted AT_NEXT_REAUTH_ID for next time. If all checks are successful, the client responds with the EAP-Response/SIM/Re-authentication packet, including the AT_COUNTER attribute with the same counter value and the AT_MAC attribute.

The server verifies the AT_MAC attribute and also verifies that the counter value is the same that it used in the EAP-Request/SIM/Re-authentication packet. If these checks are successful, the re-authentication has succeeded and the server sends the EAP-Success packet to the client.

```
   Client                                             Authenticator
          |                                                        |
          |                                 EAP-Request/Identity   |
          |<-------------------------------------------------------|
          |                                                        |
          | EAP-Response/Identity                                  |
          | (Includes a re-authentication identity)                |
          |------------------------------------------------------->|
          |                                                        |
          |                          +-------------------------------+
          |                          | Server recognizes the identity |
          |                          | and agrees on using fast       |
          |                          | re-authentication              |
          |                          +-------------------------------+
          |                                                        |
          |  EAP-Request/SIM/Re-authentication                     |
          |  (AT_IV, AT_ENCR_DATA, *AT_COUNTER,                    |
          |   *AT_NONCE_S, *AT_NEXT_REAUTH_ID, AT_MAC)             |
          |<-------------------------------------------------------|
          |                                                        |
          |                                                        |
     +-------------------------------------------------+           |
     | Client verifies AT_MAC and the freshness of   |             |
     | the counter. Client MAY store the new re-     |             |
     | authentication identity for next re-auth.     |             |
     +-------------------------------------------------+           |
          |                                                        |
          | EAP-Response/SIM/Re-authentication                     |
          | (AT_IV, AT_ENCR_DATA, *AT_COUNTER with same value,     |
          |  AT_MAC)                                               |
          |------------------------------------------------------->|
          |                                                        |
          |                          +-------------------------------+
          |                          | Server verifies AT_MAC and     |
          |                          | the counter                    |
          |                          +-------------------------------+
          |                                                        |
          |                                          EAP-Success   |
          |<-------------------------------------------------------|
          |                                                        |
```

If the client does not accept the counter value of EAP-Request/SIM/Re-authentication, it indicates the counter synchronization problem by including the encrypted AT_COUNTER_TOO_SMALL in EAP-Response/SIM/Re-authentication. The server responds with EAP-Request/SIM/Start to initiate a normal full authentication procedure. This is illustrated in the following figure. Encrypted attributes are denoted with '*'.

```
   Client                                           Authenticator
          |                                                     |
          |                            EAP-Request/Identity     |
          |<----------------------------------------------------|
          |                                                     |
          | EAP-Response/Identity                               |
          | (Includes a re-authentication identity)             |
          |---------------------------------------------------->|
          |                                                     |
          |  EAP-Request/SIM/Re-authentication                  |
          |  (AT_IV, AT_ENCR_DATA, *AT_COUNTER,                 |
          |   *AT_NONCE_S, *AT_NEXT_REAUTH_ID, AT_MAC)          |
          |<----------------------------------------------------|
          |                                                     |
     +------------------------------------------------+         |
     | AT_MAC is valid but the counter is not fresh. |          |
     +------------------------------------------------+         |
          |                                                     |
          | EAP-Response/SIM/Re-authentication                  |
          | (AT_IV, AT_ENCR_DATA, *AT_COUNTER_TOO_SMALL,        |
          |  *AT_COUNTER, AT_MAC)                               |
          |---------------------------------------------------->|
          |                                                     |
          |            +----------------------------------------------+
          |            | Server verifies AT_MAC but detects           |
          |            | That client has included AT_COUNTER_TOO_SMALL|
          |            +----------------------------------------------+
          |                                                     |
          |                        EAP-Request/SIM/Start        |
          |                        (AT_VERSION_LIST)            |
          |<----------------------------------------------------|
          |                                                     |
     +----------------------------------------------------------------+
     |                Normal full authentication follows.             |
     +----------------------------------------------------------------+
          |                                                     |
```

In the figure above, the first three messages are similar to the basic re-authentication case. When the client detects that the counter value is not fresh, it includes the AT_COUNTER_TOO_SMALL attribute in EAP-Response/SIM/Re-authentication. This attribute doesn’t contain any data but it is a request for the server to initiate full authentication. In this case, the client MUST ignore the contents of the server’s AT_NEXT_REAUTH_ID attribute.

On receipt of AT_COUNTER_TOO_SMALL, the server verifies AT_MAC and verifies that AT_COUNTER contains the same as in the EAP-Request/SIM/Re-authentication packet. If not, the server silently discards the EAP-Response/SIM/Re-authentication packet. If all checks on the packet are successful, the server transmits a new EAP-Request/SIM/Start packet and the full authentication procedure is performed as usual. Since the server already knows the subscriber identity, it MUST NOT include AT_ANY_ID_REQ, AT_FULLAUTH_ID_REQ or AT_PERMANENT_ID_REQ in the EAP-Request/SIM/Start.

9. Message Format

The Type-Data of the EAP/SIM packets begins with a 1-octet Subtype field, which is followed by a 2-octet reserved field. The rest of the Type-Data consists of attributes that are encoded in Type, Length, Value format. The figure below shows the generic format of an attribute.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |    Length     | Value...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Attribute Type

Indicates the particular type of attribute. The attribute type values are listed in Section 20.

Length

Indicates the length of this attribute in multiples of four bytes. The maximum length of an attribute is 1024 bytes. The length includes the Attribute Type and Length bytes.

Value

The particular data associated with this attribute. This field is always included and it may be two or more bytes in length. The type and length fields determine the format and length of the value field.

When an attribute numbered within the range 0 through 127 is encountered but not recognized, the EAP/SIM message containing that attribute MUST be silently discarded. These attributes are called non-skippable attributes.

When an attribute numbered in the range 128 through 255 is encountered but not recognized that particular attribute is ignored, but the rest of the attributes and message data MUST still be processed. The Length field of the attribute is used to skip the attribute value in searching for the next attribute. These attributes are called skippable attributes.

Unless otherwise specified, the order of the attributes in an EAP/SIM message is insignificant, and an EAP/SIM implementation should not assume a certain order to be used.

Attributes can be encapsulated within other attributes. In other words, the value field of an attribute type can be specified to contain other attributes.

10. Message Authentication and Encryption

This section specifies EAP/SIM attributes for attribute encryption and EAP/SIM message authentication.

Because the K_encr and K_aut keys derived from the RAND challenges (as specified in Section 19) are required to process the integrity protection and encryption attributes, these attributes can only be used in the EAP-Request/SIM/Challenge message and any EAP/SIM messages sent after EAP-Requets/SIM/Challenge. For example, these attributes cannot be used in EAP-Request/SIM/Start.

10.1. AT_MAC Attribute

The AT_MAC attribute is used for EAP/SIM message authentication. The AT_MAC attribute MUST be included in all EAP/SIM packets except those with the EAP/SIM message Subtype Start or Notification. Messages that do not meet these conditions MUST be silently discarded.

The value field of the AT_MAC attribute contains two reserved bytes followed by a message authentication code (MAC). The MAC is calculated over the whole EAP packet, concatenated with optional message-specific data, with the exception that the value field of the MAC attribute is set to zero when calculating the MAC. The reserved bytes are set to zero when sending and ignored on reception.

The contents of the message-specific data, if present, are specified separately for each EAP/SIM message. The message-specific data is included in order to protect data that is not transmitted with the EAP packet.

The format of the AT_MAC attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_MAC    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           MAC                                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The MAC algorithm is HMAC-SHA1-128 [13] keyed hash value. (The HMAC-SHA1-128 value is obtained from the 20-byte HMAC-SHA1 value by truncating the output to 16 bytes. Hence, the length of the MAC is 16 bytes.) The derivation of the authentication key (K_aut) used in the calculation of the MAC is specified in Section 19.

10.2. AT_IV, AT_ENCR_DATA and AT_PADDING Attributes

AT_IV and AT_ENCR_DATA attributes can be optionally used to transmit encrypted information between the EAP/SIM client and server.

The value field of AT_IV contains two reserved bytes followed by a 16-byte initialization vector required by the AT_ENCR_DATA attribute. The reserved bytes are set to zero when sending and ignored on reception. The AT_IV attribute MUST be included if and only if the AT_ENCR_DATA is included. Messages that do not meet this condition MUST be silently discarded.

The sender of the AT_IV attribute chooses the initialization vector by random. The sender MUST NOT reuse the initialization vector value from previous EAP SIM packets but the sender MUST choose it freshly for each AT_IV attribute. The sends SHOULD use a good source of randomness to generate the initialization vector. The format of AT_IV is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_IV     | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The value field of the AT_ENCR_DATA attribute consists of two reserved bytes followed by bytes encrypted using the Advanced Encryption Standard (AES) [7] in the Cipher Block Chaining (CBC) mode of operation, using the initialization vector from the AT_IV attribute. The reserved bytes are set to zero when sending and ignored on reception. Please see [8] for a description of the CBC mode. The format of the AT_ENCR_DATA attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Encrypted Data                             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The derivation of the encryption key (K_encr) is specified in Section 19.

The plaintext consists of nested EAP/SIM attributes.

The encryption algorithm requires the length of the plaintext to be a multiple of 16 bytes. The sender may need to include the AT_PADDING attribute as the last attribute within AT_ENCR_DATA. The AT_PADDING attribute is not included if the total length of other nested attributes within the AT_ENCR_DATA attribute is a multiple of 16 bytes. As usual, the Length of the Padding attribute includes the Attribute Type and Attribute Length fields. The Length of the Padding attribute is 4, 8 or 12 bytes. It is chosen so that the length of the value field of the AT_ENCR_DATA attribute becomes a multiple of 16 bytes. The actual pad bytes in the value field are set to zero (0x00) on sending. The recipient of the message MUST verify that the pad bytes are set to zero, and silently drop the message if this verification fails. The format of the AT_PADDING attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

11. EAP-Request/SIM/Start

The first SIM specific EAP Request is of subtype Start. The EAP/SIM/Start roundtrip is used for two purposes. On full authentication, the this packet is used to request the client to send the AT_NONCE_MT attribute to the server. In addition, as specified in Section 6, the Start round trip may be used for obtaining the client identity to the server. The format of the EAP Request/SIM/Start packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_PERM..._REQ | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_FULL..._REQ | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_ANY_ID_REQ  | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_VERSION_L..| Length        | Actual Version List Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Supported Version 1          |  Supported Version 2          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Supported Version N           |     Padding                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1 for Request

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

10

Reserved

Set to zero on sending, ignored on reception

AT_PERMANENT_ID_REQ

The AT_PERMANENT_ID_REQ attribute is optional to include and it is included in the cases defined in Section 7. It MUST NOT be included if AT_ANY_ID_REQ or AT_FULLAUTH_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

AT_FULLAUTH_ID_REQ

The AT_FULLAUTH_ID_REQ attribute is optional to include and it is included in the cases defined in Section 7. It MUST NOT be included if AT_ANY_ID_REQ or AT_PERMANENT_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

AT_ANY_ID_REQ

The AT_ANY_ID_REQ attribute is optional and it is included in the cases defined in Section 6. It MUST NOT be included if AT_PERMANENT_ID_REQ or AT_FULLAUTH_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

AT_VERSION_LIST

The AT_VERSION_LIST attribute MUST be included. This attribute is used in version negotiation, as specified in Section 4. The value field of this attribute begins with 2-byte Actual Version List Length, which specifies the length of the Version List in bytes. This field is followed by the list of supported version, each 2 bytes. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the value field with zero bytes when necessary.

12. EAP-Response/SIM/Start

The format of the EAP Response/SIM/Start packet is shown below.

```
   0                   1                   2                   3
   0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |     Code      |  Identifier   |            Length             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |     Type      |    Subtype    |           Reserved            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |AT_NONCE_MT    | Length = 5    |           Reserved            |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  |                           NONCE_MT                            |
  |                                                               |
  |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | AT_IDENTITY   | Length        | Actual Identity Length        |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |                                                               |
  .                       Identity (optional)                     .
  .                                                               .
  |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  | AT_SELECTED...| Length = 1    |    Selected Version           |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2 for Response

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

10

Reserved

Set to zero when sending, ignored on reception.

AT_NONCE_MT

The AT_NONCE_MT attribute MUST NOT be included on re-authentication, that is, if the AT_IDENTITY with a re-authentication identity is included. AT_NONCE_MT MUST be included in all other cases (full authentication). The value field contains two reserved bytes followed by a random number generated by the client (16 bytes) freshly for this EAP/SIM authentication. The random number is used as a seed value for the new keying material. The reserved bytes are set to zero upon sending and ignored upon reception.

AT_IDENTITY

The AT_IDENTITY attribute is optional to include and it is included in cases defined in Section 6 and 7. The value field of this attribute begins with 2-byte actual identity length, which specifies the length of the identity in bytes. This field is followed by the subscriber identity of the indicated actual length, in the same Network Access Identifier format that is used in EAP-Response/Identity, i.e. including the NAI realm portion. The identity is the permanent IMSI-based identity, a pseudonym identity or a re-authentication identity. The identity format is specified in Section 5. The identity does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the identity with zero bytes when necessary.

AT_SELECTED_VERSION

The AT_SELECTED_VERSION attribute MUST NOT be included on re-authentication, that is, if the AT_IDENTITY attribute with a re-authentication identity is included. In all other cases, AT_SELECTED_VERSION MUST be included (full authentication). This attribute is used in version negotiation, as specified in Section 4. The value field of this attribute contains a two-byte version number, which indicates the EAP/SIM version that the client wants to use.

13. EAP-Request/SIM/Challenge

The format of the EAP-Request/SIM/Challenge packet is shown below.

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     Code      |  Identifier   |            Length             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |     Type      |    Subtype    |           Reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | AT_RAND       | Length        |           Reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     .                            n*RAND                             .
     .                                                               .
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | AT_IV         | Length = 5    |           Reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                Initialization Vector (optional)               |
     |                                                               |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | AT_ENCR_DATA  | Length        |           Reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     .                   Encrypted Data (optional)                   .
     .                                                               .
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | AT_MAC        | Length = 5    |           Reserved            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                                                               |
     |                           MAC                                 |
     |                                                               |
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1 for Request

Identifier

See [1]

Length

The length of the EAP packet.

Type

18

Subtype

11

Reserved

Set to zero when sending, ignored on reception.

AT_RAND

The AT_RAND attribute MUST be included. The value field of this attribute contains two reserved bytes followed by n GSM RANDs (each 16 bytes long). The reserved bytes are set to zero upon sending and ignored upon reception.

The number of RAND challenges MUST be two or three. The client MAY silently ignore the EAP-Request/SIM/Challenge message, if the number of RAND challenges is two while the client’s local policy requires three challenges to be used.

AT_IV

The AT_IV attribute is optional to include. See section 10.2.

AT_ENCR_DATA

The AT_ENCR_DATA attribute is optional to include. See section 10.2. The plaintext consists of nested attributes as described below.

AT_MAC

AT_MAC MUST be included. For EAP-Request/SIM/Challenge, the MAC code is calculated over the following data:

EAP packet| NONCE_MT

The EAP packet is represented as specified in Section 10.1. It is followed by the 16-byte NONCE_MT value from the client’s AT_NONCE_MT attribute.

The AT_IV, AT_ENCR_DATA and AT_MAC attributes are used for identity privacy and for communicating the next re-authentication identity. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | AT_NEXT_PSEU..| Length        | Actual Pseudonym Length       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   .                           Next Pseudonym                      .
   .                                                               .
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | AT_NEXT_REAU..| Length        | Actual Re-Auth Identity Length|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   .                    Next Re-authentication Username            .
   .                                                               .
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  AT_PADDING   | Length        | Padding...                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_NEXT_PSEUDONYM

The AT_NEXT_PSEUDONYM attribute is optional to include. The value field of this attribute begins with 2-byte actual pseudonym length, which specifies the length of the pseudonym in bytes. This field is followed by a pseudonym username, of the indicated actual length, that the client can use in the next authentication, as described in Section 7. The username does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the pseudonym with zero bytes when necessary.

AT_NEXT_REAUTH_ID

The AT_NEXT_REAUTH_ID attribute is optional to include. The value field of this attribute begins with 2-byte actual re-authentication identity length, which specifies the length of the re-authentication identity in bytes. This field is followed by a re-authentication identity, of the indicated actual length, that the client can use in the next re-authentication, as described in Section 8. The re-authentication identity includes both a username portion and a realm name portion. The re-authentication identity does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the re-authentication identity with zero bytes when necessary.

AT_PADDING

The AT_PADDING attribute is optional. See section 10.2

14. EAP-Response/SIM/Challenge

The format of the EAP-Response/SIM/Challenge packet is shown below.

Later versions of this protocol MAY make use of the AT_ENCR_DATA and AT_IV attributes in this message to include encrypted (skippable) attributes. AT_ENCR_DATA and AT_IV attributes are not shown in the figure below. If present, they are processed as in EAP-Request/SIM/Challenge packet. The EAP server MUST process EAP-Response/SIM/Challenge messages that include these attributes even if the server did not implement these optional attributes.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_MAC       | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                           MAC                                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2 for Response

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

11

Reserved

Set to zero when sending, ignored on reception.

AT_MAC

AT_MAC MUST be included. For EAP-Response/SIM/Challenge, the MAC code is calculated over the following data:
    EAP packet| n*SRES
The EAP packet is represented as specified in Section 10.1. The EAP packet bytes are immediately followed by the two or three SRES values concatenated, denoted above with the notation n*SRES. The SRES values are used in the same order as the corresponding RAND challenges in AT_RAND attribute.

15. EAP-Request/SIM/Re-authentication

The format of the EAP-Request/SIM/Re-authentication packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_IV         | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Encrypted Data                             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_MAC       | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                             MAC                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1 for Request

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

13

Reserved

Set to zero when sending, ignored on reception.

AT_IV

The AT_IV attribute is MUST be included. See section 10.2.

AT_ENCR_DATA

The AT_ENCR_DATA attribute MUST be included. See section 10.2. The plaintext consists of nested attributes as described below.

AT_MAC

AT_MAC MUST be included. No message-specific data is included in the MAC calculation. See Section 10.1.

The AT_IV and AT_ENCR_DATA attributes are used for communicating encrypted attributes. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  AT_COUNTER   | Length = 1    |           Counter             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | AT_NONCE_S    | Length = 5    |           Reserved            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                                                               |
   |                            NONCE_S                            |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | AT_NEXT_REAU..| Length        | Actual Re-Auth Identity Length|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   .                   Next Re-authentication Username             .
   .                                                               .
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  AT_PADDING   | Length        | Padding...                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
   |                                                               |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_COUNTER

The AT_COUNTER attribute MUST be included. The value field consists of a 16-bit unsigned integer counter value, represented in network byte order.

AT_NONCE_S

The AT_NONCE_S attribute MUST be included. The value field contains two reserved bytes followed by a random number generated by the server (16 bytes) freshly for this EAP/SIM re-authentication. The random number is used as challenge for the client and also a seed value for the new keying material. The reserved bytes are set to zero upon sending and ignored upon reception.

AT_NEXT_REAUTH_ID

The AT_NEXT_REAUTH_ID attribute is optional to include. The attribute is described in Section 13.

AT_PADDING

The AT_PADDING attribute is optional to include. See section 10.2

16. EAP-Response/SIM/Re-authentication

The format of the EAP-Response/SIM/Re-authentication packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_IV         | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Encrypted Data                             .
.                                                               .
|                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_MAC       | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                             MAC                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2 for Response

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

13

Reserved

Set to zero when sending, ignored on reception.

AT_IV

The AT_IV attribute is MUST be included. See section 10.2.

AT_ENCR_DATA

The AT_ENCR_DATA attribute MUST be included. See section 10.2. The plaintext consists of nested attributes as described below.

AT_MAC

For EAP-Response/SIM/Re-authentication, the MAC code is calculated over the following data:

EAP packet| NONCE_S

The EAP packet is represented as specified in Section 10.1. It is followed by the 16-byte NONCE_S value from the client's AT_NONCE_S attribute.

The AT_IV and AT_ENCR_DATA attributes are used for communicating encrypted attributes. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_COUNTER   | Length = 1    |           Counter             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_COUNTER...| Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_COUNTER

The AT_COUNTER attribute MUST be included. The format of this attribute is specified in Section 15.

AT_COUNTER_TOO_SMALL

The AT_COUNTER_TOO_SMALL attribute is optional to include, and it is included in cases specified in Section 8.

AT_PADDING

The AT_PADDING attribute is optional to include. See section 10.2

17. Unsuccessful Cases

In general, if an EAP/SIM client or server implementation detects an error in a received EAP/SIM packet, the EAP/SIM implementation silently ignores the EAP packet, does not change its state and does not send any EAP messages to its peer. Examples of such errors, specified in detail elsewhere in this document, are an invalid AT_MAC value, insufficient number of RAND challenges included in AT_RAND, no acceptable version included in AT_VERSION_LIST, a mandatory attribute is missing, illegal attributes included and an unrecognized non-skippable attribute.

The rationale for this error case behavior is that an active attacker may have sent the erroneous packet. As the EAP/SIM client or server does not process the packet and does not change its state, it is possible to successfully process a valid packet if such packet is received later. If no valid packets are received, the authentication exchange will eventually time out.

As normally in EAP, the EAP server sends the EAP-Failure packet to the client when the authentication procedure fails on the EAP Server. In EAP/SIM, this may occur for example if the EAP server is not able to obtain the GSM triplets for the subscriber or the authentication exchange times out.

18. EAP/SIM Notifications

The EAP-Request/Notification, specified in [1], can be used to convey a displayable message from the authenticator to the client. Because these messages are textual messages, it may be hard for the client to present them in the user’s preferred language. Therefore, EAP/SIM uses a separate EAP/SIM message subtype to transmit localizable notification codes instead of the EAP-Request/Notification packet.

The EAP server MAY issue an EAP-Request/SIM/Notification packet to the client. The client MAY delay the processing of EAP-Request/SIM/Notification and wait for other EAP/SIM requests. If a valid EAP/SIM request of another subtype is received, the client MAY silently ignore the EAP-Request/SIM notification and process the other EAP/SIM request instead. If the client decides to process the EAP-Request/SIM/Notification, then the client MAY show a notification message to the user and the client MUST respond to the EAP server with an EAP-Response/SIM/Notification packet.

Some of the notification codes are authorization related and hence not usually considered as part of the responsibility of an EAP method. However, they are included as part of EAP/SIM because there are currently no other ways to convey this information to the user in a localizable way, and the information is potentially useful for the user. An EAP/SIM server implementation may decide never to send these EAP/SIM notifications.

The format of the EAP-Request/SIM/Notification packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_NOTIFICATION| Length = 1    |      Notification Code        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

1 for Request

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

12

Reserved

Set to zero when sending, ignored on reception.

AT_NOTIFICATION

The AT_NOTIFICATION attribute MUST be included. The value field of this attribute contains a two-byte notification code. The following code values have been reserved. The descriptions below illustrate the semantics of the notifications. The client implementation MAY use different wordings when presenting the notifications to the user. The "requested service" depends on the environment where EAP/SIM is applied.

1024 - Visited network does not have a roaming agreement with user’s home operator or a suitable roaming broker

1026 – User has been temporarily denied access to the requested service

1031 - User has not subscribed to the requested service

The format of the EAP-Response/SIM/Notification packet is shown below. Because this packet is only an acknowledgement of EAP-Request/SIM/Notification, it does not contain any mandatory attributes.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2 for Response

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

12

Reserved

Set to zero when sending, ignored on reception.

19. Key Generation

This section specifies how keying material is generated.

EAP SIM requires two keys for its own purposes, the authentication key K_aut to be used with the AT_MAC attribute, and the encryption key K_encr, to be used with the AT_ENCR_DATA attribute. The same K_aut and K_encr keys are used in full authentication and subsequent re-authentications. In addition, it is possible to derive additional application specific key material, such as a master key to be used with IEEE 802.11i. New application specific keys are derived on each re-authentication.

Key derivation is based on the random number generation specified in NIST Federal Information Processing Standards (FIPS) Publication 186-2 [11]. The pseudo-random number generator is specified in the change notice 1 (2001 October 5) of [11] (Algorithm 1). As specified in the change notice (page 74), when Algorithm 1 is used as a general-purpose pseudo-random number generator, the "mod q" term in step 3.3 is omitted. The function G used in the algorithm is constructed via Secure Hash Standard as specified in Appendix 3.3 of the standard. For convenience, the random number algorithm with the correct modification is cited in Annex C.

160-bit XKEY and XVAL values are used, so b = 160. On each full authentication, the initial secret seed-key XKEY is computed from the n GSM Kc keys and the NONCE_MT with the following formula:

```
XKEY = SHA1(Identity|n*Kc| NONCE_MT| Version List| Selected
Version)
```

In the formula above, the "|" character denotes concatenation. Identity denotes the user identity string without any terminating null characters. It is the identity from the AT_IDENTITY attribute from the last EAP-Response/SIM/Start packet, or, if AT_IDENTITY was not used, the identity from the EAP-Response/Identity packet. The notation n*Kc denotes the n Kc values concatenated. The Kc keys are used in the same order as the RAND challenges in AT_RAND attribute. NONCE_MT denotes the NONCE_MT value (not the AT_NONCE_MT attribute but just the nonce value). The Version List includes the 2-byte supported version numbers from AT_VERSION_LIST, in the same order as in the attribute. The Selected Version is the 2-byte selected version from AT_SELECTED_VERSION. Network byte order is used, just as in the attributes. The hash function SHA1 is specified in [12].

The optional user input values (XSEED_j) in step 3.1 are set to zero.

The resulting 320-bit random numbers x_0, x_1, ..., x_m-1 are concatenated and partitioned into suitable-sized chunks and used as keys in the following order: K_encr (128 bits), K_aut (128 bits), EAP application specific keys. The number of pseudo-random number generator iterations (m) depends on the amount of required keying material.

On re-authentication, the same pseudo-random number generator can be used to generate new application specific keys. The seed value XKEY’ is calculated as follows:

XKEY’ = SHA1(Identity|counter|NONCE_S|original XKEY)

In the formula above, the Identity denotes the re-authentication user identity, without any terminating null characters, from the AT_IDENTITY attribute of the EAP-Response/SIM/Start packet, or, if EAP-Response/SIM/Start was not used on re-authentication, the identity string from the EAP-Response/Identity packet. The counter denotes the counter value from AT_COUNTER attribute used in the EAP-Response/SIM/Re-authentication packet. The counter is used in network byte order. NONCE_S denotes the 16-byte NONCE_S value from the AT_NONCE_S attribute used in the EAP-Request/SIM/Re-authentication packet. The original XKEY is the XKEY value from the preceding full authentication. The pseudo-random number generator is run with the new seed value XKEY’, and the resulting 320-bit random numbers x_0, x_1, ..., x_m-1 are concatenated and partitioned into suitable-sized chunks and used as new application specific keys.

For example, the EAP application specific material can be used for packet security between the client and the authenticator. Because the required keying material depends on the EAP application and the EAP key derivation standardization has not been finalized yet, general rules of key derivation cannot be given here. However, please see Annex B for a specification of how keys for IEEE 802.11 are derived.

When generating the initial seed value XKEY, the hash function is used as a mixing function to combine several session keys (Kc's) generated by the GSM authentication procedure and the random number NONCE_MT into a single session key. There are several reasons for this. The current GSM session keys are at most 64 bits, so two or more of them are needed to generate a longer key. By using a one-way function to combine the keys, we are assured that even if an attacker managed to learn one of the EAP/SIM session keys, it wouldn’t help him in learning the original GSM Kc's. In addition, since we include the random number NONCE_MT in the calculation, the client is able to verify that the SIM authentication values it receives from the network are fresh and not a replay. (Please see also Section 21.)

20. IANA Considerations

The realm name "owlan.org" has been reserved for NAI realm names generated from the IMSI.

IANA has assigned the EAP type number 18 for this protocol.

EAP/SIM messages include a Subtype field. The following Subtypes are specified:

```
Start.........................................10
Challenge.....................................11
Notification..................................12
Re-authentication.............................13
```

The Subtype-specific data is composed of attributes, which have attribute type numbers. The following attribute types are specified:

```
AT_RAND........................................1
AT_PADDING.....................................6
AT_NONCE_MT....................................7
AT_PERMANENT_ID_REQ...........................10
AT_MAC........................................11
AT_NOTIFICATION...............................12
AT_ANY_ID_REQ.................................13
AT_IDENTITY...................................14
AT_VERSION_LIST...............................15
AT_SELECTED_VERSION...........................16
AT_FULLAUTH_ID_REQ............................17
AT_COUNTER....................................19
AT_COUNTER_TOO_SMALL..........................20
AT_NONCE_S....................................21

AT_IV........................................129
AT_ENCR_DATA.................................130
AT_NEXT_PSEUDONYM............................132
AT_NEXT_REAUTH_ID............................133
```

The AT_NOTIFICATION attribute contains a notification code value. Values 1024, 1026 and 1031 have been specified in Section 18 of this document.

The AT_VERSION_LIST and AT_SELECTED_VERSION attributes contain version numbers. Version 1 has been specified in Section 4 of this document.

All requests for value assignment from the various number spaces described in this document require proper documentation, according to the "Specification Required" policy described in [14]. Requests must be specified in sufficient detail so that interoperability between independent implementations is possible. Possible forms of documentation include, but are not limited to, RFCs, the products of another standards body (e.g. 3GPP), or permanently and readily available vendor design notes.

21. Security Considerations

The protocol in this document is intended to provide the appropriate level of security to operate Extensible Authentication Protocol using the GSM SIM application.

EAP/SIM includes optional IMSI privacy support that protects the privacy of the subscriber identity against passive eavesdropping. The mechanism cannot be used on the first connection with a given server, when the IMSI will have to be sent in the clear. The terminal SHOULD store the pseudonym in a non-volatile memory so that it can be maintained across reboots. An active attacker that impersonates the network may use the AT_IMSI_REQ attribute (Section 7) to learn the subscriber's IMSI. However, as discussed in Section 7, the terminal can refuse to send the cleartext IMSI if it believes that the network should be able to recognize the pseudonym. This is the same level of protection as in the GSM and UMTS cellular networks.

In EAP/SIM, the client believes that the network is authentic because the network can calculate a correct AT_MAC value in the EAP-Request/SIM/Challenge packet. To calculate AT_MAC, it is sufficient to know the complete GSM triplets (RAND, SRES, Kc) used in the authentication. Because the network selects the RAND challenges and hereby the triplets, an attacker that knows two or three GSM triplets for the subscriber is able to impersonate a valid network to the client. Given physical access to the SIM card, it is easy to obtain any number of GSM triplets. Another way to obtain a RAND challenge and the corresponding SRES response of a GSM triplet is to eavesdrop on the GSM network. (To obtain the Kc key from the GSM network, the attacker needs to mount a brute force attach on encrypted data to find the Kc key by exhaustive search.) Yet another way to obtain triplets is to mount an attack on the client platform via a virus or other malicious piece of software. The client SHOULD be protected against triplet querying attacks by malicious software.

EAP/SIM combines several GSM triplets in order to generate stronger session keys and stronger AT_MAC values. The actual strength of the resulting key depends, among other things, on the operator-specific authentication algorithms, the strength of the Ki key, and the quality of the RAND challenges, which is also operator specific. For example, some SIM cards generate Kc keys with 10 bits set to zero. Such restrictions may prevent the concatenation technique from yielding strong session keys. Because the strength of the Ki key is 128 bits, the ultimate strength of any derived secret key material is never more than 128 bits.

There is no known way to obtain complete GSM triplets by mounting an attack against EAP/SIM. A passive eavesdropper can learn n*RAND and AT_MAC and may be able to link this information to the subscriber identity. An active attacker that impersonates a GSM subscriber can easily obtain n*RAND and AT_MAC values from the EAP server for any given subscriber identity. However, calculating the Kc and SRES values from AT_MAC would require the attacker to reverse the keyed message authentication code function HMAC-SHA1-128.

As EAP SIM does not expose any values calculated from an individual GSM Kc keys, it is not possible to mount a brute force attack on just one of the Kc keys in EAP SIM. Therefore, when considering brute force attacks on the values exposed in EAP SIM, the effective length of EAP SIM session keys is not compromised by the fact that they are combined from several shorter keys, i.e the effective length of 128 bits may be achieved.

However, EAP SIM cannot prevent attacks over the GSM or GPRS radio networks. If the same SIM card is also used in GSM or GPRS, it is possible to mount attacks over the cellular interface. With a rogue GSM base station, an attacker can send the RAND challenges used in EAP SIM to the terminal and then mount a brute force attack to cryptanalyze the GSM or GPRS data that is encrypted with the 64-bit Kc keys. This makes it possible to attack each 64-bit key separately. In other words, by mounting attacks over GSM, the effective length of EAP SIM session keys can be reduced basically to the same level as in GSM. Because this attack requires the attacker to build a rogue GSM base station, the cost of the attack is not negligible – it is the same cost as usually in GSM.An EAP/SIM implementation SHOULD use a good source of randomness to generate the random numbers required in the protocol. Please see [15] for more information on generating random numbers for security applications.

22. Intellectual Property Right Notice

On IPR related issues, Nokia refers to the Nokia Statement on Patent licensing, see http://www.ietf.org/ietf/IPR/NOKIA.

23. Acknowledgements and Contributions

The editors thank Juha Ala-Laurila, N. Asokan, Simon Blake-Wilson, Jan-Erik Ekberg, Augustin Farrugia, Patrik Flykt, Mark Grayson, Max de Groot, Jukka-Pekka Honkanen, Antti Kuikka, Jukka Latva, Lassi Lehtinen, Valtteri Niemi, Kaisa Nyberg, Jyri Rinnemaa, Timo Takamäki and Raimo Vuonnala for their contributions and critiques.

Thanks to Greg Rose of Qualcomm for his most valuable comments [16].

The IMSI privacy support is based on the identity privacy support of [6]. The attribute format is based on the extension format of Mobile IPv4 [17].

This protocol has been partly developed in parallel with EAP AKA [18], and hence this specification incorporates many ideas from Jari Arkko.

References

[1] L. Blunk, J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998. (NORMATIVE)

[2] GSM Technical Specification GSM 03.20 (ETS 300 534): "Digital cellular telecommunication system (Phase 2); Security related network functions", European Telecommunications Standards Institute, August 1997. (NORMATIVE)

[3] S. Bradner, "Key words for use in RFCs to indicate Requirement Levels", RFC 2119, March 1997. (NORMATIVE)

[4] GSM Technical Specification GSM 03.03 (ETS 300 523): "Digital cellular telecommunication system (Phase 2); Numbering, addressing and identification", European Telecommunications Standards Institute, April 1997. (NORMATIVE)

[5] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999. (NORMATIVE)

[6] J. Carlson, B. Aboba, H. Haverinen, "EAP SRP-SHA1 Authentication Protocol", draft-ietf-pppext-eap-srp-03.txt, July 2001 (work-in-progress). (INFORMATIVE)

[7] Federal Information Processing Standard (FIPS) draft standard, "Advanced Encryption Standard (AES)", http://csrc.nist.gov/publications/drafts/dfips-AES.pdf, September 2001. (NORMATIVE)

[8] US National Bureau of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, December 1980. (NORMATIVE)

[9] GSM Technical Specification GSM 03.03 (ETS 300 523): "Digital cellular telecommunication system (Phase 2); Numbering, addressing and identification", European Telecommunications Standards Institute, April 1997. (NORMATIVE)

[10] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999. (NORMATIVE)

[11] Federal Information Processing Standards (FIPS) Publication 186-2 (with change notice), "Digital Signature Standard (DSS)", National Institute of Standards and Technology, January 27, 2000. (NORMATIVE)
Available on-line at:
http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

[12] Federal Information Processing Standard (FIPS) Publication 180-1, "Secure Hash Standard," National Institute of Standards and Technology, U.S. Department of Commerce, April 17, 1995. (NORMATIVE)

[13] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997. (NORMATIVE)

[14] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998. (NORMATIVE)

[15] D. Eastlake, 3rd, S. Crocker, J. Schiller, "Randomness Recommendations for Security", RFC 1750 (Informational), December 1994. (INFORMATIVE)

[16] Qualcomm, "Comments on draft EAP/SIM", 3rd Generation Partnership Project document 3GPP TSG SA WG3 Security — S3#22, S3-020125, February 2002. (INFORMATIVE)

[17] C. Perkins (editor), "IP Mobility Support", RFC 2002, October 1996. (INFORMATIVE)

[18] J. Arkko, H. Haverinen, "EAP AKA Authentication", draft-arkko-pppext-eap-aka-04.txt, June 2002 (work in progress). (INFORMATIVE)

Editors' Addresses

Henry Haverinen
Nokia Mobile Phones
P.O. Box 88
FIN-33721 Tampere
Finland
E-mail: henry.haverinen@nokia.com
Phone: +358 50 594 4899

Joseph Salowey
Cisco Systems
2901 Third Avenue
Seattle, WA 98121
US
E-mail: jsalowey@cisco.com
Phone: +1 206 256 3380

Annex A. Test Vectors

Test vectors for the NIST FIPS 186-2 pseudo-random number generator [11] are available at the following URL: http://csrc.nist.gov/encryption/dss/Examples-1024bit.pdf

TBD: Test vectors for EAP SIM values

Annex B. Key Derivation for IEEE 802.11

As specified in Section 19, application specific keying material can be derived with the pseudo-random function.

The key hierarchy in IEEE 802.11i currently assumes that EAP methods produce a 256-bit Pairwise Master Key (PMK). When a Pairwise Master Key is required, it is the first EAP application specific key that is derived. On full authentication, the PMK immediately follows K_aut in the key stream resulting from the key expansion scheme. On re-authentication, the PMK is the first new application specific key that is derived.

For pre 802.11i networks, the signature key used to authenticate broadcast keys [802.1x] is selected as the first 256 bits of the EAP application specific keys immediately after K_aut. (On re-authentication, the first 256 application specific key bits are used as the signature key.) The next 256 bits are used as the WEP session key. The full 256-bit key is not usually used during WEP encryption, unused bits at then end should be ignored by the implementation. When the keys are transmitted from the authenticator to the access point using the RADIUS protocol the session key is placed in an MS-MPPE-RECV-KEY attribute and the signature key is placed in an MS-MPPE-SEND-KEY attribute. These attributes are defined in RFC 2548.

Annex C. Pseudo-Random Number Generator

The "|" character denotes concatenation, and "^" denotes involution.

Step 1: Choose a new, secret value for the seed-key, XKEY

Step 2: In hexadecimal notation let

```
    t = 67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0
    This is the initial value for H0|H1|H2|H3|H4
    in the FIPS SHS [12]
```

Step 3: For j = 0 to m – 1 do

```
      3.1 XSEED_j = optional user input
      3.2 For i = 0 to 1 do
          a. XVAL = (XKEY + XSEED_j) mod 2^b
          b. w_i = G(t, XVAL)
          c. XKEY = (1 + XKEY + w_i) mod 2^b
      3.3 x_j = w_0|w_1
```

Annex B EAP AKA

J. Arkko
Ericsson
H. Haverinen
Nokia

Internet Draft
Document: draft-arkko-pppext-eap-aka-05.txt
Expires: March 2003
October 2002

EAP AKA Authentication

Status of this Memo

Abstract

This document specifies an Extensible Authentication Protocol (EAP) mechanism for authentication and session key distribution using the UMTS AKA authentication mechanism. AKA is based on symmetric keys, and runs typically in a UMTS Subscriber Identity Module, a smart card like device. AKA provides also backward compatibility to GSM authentication, making it possible to use EAP AKA for authenticating both GSM and UMTS subscribers.

EAP AKA includes optional identity privacy support and an optional re-authentication procedure.

Table of Contents

1. Introduction and Motivation

This document specifies an Extensible Authentication Protocol (EAP) mechanism for authentication and session key distribution using the UMTS AKA authentication mechanism [1]. The Universal Mobile Telecommunications System (UMTS) is a global third generation mobile network standard.

AKA is based on challenge-response mechanisms and symmetric cryptography. AKA typically runs in a UMTS Subscriber Identity Module (USIM), a smart card like device. However, the applicability of AKA is not limited to client devices with smart cards, but the AKA mechanisms could also be implemented in host software, for example. AKA also provides backward compatibility to the GSM authentication mechanism [2]. Compared to the GSM mechanism, AKA provides substantially longer key lengths and the authentication of the server side as well as the client side.

The introduction of AKA inside EAP allows several new applications. These include the following:

- The use of the AKA also as a secure PPP authentication method in devices that already contain an USIM.

- The use of the third generation mobile network authentication infrastructure in the context of wireless LANs and IEEE 801.1x technology through EAP over Wireless [3, 4].

- Relying on AKA and the existing infrastructure in a seamless way with any other technology that can use EAP.

AKA works in the following manner:

- The USIM and the home environment have agreed on a secret key beforehand.

- The actual authentication process starts by having the home environment produce an authentication vector, based on the secret key and a sequence number. The authentication vector contains a random part RAND, an authenticator part AUTN used for authenticating the network to the USIM, an expected result part XRES, a session key for integrity check IK, and a session key for encryption CK.

- The RAND and the AUTN are delivered to the USIM.

- The USIM verifies the AUTN, again based on the secret key and the sequence number. If this process is successful (the AUTN is valid and the sequence number used to generate AUTN is within the correct range), the USIM produces an authentication result, RES and sends this to the home environment.

- The home environment verifies the correct result from the USIM. If the result is correct, IK and CK can be used to protect further communications between the USIM and the home environment.

When verifying AUTN, the USIM may detect that the sequence number the network uses is not within the correct range. In this case, the USIM calculates a sequence number synchronization parameter AUTS and sends it to the network. AKA authentication may then be retried with a new authentication vector generated using the synchronized sequence number.

For a specification of the AKA mechanisms and how the cryptographic values AUTN, RES, IK, CK and AUTS are calculated, see reference [1].

It is also possible that the home environment delegates the actual authentication task to an intermediate node. In this case the authentication vector or parts of it are delivered to the intermediate node, enabling it to perform the comparison between RES and XRES, and possibly also use CK and IK. Such delivery MUST be done in a secure manner. In EAP AKA, the EAP server node is such an intermediate node.

In the third generation mobile networks, AKA is used both for radio network authentication and IP multimedia service authentication purposes. Different user identities and formats are used for these; the radio network uses the International Mobile Subscriber Identifier (IMSI), whereas the IP multimedia service uses the Network Access Identifier (NAI) [5].

2. Conventions used in this document

The following terms will be used through this document:

AAA protocol

Authentication, Authorization and Accounting protocol

AAA server

The AAA server is responsible for storing shared secrets and other credential information necessary for the authentication of users. Cf. EAP server

AKA

Authentication and Key Agreement

AuC

Authentication Centre. The mobile network element that can authenticate subscribers either in GSM or in UMTS networks.

Authenticator

The entity that terminates the protocol carrying EAP used by the client, such as a Network Access Server (NAS) terminating the PPP link. The EAP server may be co-located in the Authenticator. In this case, the Authenticator may actually authenticate the user based on information received from the AAA server.

EAP

Extensible Authentication Protocol [6].

EAP server

The network element that terminates the EAP protocol. Typically, the EAP server functionality is implemented in a AAA server.

GSM

Global System for Mobile communications.

NAI

Network Access Identifier [5].

AUTN

Authentication value generated by the AuC which together with the RAND authenticates the server to the client, 128 bits [1].

AUTS

A value generated by the client upon experiencing a synchronization failure, 112 bits.

RAND

Random number generated by the AuC, 128 bits [1].

RES

Authentication result from the client, which together with the RAND authenticates the client to the server, 128 bits [1].

SQN

Sequence number used in the authentication process, 48 bits [1].

SIM

Subscriber Identity Module. The SIM is an application traditionally resident on smart cards distributed by GSM operators.SRES

The authentication result parameter in GSM, corresponds to the RES parameter in UMTS aka, 32 bits.

USIM

UMTS Subscriber Identity Module. USIM is an application that is resident e.g. on smart cards distributed by UMTS operators.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [7]

3. Protocol Overview

In this document, the term EAP Server refers to the network element that terminates the EAP protocol. Usually the EAP server is separate from the authenticator device, which is the network element closest to the client, such as a Network Access Server (NAS) or an IEEE 802.1X bridge. Alternatively, the EAP server functionality may be co-located in the authenticator although typically, the the EAP server functionality is implemented on a separate AAA server with whom the authenticator communicates using an AAA protocol. (The exact AAA communications are outside the scope of this document, however.)

The below message flow shows the basic successful full authentication case with the EAP AKA. The EAP AKA uses two roundtrips to authorize the user and generate session keys. As in other EAP schemes, first an identity request/response message pair is exchanged. (As specified in [6], the initial identity request is not required, and MAY be bypassed in cases where the authenticator can presume the identity, such as when using leased lines, dedicated dial-ups, etc. Please see also Section 5 for specification how to obtain the identity via EAP AKA messages.)

Next, the EAP server starts the actual AKA protocol by sending an EAP-Request/AKA-Challenge message. EAP AKA packets encapsulate parameters in attributes, encoded in a Type, Length, Value format. The packet format and the use of attributes are specified in Section 8. The EAP-Request/AKA-Challenge message contains a random number (AT_RAND) and an authorization vector (AT_AUTN), and a message authentication code AT_MAC. The EAP-Request/AKA-Challenge message MAY optionally contain encrypted data, which is used for IMSI privacy support, as described in Section 6. The AT_MAC attribute contains a message authentication code covering the EAP packet. The encrypted data is not shown in the figures of this section.

The client runs the AKA algorithm (perhaps inside an USIM) and verifies the AUTN. If this is successful, the client is talking to a legitimate EAP server and proceeds to send the EAP-Response/AKA-Challenge. This message contains a result parameter that allows the EAP server in turn to verify that the client is a legitimate one, and the AT_MAC attribute to integrity protect the EAP message.

```
 Client                                                   Authenticator
    |                                                        |
    |                      EAP-Request/Identity              |
    |<-------------------------------------------------------|
    |                                                        |
    | EAP-Response/Identity                                  |
    | (Includes user's NAI)                                  |
    |------------------------------------------------------->|
    |                                                        |
    |                            +-------------------------------+
    |                            | Server runs UMTS algorithms, |
    |                            | generates RAND and AUTN.     |
    |                            +-------------------------------+
    |                                                        |
    |                         EAP-Request/AKA-Challenge      |
    |                         (AT_RAND, AT_AUTN, AT_MAC)     |
    |<-------------------------------------------------------|
    |                                                        |
+-------------------------------------+                      |
| Client runs UMTS algorithms on USIM,|                      |
| verifies AUTN and MAC, derives RES  |                      |
| and session key                     |                      |
+-------------------------------------+                      |
    |                                                        |
    | EAP-Response/AKA-Challenge                             |
    | (AT_RES, AT_MAC)                                       |
    |------------------------------------------------------->|
    |                                                        |
    |                          +---------------------------------+
    |                          | Server checks the given RES,   |
    |                          | and MAC and finds them correct.|
    |                          +---------------------------------+
    |                                                        |
    |                                           EAP-Success  |
    |<-------------------------------------------------------|
```

When EAP AKA is run in the GSM compatible mode, the message flow is otherwise identical to the message flow below except that the AT_AUTN attribute is not included in EAP-Request/AKA-Challenge packet and AT_MAC attribute is not included in any attribute.

The second message flow shows how the EAP server rejects the Client due to failed authentication. The same flow is also used in the GSM compatible mode, except that the AT_AUTN attribute and AT_MAC attribute are not used in the messages.

```
    Client                                              Authenticator
       |                                                          |
       |                    EAP-Request/Identity                  |
       |<---------------------------------------------------------|
       |                                                          |
       | EAP-Response/Identity                                    |
       | (Includes user's NAI)                                    |
       |--------------------------------------------------------->|
       |                                                          |
       |                               +------------------------------+
       |                               | Server runs UMTS algorithms, |
       |                               | generates RAND and AUTN.     |
       |                               +------------------------------+
       |                                                          |
       |                    EAP-Request/AKA-Challenge             |
       |                    (AT_RAND, AT_AUTN, AT_MAC)            |
       |<---------------------------------------------------------|
       |                                                          |
   +--------------------------------------+                       |
   | Client runs UMTS algorithms on USIM,|                       |
   | possibly verifies AUTN, and sends an|                       |
   | invalid response                    |                       |
   +--------------------------------------+                       |
       |                                                          |
       | EAP-Response/AKA-Challenge                               |
       | (AT_RES, AT_MAC)                                         |
       |--------------------------------------------------------->|
       |                                                          |
       |              +--------------------------------------------+
       |              | Server checks the given RES and the MAC,  |
       |              | and finds one of them incorrct.           |
       |              +--------------------------------------------+
       |                                                          |
       |                                             EAP-Failure  |
       |<---------------------------------------------------------|
```

The next message flow shows the client rejecting the AUTN of the EAP server. This flow is not used in the GSM compatible mode.

The client sends an explicit error message (EAP-Response/AKA-Authentication-Reject) to the Authenticator, as usual in AKA when AUTN is incorrect. This allows the EAP server to produce the same error statistics as AKA in general produces in UMTS. Please note that this behavior is different from other EAP/AKA error cases, such as when encountering an incorrect AT_MAC attribute, when the client silently discards the EAP/AKA message.

```
  Client                                              Authenticator
     |                                                        |
     |                      EAP-Request/Identity              |
     |<-------------------------------------------------------|
     |                                                        |
     | EAP-Response/Identity                                  |
     | (Includes user's NAI)                                  |
     |------------------------------------------------------->|
     |                                                        |
     |                                 +-------------------------------+
     |                                 | Server runs UMTS algorithms,  |
     |                                 | generates RAND and a bad AUTN |
     |                                 +-------------------------------+
     |                                                        |
     |                         EAP-Request/AKA-Challenge      |
     |                         (AT_RAND, AT_AUTN, AT_MAC)     |
     |<-------------------------------------------------------|
     |                                                        |
 +--------------------------------------+                     |
 | Client runs UMTS algorithms on USIM  |                     |
 | and discovers AUTN that can not be   |                     |
 | verified                             |                     |
 +--------------------------------------+                     |
     |                                                        |
     | EAP-Response/AKA-Authentication-Reject                 |
     |------------------------------------------------------->|
     |                                                        |
     |                                                        |
     |                                          EAP-Failure   |
     |<-------------------------------------------------------|
```

Networks that are not UMTS aware use the GSM compatible version of this protocol even for UMTS subscribers. In this case, the AUTN parameter is not included in the EAP-Request/AKA-Challenge packet. If a UMTS capable client does not want to accept the use of the GSM compatible mode, the client can reject the authentication by silently ignoring any EAP-Request/AKA-Challenge packets that do not include the AUTN parameter.

The AKA uses shared secrets between the Client and the Client's home operator together with a sequence number to actually perform an authentication. In certain circumstances it is possible for the sequence numbers to get out of sequence. Here's what happens then:

```
  Client                                                  Authenticator
     |                                                          |
     |                      EAP-Request/Identity                |
     |<---------------------------------------------------------|
     |                                                          |
     | EAP-Response/Identity                                    |
     | (Includes user's NAI)                                    |
     |--------------------------------------------------------->|
     |                                                          |
     |                            +-------------------------------+
     |                            | Server runs UMTS algorithms, |
     |                            | generates RAND and AUTN.      |
     |                            +-------------------------------+
     |                                                          |
     |                         EAP-Request/AKA-Challenge        |
     |                         (AT_RAND, AT_AUTN, AT_MAC)       |
     |<---------------------------------------------------------|
     |                                                          |
 +---------------------------------------+                      |
 | Client runs UMTS algorithms on USIM   |                      |
 | and discovers AUTN that contains an   |                      |
 | inappropriate sequence number         |                      |
 +---------------------------------------+                      |
     |                                                          |
     | EAP-Response/AKA-Synchronization-Failure                 |
     | (AT_AUTS)                                                |
     |--------------------------------------------------------->|
     |                                                          |
     |                              +---------------------------+
     |                              | Perform resynchronization |
     |                              | Using AUTS and            |
     |                              | the sent RAND             |
     |                              +---------------------------+
     |                                                          |
```

After the resynchronization process takes place in the server and AAA side, the process continues by the server side sending a new EAP-Request/AKA-Challenge message.

In addition to the full authentication scenarios described above, EAP AKA includes a re-authentication procedure, which is specified in Section 7.

4. User identity in EAP-Response/Identity

In the beginning of EAP authentication, the Authenticator issues the EAP-Request/Identity packet to the client. The client responds with EAP-Response/Identity, which contains the user's identity. The formats of these packets are specified in [6].

UMTS and GSM subscribers are identified with the International Mobile Subscriber Identity (IMSI) [12]. The IMSI is composed of a three digit Mobile Country Code (MCC), a two or three digit Mobile Network Code (MNC) and a not more than 10 digit Mobile Subscriber

Identification Number (MSIN). In other words, the IMSI is a string of not more than 15 digits. MCC and MNC uniquely identify the operator.

Internet AAA protocols identify users with the Network Access Identifier (NAI) [5]. When used in a roaming environment, the NAI is composed of a username and a realm, separated with "@" (username@realm). The username portion identifies the subscriber within the realm. The AAA nodes use the realm portion of the NAI to route AAA requests to the correct AAA server. The realm name used in this protocol MAY be chosen by the operator and it MAY a configurable parameter in the EAP/AKA client implementation. In this case, the client is typically configured with the NAI realm of the home operator. Operators MAY reserve a specific realm name for EAP/AKA users. This convention makes it easy to recognize that the NAI identifies a subscriber that uses EAP/AKA. Such reserved NAI realm may be useful as a hint as to the first authentication method to use during method negotiation.

There are three types of NAI username portions in EAP/AKA: non-pseudonym permanent usernames that are based on the IMSI, pseudonym usernames and re-authentication usernames. The first two are only used on full authentication and the last one only on re-authentication. When the optional IMSI privacy support is not used, the non-pseudonym permanent username is used. The non-pseudonym permanent username is of the format "0imsi". In other words, the first character of the username is the digit zero (ASCII value 0x30), followed by the IMSI. The IMSI is an ASCII string that consists of not more than 15 decimal digits (ASCII values between 0x30 and 0x39) as specified in [12]

The EAP server MAY use the leading "0" as a hint to try EAP/AKA as the first authentication method during method negotiation, rather than for example EAP/SIM. The EAP/AKA server MAY propose EAP/AKA even if the leading character was not "0".

When the optional identity privacy support is used on full authentication, the client MAY use the pseudonym received as part of the previous full authentication sequence as the username portion of the NAI, as specified in Section 6. The client MUST NOT modify the pseudonym received in AT_NEXT_PSEUDONYM. For example, the client MUST NOT append any leading characters in the pseudonym.

On re-authentication, the client uses the re-authentication identity received as part of the previous authentication sequence as the NAI. A new re-authentication identity may be delivered as part of both full authentication and re-authentication. The client MUST NOT modify the re-authentication identity received in AT_NEXT_REAUTH_ID. For example, the client MUST NOT append any leading characters in the re-authentication identity.

If no configured realm name is available, the client MAY derive the realm name from the MCC and MNC portions of the IMSI. In this case, the realm name is obtained by concatenating "mnc", the MNC digits of

IMSI, ".mcc", the MCC digits of IMSI and ".owlan.org". For example, if the IMSI is 123456789098765, and the MNC is three digits long, then the derived realm name is "mnc456.mcc123.owlan.org".
If the client is not able to determine whether the MNC is two or three digits long, the client MAY use a 3-digit MNC. If the correct length of the MNC is two, then the MNC used in the realm name will include the first digit of MSIN. Hence, when configuring AAA networks for operators that have 2-digit MNC's, the network SHOULD also be prepared for realm names with incorrect 3-digit MNC's.

5. Obtaining Subscriber Identity via EAP AKA Messages

It may be useful to obtain the identity of the subscriber through means other than EAP Request/Identity. This can eliminate the need for an identity request when using EAP method negotiation. If this was not possible then it might not be possible to negotiate EAP/AKA as the second method since it is not specified how to deal with a new EAP Request/Identity.

If the EAP server does not have any identity (IMSI, pseudonym or re-authentication username) available when sending the first EAP/AKA request, then the EAP server may issue the EAP-Request/AKA-Identity packet and includes the AT_ANY_ID_REQ attribute (specified in Section 10.5). This attribute does not contain any data.

The AT_ANY_ID_REQ attribute requests the client to include the AT_IDENTITY attribute (specified in Section 10.6) in the EAP-Response/AKA-Identity packet. The identity format in the AT_IDENTITY attribute is the same as in the EAP-Response/Identity packet. The AT_IDENTITY attribute contains an IMSI-based permanent identity, a pseudonym identity or a re-authentication identity. If the server does not support re-authentication, it uses the AT_FULLAUTH_ID_REQ attribute instead of the AT_ANY_ID_REQ attribute to directly request for a full authentication identity (either the permanent identity or a pseudonym identity). If the server uses the AT_FULLAUTH_ID_REQ attribute, the client MUST NOT use a re-authentication identity in the AT_IDENTITY attribute.

The use of pseudonyms for anonymity is specified in Section 6. The use of re-authentication identities is specified in Section 7.

This case for full authentication is illustrated in the figure below. In this case, AT_IDENTITY contains either the permanent identity or a pseudonym identity. The same sequence is also used in case the server uses the AT_FULLAUTH_ID_REQ in EAP-Request/AKA-Identity

```
    Client                                               Authenticator
       |                                                       |
       |                              +-------------------------------+
       |                              | Server does not have any      |
       |                              | Subscriber identity available|
       |                              | When starting EAP/AKA         |
       |                              +-------------------------------+
       |                                                       |
       |          EAP-Request/AKA-Identity                     |
       |          (AT_ANY_ID_REQ)                              |
       |<------------------------------------------------------|
       |                                                       |
       |                                                       |
       | EAP-Response/AKA-Identity                             |
       | (AT_IDENTITY)                                         |
       |------------------------------------------------------>|
       |                                                       |
```

If the client wants to perform full authentication, it includes the permanent identity or a pseudonym identity in the AT_IDENTITY attribute. The client may use these identities in response to either AT_ANY_ID_REQ or AT_FULLAUTH_ID_REQ. If the server uses the AT_ANY_ID_REQ and the client wants to perform re-authentication, then the client includes a re-authentication identity in the AT_IDENTITY attribute.

If the client uses its full authentication identity and the AT_IDENTITY attribute contains a valid permanent identity or a valid pseudonym identity that the EAP server is able to decode to the permanent identity, then the full authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/AKA-Challenge message.

On re-authentication, if the AT_IDENTITY attribute contains a valid re-authentication identity and the server agrees on using re-authentication, then the server proceeds with the re-authentication sequence and issues the EAP-Request/AKA-Reauthentication packet, as specified in Section 7. If the server does not recognize the re-authentication identity, then the server issues a second EAP-Request/AKA-Identity message and includes the AT_FULLAUTH_ID_REQ attribute. In this case, a second EAP/AKA-Identity round trip is required. The messages used on the first roundtrip are ignored. This is illustrated below.

```
    Client                                                   Authenticator
       |                                                          |
       |                             +-------------------------------+
       |                             | Server does not have any      |
       |                             | Subscriber identity available|
       |                             | When starting EAP/AKA         |
       |                             +-------------------------------+
       |                                                          |
       |         EAP-Request/AKA-Identity                         |
       |         (AT_ANY_ID_REQ)                                  |
       |<---------------------------------------------------------|
       |                                                          |
       |                                                          |
       | EAP-Response/AKA-Identity                                |
       | (AT_IDENTITY containing a re-authentication identity)    |
       |--------------------------------------------------------->|
       |                                                          |
       |                             +-------------------------------+
       |                             | Server does not recognize     |
       |                             | The re-authentication         |
       |                             | Identity                      |
       |                             +-------------------------------+
       |                                                          |
       |     EAP-Request/AKA-Identity                             |
       |     (AT_FULLAUTH_ID_REQ)                                 |
       |<---------------------------------------------------------|
       |                                                          |
       |                                                          |
       | EAP-Response/AKA-Identity                                |
       | (AT_IDENTITY with a full-auth. Identity)                 |
       |--------------------------------------------------------->|
       |                                                          |
```

If the server recognizes the re-authentication identity, but still wants to fall back on full authentication, the server may issue the EAP-Request/AKA-Challenge packet. In this case, the full authentication procedure proceeds as usual.

An extra EAP/AKA-Identity round trip is also required in cases when the AT_IDENTITY attribute contains a pseudonym identity that the EAP server fails to decode. The operation in this case is specified in Section 6.

6. Identity Privacy Support

EAP/AKA includes optional identity privacy (anonymity) support that can be used to hide the cleartext IMSI and to make the subscriber’s connections unlinkable to eavesdroppers. Identity privacy is based on temporary identities, or pseudonyms, which are equivalent to but separate from the Temporary Mobile Subscriber Identities (TMSI) that are used on cellular networks.

If identity privacy is not used or if the client does not have any pseudonyms or re-authentication identities are available, the client transmits the permanent identity (based on IMSI) in the EAP-Response/Identity packet or in the AT_IDENTITY attribute.

The EAP-Request/AKA-Challenge message MAY include an encrypted pseudonym in the value field of the AT_ENCR_DATA attribute. The AT_IV and AT_MAC attributes are also used to transport the pseudonym to the client, as described in Section 10.1. Because the identity privacy support is optional to implement, the client MAY ignore the AT_IV and AT_ENCR_DATA attributes and always transmit the IMSI-based permanent identity in the EAP-Response/Identity packet and in the AT_IDENTITY attribute.

On receipt of the EAP-Request/AKA-Challenge, the client verifies the AT_MAC attribute before looking at the AT_ENCR_DATA attribute. If the AT_MAC is invalid, then the client MUST silently discard the EAP packet. If the AT_MAC attribute is valid, then the client MAY decrypt the encrypted data in AT_ENCR_DATA and use the obtained pseudonym on the next full authentication.

If the client does not receive a new pseudonym in the EAP-Request/AKA-Challenge message, the client MAY use an old pseudonym instead of the permanent identity on next full authentication.

The EAP server produces pseudonyms in an implementation-dependent manner. Please see [8] for examples on how to produce pseudonyms. Only the EAP server needs to be able to map the pseudonym to the permanent identity. Regardless of construction method, the pseudonym MUST conform to the grammar specified for the username portion of an NAI. The EAP AKA server MAY produce pseudonyms that begin with a leading "0" character in order to be able to use the leading character as a hint in EAP method negotiation during next authentication.

On the next full authentication with the EAP server, the client MAY transmit the received pseudonym in the first EAP-Response/Identity packet. The client concatenates the received pseudonym with the "@" character and the NAI realm portion. The client selects the realm name portion similarly as it select the realm name portion when using the permanent identity. If the EAP server successfully decodes the pseudonym received in the EAP-Response/Identity packet to a known client identity (IMSI), the authentication proceeds with the EAP-Request/AKA-Challenge message as usual.

Because the client may fail to save a pseudonym sent to in an EAP-Request/AKA-Challenge, for example due to malfunction, the EAP server SHOULD maintain at least one old pseudonym in addition to the most recent pseudonym.

If the EAP server requests the client to include its identity in the EAP-Response/AKA-Identity packet, as specified in Section 5, the client MAY transmit the received pseudonym in the AT_IDENTITY attribute. If the EAP server successfully decodes the pseudonym to a known identity, then the authentication proceeds with the EAP-Request/AKA-Challenge packet as usual.

If the EAP server fails to decode the pseudonym to a known identity, then the EAP server requests the permanent identity (non-pseudonym identity) by including the AT_PERMANENT_ID_REQ attribute (Section 10.5) in the EAP-Request/AKA-Challenge message.

The EAP server issues the EAP-Request/AKA-Identity message also in the case when it received the undecodable pseudonym in AT_IDENTITY included the EAP-Response/AKA-Identity packet. In this case, a second EAP/AKA-Identity round trip is required.

A received AT_PERMANENT_ID_REQ does not necessarily originate from the valid network, but an active attacker may transmit an EAP-Request/AKA-Identity packet with an AT_PERMANENT_ID_REQ attribute to the client, in an effort to find out the true identity of the user. On receipt of EAP-Request/AKA-Identity that includes AT_PERMANENT_ID_REQ, the client MAY delay the processing of the message for a while in order to wait for another EAP AKA message that does not include the AT_PERMANENT_ID_REQ attribute.

Basically, there are two different policies that the client can employ with regard to AT_PERMANENT_ID_REQ. A "conservative" client assumes that the network is able to maintain pseudonyms robustly. Therefore, if a conservative client has a pseudonym, the client silently ignores the EAP packet with AT_PERMANENT_ID_REQ, because the client believes that the valid network is able to decode the pseudonym. (Alternatively, the conservative client may respond to AT_PERMANENT_ID_REQ in certain circumstances, for example if the pseudonym was received a long time ago.) The benefit of this policy is that it protects the client against active attacks on anonymity. On the other hand, a "liberal" client always accepts the AT_PERMANENT_ID_REQ and responds with the IMSI-based permanent identity. The benefit of this policy is that it works even if the valid network sometimes loses pseudonyms and is not able to decode them to the permanent identity.

The value field of the AT_PERMANENT_ID_REQ does not contain any data but the attribute is included to request the client to include the AT_IDENTITY attribute (Section 10.6) with the permanent authentication identity in the EAP-Response/AKA-Identity message. In this case, the AT_IDENTITY attribute contains the client’s permanent identity in the clear.

Please note that the EAP/AKA client and the EAP/AKA server only process the AT_IDENTITY attribute and entities that only pass through EAP packets do not process this attribute. Hence, if the EAP server is not co-located in the authenticator, then the authenticator and other intermediate AAA elements (such as possible AAA proxy servers) will continue to refer to the client with the original identity from the EAP-Response/Identity packet regardless if the decoding fails in the EAP server.

The figure below illustrates the case when the EAP server fails to decode the pseudonym included in the EAP-Response/Identity packet.

```
    Client                                                 Authenticator
       |                                                        |
       |                              EAP-Request/Identity      |
       |<-------------------------------------------------------|
       |                                                        |
       | EAP-Response/Identity                                  |
       | (Includes a pseudonym)                                 |
       |------------------------------------------------------->|
       |                                                        |
       |                            +-------------------------------+
       |                            | Server fails to decode the    |
       |                            | Pseudonym.                    |
       |                            +-------------------------------+
       |                                                        |
       |  EAP-Request/AKA-Identity                              |
       |  (AT_PERMANENT_ID_REQ)                                 |
       |<-------------------------------------------------------|
       |                                                        |
       |                                                        |
       | EAP-Response/AKA-Identity                              |
       | (AT_IDENTITY with permanent identity)                  |
       |------------------------------------------------------->|
       |                                                        |
```

After the EAP-Response/AKA-Identity message, the authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/AKA-Challenge message.

The figure below illustrates the case when the EAP server fails to decode the pseudonym included in the AT_IDENTITY attribute.

```
    Client                                              Authenticator
       |                                                        |
       |                            +-------------------------------+
       |                            | Server does not have any      |
       |                            | Subscriber identity available |
       |                            | When starting EAP/AKA         |
       |                            +-------------------------------+
       |                                                        |
       |        EAP-Request/AKA-Identity                        |
       |        (AT_ANY_ID_REQ)                                 |
       |<-------------------------------------------------------|
       |                                                        |
       |                                                        |
       |EAP-Response/AKA-Identity                               |
       |(AT_IDENTITY with a pseudonym identity)                 |
       |------------------------------------------------------->|
       |                                                        |
       |                                                        |
       |                            +-------------------------------+
       |                            | Server fails to decode the    |
       |                            | Pseudonym in AT_IDENTITY      |
       |                            +-------------------------------+
       |                                                        |
       |                EAP-Request/AKA-Identity                |
       |                (AT_PERMANENT_ID_REQ)                   |
       |<-------------------------------------------------------|
       |                                                        |
       |                                                        |
       | EAP-Response/AKA-Identity                              |
       | (AT_IDENTITY with permanent identity)                  |
       |------------------------------------------------------->|
       |                                                        |
```

In the worst case, there are three EAP/AKA-Identity round trips before the server has obtained an acceptable identity: on the first round, the client sends its re-authentication identity in AT_IDENTITY. The server fails to accept it and request a full authentication identity with a second EAP-Request/AKA-Identity. The client responds with a pseudonym identity in AT_IDENTITY. The server fails to decode the pseudonym and has to issue a third EAP-Request/AKA-Identity, including AT_PERMANENT_ID_REQ. Finally, the server accepts the client’s EAP-Response/AKA-Identity with the AT_IDENTITY attribute and proceeds with full authentication. This is illustrated in the figure below.

```
    Client                                             Authenticator
       |                                                       |
       |                            +------------------------------+
       |                            | Server does not have any     |
       |                            | Subscriber identity available|
       |                            | When starting EAP/AKA        |
       |                            +------------------------------+
       |                                                       |
       |        EAP-Request/AKA-Identity                       |
       |        (AT_ANY_ID_REQ)                                |
       |<------------------------------------------------------|
       |                                                       |
       | EAP-Response/AKA-Identity                             |
       | (AT_IDENTITY with re-authentication identity)         |
       |------------------------------------------------------>|
       |                                                       |
       |                            +------------------------------+
       |                            | Server does not accept       |
       |                            | The re-authentication        |
       |                            | Identity                     |
       |                            +------------------------------+
       |                                                       |
       |     EAP-Request/AKA-Identity                          |
       |     (AT_FULLAUTH_ID_REQ)                              |
       |<------------------------------------------------------|
       |                                                       |
       |EAP-Response/AKA-Identity                              |
       |(AT_IDENTITY with a pseudonym identity)                |
       |------------------------------------------------------>|
       |                                                       |
       |                            +------------------------------+
       |                            | Server fails to decode the   |
       |                            | Pseudonym in AT_IDENTITY     |
       |                            +------------------------------+
       |                                                       |
       |           EAP-Request/AKA-Identity                    |
       |           (AT_PERMANENT_ID_REQ)                       |
       |<------------------------------------------------------|
       |                                                       |
       |                                                       |
       | EAP-Response/AKA-Identity                             |
       | (AT_IDENTITY with permanent identity)                 |
       |------------------------------------------------------>|
       |                                                       |
```

After the last EAP-Response/AKA-Identity message, the full authentication sequence proceeds as usual with the EAP Server issuing the EAP-Request/AKA-Challenge message.

Because the keys that are used to protect the pseudonym are derived from the AKA cipher key (CK) and the AKA integrity key (IK), the identity privacy support is not available when EAP AKA is used in the GSM compatible mode.

7. Re-authentication

In some environments, EAP authentication may be performed frequently. Because the EAP AKA full authentication procedure makes use of the UMTS AKA algorithms, and it therefore requires fresh authentication vectors from the Authentication Centre, the full authentication procedure is not very well suitable for frequent use. Therefore, EAP AKA includes a more inexpensive re-authentication procedure that does not make use of the UMTS AKA algorithms and does not need new vectors from the Authentication Centre.

Re-authentication is optional to implement for both the EAP AKA server and client. On each EAP authentication, either one of the entities may also fall back on full authentication if they do not want to use re-authentication.

Re-authentication is based on the keys derived on the preceding full authentication. The same K_aut and K_encr keys as in full authentication are used to protect EAP AKA packets and attributes, and the original XKEY seed value from full authentication is used to generate fresh application specific keys, as specified in Section 12.

On re-authentication, the client protects against replays with an unsigned 16-bit counter, included in the AT_COUNTER attribute. On full authentication, both the server and the client initialize the counter to one. The counter value of at least one is used on the first re-authentication. On subsequent re-authentications, the counter MUST be greater than on any of the previous re-authentications. For example, on the second re-authentication, counter value is two or greater etc. The AT_COUNTER attribute is encrypted.

The server includes an encrypted server nonce (AT_NONCE_S) in the re-authentication request. The AT_MAC attribute in the client’s response is calculated over NONCE_S to provide a challenge/response authentication scheme. The NONCE_S also contributes to the new application specific keys.

As discussed in Section 6, in some environments the client may assume that the network can reliably store pseudonyms and therefore the client may fail to respond to the AT_PERMANENT_ID_REQ attribute. The network SHOULD store pseudonyms on a reliable database. Because one of the objectives of the re-authentication procedure is to reduce load on the network, the re-authentication procedure does not require the EAP server to contact a reliable database. Therefore, the re-authentication procedure makes use of separate re-authentication user identities. Pseudonyms and the permanent IMSI-based identity are reserved for full authentication only. The network does not need to store re-authentication identities as carefully as pseudonyms. If a re-authentication identity is lost and the network does not recognize it, the EAP server can fall back on full authentication.

If the EAP server supports re-authentication, it MAY include the skippable AT_NEXT_REAUTH_ID attribute in the encrypted data of EAP-Request/AKA-Challenge message. This attribute contains a new re-authentication identity for the next re-authentication. The client MAY ignore this attribute, in which case it will use full authentication next time. If the client wants to use re-authentication, it uses this re-authentication identity on next authentication. Even if the client has a re-authentication identity, the client MAY discard the re-authentication identity and use a pseudonym or the IMSI-based permanent identity instead, in which case full authentication will be performed.

The re-authentication identity received in AT_NEXT_REAUTH_ID contains both the username portion and the realm portion of the Network Access Identifier. The EAP Server can choose an appropriate realm part in order to have the AAA infrastructure route subsequent re-authentication related requests to the same AAA server. For example, the realm part MAY include a portion that is specific to the AAA server. Hence, it is sufficient to store the context required for re-authentication in the AAA server that performed the full authentication.

The client MAY use the re-authentication identity in the EAP-Response/Identity packet or, in response to server's AT_ANY_ID_REQ attribute, the client MAY use the re-authentication identity in the AT_IDENTITY attribute of the EAP-Response/AKA-Identity packet.

Even if the client uses a re-authentication identity, the server may want to fall back on full authentication, for example because the server does not recognize the re-authentication identity or does not want to use re-authentication. If the server was able to decode the re-authentication identity to the permanent identity, the server issues the EAP-Request/AKA-Challenge packet to initiate full authentication. If the server was not able to recover the client's identity from the re-authentication identity, the server starts the full authentication procedure by issuing an EAP-Request/AKA-Identity packet. This packet always starts a full authentication sequence if it does not include the AT_ANY_ID_REQ attribute. (As specified in Sections 5 and 6, the server MAY use AT_ANY_ID_REQ, AT_FULLAUTH_ID_REQ or AT_PERMANENT_ID_REQ attributes if it does not know the client's identity.)

Both the client and the server SHOULD have an upper limit for the number of subsequent re-authentications allowed before a full authentication needs to be performed. Because a 16-bit counter is used in re-authentication, the theoretical maximum number of re-authentications is reached when the counter value reaches 0xFFFF.

In order to use re-authentication, the client and the server need to store the following values: original XKEY, K_aut, K_encr, latest counter value and the next re-authentication identity.

The following figure illustrates the re-authentication procedure. Encrypted attributes are denoted with '*'. The client uses its re-authentication identity in the EAP-Response/Identity packet. As discussed above, an alternative way to communicate the re-authentication identity to the server is for the client to use the AT_IDENTITY attribute in the EAP-Response/AKA-Identity message. This latter case is not illustrated in the figure below, and it is only possible when the server requests the client to send its identity by including the AT_ANY_ID_REQ attribute in the EAP-Request/AKA-Identity packet.

If the server recognizes the re-authentication identity and agrees on using re-authentication, then the server sends the EAP-Request/AKA-Reauthentication packet to the client. This packet MUST include the encrypted AT_COUNTER attribute, with a fresh counter value, the encrypted AT_NONCE_S attribute that contains a random number chosen by the server, the AT_ENCR_DATA and the AT_IV attributes used for encryption, and the AT_MAC attribute that contains a message authentication code over the packet. The packet MAY also include an encrypted AT_NEXT_REAUTH_ID attribute that contains the next re-authentication identity.

Re-authentication identities are one-time identities. If the client does not receive a new re-authentication identity, it MUST use either the permanent identity or a pseudonym identity on the next authentication to initiate full authentication.

The client verifies that the counter value is fresh (greater than any previously used value). The client also verifies that AT_MAC is correct. The client MAY save the next re-authentication identity from the encrypted AT_NEXT_REAUTH_ID for next time. If all checks are successful, the client responds with the EAP-Response/AKA-Reauthentication packet, including the AT_COUNTER attribute with the same counter value and the AT_MAC attribute.

The server verifies the AT_MAC attribute and also verifies that the counter value is the same that it used in the EAP-Request/AKA-Reauthentication packet. If these checks are successful, the re-authentication has succeeded and the server sends the EAP-Success packet to the client.

```
Client                                                  Authenticator
   |                                                        |
   |                              EAP-Request/Identity      |
   |<-------------------------------------------------------|
   |                                                        |
   | EAP-Response/Identity                                  |
   | (Includes a re-authentication identity)                |
   |------------------------------------------------------->|
   |                                                        |
   |                          +--------------------------------+
   |                          | Server recognizes the identity |
   |                          | and agrees on using fast       |
   |                          | re-authentication              |
   |                          +--------------------------------+
   |                                                        |
   |  EAP-Request/AKA-Reauthentication                      |
   |  (AT_IV, AT_ENCR_DATA, *AT_COUNTER,                    |
   |   *AT_NONCE_S, *AT_NEXT_REAUTH_ID, AT_MAC)             |
   |<-------------------------------------------------------|
   |                                                        |
   |                                                        |
+-----------------------------------------------+          |
| Client verifies AT_MAC and the freshness of   |          |
| the counter. Client MAY store the new re-     |          |
| authentication identity for next re-auth.     |          |
+-----------------------------------------------+          |
   |                                                        |
   | EAP-Response/AKA-Reauthentication                      |
   | (AT_IV, AT_ENCR_DATA, *AT_COUNTER with same value,     |
   |  AT_MAC)                                               |
   |------------------------------------------------------->|
   |                                                        |
   |                          +--------------------------------+
   |                          | Server verifies AT_MAC and     |
   |                          | the counter                    |
   |                          +--------------------------------+
   |                                                        |
   |                                          EAP-Success   |
   |<-------------------------------------------------------|
   |                                                        |
```

If the client does not accept the counter value of EAP-Request/AKA-Reauthentication, it indicates the counter synchronization problem by including the encrypted AT_COUNTER_TOO_SMALL in EAP-Response/AKA-Reauthentication. The server responds with EAP-Request/AKA-Challenge to initiate a normal full authentication procedure. This is illustrated in the following figure. Encrypted attributes are denoted with '*'.

```
    Client                                                Authenticator
       |                                                        |
       |                                EAP-Request/Identity    |
       |<-------------------------------------------------------|
       |                                                        |
       | EAP-Response/Identity                                  |
       | (Includes a re-authentication identity)                |
       |------------------------------------------------------->|
       |                                                        |
       |  EAP-Request/AKA-Reauthentication                      |
       |  (AT_IV, AT_ENCR_DATA, *AT_COUNTER,                    |
       |   *AT_NONCE_S, *AT_NEXT_REAUTH_ID, AT_MAC)             |
       |<-------------------------------------------------------|
       |                                                        |
  +-----------------------------------------------+             |
  | AT_MAC is valid but the counter is not fresh. |             |
  +-----------------------------------------------+             |
       |                                                        |
       | EAP-Response/AKA-Reauthentication                      |
       | (AT_IV, AT_ENCR_DATA, *AT_COUNTER_TOO_SMALL,           |
       |  *AT_COUNTER, AT_MAC)                                  |
       |------------------------------------------------------->|
       |                                                        |
       |            +----------------------------------------------+
       |            | Server verifies AT_MAC but detects           |
       |            | That client has included AT_COUNTER_TOO_SMALL|
       |            +----------------------------------------------+
       |                                                        |
       |                         EAP-Request/AKA-Challenge      |
       |<-------------------------------------------------------|
       |                                                        |
  +----------------------------------------------------------------+
  |                Normal full authentication follows.             |
  +----------------------------------------------------------------+
       |                                                        |
```

In the figure above, the first three messages are similar to the basic re-authentication case. When the client detects that the counter value is not fresh, it includes the AT_COUNTER_TOO_SMALL attribute in EAP-Response/AKA-Reauthentication. This attribute doesn’t contain any data but it is a request for the server to initiate full authentication. In this case, the client MUST ignore the contents of the server’s AT_NEXT_REAUTH_ID attribute.

On receipt of AT_COUNTER_TOO_SMALL, the server verifies AT_MAC and verifies that AT_COUNTER contains the same as in the EAP-Request/AKA-Reauthentication packet. If not, the server silently discards the EAP-Response/AKA-Reauthentication packet. If all checks on the packet are successful, the server transmits a EAP-Request/AKA-Challenge packet and the full authentication procedure is performed as usual. Since the server already knows the subscriber identity, it MUST NOT use the EAP-Request/AKA-Identity packet to request the subscriber identity.

8. Message Format

The Type-Data of the EAP AKA packets begins with a 1-octet Subtype field, which is followed by a 2-octet reserved field. The rest of the Type-Data consists of attributes that are encoded in Type, Length, Value format. The figure below shows the generic format of an attribute.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Attribute Type |    Length     | Value...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Attribute Type

Indicates the particular type of attribute. The attribute type values are listed in Section 14.

Length

Indicates the length of this attribute in multiples of 4 bytes. The maximum length of an attribute is 1024 bytes. The length includes the Attribute Type and Length bytes.

Value

The particular data associated with this attribute. This field is always included and it may be two or more bytes in length. The type and length fields determine the format and length of the value field.

When an attribute numbered within the range 0 through 127 is encountered but not recognized, the EAP/AKA message containing that attribute MUST be silently discarded. These attributes are called non-skippable attributes.

When an attribute numbered in the range 128 through 255 is encountered but not recognized that particular attribute is ignored, but the rest of the attributes and message data MUST still be processed. The Length field of the attribute is used to skip the attribute value in searching for the next attribute. These attributes are called skippable attributes.

EAP/AKA packets do not include a version field. However, should there be reason to revise this protocol in the future, new non-skippable or skippable attributes could be specified in order to implement revised EAP/AKA versions in a backward-compatible manner.

Unless otherwise specified, the order of the attributes in an EAP AKA message is insignificant, and an EAP AKA implementation should not assume a certain order to be used.

Attributes can be encapsulated within other attributes. In other words, the value field of an attribute type can be specified to contain other attributes.

9. Message Authentication and Encryption

This section specifies EAP/AKA attributes for attribute encryption and EAP/AKA message authentication.

Encryption and integrity protection are based on the AKA session keys CK and IK. Because the CK and IK keys are derived from the RAND challenge, these attributes can only be used in the EAP-Request/AKA-Challenge message and any EAP/AKA messages sent after EAP-Request/AKA-Challenge. For example, these attributes cannot be used in EAP-Request/AKA-Identity, because the RAND challenge has not yet been transmitted at that point. As there is no key derivation specification for the GSM mode, attribute encryption and message integrity protection are not available in the GSM mode.

9.1. AT_MAC Attribute

The AT_MAC attribute can optionally be used for EAP/AKA message integrity protection. Whenever AT_ENCR_DATA (Section 9.2) is included in an EAP message, it MUST be followed (not necessarily immediately) by an AT_MAC attribute. Messages that do not meet this condition MUST be silently discarded.

The value field of the AT_MAC attribute contains two reserved bytes followed by a message authentication code (MAC). The MAC is calculated over the whole EAP packet, concatenated with optional message-specific data, with the exception that the value field of the MAC attribute is set to zero when calculating the MAC. The reserved bytes are set to zero when sending and ignored on reception.

The contents of the message-specific data, if present, are specified separately for each EAP/AKA message. The message-specific data is included in order to protect data that is not transmitted with the EAP packet.

The format of the AT_MAC attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_MAC    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           MAC                                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The MAC algorithm is HMAC-SHA1-128 [9] keyed hash value. (The HMAC-SHA1-128 value is obtained from the 20-byte HMAC-SHA1 value by truncating the output to 16 bytes. Hence, the length of the MAC is 16 bytes.) The message authentication key (K_aut) used in the calculation of the MAC is derived from the AKA integrity key (IK) and cipher key (CK), as specified in Section Error! Reference source not found..

9.2. AT_IV, AT_ENCR_DATA and AT_PADDING Attributes

AT_IV and AT_ENCR_DATA attributes can be optionally used to transmit encrypted information between the EAP/AKA client and server.

The value field of AT_IV contains two reserved bytes followed by a 16-byte initialization vector required by the AT_ENCR_DATA attribute. The reserved bytes are set to zero when sending and ignored on reception. The AT_IV attribute MUST be included if and only if the AT_ENCR_DATA is included. Messages that do not meet this condition MUST be silently discarded.

The sender of the AT_IV attribute chooses the initialization vector by random. The sender MUST NOT reuse the initialization vector value from previous EAP AKA packets but the sender MUST choose it freshly for each AT_IV attribute. The sends SHOULD use a good source of randomness to generate the initialization vector. The format of AT_IV is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_IV     | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The value field of the AT_ENCR_DATA attribute consists of two reserved bytes followed by bytes encrypted using the Advanced Encryption Standard (AES) [10] in the Cipher Block Chaining (CBC) mode of operation, using the initialization vector from the AT_IV attribute. The reserved bytes are set to zero when sending and ignored on reception. Please see [11] for a description of the CBC mode. The format of the AT_ENCR_DATA attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Encrypted Data                             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The encryption key (K_encr) is derived is derived from the AKA integrity key (IK) and cipher key (CK), as specified in Section Error! Reference source not found..
The plaintext consists of nested EAP/AKA attributes.

The encryption algorithm requires the length of the plaintext to be a multiple of 16 bytes. The sender may need to include the AT_PADDING attribute as the last attribute within AT_ENCR_DATA. The AT_PADDING attribute is not included if the total length of other nested attributes within the AT_ENCR_DATA attribute is a multiple of 16 bytes. As usual, the Length of the Padding attribute includes the Attribute Type and Attribute Length fields. The Length of the Padding attribute is 4, 8 or 12 bytes. It is chosen so that the length of the value field of the AT_ENCR_DATA attribute becomes a multiple of 16 bytes. The actual pad bytes in the value field are set to zero (0x00) on sending. The recipient of the message MUST verify that the pad bytes are set to zero, and silently drop the message if this verification fails. The format of the AT_PADDING attribute is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

10. Messages

AT_NEXT_PSEUDONYMEAP-Request/AKA-Challenge

The format of the EAP-Request/AKA-Challenge packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AT_RAND    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                             RAND                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    AT_AUTN    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                        AUTN (optional)                        |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_IV     | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector (optional)              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                    Encrypted Data (optional)                  |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_MAC    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                           MAC (optional)                      |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

1 for Request

Identifier

See [6]

Length

The length of the EAP Request packet.

Type

23

Subtype

1 for AKA-Challenge

Reserved

Set to zero when sending, ignored on reception.

AT_RAND

The value field of this attribute contains two reserved bytes followed by the AKA RAND parameter, 16 bytes (128 bits). The reserved bytes are set to zero when sending and ignored on reception. The AT_RAND attribute MUST be present in EAP-Request/AKA-Challenge.

AT_AUTN

The value field of this attribute contains two reserved bytes followed by the AKA AUTN parameter, 16 bytes (128 bits). The reserved bytes are set to zero when sending and ignored on reception. The AT_AUTN attribute MUST NOT be included in the GSM compatible mode of this protocol; otherwise it MUST be included.

AT_IV

See Section 9.2.

AT_ENCR_DATA

See Section 9.2. The nested attributes that are included in the plaintext of AT_ENCR_DATA are described below.

AT_MAC

AT_MAC MUST NOT be included in GSM compatible mode; otherwise it MUST be included. In EAP-Request/AKA-Challenge, there is no message-specific data covered by the MAC. See Section 9.1.

In the EAP-Request/AKA-Challege message, the AT_IV, AT_ENCR_DATA and AT_MAC attributes are used for IMSI privacy and for communicating the next re-authentication identity. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below. Later versions of this protocol MAY specify additional attributes to be included within the encrypted data.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_NEXT_PS... | Length        | Actual Pseudonym Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                         Next Pseudonym                        .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_NEXT_REAU..| Length        | Actual Re-Auth Identity Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Next Re-authentication Username            .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_NEXT_PSEUDONYM

This attribute is optional. The value field of this attribute begins with 2-byte actual pseudonym length, which specifies the length of the pseudonym in bytes. This field is followed by a pseudonym user name, of the indicated actual length, that the client can use in the next authentication, as described in Section 6. The user name does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the pseudonym with zero bytes when necessary.

AT_NEXT_REAUTH_ID

The AT_NEXT_REAUTH_ID attribute is optional to include. The value field of this attribute begins with 2-byte actual re-authentication identity length, which specifies the length of the re-authentication identity in bytes. This field is followed by a re-authentication identity, of the indicated actual length, that the client can use in the next re-authentication, as described in Section 7. The re-authentication identity includes both a username portion and a realm name portion. The re-authentication identity does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the re-authentication identity with zero bytes when necessary.

AT_PADDING

AT_PADDING is optional to include. See Section 9.2.

10.2. EAP-Response/AKA-Challenge

The format of the EAP-Response/AKA-Challenge packet is shown below.

Later versions of this protocol MAY make use of the AT_ENCR_DATA and AT_IV attributes in this message to include encrypted (skippable) attributes. AT_MAC, AT_ENCR_DATA and AT_IV attributes are not shown in the figure below. If present, they are processed as in EAP-Request/AKA-Challenge packet. The EAP server MUST process EAP-Response/AKA-Challenge messages that include these attributes even if the server did not implement these optional attributes.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Subtype     |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_RES    |    Length     |          RES Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|                                                               |
|                              RES                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     AT_MAC    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                            MAC (optional)                     |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

2 for Response

Identifier

See [6]

Length

The length of the EAP Response packet.

Type

23

Subtype

1 for AKA-Challenge

Reserved

Set to zero when sending, ignored on reception.

AT_RES

This attribute MUST be included in EAP-Response/AKA-Challenge. The value field of this attribute begins with the 2-byte RES Length, which is identifies the exact length of the RES (or SRES) in bits. The RES length is followed by the UMTS AKA RES or GSM SRES parameter. According to the specification [13] the length of the AKA RES can vary between 32 and 128 bits. The GSM SRES parameter is always 32 bits long. Because the length of the AT_RES attribute must be a multiple of 4 bytes, the sender pads the RES with zero bits where necessary.

AT_MAC

AT_MAC MUST NOT be included in GSM compatible mode; otherwise it MUST be included. In EAP-Response/AKA-Challenge, there is no message-specific data covered by the MAC. See Section 9.1.

10.3. EAP-Response/AKA-Authentication-Reject

The format of the EAP-Response/AKA-Authentication-Reject packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

2 for Response

Identifier

See [6]

Length

The length of the EAP Response packet.

Type

23

Subtype

2 for AKA-Authentication-Reject

Reserved

Set to zero on sending, ignored on reception.

10.4. EAP-Response/AKA-Synchronization-Failure

The format of the EAP-Response/AKA-Synchronization-Failure packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Subtype     |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+|
|    AT_AUTS    | Length = 4    |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                             AUTS                              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

2 for Response

Identifier

See [6]

Length

The length of the EAP Response packet, 20.

Type

23

Subtype

4 for AKA-Synchronization-Failure

AT_AUTS

This attribute MUST be included in EAP-Response/AKA-Synchronization-Failure. The value field of this attribute contains the AKA AUTS parameter, 112 bits (14 bytes).

10.5. EAP-Request/AKA-Identity

The format of the EAP-Request/AKA-Identity packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Subtype     |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_PERM..._REQ | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_FULL..._REQ | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|AT_ANY_ID_REQ  | Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

1 for Request

Identifier

See [6]

Length

The length of the EAP Request packet.

Type

23

Subtype

5 for AKA-Identity

Reserved

Set to zero on sending, ignored on reception.

AT_PERMANENT_ID_REQ

The AT_PERMANENT_ID_REQ attribute is optional to include and it is included in the cases defined in Section 6. It MUST NOT be included if AT_ANY_ID_REQ or AT_FULLAUTH_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

AT_FULLAUTH_ID_REQ

The AT_FULLAUTH_ID_REQ attribute is optional to include and it is included in the cases defined in Section 5. It MUST NOT be included if AT_ANY_ID_REQ or AT_PERMANENT_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

AT_ANY_ID_REQ

The AT_ANY_ID_REQ attribute is optional and it is included in the cases defined in Section 5. It MUST NOT be included if AT_PERMANENT_ID_REQ or AT_FULLAUTH_ID_REQ is included. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

10.6. EAP-Response/AKA-Identity

The format of the EAP-Response/AKA-Identity packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |   Subtype     |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_IDENTITY   | Length        | Actual Identity Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                 Current Identity                              .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The semantics of the fields is described below:

Code

2 for Response

Identifier

See [6]

Length

The length of the EAP Response packet.

Type

23

Subtype

5 for AKA-Identity

Reserved

Set to zero on sending, ignored on reception.

AT_IDENTITY

The AT_IDENTITY attribute is optional to include and it is included in cases defined in Section 5 and 6. The value field of this attribute begins with 2-byte actual identity length, which specifies the length of the identity in bytes. This field is followed by the subscriber identity of the indicated actual length, in the same Network Access Identifier format that is used in EAP-Response/Identity, i.e. including the NAI realm portion. The identity is the permanent IMSI-based identity, a pseudonym identity or a re-authentication identity. The identity format is specified in Section 4. The identity does not include any terminating null characters. Because the length of the attribute must be a multiple of 4 bytes, the sender pads the identity with zero bytes when necessary.

10.7. EAP-Request/AKA-Reauthentication

The format of the EAP-Request/AKA-Reauthentication packet is shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Subtype    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_IV         | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                 Initialization Vector                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_ENCR_DATA  | Length        |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                    Encrypted Data                             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_MAC       | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                              MAC                              |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

   1 for Request

Identifier

   See [6].

Length

   The length of the EAP packet.

Type

   23

Subtype

   13

Reserved

   Set to zero when sending, ignored on reception.

AT_IV

The AT_IV attribute is MUST be included. See Section 9.2.

AT_ENCR_DATA

The AT_ENCR_DATA attribute MUST be included. See Section 9.2. The plaintext consists of nested attributes as described below.

AT_MAC

AT_MAC MUST be included. No message-specific data is included in the MAC calculation. See Section 9.1.

The AT_IV and AT_ENCR_DATA attributes are used for communicating encrypted attributes. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_COUNTER   | Length = 1    |           Counter             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_NONCE_S    | Length = 5    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                            NONCE_S                            |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| AT_NEXT_REAU..| Length        | Actual Re-Auth Identity Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
.                   Next Re-authentication Username             .
.                                                               .
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_COUNTER

The AT_COUNTER attribute MUST be included. The value field consists of a 16-bit unsigned integer counter value, represented in network byte order.

AT_NONCE_S

The AT_NONCE_S attribute MUST be included. The value field contains two reserved bytes followed by a random number generated by the server (16 bytes) freshly for this EAP/AKA re-authentication. The random number is used as challenge for the client and also a seed value for the new keying material. The reserved bytes are set to zero upon sending and ignored upon reception.

AT_NEXT_REAUTH_ID

The AT_NEXT_REAUTH_ID attribute is optional to include. The attribute is described in Section 10.1.

AT_PADDING

The AT_PADDING attribute is optional to include. See section 9.2

10.8. EAP-Response/AKA-Reauthentication

The format of the EAP-Response/AKA-Reauthentication packet is shown below.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Code      |  Identifier   |            Length             |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     Type      |    Subtype    |           Reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | AT_IV         | Length = 5    |           Reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                 Initialization Vector                         |
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | AT_ENCR_DATA  | Length        |           Reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 .                    Encrypted Data                             .
 .                                                               .
 |                                                               |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  AT_MAC       | Length = 5    |           Reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                                                               |
 |                                                               |
 |                             MAC                               |
 |                                                               |
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Code

2 for Response

Identifier

See [6].

Length

The length of the EAP packet.

Type

23

Subtype

13

Reserved

Set to zero when sending, ignored on reception.

AT_IV

The AT_IV attribute is MUST be included. See Section 9.2.

AT_ENCR_DATA

The AT_ENCR_DATA attribute MUST be included. See Section 9.2. The plaintext consists of nested attributes as described below.

AT_MAC

For EAP-Response/AKA-Reauthentication, the MAC code is calculated over the following data:

```
EAP packet| NONCE_S
```

The EAP packet is represented as specified in Section 9.1. It is followed by the 16-byte NONCE_S value from the client’s AT_NONCE_S attribute.

The AT_IV and AT_ENCR_DATA attributes are used for communicating encrypted attributes. The plaintext of the AT_ENCR_DATA value field consists of nested attributes, which are shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_COUNTER   | Length = 1    |           Counter             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_COUNTER...| Length = 1    |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AT_PADDING   | Length        | Padding...                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

AT_COUNTER

The AT_COUNTER attribute MUST be included. The format of this attribute is specified in Section 10.7.

AT_COUNTER_TOO_SMALL

The AT_COUNTER_TOO_SMALL attribute is optional to include, and it is included in cases specified in Section 7.

AT_PADDING

The AT_PADDING attribute is optional to include. See section 9.2

11. Unsuccessful Cases

In general, if an EAP/AKA client or server implementation detects an error in a received EAP/AKA packet, the EAP/AKA implementation silently ignores the EAP packet, does not change its state and does not send any EAP messages to its peer. Examples of such errors, specified in detail elsewhere in this document, are an invalid AT_MAC value, a mandatory attribute is missing, illegal attributes included and an unrecognized non-skippable attribute. If no valid packets are received, the authentication exchange will eventually time out.

As normally in EAP, the EAP server sends the EAP-Failure packet to the client when the authentication procedure fails on the EAP Server. In EAP/AKA, this may occur for example if the EAP server is not able to obtain authentication vectors for the subscriber or the authentication exchange times out.

12. Key Derivation

This section specifies how EAP AKA keying material is derived from the IK and CK keys. Because IK and CK are not available in the GSM mode, this key derivation specification can only be applied in the UMTS AKA mode.

EAP AKA requires two keys for its own purposes, a message authentication key K_aut and an encryption key K_encr, to be used with the AT_MAC and AT_ENCR_DATA attributes. The same K_aut and K_encr keys are used in full authentication and subsequent re-authentications. In addition, it is possible to derive additional application specific key material, such as a master key to be used with IEEE 802.11i.

Key derivation is based on the pseudo-random number generator specified in NIST Federal Information Processing Standards Publication 186-2 [14]. The pseudo-random number generator is specified in the change notice 1 (2001 October 5)of [14] (Algorithm 1). As specified in the change notice (page 74), when Algorithm 1 is used as a general-purpose random number generator, the "mod q" term in step 3.2 is omitted. The function G used in the algorithm is constructed via Secure Hash Standard as specified in Appendix 3.3 of the standard. For convenience, the pseudo-random number algorithm with the correct modification is cited in Annex B.

160-bit XKEY and XVAL values are used, so b = 160. The initial secret seed value XKEY is computed from the AKA integrity key IK and cipher key CK with the following formula:

```
XKEY = SHA1(Identity|IK|CK)
```

In the formula above, the "|" character denotes concatenation. Identity denotes the user identity string without any terminating null characters. It is the identity from the AT_IDENTITY attribute from the last EAP-Response/AKA-Identity packet, or, if AT_IDENTITY was not used, the identity from the EAP-Response/Identity packet.

The optional user input values (XSEED_j) in Step 3.1 are set to zero.

The resulting 320-bit random numbers x_0, x_1, ..., x_m-1 are concatenated and partitioned into suitable-sized chunks and used as keys in the following order: K_encr (128 bits), K_aut (128 bits), EAP application specific keys. The number of pseudo-random number generator iterations (m) depends on the amount of required keying material. The EAP application specific material immediately follows K_aut.

On re-authentication, the same pseudo-random number generator can be used to generate new application specific keys. The seed value XKEY' is calculated as follows:

```
XKEY' = SHA1(Identity|counter|NONCE_S|original XKEY)
```

In the formula above, the Identity denotes the re-authentication user identity, without any terminating null characters, from the AT_IDENTITY attribute of the EAP-Response/AKA-Identity packet, or, if EAP-Response/AKA-Identity was not used on re-authentication, the identity string from the EAP-Response/Identity packet. The counter denotes the counter value from AT_COUNTER attribute used in the EAP-

Response/AKA-Reauthentication packet. The counter is used in network byte order. NONCE_S denotes the 16-byte NONCE_S value from the AT_NONCE_S attribute used in the EAP-Request/AKA-Reauthentication packet. The original XKEY is the XKEY value from the preceding full authentication. The pseudo-random number generator is run with the new seed value XKEY’, and the resulting 320-bit random numbers x_0, x_1, ..., x_m-1 are concatenated and partitioned into suitable-sized chunks and used as new application specific keys.

For example, the EAP application specific material can be used for packet security between the client and the authenticator. Because the required keying material depends on the EAP application and the EAP key derivation standardization has not been finalized yet, rules of key derivation cannot be given here.). However, please see Annex A for a specification of how keys for IEEE 802.11 are derived.

13. Interoperability with GSM

The EAP AKA protocol is able to authenticate both UMTS and GSM users, if the subscriber's operator's network is UMTS aware. This is because the home network will be able to determine from the subscriber records whether the subscriber is equipped with a UMTS USIM or a GSM SIM. A UMTS aware home network will hence always use UMTS AKA with UMTS subscribers and GSM authentication with GSM subscribers. With GSM subscribers, the EAP AKA protocol is always used in the GSM compatible mode.

It is not possible to use a GSM AuC to authenticate UMTS subscribers. (Note that if the home network doesn't support an authentication method it should not distribute SIMs for that method.)

However, it is possible that the node actually terminating EAP and the node that stores the authentication keys (AuC) are separate, and support different authentication types. If the node terminating EAP is GSM-only but AuC is UMTS-aware, then authentication can still be achieved using the GSM compatible version of EAP AKA. This authentication will be weaker, since the GSM compatible mode does not provide for mutual authentication. Section 6.8.1.1 in [1] specifies how the GSM SRES parameter and the Kc key can be calculated on the USIM and the AuC. If a UMTS terminal does not want to accept the GSM compatible version of this protocol, then it can reject GSM authentication by silently ignoring the GSM mode EAP-Request/AKA-Challenge packet.

In conclusion, the following table shows which variant of the EAP AKA protocol should be run under different conditions:

SIM	EAP node	AuC	EAP AKA mode
GSM	(any)	(any)	GSM
UMTS	(any)	GSM	(illegal)
UMTS	GSM	GSM+UMTS	GSM
UMTS	GSM+UMTS	GSM+UMTS	UMTS

14. IANA and Protocol Numbering Considerations

The realm name "owlan.org" has been reserved for NAI realm names generated from the IMSI.

IANA has assigned the number 23 for EAP AKA authentication.

EAP AKA messages include a Subtype field. The following Subtypes are specified:

```
AKA-Challenge...................................1
AKA-Authentication-Reject.......................2
AKA-Synchronization-Failure.....................4
AKA-Identity....................................5
AKA-Reauthentication...........................13
```

The Subtype-specific data is composed of attributes, which have attribute type numbers. The following attribute types are specified:

```
AT_RAND.........................................1
AT_AUTN.........................................2
AT_RES..........................................3
AT_AUTS.........................................4
AT_PADDING......................................6
AT_PERMANENT_ID_REQ............................10
AT_MAC.........................................11
AT_ANY_ID_REQ..................................13
AT_IDENTITY....................................14
AT_FULLAUTH_ID_REQ.............................17
AT_COUNTER.....................................19
AT_COUNTER_TOO_SMALL...........................20
AT_NONCE_S.....................................21

AT_IV.........................................129
AT_ENCR_DATA..................................130
AT_NEXT_PSEUDONYM.............................132
AT_NEXT_REAUTH_ID.............................133
```

All requests for value assignment from the various number spaces described in this document require proper documentation, according to the "Specification Required" policy described in [15]. Requests must be specified in sufficient detail so that interoperability between independent implementations is possible. Possible forms of documentation include, but are not limited to, RFCs, the products of another standards body (e.g. 3GPP), or permanently and readily available vendor design notes.

15. Security Considerations

Implementations running the EAP AKA protocol will rely on the security of the AKA scheme, and the secrecy of the symmetric keys stored in the USIM and the AuC.

16. Intellectual Property Right Notices

On IPR related issues, Nokia and Ericsson refer to the their respective statements on patent licensing. Please see http://www.ietf.org/ietf/IPR/NOKIA and http://www.ietf.org/ietf/IPR/ERICSSON-General

Acknowledgements and Contributions

The authors wish to thank Rolf Blom of Ericsson, Bernard Aboba of Microsoft, Arne Norefors of Ericsson, N.Asokan of Nokia, Valtteri Niemi of Nokia, Kaisa Nyberg of Nokia, Jukka-Pekka Honkanen of Nokia and Olivier Paridaens of Alcatel for interesting discussions in this problem space.

The identiy privacy support is based on the identity privacy support of [8]. The attribute format is based on the extension format of Mobile IPv4 [16].

Authors' Addresses

Jari Arkko
Ericsson
02420 Jorvas Phone: +358 40 5079256
Finland Email: jari.arkko@ericsson.com

Henry Haverinen
Nokia Mobile Phones
P.O. Box 88
33721 Tampere Phone: +358 50 594 4899
Finland E-mail: henry.haverinen@nokia.com

Annex A. Key Derivation for IEEE 802.11

As specified in Section 12, application specific keying material can be derived with the pseudo-random function.

The key hierarchy in IEEE 802.11i currently assumes that EAP methods produce a 256-bit Pairwise Master Key (PMK). When a Pairwise Master Key is required, it is the first EAP application specific key that is derived. On full authentication, the PMK immediately follows K_aut in the key stream resulting from the key expansion scheme. On re-authentication, the PMK is the first new application specific key that is derived.

For pre 802.11i networks, the signature key used to authenticate broadcast keys in IEEE 802.1x is selected as the first 256 bits of the EAP application specific keys immediately after K_aut. (On re-authentication, the first 256 application specific key bits are used as the signature key.) The next 256 bits are used as the WEP session key. The full 256-bit key is not usually used during WEP encryption, unused bits at then end should be ignored by the implementation. When the keys are transmitted from the authenticator to the access point using the RADIUS protocol the session key is placed in an MS-MPPE-RECV-KEY attribute and the signature key is placed in an MS-MPPE-SEND-KEY attribute. These attributes are defined in RFC 2548.

Annex B. Pseudo-Random Number Generator

The "|" character denotes concatenation, and "^" denotes involution.

```
Step 1: Choose a new, secret value for the seed-key, XKEY

Step 2: In hexadecimal notation let
    t = 67452301 EFCDAB89 98BADCFE 10325476 C3D2E1F0
    This is the initial value for H0|H1|H2|H3|H4
    in the FIPS SHS [12]

Step 3: For j = 0 to m – 1 do
      3.1 XSEED_j = optional user input
      3.2 For i = 0 to 1 do
          a. XVAL = (XKEY + XSEED_j) mod 2^b
          b. w_i = G(t, XVAL)
          c. XKEY = (1 + XKEY + w_i) mod 2^b
      3.3 x_j = w_0|w_1
```

References

[1] 3GPP Technical Specification 3GPP TS 33.102 V3.6.0: "Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 1999)", 3rd Generation Partnership Project, November 2000. (NORMATIVE)

[2] GSM Technical Specification GSM 03.20 (ETS 300 534): "Digital cellular telecommunication system (Phase 2); Security related network functions", European Telecommunications Standards, Institute, August 1997. (NORMATIVE)

[3] IEEE P802.1X/D11, "Standards for Local Area and Metropolitan Area Networks: Standard for Port Based Network Access Control", March 2001. (INFORMATIVE)

[4] IEEE Draft 802.11eS/D1, "Draft Supplement to STANDARD FOR Telecommunications and Information Exchange between Systems - LAN/MAN Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Specification for Enhanced Security", March 2001. (INFORMATIVE)

[5] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999. (NORMATIVE)

[6] L. Blunk, J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998. (NORMATIVE)

[7] S. Bradner, "Key words for use in RFCs to indicate Requirement Levels", RFC 2119, March 1997. (NORMATIVE)

[8] J. Carlson, B. Aboba, H. Haverinen, "EAP SRP-SHA1 Authentication Protocol", draft-ietf-pppext-eap-srp-03.txt, July 2001 (work-in-progress). (INFORMATIVE)

[9] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC2104, February 1997. (NORMATIVE)

[10] Federal Information Processing Standard (FIPS) draft standard, "Advanced Encryption Standard (AES)", http://csrc.nist.gov/publications/drafts/dfips-AES.pdf, September 2001. (NORMATIVE)

[11] US National Bureau of Standards, "DES Modes of Operation", Federal Information Processing Standard (FIPS) Publication 81, December 1980. (NORMATIVE)

[12] GSM Technical Specification GSM 03.03 (ETS 300 523): "Digital cellular telecommunication system (Phase 2); Numbering,

addressing and identification", European Telecommunications Standards Institute, April 1997. (NORMATIVE)

[13] 3GPP Technical Specification 3GPP TS 33.105 V3.5.0: "Technical Specification Group Services and System Aspects; 3G Security; Cryptographic Algorithm Requirements (Release 1999)", 3rdGeneration Partnership Project, October 2000 (NORMATIVE)

[14] Federal Information Processing Standards (FIPS) Publication 186-2 (with change notice), "Digital Signature Standard (DSS)", National Institute of Standards and Technology, January 27, 2000, (NORMATIVE)
Available on-line at:
http://csrc.nist.gov/publications/fips/fips186-2/fips186-2-change1.pdf

[15] T. Narten, H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998. (NORMATIVE)

[16] C. Perkins (editor), "IP Mobility Support", RFC 2002, October 1996. (INFORMATIVE)