3GPP TSG SA WG3 Security — S3#32 **S3-040120**
09 - 13 February 2004
Edinburgh, Scotland, UK

EP SCP-WG Meeting #16 **Tdoc SCP-040101**
Barcelona, Spain, 04-06 February 2004.

Title: L.S. on ETSI TS 102.310 for information
Response to:

Source: EP-SCP
To: 3GPP-T3, 3GPP-SA1, 3GPP-SA3
Cc:

Contact Person:
Name: Patrice Beaudou (Axalto)
Tel. Number:
E-mail Address: PBeaudou@axalto.com

Attachments: SCP-040078

---

1. Overall Description:

EP-SCP would like to inform 3GPP-T3, 3GPP-SA1, and 3GPP-SA3 on the availability of the attached document, ETSI TS 102 310, Extensible Authentication Protocol support in the UICC.

This TS defines features that shall be provided by the UICC to support EAP authentication capabilities. The goal of these new features is to enable the UICC to provide support of different EAP methods, ensuring interoperability between the UICC and any terminal, independently of their respective manufacturers.

EP-SCP would like to thank 3GPP-T3, 3GPP-SA1, and 3GPP-SA3 for their consideration of this TS and invite comments, if possible, prior to the WG1 meeting (dates below).

2. Actions:

To 3GPP-T3, 3GPP-SA1, 3GPP-SA3:

ACTION:

- EP SCP kindly asks 3GPP-T3, 3GPP-SA1, 3GPP-SA3 to provide comments if required.

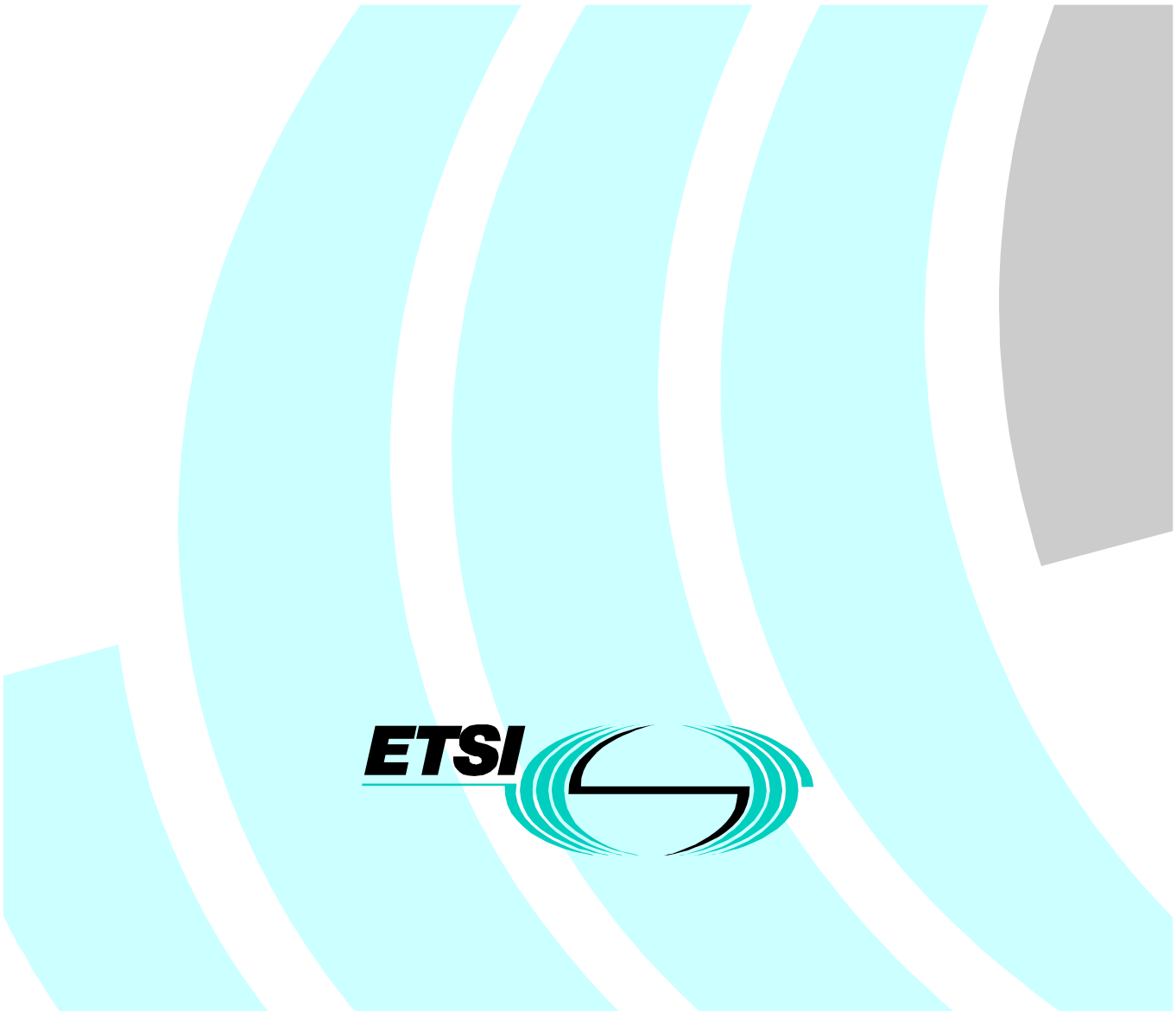
3. Date of Next EP SCP Meetings:

EP SCP WG1#10	29 March – 1 April 2004	Sophia Antipolis, France
EP SCP Meeting #17	05-07 May 2004	Sophia Antipolis, France

ETSI TS 102.310 V 1.0.0 (2004-02)

Technical Specification

Smart Cards;
Extensible Authentication Protocol support in the UICC;
(Release 6)



Reference

Keywords
EAP, UICC

ETSI

Postal address
F-06921 Sophia Antipolis Cedex - FRANCE

Office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16
Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Internet
secretariat@etsi.fr
Individual copies of this ETSI deliverable
can be downloaded from
http://www.etsi.org
If you find errors in the present document, send your
comment to: editor@etsi.fr

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by the ETSI Project Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within ETSI SCP and may change following formal ETSI SCP approval. Should ETSI SCP modify the contents of the present document it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x indicates the release (3 indicates Release 1999 and 4 indicates the subsequent release (called "Release 4").

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification.

1 Scope

The present document defines additional features that shall be provided by the UICC to support EAP authentication capabilities.

The goal of these new features is to adapt the UICC to provide support of different EAP methods, ensuring interoperability between the UICC and any terminal independently of their respective manufacturers.

The present document defines:

- The architectural framework
- The additional commands required;

2 References

The following documents contain provisions that, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

[1] Extensible Authentication Protocol (EAP) (IETF <draft-ietf-eap-rfc2284bis-06.txt>)

[2] IETF RFC 2284 "PPP Extensible Authentication Protocol (EAP) (http://www.ietf.org/rfc/rfc2284.txt)

[3] EAP-support in smartcards (http://www.ietf.org/internet-drafts/draft-urien-eap-smartcard-03.txt)

[4] ETSI TS 102 221 "Smart cards; UICC-Terminal interface; Physical and logical characteristics

[5] draft-arkko-pppext-eap-aka-11, "EAP AKA Authentication". http://www.ietf.org/internet-drafts/draft-arkko-pppext-eap-aka-11.txt

[6] draft-haverinen-pppext-eap-sim-12, "EAP SIM Authentication".http://www.ietf.org/internet-drafts/draft-haverinen-pppext-eap-sim-12.txt

[7] IETF RFC 2284 "PPP EAP TLS Authentication Protocol"

[8] 3GPP TS 51.011: "Specification of the Subscriber Identity Module – Mobile Equipment (SIM – ME) interface".

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

3 Definitions and Acronyms

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

Authenticator: The end of the EAP link initiating EAP authentication

Peer or Supplicant: The end of the EAP Link that responds to the authenticator.

3.2 Acronyms

4 Introduction

The Extensible Authentication Protocol (EAP) [1] is a general authentication framework, which supports multiple authentication methods. EAP typically may run directly over data link layers such as PPP or IEEE 802.

As described in EAP [1], EAP implementations consist of three main components:

-A **lower layer** that is responsible for transmitting and receiving EAP frames between the peer and the authenticator. (EAP has been run over a variety of lower layers including PPP; wired IEEE 802 LANs [IEEE-802.1X]; IEEE 802.11 wireless LANs [IEEE-802.11]; UDP (L2TP [RFC2661] and ISAKMP [PIC]); and TCP [PIC]).

-An **EAP layer** that receives and transmits EAP packets via the lower layer, implements duplicate detection and retransmission, and delivers and receives EAP messages to and from EAP methods.

-EAP methods that implement the authentication algorithms and receive/transmit EAP messages via the EAP layer.

The UICC offers suitable possibilities for the implementation of some of these EAP methods in the peer side, since it provides the required protection of credentials and authentication algorithms. This is even more important when the following conditions apply:

-The authentication methods require the usage of credentials that are stored in the UICC.

-For security reasons, these credentials shall not be revealed in clear in an unprotected peer environment (e.g. a laptop or mobile terminal).

The IETF draft "EAP support in smartcards" [3] specifies some principles on how it is possible to carry out a particular EAP method inside a smart card.

The present document defines how these principles shall be implemented in the UICC in order to enable that UICC applications may support one or more of these EAP methods.

Examples of EAP methods that can be implemented in the UICC are EAP SIM [6], EAP AKA [5] and EAP TLS [7].

Note: This document refers to the EAP-bis draft [1], which is intended to make obsolete the RFC 2284 [2] once approved. However, all statements contained in this document may be also applied for devices compliant with RFC 2284 [2].

5 Architecture

5.1 Architectural Principles

The following architectural principles are applied:

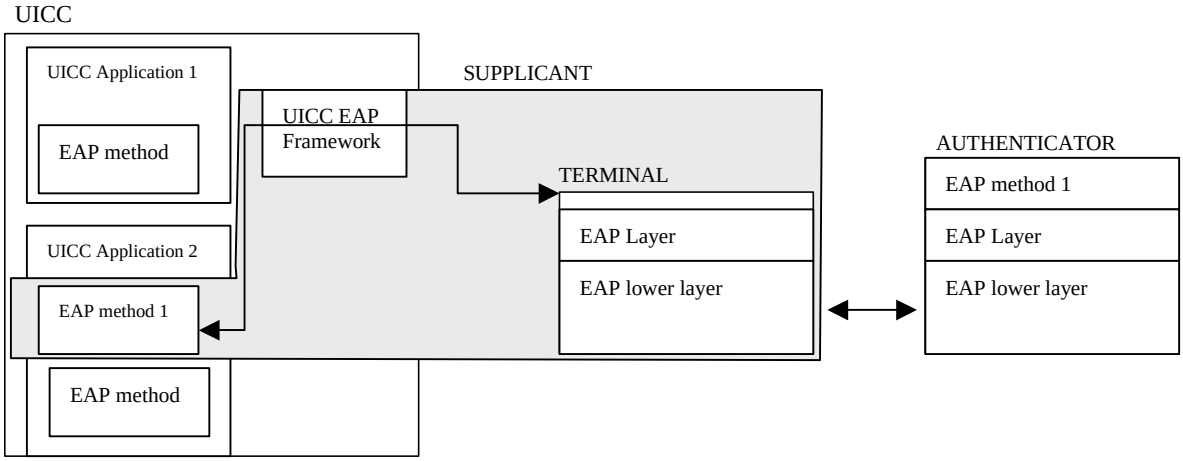
-The authenticator is able to perform an EAP authentication process (using an specific EAP method) with a UICC application implementing this method. That means that the authentication is performed end-to-end between the authenticator and the UICC application.

-The peer is composed of several components:

-**The UICC EAP Framework** provides information to the terminal about the existing UICC applications that provide UICC EAP clients.

-A **UICC application** provides one or more UICC EAP clients.

-A UICC EAP client implements one specific EAP method.



-

Figure 1: EAP architecture when supplicant is split between a UICC and a terminal.

5.2 EAP clients discovery

When a UICC application implements one or more EAP clients, its corresponding record in EF_{DIR} shall contain the following EAP related Data Objects.

-Application EAP support types list: Defining the EAP methods supported by the corresponding UICC application

-Application EAP Dedicated File list: Defining the list of Dedicated Files associated to a particular supported EAP method. Each of this DF are hereafter referred as DF_{EAP}

-Application EAP Label: Defining a user readable label defining the EAP clients

Table 1: Coding of EAP related DOs

Length	Description	Status
1	Discretionary template tag = '73'	M
1	Length of the discretionary template	M
1	Application Type Tag = ‘80’ Editor’ s Note: To be Reserved in 102.221	M
1	Application Type length	M
1	Application Type Value (EAP=’81’) Editor’ s Note: To be Reserved in 102.221	M
1	Application specific data content tag (=”A0”) Editor’ s Note: To be Reserved in 102.221	M
1	Application specific data content length	M
1	Application EAP supported types list tag = '01'	M
1	Length of the Application EAP supported types list	M
A	Application EAP supported types list	M
1	Application EAP Dedicated file list tag = '02'	M
1	Length of Application EAP Dedicated file list	M
B	Application EAP Dedicated File list	M
1	Application EAP Label tag = '03'	M
1	Length of the Application EAP Label	M
C	Application EAP Label	M

Coding:

-Application EAP support types list:

Contain a list of supported EAP type (as defined in [1]) each of them coded in one byte.

Example: An UICC application supporting EAP-SIM and EAP-TLS provides the following "Application EAP supported types list":

' 120D'corresponding to EAP-SIM (Type=18) and EAP-TLS (Type=13)

-Application EAP Dedicated Files list:

Contain a list of file identifiers of each DF_{EAP} associated to a particular supported EAP type. Each of them coded in two bytes.

Example: Using the previous example, A DF '6D34' for EAP-SIM and the same for EAP-TLS will result in the following EAP Dedicated Files list:

'6D346D34'

-Application EAP label:

The application label is a DO that contains a string of bytes provided by the application provider to be shown to the user for information.

5.3 EAP-capable-application selection

The terminal shall use the information in EF_{DIR} file if available to present the list of EAP-capable applications to the user or to any application that may request an EAP authentication.

The terminal shall then select the corresponding EAP-capable-application to start an EAP authentication. Once selected, all EAP-Client state machines of the application are reset.

5.4 Key derivation

It is possible for many EAP methods to derive key material after successful authentications. These keys may be used for subsequent processes (e.g. for WEP encryption in 802.11).

Keys derived from an authentication shall be retrieved by the terminal by inspecting the mandatory file $EF_{EAPKEYS}$

5.5 Authentication Status

The terminal may retrieve the authentication status of the EAP client in the selected UICC application by inspecting the mandatory file $EF_{EAPSTATUS}$

6 EAP related Commands

The following sections specify the additional commands needed to implement the EAP framework in the UICC. These commands are described in [3]

6.1 EAP Authenticate

6.1.1 Command description

The function is used to transfer the EAP packets from the terminal to the selected UICC EAP client (i.e. EAP client in the selected UICC application that corresponds to the given EAP type)

The UICC EAP client shall provide a response EAP packet or a warning status word according to the authentication method being used.

The UICC EAP client shall maintain the state machine of the authentication process as described for the particular EAP method used.

The function is related to a particular UICC application supporting EAP and shall not be executable unless this application has been selected and activated, and the current directory is a DF_{EAP} related to a specific EAP method.

Each UICC application implementing a UICC EAP client may require different security conditions to execute this command (e.g. user PIN verification).

The format of the EAP packet is defined by the application implementing the EAP client and shall respect the conventions corresponding for the EAP method.

Input:

- EAP Packet
- EAP type

Output

- Either none (i.e. if authentication successful: EAP success packet received)

or

- EAP Response Packet

6.1.1.2 Command parameters and data

Code	Value
CLA	As specified in ETSI SCP 102 221 [4]
INS	'88'
P1	EAP type (coded as defined in EAP related Data Objects)
P2	See table 9.1
Lc	Length of subsequent EAP packet
Data	See below
Le	Length of the response data

NOTE: Parameter P1 indicates the targeted EAP client in the selected application.

Table 6.1: Coding of P2

b8	b7	b6	b5	b4	b3	b2	B1	Meaning
1	-	-	-	-	-	-	-	Specific reference data (DF $_{EAP}$ application dependent KEY)
-	X	X	-	-	-	-	-	'00' (other values are RFU)
-	-	-	X	X	X	X	X	Reference data number ('01' to '1F')

Command data:

Byte(s)	Description	Length
1 - Lc	EAP Packet (see note)	Lc
NOTE: EAP packet coded as defined for the method of EAP used as defined in [1]		

Response data :

Byte(s)	Description	Length
1 - Le	EAP Packet (see note)	Le
NOTE: EAP packet coded as defined for the method of EAP used as defined in [1]		

6.2 Specific status conditions returned

This clause specifies the coding of the specific status words SW1 and SW2.

6.2.1 Status words

The following table shows the meaning of possible status conditions returned.

Table 6.2: Status byte coding - warnings

SW1	SW2	Description
'62'	'00'	- No information given, state of non volatile memory unchanged (EAP Packet silently ignored)

Table 6.3: Status byte coding - application errors

SW1	SW2	Description
'98'	'62'	- Authentication error (EAP Failure Packet received)

7 EAP Files

This clause describes the files present in an application supporting an EAP type. The following files are situated under the corresponding DF_{EAP} of a particular UICC application:

7.1 $EF_{EAPKEYS}$ (EAP derived keys)

This EF contains the key material derived after a successful EAP authentication.

Structure of $EF_{EAPKEYS}$

Identifier: '4F01'	Structure: transparent	Conditional (see Note)
File size: n	Update activity: high	
Access Conditions: READ PIN UPDATE ADM/NEVER DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1	1st Key Tag	O	1 bytes
2	1st Key Length	O	1 bytes
3-L1+3	1st Key Value	O	L1 bytes
	...		
	Kst Key Tag	O	1 bytes
	Kst Key Length	O	1 bytes
	Kst Key Value	O	LK bytes

Note: The presence of this file depends on the supported EAP methods.

- Key Tag

  Contents:

  -Identifier of the derived key

  Coding:

  Editors Note: key tags reserved are FFS.

- Key Length

  Contents:

  -Length of the derived key

- Key Value

  Contents:

  -Derived key

7.2 $EF_{EAPSTATUS}$ (EAP Authentication STATUS)

This EF contains the authentication status corresponding to each of the EAP clients supported by the application.

Structure of $EF_{EAPSTATUS}$

Identifier: '4F02'	Structure: transparent	Mandatory
File size: n	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	ADM/NEVER
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	EAP type	M	1 bytes
2	Authentication Status	M	1 bytes
	...		
K	EAP type (K supported type)	O	1 bytes
K+1	Authentication Status	O	1 bytes

- EAP Type

Contents:

-Type of EAP supported by the application

Coding:

-As defined for EAP related DOs

- Authentication Status

Contents:

-Status of the corresponding EAP authentication

Coding:

-Authentication Status coded in one byte as below:

Value	Meaning
'00'	No authentication started
'01'	Authenticating
'02'	Authenticated
'03'	Held (Authentication failure)

7.3 EF_{PUId} (Permanent User Identity)

This EF contains the permanent user identity. Permanent User identity may be used as the username part of the Network Access Identifier (NAI)

This File is not mandatory if the Permanent user identity is derived by other means. (e.g. as defined in EAP SIM [6] or EAP-AKA [5]).

Structure of EF_{PUid}

Identifier: '4F03'	Structure: transparent	Optional
File size: n (where n ≥10 bytes)	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1 to n	Permanent user identity	M	n bytes

Permanent user identity

Contents:

-user identity to be used as the username part of the NAI

Coding:

-Binary. Unused bytes shall be set to ‘FF’ and shall not be considered as a part of the value.

7.4 EF_{PsL} (Pseudonym List)

This EF contains a list of temporary user identifiers (pseudonyms) for subscriber identification. Pseudonyms may be provided as part of a previous authentication sequence. This may be used as the username part of the Network Access Identifier (NAI).

This File is not mandatory if pseudonyms are not managed by the application or they are derived by other means.

Structure of EF_{PsL}

Identifier: '4F04'	Structure: Cyclic	Optional
Record length: n	Update activity: high	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to n	Pseudonym	M	n bytes

- Pseudonym.

Contents:

-Pseudonym to be used as the username part of the NAI

Coding:

-Binary Unused bytes shall be set to ‘FF’ and shall not be considered as a part of the value.

7.5 EF_{RIL} (Re-authentication Identity List)

This EF contains a list of re-authentication identities for subscriber identification. Re-authentication identities may be provided as part of a previous authentication sequence. This may be used as the username part of the Network Access Identifier (NAI).

This file is not mandatory if re-authentication identities are not managed by the application or they are provided by other means.

Structure of EF_{RIL}

Identifier: '4F05'	Structure: Cyclic	Optional
Record length: n (where n ≥10 bytes)	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	PIN
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1 to n	Re-authentication Identity	M	n bytes

- Re-authentication Identity.

Contents:

- Re-authentication identity to be used as the username part of the NAI

Coding:

- Binary Unused bytes shall be set to ‘FF’ and shall not be considered as a part of the value.

Annex A History (Informative)

Change History

This annex lists all change requests approved for the present document by ETSI SCP.

SCP#	SCP tdoc	WG tdoc	VERS	CR	RV	PH	CAT	SUBJECT	Resulting Version

Document History

Document history		
V0.0.1	September 2003	Initial draft.
V0.0.2	November 2003	Second Draft (Editorials + added 7.4, 7.5) Removal of some EAP commands Figure 1 added
V0.0.3	January 2004	Update related to the GSM session Supplicant software replaced by terminal
V0.0.4	January 2004	Authenticate Command Files for Key Derivation and Status
V0.0.5	January 2004	Definition of DF_{EAP} New EF DIR DO structure Defined Files under DF_{EAP}
V1.0.0	February 2004	For information