3GPP TSG SA WG3 Security — S3#21 **S3-010663**

27 - 30 November, 2001

Sophia Antipolis, France

Point-to-Point Extensions Working Group H. Haverinen
Internet Draft Nokia
November 2001

EAP SIM Authentication
draft-haverinen-pppext-eap-sim-02.txt

Status of this Memo

Abstract

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

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 authentication algorithm that runs on the SIM can be given a 128-bit random number (RAND) as a challenge. The SIM runs an operator-specific confidential 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.

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 authorize the subscriber.

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. SIM cards are smart cards distributed by GSM operators.

3. Overview

Figure 1 shows an overview of the EAP/SIM authentication procedure. This version of EAP/SIM exchange uses three roundtrips to authorize 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 clients response includes the user's International Mobile Subscriber Identity (IMSI) (Section 6).

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. The first EAP-Request/SIM packet is of the Subtype 10 (Start). Usually this packet contains no attributes. (However, see Section 4for an exception.) The client responds with the EAP-Response/SIM/Start packet, which includes the AT_NONCE_MT attribute that contains a random number NONCE_MT, picked up by the client.

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. From the triplets, the EAP server derives MAC_RAND and the keying material. Section 13 specifies how these cryptographic values are calculated.

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 for the challenges (MAC_RAND). On receipt of this message, the client runs the GSM authentication algorithm on the SIM card and calculates a copy of MAC_RAND. The client then verifies that the calculated MAC_RAND equals the received MAC_RAND. If the MAC_RAND's do not match, then the client silently ignores the EAP packet and does not send any authentication values calculated on the SIM to the network. Eventually, if another EAP-Request/SIM/Challenge packet with a valid MAC_RAND is not received, the connection establishment will time out.

Since the RAND's given to a client are accompanied with the message authentication code MAC_RAND, 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 client's response MAC_SRES (Section 13). The EAP server verifies that the MAC_SRES 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.

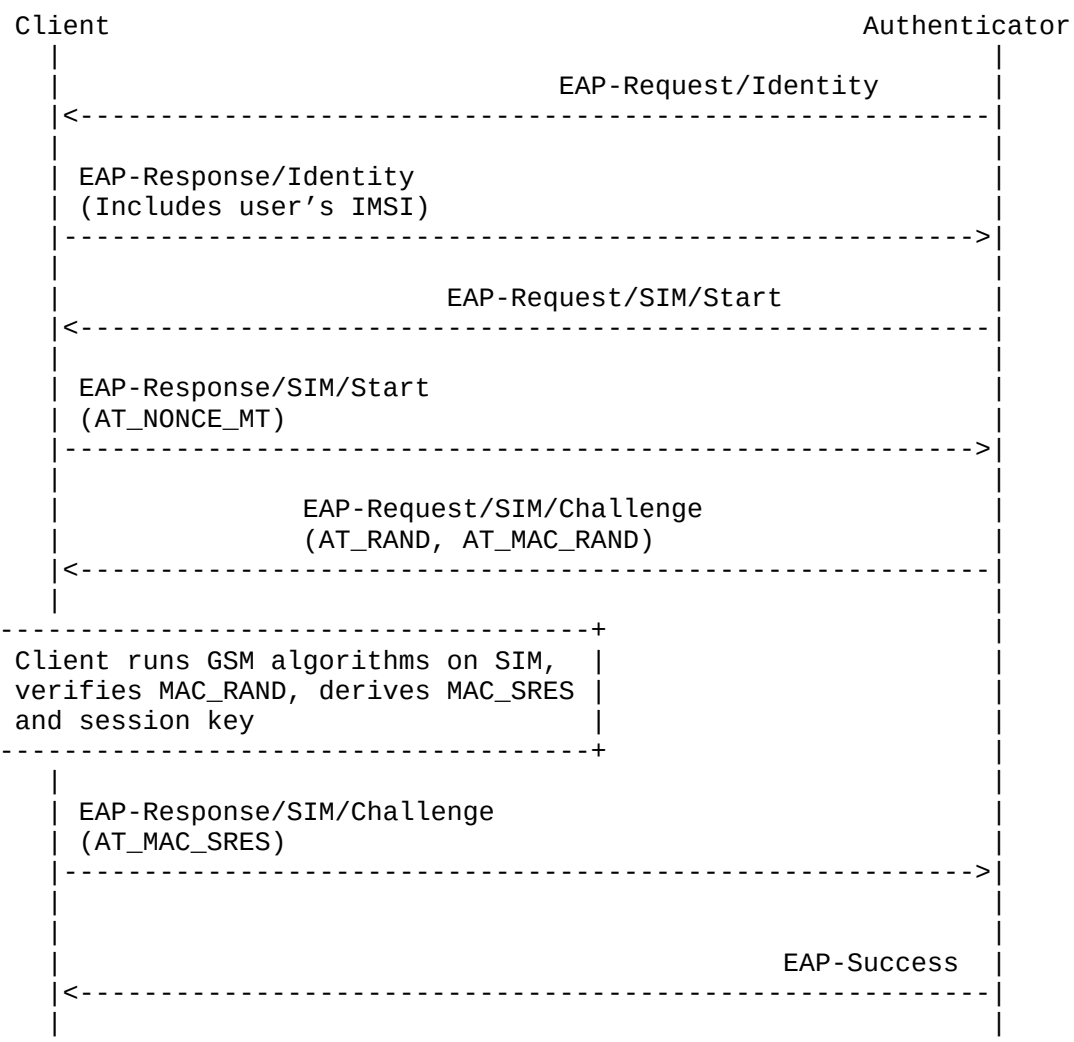


Figure 1 EAP/GSM SIM authentication procedure

4. IMSI Privacy Support

In the very first connection to an EAP server, the client always transmits the cleartext IMSI in the EAP-Response/Identity packet. In subsequent connections, the optional IMSI privacy support can be used to hide the IMSI and to make the connections unlinkable to a passive eavesdropper.

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 9. Because the IMSI privacy support is optional to implement, the client MAY ignore the AT_IV,

AT_ENCR_DATA, and AT_MAC attributes and always transmit the IMSI in the EAP-Response/Identity packet.

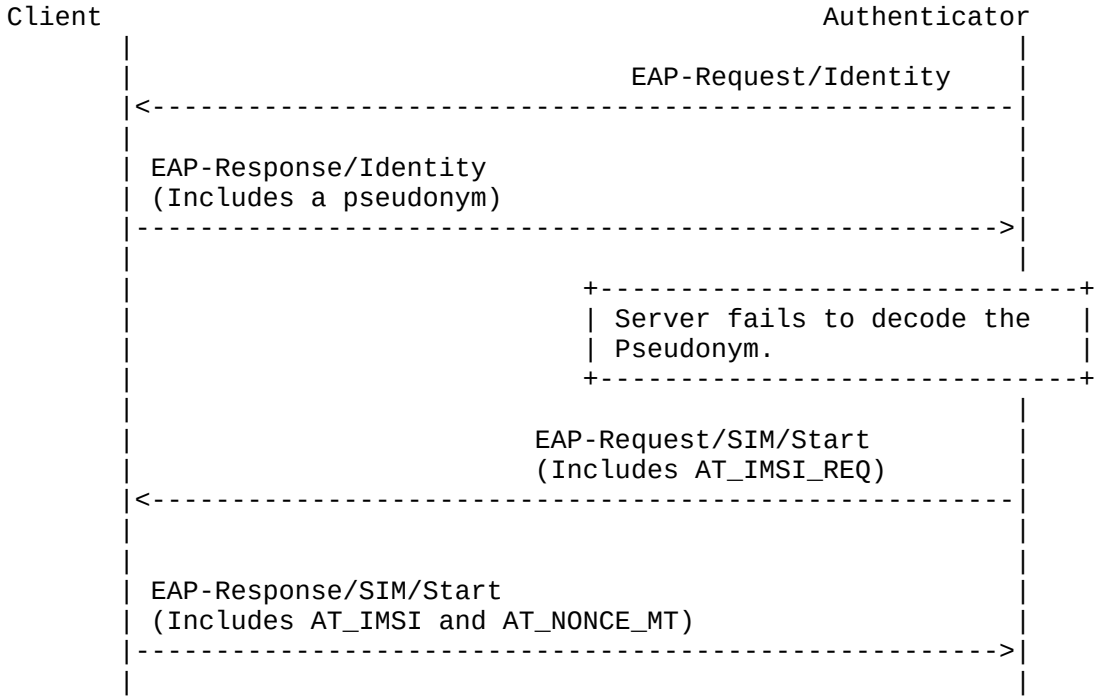
On receipt of the EAP-Request/SIM/Challenge, the client verifies the AT_MAC_RAND attribute before looking at the AT_ENCR_DATA or AT_MAC attributes. If the MAC_RAND is invalid, then the client MUST ignore the AT_IV, AT_ENCR_DATA and AT_MAC attributes. If MAC_RAND is valid, then the client MAY verify the AT_MAC attribute. If the AT_MAC attribute is valid, then the client MAY decrypt the encrypted data and use the obtained pseudonym used in the next authentication. If the MAC is invalid, then the encrypted data MUST be ignored and the whole EAP packet MAY be silently ignored.

The EAP server produces pseudonyms in an implementation-dependent manner. Please see [4] for examples on how to produce pseudonyms. The pseudonyms need to be reversible to the IMSI only on the EAP server. Regardless of construction method, the pseudonym MUST conform to the grammar specified for the username portion of an NAI.

On the next connection to 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 MUST use the same realm portion that it used in the connection when it received the pseudonym.

If the EAP server successfully decodes the pseudonym to a known client identity (IMSI), the authentication proceeds with the EAP-Request/SIM/Start message as usual.

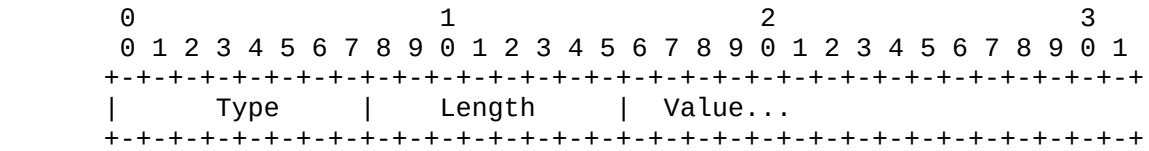
If the EAP server fails to decode the pseudonym to a known identity, then the EAP server requests the regular IMSI (non-pseudonym identity) by including the AT_IMSI_REQ attribute (Section 7) in the EAP-Request/SIM/Start message. The value field of the AT_IMSI_REQ does not contain any data but the attribute is included to request the client to include the AT_IMSI attribute (Section 8) in the EAP-Response/SIM/Start message. The AT_IMSI attribute contains the client’s IMSI in the clear. This case is illustrated in the figure below.



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

5. 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.



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 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.

6. 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) [5]. The IMSI is composed of a three digit Mobile Country Code (MCC), a two 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) [6]. When used in a roaming environment, the NAI is composed of a username and a realm, separated with "@". 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. Operators SHOULD reserve the realm portion of

NAI for EAP/SIM users exclusively, so that exactly the same realm is not used with other authentication methods. This convention makes it easy to recognize that the NAI identifies a GSM subscriber of this operator, which may be useful when configuring the routing rules in the visited AAA networks.

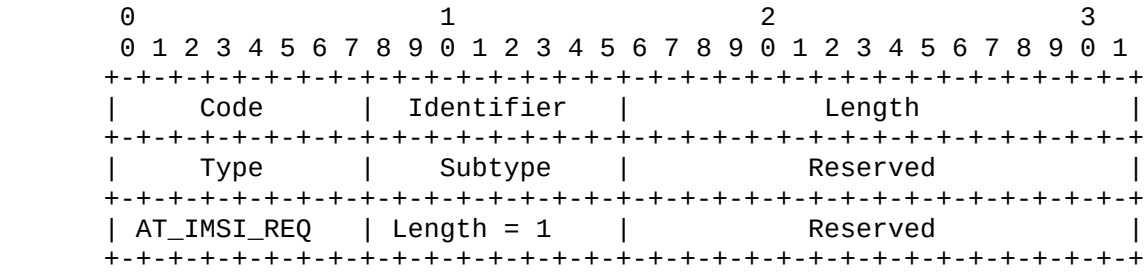
When the optional IMSI privacy support is not used, the client transmits the user's IMSI as a NAI in the EAP Response/Identity packet. The NAI is of the format "1imsi@realm". In other words, the first character is the digit one (ASCII value 0x31), followed by the IMSI, followed by the @ character and the realm. The IMSI is an ASCII string that consists of not more than 15 decimal digits (ASCII values between 0x30 and 0x39) as specified in [5].

When the optional IMSI privacy support is used, the client MAY use the pseudonym received as part of the previous authentication sequence as the user name portion of the NAI, as specified in Section 4.

The AAA network routes the AAA request to the correct AAA server using the realm part of the NAI.

7. EAP-Request/SIM/Start

The first SIM specific EAP Request is of subtype Start. The format of the EAP Request/SIM/Start packet is shown below.



Code

1 for Request

Identifier

See [1].

Length

The length of the EAP packet.

Type

18

Subtype

10

Reserved

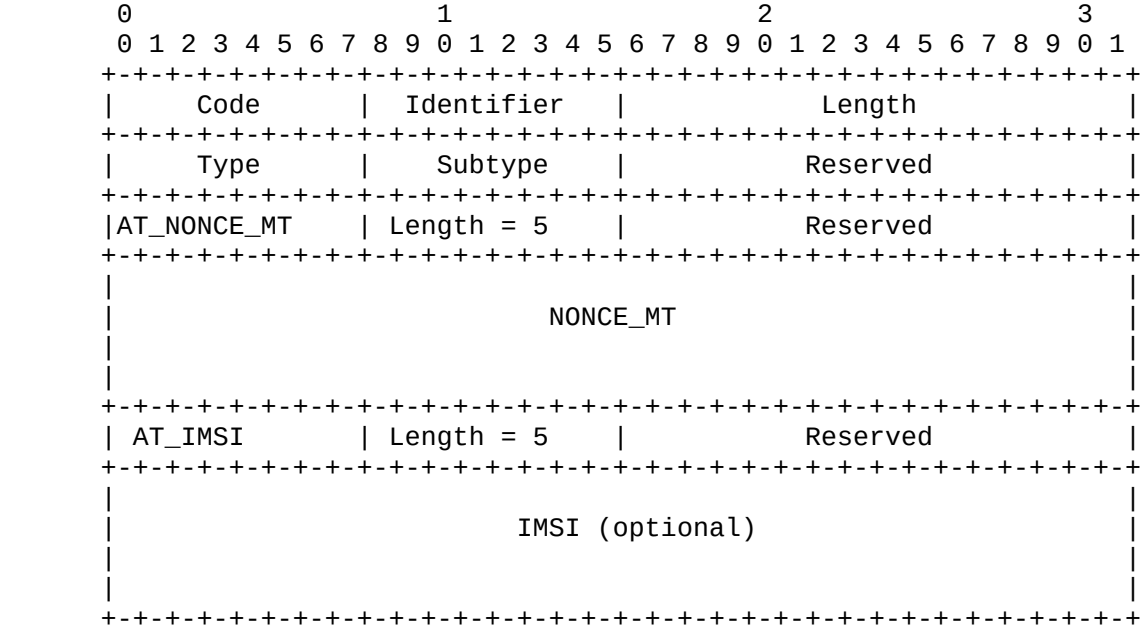
Set to zero on sending, ignored on reception

AT_IMSI_REQ

The AT_IMSI_REQ attribute is optional and it is included in the cases defined in Section 4. The value field only contains two reserved bytes, which are set to zero on sending and ignored on reception.

8. EAP-Response/SIM/Start

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



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

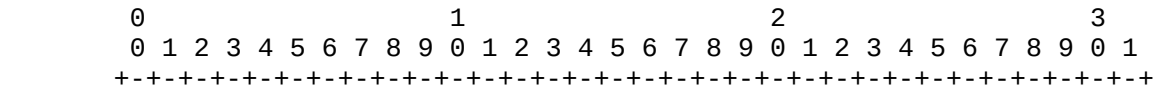
The AT_NONCE_MT attribute MUST be included. The value field contains two reserved bytes followed by a random number generated by the client (16 bytes), which is used as a seed value for the new key. The reserved bytes are set to zero upon sending and ignored upon reception.

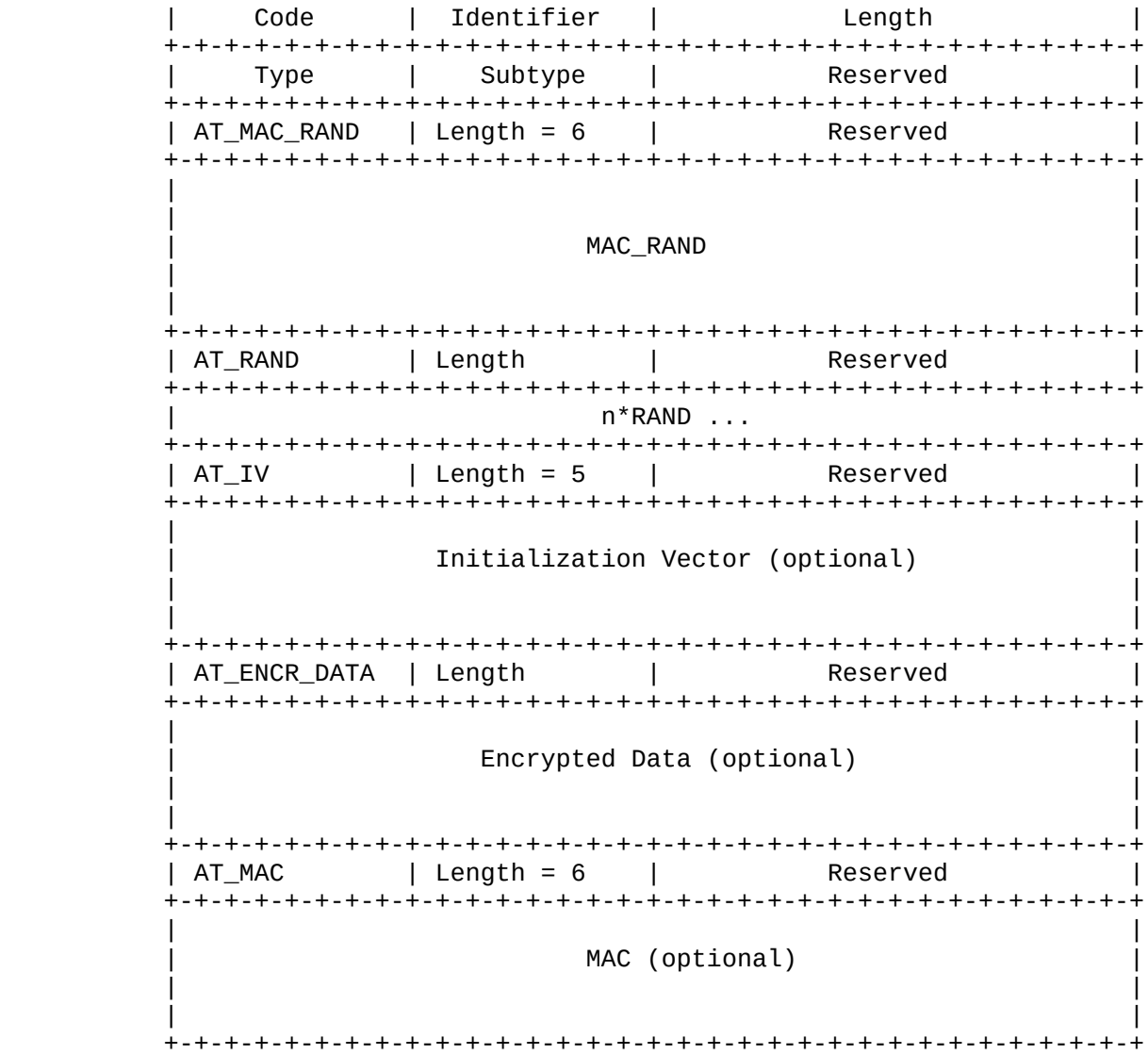
AT_IMSI

The AT_IMSI attribute is optional and it is included in cases defined in Section 4. The value field contains two reserved bytes followed by the IMSI, represented as an ASCII string that consists of not more than 15 decimal digits (ASCII values between 0x30 and 0x39) [5]. The reserved bytes are set to zero on sending and ignored on reception. The IMSI characters are followed by one or more "F" characters (ASCII value 0x46). They are included to make the length of the value field 16 bytes.

9. EAP-Request/SIM/Challenge

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





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_MAC_RAND

The AT_MAC_RAND attribute MUST be included. The value field of this attribute contains two reserved bytes followed by a message authentication code MAC_RAND (Section 13), 20 bytes. The reserved bytes are set to zero upon sending and ignored upon 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 SHOULD be at least two. The client MAY silently ignore the EAP-Request/SIM/Challenge message, if the number of RAND challenges is not in accordance with its local policy.

AT_IV

The value field 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. This 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.

AT_ENCR_DATA

The AT_ENCR_DATA attribute is optional. The value field of this 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 derivation of the encryption key K_encr used for this attribute is specified in Section 13. The plaintext consists of nested attributes as described below.

AT_MAC

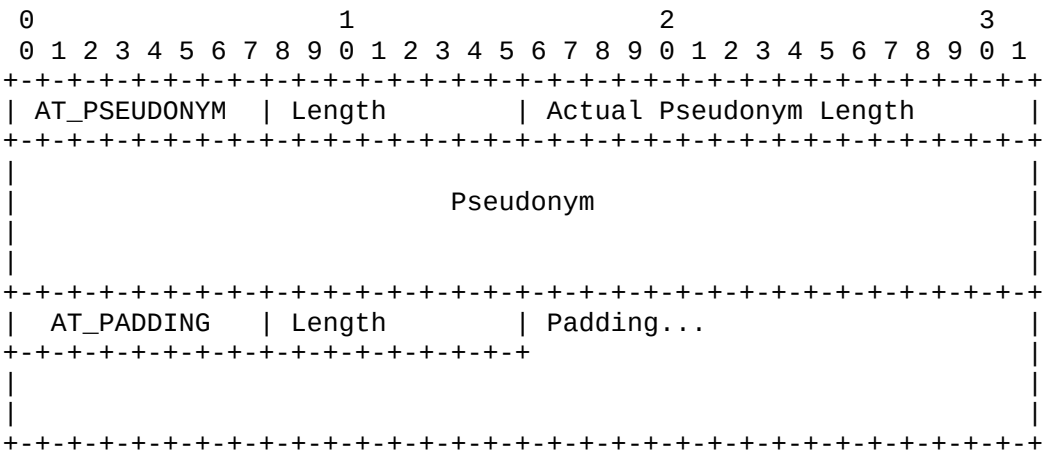
This attribute is optional, but it MUST be included whenever the AT_ENCR_DATA attribute is included. Messages that do not meet this condition MUST be silently discarded.

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 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 MAC algorithm is HMAC-SHA1 [9] keyed hash value, so the length of the MAC is 20 bytes.

The derivation of the integrity protection key (K_int) used in the calculation of the MAC is specified in Section 13.

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



AT_PSEUDONYM

The AT_PSEUDONYM 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 4. 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_PADDING

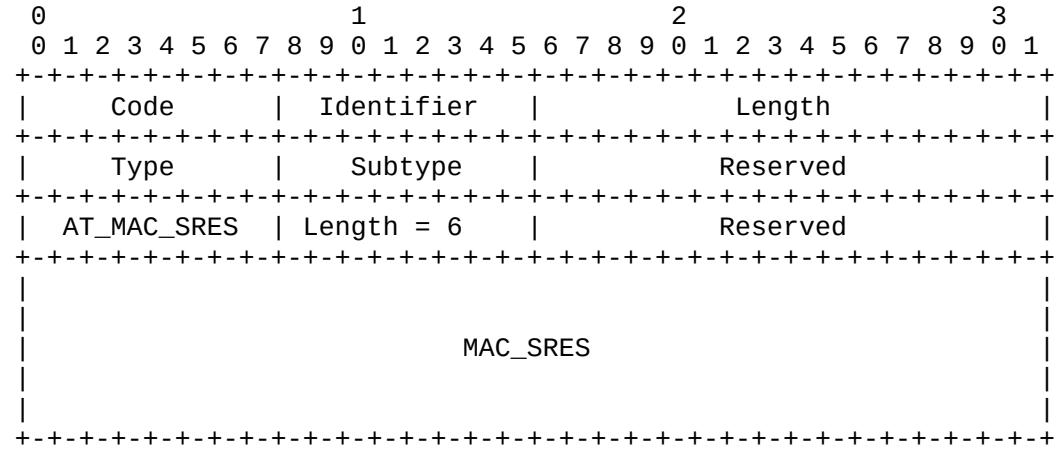
The encryption algorithm requires the length of the plaintext to be a multiple of 16 bytes. The sender may need to include the AT_PADDING attribute as the last attribute within AT_ENCR_DATA. The AT_PADDING attribute is not included if the total length of

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.

10. EAP-Response/SIM/Challenge

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

EAP-Response/SIM/Challenge MAY include the AT_MAC attribute to integrity protect the EAP packet. 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/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.



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_SRES

The AT_MAC_SRES attribute MUST be included. The value field of this attribute contains two reserved bytes followec by the MAC_SRES response calculated by the client (Section 13), 20 bytes. The reserved bytes are set to zero upon sending and ignored upon reception.

11. Unsuccessful Cases

As normally in EAP, the client is sent the EAP-Failure packet 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 EAP server receives an incorrect MAC_SRES.

In general, if an error occurs on the client while processing a received EAP-Request packet, the client silently ignores the EAP packet and does not send any EAP messages to the network. Examples of such errors, specified in detail elsewhere in this document, are an invalid MAC_RAND value, insufficient number of RAND challenges included in AT_RAND, and an unrecognized non-skippable attribute.

As specified in [1], the EAP client must respond with EAP-Response/Nak when it receives an EAP Request of an undesired or unrecognized authentication type.

12. Localization of EAP/SIM Specific Notifications

The EAP-Request/Notification, specified in [1], can be used to convey a displayable message from the authenticator to the client. In EAP/SIM, the displayable message of EAP-Request/Notification MAY begin with one of the following notification numbers, which can be

used to localize the notification message in the client device. The notification number is usually followed by a textual message, which the client MAY ignore if it uses the notification number to display the message.

1024

Visited network does not have a roaming agreement with user’s

home operator

1026

User’s calls are barred

1031

User has not subscribed to the requested service

13. Calculation of Cryptographic Values

This section specifies how keying material is generated and how the message authentication codes MAC_RAND and MAC_SRES are calculated.

When calculating these values, the IMSI is packed into 8 bytes. The most significant nibble of the first byte is the first digit in the IMSI, the least significant nibble the second digit in IMSI etc. The least significant nibble of the 8th byte is ’F’ as the IMSI typically is 15 digits. Unused nibbles are filled with ’F’ in case the IMSI is less than 15 digits. For example, the IMSI 244070100000112 is coded as follows: the first byte is 0x24, the second byte is 0x40, ..., and the eighth byte is 0x2F.

In the formulae, the notation prf(key, msg) denotes the keyed pseudo-random function used to generate a deterministic output that appears pseudo-random. The prf() is used both for key derivations and for authentication (i.e. as a keyed MAC). The notation hash(msg) denotes a one-way hash function of a message. In this version of EAP/SIM, the prf () is HMAC-SHA1 [9], and the hash() is SHA-1 [10].

First, a master key K_master is calculated as follows:

K_master

hash (n*Kc| NONCE_MT)

The master key is only used to derive other keying material with the following key expansion scheme, which is similar to the keying material derivation of Internet Key Exchange [11]. The following formulae are used:

Key_block_0

prf(K_master, Client Identity | 0)

Key_block_i, where i = 1, 2...

prf(K_master, Key_block_i-1 | Client Identity | i)

The values of 0, 1, and 2 etc. above are represented by a single octet. The client identity represents the string used as the client identity in the EAP-Response/Identity message (Section 6). If a pseudonym was used in the EAP-Response/Identity message, it is used in this formula regardless of whether the EAP server recognized the

pseudonym.

The resulting material Key_block_0, Key_block_1, ... is then partitioned into suitable-sized chunks and used as keys in the following order:

K_randsres (20 octets),
K_encr (16 octets),
K_int (20 octets),
EAP application specific keys

K_randsres is used in the calculation of MAC_RAND and MAC_SRES as follows:

MAC_RAND

prf (K_randsres, n*RAND | IMSI | Message Subtype)

MAC_SRES

prf (K_randsres, n*SRES | IMSI | Message Subtype)

Message subtype above contains the contents of the Subtype field of the EAP/SIM message (one octet), in which the MAC_RAND or MAC_SRES parameter is included.

The K_encr and K_int keys are the encryption and integrity protection keys required for AT_ENCR_DATA and AT_MAC attributes.

The keying material following K_int can be used as required by the EAP application. Even if K_encr or K_int were not used in the particular authentication sequence, they are derived and the EAP application specific material begins after K_int.

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, exact rules of key derivation cannot be given here. As a guideline, which can be used applicable, the EAP application specific keys resulting from the key expansion scheme is used in the following order:

any master session keys required,
any encryption keys required,

any integrity protection keys required,
any initialization vectors required

If separate keys or IV's are required for each direction, then the downlink material (to protect traffic to user) is taken before the uplink material (to protect traffic from user).

When generating K_master, 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 manages to learn one of the EAP/SIM session keys, it doesn'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 15.)

14. IANA Considerations

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
```

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

```
AT_RAND...................................1
AT_IMSI...................................5
AT_PADDING................................6
AT_NONCE_MT...............................7
AT_MAC_RAND...............................8
AT_MAC_SRES...............................9
AT_IMSI_REQ..............................10

AT_IV...................................129
AT_ENCR_DATA............................130
AT_MAC..................................131
AT_PSEUDONYM............................132
```

15. 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.

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. EAP/SIM does not protect the privacy of the IMSI against active attacks. An active attacker that impersonates the network can easily learn the subscriber's IMSI. 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 MAC_RAND value. To

calculate MAC_RAND, 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 a GSM triplet 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. The corresponding Kc key could be obtained for example by cryptanalysing encrypted GSM traffic. (Of course, this can be used to attack EAP/SIM only if the same SIM card is used both for GSM network access and for EAP/SIM.) For these reasons, network authentication of EAP/SIM SHOULD NOT be used exclusively if strong network authentication is a concern.

There is no known way to obtain complete GSM triplets by mounting an attack against EAP/SIM. A passive eavesdropper can learn n*RAND, MAC_RAND and MAC_SRES, 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 MAC_RAND values from the EAP server for any given subscriber identity. However, calculating the Kc and SRES values from MAC_RAND and MAC_SRES would require the attacker to reverse the keyed message authentication code function HMAC-SHA1.

EAP/SIM combines several GSM triplets in order to generate a stronger session key and stronger MAC_RAND and MAC_SRES 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.

An EAP/SIM implementation SHOULD use a good source of randomness to generate the random numbers required in the protocol. Please see [12] for more information on generating random numbers for security applications.

16. 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.

17. Acknowledgements

The author thanks Juha Ala-Laurila, N. Asokan, Simon Blake-Wilson, Jan-Erik Ekberg, Patrik Flykt, Jukka-Pekka Honkanen, Antti Kuikka, Jukka Latva, Lassi Lehtinen, Jyri Rinnemaa, Timo Takam„ki and Raimo Vuonnala for theirs contributions and critiques.

The IMSI privacy support is based on the identity privacy support of [4]. The attribute format is based on the extension format of Mobile IPv4 [13].

This protocol has been partly developed in parallel with EAP AKA [14], 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

[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

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

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

[5] 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

[6] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486, January 1999.

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

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

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

[10] 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.

[11] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC

2409, November 1998

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

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

[14] J. Arkko, H. Haverinen, "EAP AKA Authentication", draft-arkko-pppext-eap-aka-01.txt, November 2001 (work in progress)

Author's Address

Henry Haverinen
Nokia Mobile Phones
P.O. Box 88
FIN-33721 Tampere
Finland
E-mail: henry.haverinen@nokia.com
Phone: +358 50 594 4899
Fax: +358 3 318 3690