International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1034

(02/2011)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Information and network security – Network security

Guidelines on extensible authentication protocol based authentication and key management in a data communication network

Recommendation ITU-T X.1034

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	**X.1030–X.1049**
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1034

Guidelines on extensible authentication protocol based authentication and key management in a data communication network

Summary

The extensible authentication protocol (EAP) is an authentication framework that supports multiple authentication mechanisms between a supplicant and an authentication server in a data communication network. EAP can be used as a basic tool for enabling user authentication and distribution of session keys in a data communication network. Since there are several EAP methods, the application designer should select the optimal EAP method among them.

This revision of Recommendation ITU-T X.1034 describes a framework for EAP-based authentication and key management for securing the lower layer in a communication network. It provides guidance on the selection of EAP methods and describes the mechanism for key management for the lower layer of a data communication network. The framework described in this Recommendation can be applied to protect data communication networks with wireless or wired access networks with a shared medium.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T X.1034	2008-04-06	17
2.0	ITU-T X.1034	2011-02-13	17

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

© ITU 2012

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Recommendation ITU-T X.1034

Guidelines on extensible authentication protocol based authentication and key management in a data communication network

1 Scope

The extensible authentication protocol (EAP) is an authentication framework that supports multiple authentication mechanisms between a supplicant and an authentication server. EAP can work directly over lower layers, e.g., the data link layer, such as the point-to-point protocol (PPP), IEEE 802, CDMA2000, UMTS, or VDSL/ADSL. For example, IEEE 802.1X is a typical transport mechanism for EAP over 802 LANs. The EAP basically performs authentication for a device attached to a LAN, establishing a secure point-to-point connection or preventing access by an unauthorized device. In other words, EAP can be used to authenticate the supplicant wishing to access the network. The AAA function may be used as one of the key functions for lower-layer security of a data communication network. AAA enables transporting the secret key from the authentication server to the authenticator. Thus, defining the requirements of the EAP method and key management protocol, establishing criteria for selecting an optimal EAP method among several existing EAP methods, and defining a suitable framework for EAP and an optimal key management protocol including key derivation methods for lower-layer security in end-to-end data communication are essential. This Recommendation applies mainly to EAP-based authentication and key management protocol for data communication with a wireless access network where communication through the wireless access network should be protected by the key material derived from the key management protocol.

This Recommendation describes a framework for authentication and key management to secure the lower layer in data communication. It also provides guidance on the selection of EAP methods for a data communication network and describes the mechanism for key management and possible key hierarchy for lower-layer security in a data communication network. This Recommendation is to provide complete sets for EAP-based authentication itself but also the key management, from threat analysis to requirements, allowing the network operator to choose an adequate EAP method by using some criteria described for a specific network environment.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.805] Recommendation ITU-T X.805 (2003), *Security architecture for systems providing end-to-end communications.*

[ITU-T X.1121] Recommendation ITU-T X.1121 (2004), *Framework of security technologies for mobile end-to-end data communications.*

[ITU-T X.1151] Recommendation ITU-T X.1151 (2007), *Guideline on secure password-based authentication protocol with key exchange.*

[IETF RFC 3748] IETF RFC 3748 (2004), *Extensible Authentication Protocol (EAP*).

[IETF RFC 4017] IETF RFC 4017 (2005), *Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs.*

[IETF RFC 5216] IETF RFC 5216 (2008), *The EAP-TLS Authentication Protocol.*

[ISO/IEC 8802-11] ISO/IEC 8802-11:2005/Amd.6:2006, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 6: Medium Access Control (MAC) Security Enhancements).*

3 Terms and definitions

3.1 Terms defined elsewhere

3.1.1 passive attack [ITU-T X.1151]: This refers to an attack that involves listening, i.e., eavesdropping, without modifying or supplementing information.

3.1.2 server-compromised attack [ITU-T X.1151]: This refers to an attack wherein an attacker obtains verifier information from the server and launches a dictionary attack on the password file.

3.1.3 temporal key (TK) [ISO/IEC 8802-11]: This pertains to the keying materials for the encryption and integrity of messages during later data sessions. TK generally resides in the part of PTK.

3.2 Terms defined in this Recommendation

3.2.1 4-way handshake [adapted from IETF RFC 4017]: A 4-way handshake is a process consisting of 4 messages exchanged by two parties, where a pair-wise master key is involved. As a Pair-wise Authentication and Key Management Protocol (AKMP) defined in [ISO/IEC 8802-11], it confirms the mutual possession of a Pair-wise Master Key by two parties and distributes a Group Key.

3.2.2 authentication, authorization, accounting (AAA) [adapted from IETF RFC 4017]: The AAA protocol can be used as transport mechanism for the EAP message; it consists of RADIUS and Diameter. In general, the terms "AAA server" and "backend authentication server" are used interchangeably.

3.2.3 authenticator [adapted from IETF RFC 4017]: The authenticator refers to the endpoint of the link initiating EAP authentication when a supplicant wants to access the network.

3.2.4 backend authentication server [adapted from IETF RFC 4017]: A backend authentication server, i.e., authentication server, pertains to an entity providing authentication service to an authenticator. A typical backend authentication server is the AAA server.

3.2.5 credentials: A set of security-related information comprising keys, keying material and cryptographic algorithm-related parameters that can be used to establish the identity of an entity, or to help that entity communicate securely.

3.2.6 EAP server [adapted from IETF RFC 4017]: This entity executes the EAP authentication method with the supplicant. In case no backend authentication server is used, the EAP server plays the role of the authenticator. In case a backend authentication server is used, that is, if the authenticator operates in pass-through mode, i.e., the authenticator forwards the EAP message without any modification to the supplicant or vice versa, the EAP server is placed on the backend authentication server.

3.2.7 key confirmation: A procedure to prove one entity that another entity established the correct secret keying material as a result of a key establishment.

3.2.8 man-in-the-middle attack [adapted from ITU-T X.1151]: This refers to an attack wherein an attacker intercepts the public key being exchanged by two entities and substitutes his/her own public key to impersonate the recipient, where the attacker can own the public key or take a copy of it while being exchanged. This attack compromises the security of the cryptosystem.

3.2.9 master key (MK): Top-level keying material is shared between the supplicant and the authentication server to derive the master session key. In general, a master key is different from the master session key. This is because the MK represents a positive access decision for a supplicant by the authentication server.

3.2.10 master session key (MSK) [adapted from IETF RFC 4017]: This refers to the keying material derived between the EAP peer and server and exported to the authenticator using the EAP method. MSK is at least 64 octets long. In existing implementations, an AAA server acting as an EAP server transports the MSK to the authenticator. It refers to the privilege given to a supplicant by an authenticator to access the lower layer of a data communication network. In this Recommendation, MSK is used interchangeably with the Pair-wise Master Key (PMK).

3.2.11 mutual authentication [adapted from ITU-T X.1151]: This pertains to a type of authentication wherein the supplicant authenticates the server and the server authenticates the supplicant. Mutual authentication can prevent phishing and pharming attacks.

3.2.12 pair-wise master Key (PMK) [adapted from ISO/IEC 8802-11]: This pertains to the keying material derived between the EAP peer and server and exported to the authenticator using the EAP method. In this Recommendation, the PMK is used interchangeably with the master session key (MSK).

3.2.13 pair-wise transient key (PTK) [adapted from ISO/IEC 8802-11]: This refers to the keying material derived between the EAP peer and authenticator based on the pair-wise master key. This keying material is shared by both the peer and the authenticator.

3.2.14 perfect forward secrecy (PFS) [adapted from ITU-T X.1151]: In the cryptography of a key establishment protocol, this pertains to the condition wherein a compromised long-term private key after a given session does not compromise any earlier session.

3.2.15 server compromised-based dictionary attack [adapted from ITU-T X.1151]: For the password-based EAP method, the attacker is unable to impersonate the supplicant by obtaining a user password even after obtaining the hidden password file. Once the attacker compromises the server, he/she can obtain the hidden password file, i.e., hashed password file, and perform the offline dictionary attack against the hidden password file to obtain the password which can be used to impersonate the supplicant. However, this kind of attack can be prevented by encrypting the hidden password file by the secret key which is stored in the external hardware token or using some sophisticated cryptographic schemes, secret sharing schemes between the server and the hardware token. The resistance to the server compromised-based dictionary can be regarded as a way to mitigate the server-compromised attack. As a conclusion, this capability can be obtained by using a hardware token to store the server's secret materials.

3.2.16 successful authentication [adapted from IETF RFC 4017]: This is referred to as a successful exchange of EAP messages wherein the authentication server decides to allow the supplicant access and the supplicant decides to use such access.

3.2.17 supplicant [adapted from IETF RFC 4017]: This pertains to the endpoint responding to the authenticator. In this Recommendation, the supplicant is used interchangeably with the peer. The peer pertains to the end of the link responding to the authenticator. In [ISO/IEC 8802-11], this end is also known as the supplicant.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations:

AAA	Authentication, Authorization, and Accounting
ADSL	Asymmetric Digital Subscriber Line
CDMA	Code Division Multiple Access
DoS	Denial of Service
EAP	Extensible Authentication Protocol
EMSK	Extended Master Session Key
LAN	Local Area Network
MIC	Message Integrity Code
MK	Master Key
MSK	Master Session Key
MTU	Maximum Transmission Unit
NAS	Network Access Server
PAC	Protected Access Credential
PFS	Perfect Forward Secrecy
PMK	Pair-wise Master Key
PPP	Point-to-Point Protocol
PTK	Pair-wise Transient Key
TCP	Transmission Control Protocol
TEK	Transient Encryption Key
TIK	Transient Integrity protection Key
TK	Transient Key
TLS	Transport Layer Security
SCTP	Stream Control Transmission Protocol
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunication System
VDSL	Very High Speed Digital Subscriber Line
3GPP	3rd-Generation Partnership Project
3GPP-2	3rd-Generation Partnership Project 2

5 Conventions

None.

6 EAP-based authentication and key management framework

6.1 Introduction

A supplicant wishing to access the network should be authenticated by the network operator to use the network services or resources of the network operator. Moreover, when the network being accessed uses wireless transmission technology or a wired access network with a shared medium, the supplicant should share the common secret with the network to protect the exchanged message in later sessions against eavesdropping, modifying, or listening. The authentication and key management framework can be used to perform mutual authentication between the supplicant and the authentication server and share the common secret between the supplicant and network access server (NAS) acting as a gateway in the access network as well. This refers to the gateway node enabling the peer to gain access to the network. The function of the authenticator generally resides in the network access server.

There are three entities required for authentication and key management: a supplicant (or peer), an authenticator, and an authentication server. The supplicant functions as an end-user or a supplicant wishing to access the network in the end-user station. The authenticator acts as a policy enforcement point mediating EAP messages between the supplicant and the authentication server. The authentication server acts as a sub-function of the AAA server, authenticating the supplicant, optionally sharing a secret that can be used to derive cryptographic keys, posting the result of authentication of an end-user to the authenticator, and forwarding the shared secret to an authenticator that can be used to derive cryptographic keys between the authenticator and the supplicant to ensure confidentiality and integrity and enable message authentication. The detailed description of a policy model for key management and key derivation is given in clause 9.1.

The path between the supplicant and the authenticator may be the wireless or wired medium used by more than one peer to exchange messages; hence the need for this path to be protected with adequate protection methods. Authentication messages for mutual authentication should be exchanged between the supplicant and authentication server using the EAP transport mechanism via the authenticator. When operating in pass-through mode, the authenticator only relays EAP messages from the supplicant to the authentication server or vice versa. There are many EAP methods that are being used in a variety of applications. Therefore, the network designer should select an adequate EAP method using some criteria for evaluating the existing EAP methods. The type and syntax of an EAP message should also be defined for authentication.

The backend protocol that transfers authentication messages from the authenticator to the authentication server should use the existing AAA protocol. There are two well-known AAA protocols: RADIUS and Diameter. A specific AAA protocol should be selected by defining the criteria for evaluating AAA protocols for authentication.

Authentication and key management generally consists of four operational phases: security capability discovery, EAP authentication, AAA-based key distribution, and key management (see Figure 1). In the security capability phase, a supplicant negotiates on the security capabilities and the various parameters of the protocol to be used with the authenticator. On the other hand, in the EAP phase, the authentication server authenticates a supplicant and derives a master secret shared with the supplicant as a result of the EAP protocol. In an AAA-based key distribution phase, the authentication server transports the master secret to an authenticator to allow authentication to derive various cryptographic keys for a subsequent session between a supplicant and an authenticator. To prevent the use of the same secret key over and over and a security hole as a result of such, fresh cryptographic keys should be used in every session. Finally, in the key management phase, the authenticator exchanges random numbers with the supplicant to obtain a fresh cryptographic key.

In case the authenticator keeps the authentication-related information of a user, the authentication server is not required, i.e., the authentication server can act as part of the authenticator.

The clause is to describe an overview of the framework of the authentication and key management. The detailed operation for the key management protocol is described in clause 8.2. Since the key management function can be performed based on the policy model in clause 9.1 and the key hierarchy is constructed based on the policy model, the example of key hierarchy is described in detail in clause 9.2.

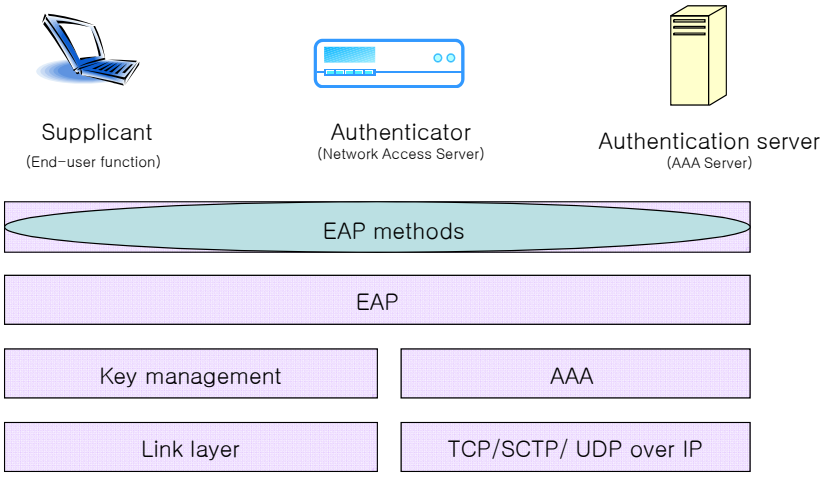

Figure 1 – EAP-based authentication and key management framework

6.2 General features of EAP

EAP should have the following properties:

- **Simplicity**: Implementation should be simple, and deployment with minimal pre-existing infrastructure.
- **Wide applicability**: EAP [IETF RFC 3748] should be applicable as much as possible to any network such as wireless access networks and wired access networks as well as to any type of access network such as IEEE 802 wireless LANs [ISO/IEC 8802-11], 3GPP [b-3GPP], and 3GPP2 [b-3GPP2] mobile networks.
- **Security**: All kinds of major attacks should be resisted, such as eavesdropping, man-in-the-middle attack, modifications and replay attack, as well as any other fabrication.
- **Extensibility**: Adding to the method possible future extensions on a per-need basis should be enabled.

The following are the typical advantages of an EAP protocol:

- An EAP protocol can work with multiple authentication mechanisms. This suggests its independence from any specific authentication mechanism.
- As authenticator, the NAS (network access server) does not need to understand the details of each authentication method since it only acts as a mediator between the supplicant and the authentication server. In case a backend authentication server is used, NAS simply acts as a pass-through agent, i.e., all packets are forwarded without any modification. In some cases wherein no backend authentication server is used, a local supplicant may be authenticated by the authenticator using the supplicant's credentials as stored in the authenticator.
- The separation of the authenticator from the backend authentication server simplifies credentials management and policy decision making.

As a typical disadvantage of the EAP protocol, proving the security of the EAP protocol and key management protocol may be somewhat difficult in case the authenticator is separated from the backend authentication server.

6.3 Basic operational procedures for authentication and key management protocols

An EAP authentication usually starts with an EAP-request packet, and terminates with an EAP-success/failure message. An EAP authentication takes place through the following steps:

- The authenticator sends a request packet known as authentication request to authenticate the supplicant.
- The supplicant sends a response packet known as authentication response in response to a valid request.
- The authenticator sends additional request packets, and the supplicant replies with a response packet.
- The conversation continues until the authenticator can no longer authenticate the supplicant or successful authentication is deemed completed.

After a user is authenticated by the authenticator, an optional key management protocol mainly based on a 4-way handshake process between the supplicant and the authenticator should be executed to derive or share a common session key for subsequent communication sessions.

7 EAP protocols

7.1 Vulnerabilities in EAP

The general threat model for data communication and mobile data communication can be applied to the threat models of Recommendations [ITU-T X.805] and [ITU-T X.1121]. Note, however, that there are several practical vulnerabilities associated with the EAP protocol:

- **Eavesdropping**: An attacker may try to obtain useful information by eavesdropping on authentication traffic.
- **Modification or fabrication**: This attack can be regarded as one of the sort of attacks resulting from man-in-the middle attack. An attacker may try to modify or send fake EAP packets.
- **DoS**: An attacker may launch denial of service attacks by spoofing lower-layer indications or success/failure packets, replaying EAP packets, or generating packets with overlapping Identifiers.
- **Online dictionary attack**: When the password-based EAP method is used, an attacker may attempt to launch an online dictionary attack to try to guess the password and pass authentication. This way the attacker may obtain an adequate password in the message received from the authentication protocol with a successful result. As a form of protection, the failed authentication trials by the server can be taken into account.
- **Offline dictionary attack**: In case the password-based EAP method is used, an attacker may attempt to recover the password by launching an offline dictionary attack on the message obtained during the previous successful protocol run.
- **Man-in-the-middle-attack**: An attacker may reside in the path between a supplicant and a server and attempt to convince the peer that it is a legal peer by mounting a man-in-the-middle attack.
- **Use of weak authentication**: An attacker may attempt to disrupt EAP negotiation to cause a weak authentication method to be selected. This attack can be regarded as one sort of attack resulting from the downgrading attack described below.

- **Weak key derivation**: An attacker may attempt to recover keys by taking advantage of weak key derivation techniques used within the EAP methods.
- **Weak cipher suites**: An attacker may attempt to take advantage of weak ciphersuites subsequently used after the EAP conversation is completed. If the conversation is completed, the attacker can exploit the weakness of the negotiated weak ciphersuites to compromise the supplicant or the authentication server.
- **Downgrading attack**: An attacker may attempt to perform downgrading attacks on lower-layer ciphersuite negotiation to ensure that a weaker ciphersuite is selected subsequently for EAP authentication. An attacker acting as an authenticator may provide incorrect information to the EAP peer and/or server using out-of-band mechanisms (e.g., through AAA or lower-layer protocol). This involves impersonating another authenticator or providing inconsistent information to the peer and EAP server.
- **Identity exposure**: The attacker learns the identity of the supplicant by eavesdropping on exchanged messages during a successful protocol run. This attack can be regarded as one of the sort of attacks resulting from eavesdropping and usually takes place as a result of the "eavesdropping" attack.
- **Channel hijacking**: The attacker hijacks the session established between the supplicant and the authentication server.
- **Server compromised dictionary attack**: For a password-based EAP method [b-IETF RFC 5433], the attacker is unable to impersonate the supplicant by obtaining a user password even after obtaining the password file. When the attacker compromises the server, he/she can obtain the hidden password file, i.e., the hashed password file, and perform the offline dictionary attack against the hidden password file to obtain the password which can then be used to impersonate the supplicant. However, this kind of attack can be prevented by encrypting the hidden password file with a secret key stored in an external hardware token or by using a sophisticated cryptographic scheme, i.e., the secret sharing scheme between the server and the hardware token. As a conclusion, this capability may be obtained by using a hardware token to store the server's secret materials.

7.2 Set of requirements for EAP

Since EAP can be performed over a wired or a wireless medium depending on the specific access network, several requirements for EAP methods were derived taking into account the requirements of WLAN [b-IETF RFC 5247] as follows:

R.1 **Secure generation of symmetric keying material**. This refers to the ability of the EAP to generate keying material to protect the subsequent EAP session or subsequent data session. In other words, the supplicant and the authentication server share a common secret: the top-level key. The top-level key is referred to as a master key (MK). All cryptographic symmetric keying material of lower-layer security may be derived from the master key. In this case, it should be generated securely from the master key by using a secure key derivation function.

R.2 **Minimum key strength**. An EAP method should be capable of generating the keying material of a master key with at least 128-bit effective key strength.

R.3 **Mutual authentication**. This pertains to an ability of the EAP method wherein an authentication server authenticates a supplicant and a supplicant authenticates an authentication server at the same time. A supplicant and an authentication server should authenticate each other.

R.4 **Maintenance of synchronized state between two entities**. Once the EAP method is successfully completed on the EAP peer and the server, the shared EAP method state of both sides is synchronized. The supplicant should maintain the synchronized state with an authentication server in order to perform the authentication successfully.

R.5 **Resistance to dictionary attacks**. This refers to the immunity to dictionary attacks. There are two kinds of dictionary attacks: the online dictionary attack and the offline dictionary attack. When password authentication-based EAP is used, passwords are commonly selected from a small set; thus raising concerns over dictionary attacks. If a password is used as a secret, a method may provide protection against dictionary attacks if it does not allow an offline attack with a work factor based on the number of passwords in an attacker's dictionary. Password-based EAP should be resistant to dictionary attacks.

R.6 **Protection against man-in-the-middle attacks**. The EAP should be protected from a man-in-the-middle attack through "cryptographic binding", "integrity protection of exchanged messages", "replay protection", and "session independence".

R.7 **Protection against server-compromised attack**. This pertains to the ability of the EAP method to resist a server-compromised attack. Specifically, even after obtaining the password file, the attacker is not able to impersonate the supplicant without performing an exhaustive dictionary attack on the compromised password file to obtain a user password.

R.8 **Prevention of domino effect or Denning-Sacco attack**. Compromising a single authenticator is not tantamount to compromising session keys and long-term secrets.

R.9 **Replay protection**. All messages exchanged by EAP must be replay-protected by using non-repeating nonces.

R.10 **Protected ciphersuite negotiation of the EAP procedure**. This refers to the ability of an EAP method to negotiate the ciphersuite used to protect the EAP conversation as well as to protect the negotiation, not the ability to negotiate the ciphersuite used to protect data. If the EAP method negotiates on the ciphersuite used to protect the EAP conversation, the "protected ciphersuite negotiation" security claim must be supported. The protected ciphersuite negotiation should be negotiated during each EAP trial to avoid compromising a particular cryptographic algorithm. The EAP method supporting negotiations of ciphersuites, protocol versions, and features should include post-verification, that is, the only practical way to detect the ciphersuite downgrading attack described in clause 7.1 is to perform post-verification of the negotiation, in which once both parties obtain a transient integrity protection key *TIK*, they send each other integrity-protected verification messages, which include the sent and received messages prior to *TIK* establishment.

R.11 **Strong, fresh session keys**. Session keys may prove to be strong and fresh in all circumstances, at the same time maintaining algorithm independence.

R.12 **Confidentiality of master keys**. The confidentiality of master keys must be maintained by the EAP peer and the authentication server. The peer can store MKs using a secure hardware token such as a smartcard.

R.13 **Authorization**. The authorization is a procedure to verify whether an entity is eligible to access a requested network or service. The authorization information is communicated from the authentication server to the authenticator based on the identity authenticated by EAP. The authenticator can use the authorization information to provide classified services to the peer. Authorization information should be kept securely in the database.

R.14 **User identity privacy**. This involves protecting the privacy of user identity. This can be obtained using the confidentiality algorithm and temporary ID of a user. In general, the temporary ID is exchanged through an encrypted message. Additional ciphersuite negotiation is required in maintaining confidentiality in the EAP procedure to ensure user identity privacy. The EAP method supports identity protection.

R.15 **Unique naming and identifying**. Session keys could be uniquely named or identified.

R.16 **Protection against server-compromised dictionary attack**. This can be obtained by using a tamper-free token such as a smartcard. An attacker compromising a server compromises the password file as well. In such case, the compromised password file may be used to discover a password by launching a dictionary attack. Note, however, that this type of vulnerability can be protected in a system using an EAP method wherein the password file is encrypted and the encrypting key is stored in the tamper-free module.

R.17 **Channel binding**. This pertains to communication within an EAP method for integrity-protected channel properties, such as endpoint identifiers, that can be compared to values communicated via out-of-band mechanisms (e.g., through an AAA or a lower-layer protocol). It needs secure mechanisms for exchanging lower-layer EAP parameters, which enable the authenticated exchange of data. In case confidentiality is required, an additional symmetric-key ciphersuite would be negotiated.

R.18 **Fragmentation**. This refers to whether or not an EAP method supports fragmentation and reassembly. EAP methods support fragmentation and reassembly if EAP packets exceed the arbitrary length of the minimum MTU (maximum transmission unit), which refers to the size (in bytes) of the largest packet that can be passed onwards by a given layer of communication protocol.

7.3 Criteria for evaluating and classifying EAP methods

The requirements given in clause 7.2 can be classified into three categories: basic requirement, threat-related requirement, and supplemental requirement. Some criteria for classifying EAP protocols may be established as follows:

- Basic requirements:
  - secure generation of symmetric keying material;
  - minimum key strength;
  - mutual authentication;
  - strong, fresh session keys;
  - confidentiality of the master key;
  - maintenance of synchronized state between two entities;
  - protected cipher suite negotiation of the EAP procedure.
- Threat-related requirements:
  - resistance to dictionary attacks;
  - protection against man-in-the-middle attacks;
  - protection against the server-compromised attack for the password-based EAP method;
  - prevention of the domino effect;
  - replay protection;
  - protection against the server compromised dictionary attack for the password-based EAP method.

- Supplemental requirements:
  - authorization;
  - user identity privacy;
  - unique naming and identifying;
  - channel binding;
  - fragmentation.

The objective of the classification of EAP methods in Table 1 is designed to be applicable to EAP methods developed in the future, not to existing EAP methods. EAP methods can be classified into three categories: fundamental-level EAP class, middle-level EAP class, and high-level EAP class. The network operator should use one of the three EAP classes. The system designer may use an EAP method of a certain level considering the security requirements of the application. Fundamental-level EAP methods satisfy the requirements listed in Table 1, e.g., the secure generation of symmetric keying material. The middle-level EAP satisfies the requirements of the fundamental EAP class and adds four more requirements, i.e., user identity privacy, authorization, unique naming and identifying, and protection against a server compromised attack. The high-level EAP satisfies all the requirements of the middle-level EAP class and adds three more requirements, protection against the server compromised dictionary attack or use of hard token for the password-based EAP method, channel binding, and fragmentation. The difference between a fundamental-level EAP method and a middle-level EAP method lies mainly in the capability of the attacker to impersonate the user, compromising the server without a dictionary attack or any further effort. On the other hand, the difference between a middle-level EAP method and a high-level EAP method lies mainly in the capability of the server using a hardware token to keep the secret to protect the user's authentication information. Therefore, EAP methods can be classified into one of the three EAP classes according to their capabilities. In Table 1, "Y" means that a certain requirement meets a certain level EAP, while "N" means that it does not meet it.

Table 1 – Classification of EAP methods

Criteria	Fundamental-level EAP	Middle-level EAP	High-level EAP
Secure generation of symmetric keying material	Y	Y	Y
Minimum key strength	Y	Y	Y
Mutual authentication	Y	Y	Y
Maintenance of a synchronized state between two entities	Y	Y	Y
Resistance to dictionary attacks	Y	Y	Y
Protection against man-in-the-middle attacks	Y	Y	Y
Prevention of domino effect or Denning-Sacco attack	Y	Y	Y
Replay protection	Y	Y	Y
Strong, fresh session keys	Y	Y	Y
Confidentiality of the master key	Y	Y	Y
Protected ciphersuite negotiation of the EAP procedure	Y	Y	Y
Authorization	N	Y	Y

Table 1 – Classification of EAP methods

Criteria	Fundamental-level EAP	Middle-level EAP	High-level EAP
Protection against the server-compromised attack for the password-based EAP method	N	Y	Y
User identity privacy	N	Y	Y
Unique naming and identifying	N	Y	Y
Protection against the server-compromised dictionary attack or use of a hard token for the password-based EAP method	N	N	Y
Channel binding	N	N	Y
Fragmentation	N	N	Y

7.4 EAP method

A suitable EAP method can be selected by applying the criteria in Table 1. For example, EAP-TLS [IETF RFC 5216] is a de facto standard for use in EAP-based authentication following the IEEE 802.1x authentication model [b-IEEE 802.1X]. If some requirements of EAP are not met, a new EAP method that meets all the requirements for the application should be developed. That is, the EAP method should have user identity privacy, protection against the server compromised dictionary attack, and channel binding. Therefore, a specific EAP method satisfying all the above-mentioned requirements can be regarded as a high-level EAP method.

7.5 Evaluation of existing EAP methods

The evaluation result for the existing EAP methods is presented in Appendix I, which can be used by the network operator for selecting the adequate EAP method among the many existing EAP methods.

8 Key management

The following should be considered when designing the key management of lower-layer security:

- Consider several access networks such as IEEE 802.11 [ISO/IEC 8802-11], 3GPP, 3GPP2, VDSLs, and other fixed networks and work smoothly with them. In other words, since the access network may use a wireless or a wired medium, the key management protocol should consider all kinds of transmission methods for secure key management.
- Compliant with the existing authentication methods; an access network with its own authentication method supports rather than excludes the existing authentication method (in case the access network does not have its own authentication method, this specification must be applied).

8.1 Practical threats to a specific wireless access network

The general threat model for a mobile network can be applied to the threat model of [ITU-T X.1121]. In addition, the following are several practical threats exclusively associated with the wireless access network:

- **DoS**: An attacker may launch denial of service attacks by interfering with the frequency spectrum through an external radio frequency source or by sending several messages to the network element in the wireless network with the intention of overloading it and denying other subscribers or devices further access.

- **Man-in-the-middle-attack**: An attacker may reside in the path between a supplicant and an authenticator posing as a legal authenticator or supplicant when intercepting the communication.
- **Rogue network access server**: Without the authentication of the supplicant by the authenticator or the authentication server, the rogue network access server can pretend to be a legal node; thus giving rise to major security concerns.
- **Illegal supplicant**: Without proper authentication or authorization, the illegal supplicant tries to succeed in the authentication procedure and gains network access in the process.

8.2 General operational phases for key management

As in wireless LANs, authentication and key management may consist of four operational phases (Figure 2): security capability discovery, EAP authentication, AAA-based key distribution, and key management of the lower layer. [b-Cam-Winget]

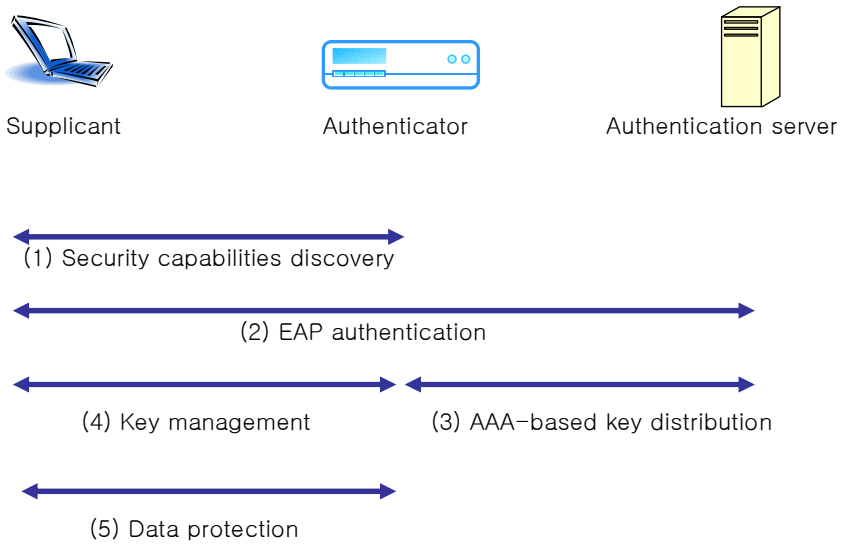

Figure 2 – Four operational phases for the authentication and key management of the lower layer

The security capability discovery phase determines the correct peer for communication, with the authenticator publishing its security capability to all supplicants periodically. At the end of the discovery phase, the supplicant is aware of the alleged network ID, alleged authentication and ciphersuites the network wants to use, and correct credentials for the network, and the authenticator, aware of the types of authentication and cipher suites. The ciphersuite negotiation between a supplicant and an authenticator is performed as part of the security capability discovery to enable crypto-agility and backward-compatibility. EAP authentication involves centralizing network access policy decisions at the authentication server, with the supplicant identified by the authentication server. The supplicant and the authentication server mutually authenticate each other, and an authentication server generates the master key as a side effect of authentication by using an EAP method and distributes the derived pair-wise master key (PMK) to the authenticator.

AAA-based key distribution involves distributing the derived master key (pair-wise master key) from the authentication server to the authenticator. The detailed AAA operation is given in Appendix II.

There are two methods for sharing the PMK between the supplicant and the authenticator: the pre-distribution method and the transported method. In a pre-distribution method, the PMK is shared by a supplicant and an authenticator in advance. In the transported method, the pair-wise master key is imported from the authentication server to the authenticator. If the pre-distribution method is used, EAP authentication and AAA-based key distribution are not required.

The key management phase involves sharing the fresh session key (pair-wise transient key) from the derived master key (PMK) between the supplicant and the authenticator, proving to each other that each peer is alive and deriving all the necessary session keys (pair-wise transient keys) for protecting both message exchange during the key management protocol and subsequent sessions between the authenticator and the supplicant. In other words, PTK may contain cryptographic keys, e.g., keys for integrity and confidentiality, for the key management protocol.

8.3 Set of requirements for key management

The key management protocol must be executed between an authenticator and a supplicant. The EAP key management protocol in a data communication network can be said to be similar to that for WLAN in IEEE. This clause describes the requirements of key management derived taking into account the requirements of WLAN [b-IETF RFC 5247].

r.1 **Mutual proof of possession of EAP keying material (mutual authentication)**. The supplicant and authenticator should prove possession of keying material to each other in a secure manner. For the key management protocol, the EAP peer and authenticator should prove possession of the pair-wise master key transported from the backend authentication server to the authenticator to demonstrate that the peer and the authenticator have been authorized. For example, possession of keying material should be proven using the result of the hash function with the input of nonce and keying material, etc. This can protect against man-in-the-middle attacks, rogue network access server, and illegal supplicant. As a minimum requirement, a key management procedure should provide mutual implicit key authentication, i.e., the established keying material is only known to the peer and the authenticator.

r.2 **Generation of fresh pair-wise transient keys (PTKs)**. The supplicant or authenticator should generate a fresh pair-wise transient key from PMK for a later data session in a secure manner. Ideally, PTKs should be cached in the lower layer. Deriving PTK from a portion of PMK in a roaming case may result in the reuse of the shared PMK. In lower layers where the caching of EAP keying material is supported, the key management protocol should support the derivation of fresh unicast or multicast TKs even when the keying material provided by the backend authentication server is not fresh. This is typically supported via the exchange of nonces or counters that are then mixed with the exported keying material to generate fresh unicast session keys or even multicast session keys if possible.

r.3 **Key control**. A key management procedure should provide key control, i.e., the supplicant and the authenticator should both contribute data for the key computation.

r.4 **Key confirmation**. A key management procedure should provide key confirmation, i.e., the peer and the authenticator should both obtain assurance that they computed the PTK correctly. Key confirmation is commonly achieved by using one of the derived keys to generate a message authentication code. Mutual key confirmation, combined with mutual implicit key authentication, provides mutual explicit key authentication.

r.5 **Perfect forward secrecy**. In case public-key based key establishment schemes are employed, a key management procedure should provide perfect forward secrecy (FS), i.e., a compromise of long-term private or pre-shared secret keys does not enable an adversary to compute the PTK generated in previous EAP executions.

r.6 **Post verification**. Post-verification should be provided for all integrity-vulnerable information that has been exchanged before a transient integrity key is available.

r.7 **Protection against practical threats to a specific wireless access network**. This means that there should be protection against all the threats described in clause 8.1. Examples of such threats include DoS, man-in-the-middle attacks, rogue network access server, and rogue supplicants.

r.8 **Secure derivation of PTK**. Keys in PTK can be classified into three categories: authentication key for the key management protocol, encryption key for the key management protocol, and encryption/authentication key (TK) for subsequent secure traffic exchange. The authentication key can be used to ensure the integrity of messages exchanged during the implementation of the key management protocol. The encryption key for the key management protocol can be used to maintain confidentiality for specific messages, e.g., group key for subsequent data traffic. PTK should be derived in a secure fashion.

r.9 **Minimum key strength**. The key management protocol should generate the keying material with 128-bit effective key strength for each key type of PTK.

r.10 **Secure capabilities negotiation**. The supplicant and authenticator should negotiate on the capabilities in a secure manner. To protect against spoofing during the discovery phase, make sure the "best" ciphersuite is selected and provide protection against the forging of negotiated security parameters. The key management protocol may support secure capabilities negotiation for the key management procedure. This includes the secure negotiation of usage modes, session parameters (e.g., security association identifiers) and key lifetimes, ciphersuites, and required filters including the confirmation of security-related capabilities discovered during the key management phase.

r.11 **Secure message protection for the key management protocol**. Messages exchanged for the key management protocol should be protected by integrity and confidentiality mechanisms. Such cryptographic services should be provided using PMK derived from MK. This can protect against man-in-the-middle, rogue network access server, and illegal supplicant attacks.

r.12 **Key lifetime negotiation**. This features explicit key lifetime negotiation or seamless rekey. The key management protocol may handle the rekey and determination of the key lifetime. If key caching is supported, secure negotiation of key lifetimes may be required.

r.13 **Authorization**. The authorization information of the EAP peer transport from the authentication server may be used to provide an appropriate labelled service to the peer wishing to use a specific network service. This can protect against illegal supplicants.

r.14 **Unique entity naming**. The supplicant or authenticator should have its own identifier. A basic feature of the key management protocol should explicitly name the parties engaged in the exchange. Without explicit identification, the parties engaged in the exchange cannot be identified.

r.15 **Key naming and selection**. Since there is more than one key for a given key type, the key management protocol may explicitly name the keys used in the proof of possession exchange to prevent confusion when more than one set of keying material could potentially be used as basis for the exchange. To support correct processing, the key management protocol may support the naming of key management and associated transient session keys for the identification of the correct set of pair-wise transient keys in processing a given packet.

r.16 **Direct operation**. Since the key management protocol is concerned with the establishment of security associations between the EAP peer and authenticator including the derivation of PTKs, only those parties are on a "need to know" basis with PTKs. The key management protocol should operate directly between the supplicant and the authenticator; the backend authentication server should not be involved in such protocol.

r.17 **Bidirectional operation**. While some ciphersuites only require a single set of PTKs to protect data traffic in both directions, other ciphersuites require a unique set of PTKs in each direction. The key management protocol should support the derivation of unicast temporal keys or multicast temporal keys in each direction such that two separate exchanges are not required.

r.18 **Group key handshake protocol**. The key management protocol could be executed as an option to generate the new group key upon the completion of the key management protocol. The group key generated by the authenticator can be transmitted to the supplicant from the authentication server as an option.

8.4 Flow of the key management protocol

The key management protocol should be executed between the authenticator and the supplicant. By exchanging authentication information, the supplicant and the authenticator can share the extended session key derived from the pseudo-random function with the input of the master session key, and random numbers generated by the authenticator and the supplicant, where the master session key is known as a PMK and the extended session key a PTK. The master session key obtained after the authentication is transferred from the authentication server to the authenticator in a secure manner. The master session key is assumed to be known to the supplicant and the authenticator only. The 4-way handshake protocol may consist of four messages exchanged between the authenticator and the supplicant.

The authenticator begins by sending the authenticator nonce in Message 1. The supplicant selects the supplicant's nonce and computes the extended session key, PTK, using the algorithm described in clause 9.2. The PTK includes the key confirmation key, key encryption key, and pair-wise session keys. The supplicant sends the supplicant's nonce and computes the MIC (message integrity code) using the key confirmation key to enable message authentication and ensure message integrity in Message 2. The authenticator can compute the PTK based on the pseudo-random function with the inputs of the authenticator's nonce and supplicant's nonce.

The authenticator computes the MIC to enable message authentication and ensure message integrity, sending the MIC and the authenticator nonce (the same as the authenticator nonce in Message 1) to protect against the replay attack.

The supplicant verifies the MIC and computes it to ensure message integrity, sending the MIC back to the authenticator in Message 3. The authenticator then verifies the MIC. This concludes the 4-way handshake protocol. The illustrative diagram for the 4-way handshake protocol can be shown in Figure 3. In Figure 3, Info1/2/3/4 denote relevant accompanying information for each message, respectively, ANonce denotes the nonce generated by the authenticator, SNonce denotes the nonce generated by the supplicant, and MIC denotes the message integrity code for the exchanged message. After the 4-way handshake protocol, the authenticator and the supplicant may share PTK (pair-wise transient key) for subsequent secure sessions between them.

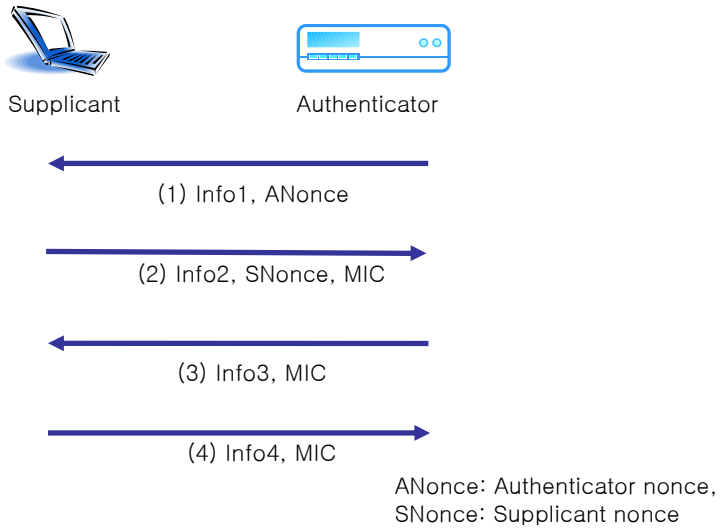

Figure 3 – Four-way handshake protocol for key management of the lower layer

8.5 Requirements classification of key management

The requirements can be classified into three categories: mandatory requirements, recommended requirements, and optional requirements. The following are the mandatory requirements of the key management protocol in a wireless access network:

- mutual proof of possession of EAP keying material (mutual authentication);
- generation of fresh pair-wise transient keys (PTKs);
- key control;
- key confirmation;
- perfect forward secrecy;
- post verification;
- protection against practical threats to a specific wireless access network;
- subsequent generation of transient session key including keys for confidentiality and data integrity;
- minimum key length;
- secure capabilities negotiation;
- secure message protection for the key management protocol.

The following are the recommended requirements of the key management protocol in a wireless access network:

- unique entity naming;
- key naming and selection;
- direct operation;
- bidirectional operation;
- authorization.

The following are the optional requirements of the key management protocol in a wireless access network:

- key lifetime negotiation;
- group key handshake protocol.

9 Cryptographic key for key management

9.1 General policy model

The policy decision point is defined as a logical component making policy decisions pertaining to the access right to the access network of a data communication network with wireless network access. The policy decision can be made together with the authentication procedure by two policy decision points by exchanging EAP messages: the supplicant and the authentication server. Note that the policy decision can be represented as a policy decision token, a fresh master key which can be shared by a supplicant and the authentication server only. This token is a symmetrical key that demonstrates authorization to make a decision. The authentication server should distribute this token to the authenticator, where it can be used to generate the policy enforcement token that represents a supplicant's access right to the access network. Both the supplicant and the authentication server must reach the same policy decision.

The policy enforcement point is defined as a logical component that enforces a policy decision by the policy decision point. The policy enforcement decision can be represented as a policy enforcement token, which is a master session key, pair-wise master key. The pair-wise master key can be generated by two policy enforcement points: the authentication server and the supplicant. The policy enforcement token should be shared by the authenticator and a supplicant only. In other words, the policy enforcement token is bound to this session between a supplicant and the authenticator. The policy enforcement token should be based on the policy decision token and Nonces between the authentication server and a supplicant. The possession of the policy enforcement token demonstrates authorization to access the access network of a data communication network.

Although the policy enforcement token can be derived from the policy decision token, the policy decision token should be independent from the policy enforcement token to prevent the authentication server from making access control decisions instead of the authenticator.

9.2 Possible cryptographic key hierarchy and key derivation

A key hierarchy is a tree structure that represents the relationship of different keys. In a key hierarchy, a node represents a key used to derive the keys represented by the descendent nodes. A key can only have one precedent, but may have multiple descendent nodes. There are at least three levels of keys in the key hierarchy for lower-layer security in a wireless access network: master key (MK), pair-wise master key (PMK), and pair-wise transient key (PTK). The master key (MK) is a top-level keying material shared between the supplicant and the authentication server; it can be used to derive a pair-wise master key. In general, a master key is different from the pair-wise master key. MK represents a positive access decision for a supplicant by the authentication server. The master key can be derived as a result of implementing the EAP protocol.

The pair-wise master key (PMK) is a keying material that can be shared between the EAP peer and the server and exported to the authenticator using the EAP method. Derived from MK, PMK is at least 64 octets long. In actual implementations, an AAA server acting as an EAP server transports PMK to the authenticator. This represents the privilege given to a supplicant by an authenticator to access the lower layer of a data communication network. The extended pair-wise master key may be an additional keying material derived between the EAP supplicant and a server and can be also exported using the EAP method.

The pair-wise transient key (PTK) is a keying material that can be derived from PMK along with the nonces of the authenticator and EAP peer. PTK is used to protect both messages of the ongoing EAP execution and subsequent session operating in unicast mode or multicast mode upon the completion of an EAP execution. PTK contains the cryptographic key for integrity and encryption for some of the EAP messages for the key management protocol and temporal key (TK) for the transfer of secure messages in later sessions. The cryptographic keys for integrity and encryption

for some of the EAP messages are called transient encryption key (TEK) and transient integrity protection key (TIK): they protect the messages of the ongoing EAP execution. That is, they are used for message protection at the EAP layer. TK is exported to the lower layers and may be transported to the authenticator to derive keys to protect the wireless data link upon the completion of an EAP execution.

For example, the pair-wise master key can be derived from the pseudo-random function with input of master key and several nonces. On the other hand, the master key is a master secret derived from the successful completion of EAP-TLS protocol, Random 1, a random number generated at the supplicant and transferred to the authentication server, and Random 2, a random number generated at the authentication server and transferred to the supplicant. For example, in the case of EAP-TLS, the master session key known as PMK can be derived as follows:

```
Pair-wise Master key (PMK) = PRF (Master Key, "Master secret" || Random 1 ||
Random 2)
```

The pair-wise transient key can be derived from the pseudo-random function with inputs of PMK, supplicant nonce, authenticator nonce, authenticator's endpoint identifier, and supplicant's endpoint identifier. PTK is a variable length key that can be extended to have the length required for the key between a supplicant and an authenticator.

```
Pair-wise Transient Key (PTK) = PRF (PMK, supplicant nonce || authenticator
nonce || supplicant endpoint identifier || authenticator endpoint identifier)
```

The specific pseudo-random function could be a TLS-PRF defined in [IETF RFC 5216] or other secure pseudo-random function. The pair-wise transient key consists of key confirmation key, key encryption key, and temporal key. The key confirmation key and key encryption key can be used during the 4-way handshake protocol to authenticate and encrypt, respectively, the exchanged messages. The temporal key can be used to protect the message during a later data session after the 4-way handshake protocol.

Appendix I

Evaluation of existing EAP methods

(This appendix does not form an integral part of this Recommendation.)

Tables I.1 and I.2 present the evaluation of well-known EAP methods based on the classification criteria in Table 1. Some example EAP methods in Appendix III are found to be noncompliant with the criteria in Table I.1. If some applications require the high-level EAP method, then new EAP methods should be developed in the future. The specific EAP method is not covered by the scope of this Recommendation, however. In Tables I.1 and I.2, "Yes" means that the requirement is satisfied by the specific EAP method, "No", that the requirement is not satisfied by the specific EAP method, and "N/A" that the requirement is not applicable to a certain EAP.

Table I.1 – Evaluation of well-known EAP methods based on the secret key

Criteria	EAP-MD5 [IETF-RFC 3748]	LEAP [b-LEAP]	EAP-AKA [b-IETF RFC 5448]	EAP-PSK [b-IETF RFC 4764]	EAP-SRP [b-IETF RFC 2945]
Secure generation of symmetric keying material	No	No	Yes	Yes	Yes
Minimum key strength	N/A	N/A	128-bits	128-bits	N/A
Mutual authentication	No	Yes	Yes	Yes	Yes
Maintenance of synchronized state between two entities	Yes	Yes	Yes	Yes	Yes
Resistance to dictionary attacks	No	No	N/A	Yes	Yes
Protection against man-in-the-middle attacks	Yes	Yes	Yes	Yes	Yes
Seamless compatibility	Yes	Yes	Yes	Yes	Yes
Strong, fresh session keys	No	No	Yes	Yes	Yes
Prevention of domino effect or Denning-Sacco attack	–	Yes	Yes	Yes	Yes
Replay protection	No	No	Yes	Yes	Yes
Confidentiality of master key	No	No	Yes	No	No
Protection against server-compromised attack	–	–	Yes	Yes	Yes
Protected ciphersuite negotiation of the EAP procedure	No	No	No	No	No
User identity privacy	No	No	Limited (using temporal ID)	No	Limited (not strong)
Unique naming	–	–	Yes	Yes	Yes

Table I.1 – Evaluation of well-known EAP methods based on the secret key

Criteria	EAP-MD5 [IETF-RFC 3748]	LEAP [b-LEAP]	EAP-AKA [b-IETF RFC 5448]	EAP-PSK [b-IETF RFC 4764]	EAP-SRP [b-IETF RFC 2945]
Protection against the server compromised-based dictionary attack	No	No	No	No	No
Channel binding	No	No	No	No	No
Fragmentation	No	No	No	No	No
Fast reconnect	No	No	Yes	No	No
Cryptographic binding	N/A	N/A	N/A	N/A	N/A
Session independence	No	No	Yes	Yes	No

Table I.2 – Evaluation of well-known EAP methods based on public key and other credentials

Criteria	EAP-TLS [IETF RFC 5216]	EAP-FAST [b-IETF RFC 4851]	EAP-IKEv2 [b-IETF RFC 5106]	EAP-TTLS [b-IETF RFC 5281]
Secure generation of symmetric keying material	Yes	Yes	Yes	Yes
Minimum key strength	2048-bits	128-bits, 2048-bits	128-bits, 2048-bits	2048-bits
Mutual authentication	Yes	Yes	Yes	Yes
Maintenance of synchronized state between two entities	Yes	Yes	Yes	Yes
Resistance to dictionary attacks	N/A	Yes	Yes	N/A
Protection against man-in-the-middle attacks	Yes	Yes	Yes	Yes
Seamless compatibility	Yes	Yes	Yes	Yes
Strong, fresh session keys	Yes	Yes	Yes	Yes
Prevention of domino effect or Denning-Sacco attack	Yes	Yes	Yes	Yes
Replay protection	Yes	Yes	Yes	Yes
Confidentiality of master key	Yes	Yes	Yes	Yes
Protection against server-compromised attack	–	Yes	–	–
Protected ciphersuite negotiation of the EAP procedure	Yes	Yes	Yes	Yes
User identity privacy	No	Yes	No	Yes
Unique naming	Yes	Yes	Yes	Yes

Table I.2 – Evaluation of well-known EAP methods based on public key and other credentials

Criteria	EAP-TLS [IETF RFC 5216]	EAP-FAST [b-IETF RFC 4851]	EAP-IKEv2 [b-IETF RFC 5106]	EAP-TTLS [b-IETF RFC 5281]
Protection against the server compromised-based dictionary attack	No	No	No	No
Channel binding	No	Yes	No	No
Fragmentation	Yes	Yes	Yes	Yes
Fast reconnect	Yes	Yes	Yes	Yes
Cryptographic binding	N/A	Yes	N/A	No
Session independence	Yes	Yes	Yes	Yes

Appendix II

AAA protocol

(This appendix does not form an integral part of this Recommendation.)

The AAA protocol is responsible for transporting authentication messages between an authenticator and an authentication server in [b-IETF RFC 2904]. There are several proposals for transporting an authentication message: RADIUS and Diameter in [b-IETF RFC 2058], [b-IETF RFC 3579] and [b-IETF RFC 3588], respectively. A possible AAA protocol must ensure the secure distribution of key material (master key). In other words, the secure distribution of key material including a secret to derive a session key for subsequent sessions must be performed between an authenticator and an authentication server. The selection of a specific AAA protocol is not covered by the scope of this Recommendation, however. Nonetheless, the AAA protocol should be selected based on the following specific criteria:

- protocol model;
- length of attribute field;
- type of transport layer protocol;
- session key distribution;
- error processing;
- distributed environment.

AAA protocols basically provide the mechanisms for exchanging EAP packets between the authenticator and the authentication server. RADIUS is known as the most widely deployed protocol, although Diameter enables a high degree of flexibility that can be used to address various requirements such as transport of AAA messages, support for mobility and roaming, and enhanced security features.

RADIUS has been known to have many problems and lack features for supporting mobility and roaming requirements, i.e., scalability problems and security problems in untrusted proxy environments. This is because this protocol only supports weak hop-by-hop security; it does not define data-object security mechanisms. Moreover, RADIUS was originally designed to support a small network with a few end-users and a specific set of access control mechanisms.

On the other hand, Diameter was designed to support roaming and mobility; it was based on the scalability and security principle, i.e., explicit support for agents by ensuring scalability and strong hop-by-hop security based on IPSec and reliable transport based on TCP.

Even though the selection of a specific AAA protocol is not covered by the scope of this Recommendation, the use of Diameter as AAA protocol for a data communication network is recommended.

Appendix III

Overview of the existing EAP methods

(This appendix does not form an integral part of this Recommendation.)

EAP methods are classified into the EAP method based on a shared secret, the EAP method based on a public key, and EAP methods based either on a secret key or a public key according to the type of credentials used. In this appendix, several features of the well-known EAP methods are described [b-EAP Youm].

III.1 Pre-shared secret-based EAP methods

EAP-MD5: EAP-MD5 is an EAP method of [IETF RFC 3748] whose implementation is mandatory and a typical example of an EAP method based on the shared secret. It is considered one of the simplest EAP methods. The peer and the EAP authentication server share the password in advance. The one-way hash algorithm, MD-5 [b-IETF RFC 1321], is used together with a pre-shared secret and a challenge to compute the hashed value to prove that the peer knows the shared secret.

It does not support mutual authentication, i.e., the authentication server only authenticates the peer. Neither does it generate any keying material as a side effect. Furthermore, it is vulnerable to dictionary attacks as well as the MITM attack. In summary, EAP-MD5 is inherently insecure, and it does not support most of the security requirements for the EAP methods.

LEAP: The Lightweight Extensible Authentication Protocol (LEAP) [b-LEAP] was developed to provide the password-based authentication protocol between the peer and the authentication server. It is considered a challenge-response protocol based on a pre-shared secret or password between the peer and the authentication server.

Unlike EAP-MD5, it supports mutual authentication and session key derivation. Note, however, that it does not provide identity privacy, and it is vulnerable to the dictionary attack.

EAP-AKA: Developed for the 3G cellular network, EAP-AKA [b-IETF RFC 5448] is an EAP method that uses the existing AKA (authentication and key agreement) mechanism developed for authentication and key exchange in the 3G cellular network. AKA is used for mutual authentication and session key derivation based on the shared symmetric key, which can be used to protect the data session in the air interface in 3G cellular networks. On the peer side, it runs in a subscriber identity module, which is either a UMTS subscriber identity module (USIM) or a (removable) user identity module ((R)UIM)) similar to a smart card. In the 3G context, an entity called HLR (home location register) acts as the authentication server; an entity called VLR (visitor location register) acts as authenticator, and a mobile station (MS), as the peer.

Basically, EAP-AKA incorporates AKA into the EAP method to perform authentication and session key derivation as well as optional identity privacy support, optional result indications, and optional fast re-authentication procedure. In addition, the peer is assumed to have access to the subscriber's USIM wherein the shared secret K is kept and the actual AKA protocol is implemented. The master key (MK) is computed from IK (Integrity key), and CK (Cipher key), during the EAP-AKA method run. MK is used to compute the transient EAP session keys (TEKs), MSK, and EMSK.

EAP-SRP: EAP-SRP is based on the SRP (secure remote password) proposed in [b-SRP]. This scheme is known as one of the typical examples of a "strong password protocol" that resists dictionary attacks. Most of the pre-shared secret-based EAP methods are known to be vulnerable to dictionary attacks. Note, however, that EAP-SRP is able to resist dictionary attacks. Basically, the SRP scheme is considered a variant of the DH key exchange scheme, allowing two entities to agree on a common secret key using public key cryptography.

Basically, EAP-SRP incorporates SRP into the EAP method to perform authentication and session key derivation. Note, however, that EAP-SRP is still in the draft document of IETF. In summary, EAP-SRP supports mutual authentication and resists dictionary attacks. Though the Internet draft of EAP-SRP cites the possibility of providing identity privacy via a hidden pseudonym, it is also said to be unable to support strong identity privacy. EAP-SRP can support limited fast reconnect.

EAP-PSK: PSK stands for "pre-shared Key". EAP-PSK supports mutual authentication based on a 16-byte, pre-shared secret between the peer and the EAP server. Mainly designed to be applied in a context with restricted computational resources, especially for mobile terminals in wireless networks, it uses only one primitive cryptographic algorithm: the AES algorithm. There are two types of EAP-PSK methods: standard EAP-PSK and extended EAP-PSK. The standard EAP-PSK method uses the protected channel to transmit a protected result indication, whereas the extended EAP-PSK uses the protected tunnel to transmit arbitrary information of variable length. It is regarded as a typical challenge/response protocol since two parties exchange their nonces, their identities, and a proof of knowledge of the secret. Authentication is enabled by sending the MAC computed with the pre-shared key over the nonces and the identities exchanged in the previous conversation. It is based on AKEP2 (authenticated key exchange protocol 2). It is assumed that two parties should have shared two keys as a prerequisite, a1 and a2, with a1 used for authentication purposes and a2 for session key derivation. It supports mutual authentication, key derivation, and dictionary attack resistance but not identity protection, fast reconnect, and protected ciphersuite negotiation.

III.2 EAP methods based on public key

EAP-TLS: EAP-TLS was published as RFC 2716 in October 1999, which was replaced by [b-IETF RFC 2716]. Considered a mature, stable, and widely deployed EAP method, it relies on transport layer security.

EAP-TLS uses a TLS handshake phase to authenticate the peer and the authentication server. Although the TLS handshake protocol actually sets up a secure tunnel between the peer and the authentication server, this tunnel is not used in the subsequent data session. Instead, since some keying materials are sent to the authenticator, the peer and the authenticator use them to protect the subsequent data session. In EAP-TLS, certificates are used to authenticate the EAP authentication server to the peer including authenticating the peer to the authentication server, albeit optionally. In other words, it supports mutual authentication based on ITU-T X.509 certificates, which results in protection against MITM attacks and use of a rogue network access server. It also generates the symmetric keying material that can be used to protect the subsequent data session. After EAP-TLS is completed, the authentication server and the peer are able to share the pre-master secret. The pre-master secret is used to generate the master secret (MS), which in turn is used to generate MSK and EMSK using the pseudo-random function.

EAP-TLS can be considered a secure EAP method; thus, it is now widely deployed in many applications. It supports fast reconnect since a new security association can be generated by using the existing security association efficiently and fast. In other words, it supports most requirements except channel binding and identity protection. Since EAP-TLS uses certificates, it inherits all certificate-related problems: the problem arising from unencrypted certificates and the problem of postponed verification of the certificate. The first problem occurs when certificates are sent in unencrypted form. This causes the identity contained in the certificate to be revealed to attackers that are able to eavesdrop on the conversation. The second problem arises when the peer is unable to verify the signature or the certificate chain. More specifically, the peer is unable to verify whether the certificate of the authentication server has been revoked in the meantime. Therefore, there are no other means of avoiding the problem except postponing verification.

III.3 EAP methods that support both shared secret and public key

This clause describes EAP methods based on a public key or a shared secret.

EAP-IKEv2: The EAP-IKEv2 [b-IETF RFC 5106] was adopted in February 2008. Based on mechanisms and payloads of IKEv2, this EAP method supports mutual authentication and session key establishment between an EAP server and an EAP peer. For mutual authentication, various authentication techniques are supported according to the type of credential: asymmetric key pairs, symmetric keys, or a combination of both. A different type of authentication credential may be used in each direction. For instance, the EAP server may authenticate itself using public key pairs, while the peer authenticates itself using a symmetric key.

III.4 Tunnel-based EAP methods

This clause describes the EAP-TTLS and EAP-FAST tunnel-based EAP method.

EAP-TTLS: EAP-TTLS is described in [b-IETF RFC 5281]. EAP-TTLS is an EAP (extensible authentication protocol) method based on the TLS (transport layer security) protocol. TTLS stands for "tunnel transport layer security". EAP-TTLS is regarded as an extension to EAP-TLS. Authentication in EAP-TLS is typically mutual, i.e., the authentication server and the peer authenticate each other. It uses the certificate to authenticate the authentication server and a simpler authentication method to authenticate the peer. It consists of two phases: the TLS handshake phase and the TLS tunnel phase. In the first phase, the authentication server is authenticated to the peer using the ITU-T X.509 certificate of the server. After the first phase is completed, the secure tunnel is established. In the second phase, all communications are protected by this secure channel. The client is authenticated to the authentication server by using legacy authentication methods such as clear-text password or challenge-response password or a more advanced authentication mechanism such as token-based authentication. EAP-TTLS supports identity protection since an attacker cannot see the user identity because the identity can be sent in the second phase. Nonetheless, EAP-TTLS is known to be vulnerable to the MITM attack. Specifically, the tunnelled protocols require the session key derived from the first phase, which is used to provide a secure tunnel. In a certain environment, a peer is allowed to skip the first phase and to proceed directly to the second phase. At this time, the active MITM attack may take place if the attacker can hijack a valid authentication session. Note, however, that a cryptographic binding scheme was proposed as a protection against the MITM attack in the tunnel-based EAP method. Therefore, EAP-TTLS can be considered to be secure if cryptographic binding is applied. In addition, the IETF EMU (EAP methods update) working group has been developing the Internet draft on the "requirements of the tunnel-based EAP method" as of December 2008.

EAP-FAST: EAP-FAST [b-IETF RFC 4851] was designed as an alternative to LEAP, which is known to be vulnerable to dictionary attacks. It was originally proposed to reduce the workload of small wireless devices. FAST stands for "flexible authentication via secure tunnelling". The primary design goals of EAP-FAST include mutual authentication, resistance to brute-force dictionary attacks, immunity to the MITM attack, and wide support for existing user databases containing credentials. In general, EAP-FAST uses the TLS handshake protocol to establish a mutually authenticated tunnel between the peer and the authentication server. Unlike EAP-TTLS, however, the secure tunnel can be established using either the public key similar to EAP-TLS or a pre-shared symmetric key known as PAC (protected access credential). PAC can be considered a security token provided to the peer by the server to establish a secure tunnel for future optimized network authentication. EAP-FAST consists of two phases. In the first phase, the peer uses a PAC to establish a secure TLS tunnel. If the peer does not have the corresponding PAC, the server requests the peer to initiate the full TLS handshake. After this full TLS handshake, the peer requests the server to issue a PAC that can be used to establish the TLS tunnel later. In the second phase, EAP-TLS authentication or legacy authentications may be used to authenticate the peer within the secure tunnel. PAC consists of three components: shared secret, opaque element, and other optional

information. The shared secret is used to establish the secure tunnel. The opaque element is provided to the peer and presented to the server when the peer wishes to obtain access to the network resource. The opaque element may include PAC as well as the peer's identity. The server uses a strong cryptographic algorithm to protect the opaque element so that the server may securely identify and authenticate the peer. Other information designed to ensure the integrity of the PAC issuer may be included.

There are three kinds of authentication methods: Certificate-based authentication, that is used in EAP-TLS; combined authentication, that is used in EAP-TTLS; and PAC (protected access credential)-based authentication. In certificate-based authentication, the peer and the authentication server use the certificates to authenticate each other. In PAC-based authentication, the peer uses PAC to establish a TLS tunnel. Therefore, EAP-FAST is considered an efficient EAP method that combines the features of EAP-TLS and EAP-TTLS and adopts the idea of using EAP-TLS with pre-shared key. In summary, EAP-FAST is a very flexible EAP method intended for the constricted mobile device since it supports mutual authentication using a pre-shared key.

Bibliography

[b-IEEE 802.1X] IEEE Std 802.1X-2004, *IEEE Standard for local and metropolitan area networks – Port-based Network Access Control*.

[b-IETF RFC 1321] IETF RFC 1321 (1992), *The MD5 Message-Digest Algorithm*.

[b-IETF RFC 2058] IETF RFC 2058 (1997), *Remote Authentication Dial In User Service (RADIUS)*.

[b-IETF RFC 2904] IETF RFC 2904 (2000), *AAA Authorization Framework*.

[b-IETF RFC 2945] IETF RFC 2945 (2000), *The SRP Authentication and Key Exchange System*.

[b-IETF RFC 3579] IETF RFC 3579 (2003), *RADIUS (Remote Authentication Dial In User Service) – Support For Extensible Authentication Protocol (*EAP*)*.

[b-IETF RFC 3588] IETF RFC 3588 (2003), *Diameter Base Protocol*.

[b-IETF RFC 4764] IETF RFC 4764 (2007), *The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method*.

[b-IETF RFC 4851] IETF RFC 4851 (2007), *The Flexible Authentication via Secure Tunneling – Extensible Authentication Protocol Method (EAP-FAST)*.

[b-IETF RFC 5106] IETF RFC 5106 (2008), *The Extensible Authentication Protocol-Internet Key Exchange Protocol Version 2 (EAP-IKEv2) Method*.

[b-IETF RFC 5247] IETF RFC 5247 (2008), *Extensible Authentication Protocol (EAP) Key Management Framework*.

[b-IETF RFC 5281] IETF RFC 5281 (2008), *Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)*.

[b-IETF RFC 5295] IETF RFC 5295 (2008), *Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)*.

[b-IETF RFC 5296] IETF RFC 5296 (2008), *EAP Extensions for EAP Re-authentication Protocol (ERP)*.

[b-IETF RFC 5433] IETF RFC 5433 (2009), *Extensible Authentication Protocol – Generalized Pre-Shared Key (EAP-GPSK) Method*.

[b-IETF RFC 5448] IETF RFC 5448 (2009), *Improved Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA')*.

[b-3GPP] The 3rd Generation Partnership Project, http://www.3gpp.org/.

[b-3GPP2] The 3rd Generation Partnership Project 2, http://www.3gpp2.org/.

[b-Cam-Winget] Cam-Winget, N., Moore, T., Nancy, Stanley, D., and Walker, J. (2004), *IEEE 802.11i Overview*, WLAN workshop.

[b-EAP Youm] Youm Heung Youl, *Extensible Authentication Protocol Overview and its Applications*, IEICE Trans. on INF. and Syst. vol.E-92 No. 5, May 2009.

[b-LEAP] Cisco, *Dictionary attack of Cisco LEAP*, Technical note, available at http://www.ciscosystems.com/warp/public/707/cisco-sn-20030802-leap.pdf, July 2004.

[b-NIST SP 800-120] NIST SP 800-120, *Recommendation for EAP Methods used in Wireless Network Access Authentication*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2012