International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1124

(11/2007)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Telecommunication security

Authentication architecture for mobile end-to-end communication

Recommendation ITU-T X.1124

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.369
IP-based networks	X.370–X.379
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	X.730–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.889
Generic applications of ASN.1	X.890–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
TELECOMMUNICATION SECURITY	**X.1000–**

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

Recommendation ITU-T X.1124

Authentication architecture for mobile end-to-end communication

Summary

Recommendation ITU-T X.1124 describes a service layer authentication architecture for mobile end-to-end data communication between mobile users and various service providers in the network. The generic negotiation mechanisms and authentication procedures specified in this Recommendation support both those entities that have miscellaneous authentication capabilities and those entities that have differentiated security requirements. The authentication addressed in this Recommendation is used for service providers and requesters and is independent of network access authentication of the mobile users.

Source

Recommendation ITU-T X.1124 was approved on 13 November 2007 by ITU-T Study Group 17 (2005-2008) under Recommendation ITU-T A.8 procedure.

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/.

CONTENTS

Recommendation ITU-T X.1124

Authentication architecture for mobile end-to-end communication

1 Scope

This Recommendation describes service layer authentication architecture in mobile end-to-end data communication between mobile users and various service providers in the network.

This Recommendation applies to three types of entities: mobile terminals in compliance with different mobile communication standards, service authentication-related network elements, and application servers in various networks including mobile networks and open networks. This Recommendation applies to three types of services: the services that are operated by mobile network operators (including the services operated by a visited network, e.g., when a user in a 3rd Generation Partnership Project (3GPP) network uses the service in a 3rd Generation Partnership Project 2 (3GPP2) network); the services provided by application servers on open networks such as the Internet for mobile terminals (e.g., web services and e-mail services); and the services provided by certain powerful mobile users acting as customized service brokers for other mobile users.

This Recommendation provides generic negotiation mechanisms and authentication procedures to support both those entities that have miscellaneous authentication capabilities and those that have differentiated security requirements. The authentication addressed herein is used for service providers and requesters and it is independent of network access authentication of the mobile users. This Recommendation builds upon the work of other standard bodies and consortia to define a more generalized authentication architecture for mobile environments.

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 M.3016] Recommendation ITU-T M.3016 (1998), *TMN Security Overview*.

[ITU-T X.509] Recommendation ITU-T X.509 (2000) | ISO/IEC 9594-8:2001, *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks.*

[ITU-T X.800] Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications.*

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

[ITU-T X.810] Recommendation ITU-T X.810 (1995) | ISO/IEC 10181-1:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Overview.*

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

[ITU-T X.1122] Recommendation ITU-T X.1122 (2004), *Guideline for implementing secure mobile systems based on PKI*.

[IETF RFC 4120] IETF RFC 4120 (2005), *The Kerberos Network Authentication Service (V5)*. <http://www.ietf.org/rfc/rfc4120.txt>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 application server [ITU-T X.1121]: An entity that connects to an open network for data communication with mobile terminals.

3.1.2 authentication [b-ITU-T X.811]: The provision of assurance of the claimed identity of an entity.

3.1.3 authentication information [ITU-T X.800]: Information used to establish the validity of a claimed identity.

3.1.4 certificate repository [ITU-T X.1122]: Database in which the certificates, CRL and other PKI-related information are stored and which is accessible online.

3.1.5 confidentiality [ITU-T X.800]: The property that information is not made available or disclosed to unauthorized individuals, entities or processes.

3.1.6 denial of service [ITU-T X.800]: The prevention of authorized access to resources or the delaying of time-critical operations.

3.1.7 eavesdropping [ITU-T M.3016]: A breach of confidentiality by monitoring communication.

3.1.8 masquerade [ITU-T X.800]: The pretence by an entity to be a different entity. For instance, an authorized entity with few privileges may use a masquerade to obtain extra privileges by impersonating an entity that has those privileges. Types include replay, relay and compromise of claim authentication information.

3.1.9 mobile network [ITU-T X.1121]: A network that provides wireless network access points to mobile terminals.

3.1.10 mobile terminal [ITU-T X.1121]: An entity that has wireless network access function and connects to a mobile network for data communication with application servers or other mobile terminals.

3.1.11 mobile user [ITU-T X.1121]: An entity (person) that uses and operates the mobile terminal for receiving various services from application service providers.

3.1.12 replay [ITU-T X.800]: A message, or part of a message, is repeated to produce unauthorized effect. For example, a valid message containing authentication information may be replayed by another entity in order to authenticate itself (as something that it is not).

3.1.13 trusted third party (TTP) [ITU-T X.810]: A security authority or its agent, trusted by other entities with respect to security-related operations.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 authentication capability information: Authentication mechanisms supported by a service subscriber or a service provider.

3.2.2 authentication mode: An identifier that specifies authentication mechanisms between a service subscriber (SS) and an entity authentication centre (EAC), a service provider (SP) and an

EAC, and authentication inquiring/derived key generation mechanism/mutual authentication mechanisms between an SS and an SP.

3.2.3 authentication negotiation procedure: A procedure that happens during the authentication procedure in which, according to the local policy, the EAC must choose an authentication mode from the authentication mechanisms supported by service providers/service subscribers and the network.

3.2.4 authentication procedure: The process of authentication between a service entity and the EAC, and authentication between two service entities, i.e., an SS and an SP. Generally, a whole authentication procedure comprises three independent integral sub-procedures: initial authentication between service entity and EAC; authentication inquiring and key generation/transportation/negotiation; and mutual authentication between service entities.

3.2.5 challenge/response: A method of protecting against replay attack. For example, if entity A wants to obtain a new message from entity B, it can first send a challenge in the form of a nonce (e.g., a cryptographic value that is used only once) to B. A then receives a response from B, based on the nonce that proves B was the intended recipient.

3.2.6 derived key: A key, indicated by *Ksp*, that is generated during the authentication inquiring and key generation/transportation/negotiation procedure. The key is shared by an SS and an SP, and it is generally derived using shared keying material *Ks*, which is the shared key between the EAC and the SS, and the identity information of service entities. It can be the base of mutual authentication between SS and SP, and be used to derive a following session key *Kt* to protect service communication between the SS and the SP. The length and lifetime of a derived key will be set according to parameters such as service type and security degree. Generally, the algorithm to derive the derived key has a default value.

3.2.7 entity authentication centre (EAC): A central network element defined in the authentication architecture, which accomplishes authentication negotiation and mutual authentication with service entities, establishes shared keying material between service entities, enquires or responds to the inquiring of authentication status of service entities, and helps to generate the derived key for the SS and SP, etc. The function of the EAC may contain other functions, e.g., the bootstrapping function in 3GPP and 3GPP2, the Kerberos server's function, or a certificate verification function.

3.2.8 entity subscription database (ESD): A database-type network element defined in the authentication architecture, which stores service entity's subscription information and authentication information binding with an entity's identifier. In addition, the ESD may have the function to compute necessary authentication data or to achieve such authentication data from another network element, e.g., similar to the function of the authentication centre in 3GPP. The ESD may contain a certificate repository, where the certificates of mobile terminals and network elements in mobile network are stored, and a certificate revocation list (CRL) to perform public key crypto functions.

3.2.9 home network: The mobile network to which the SS subscribes.

3.2.10 interim authentication check identifier (IAC-ID): A temporary identifier for an SP assigned by the EAC during the initial entity authentication of the service provider with the EAC. The identifier can be a local index of the shared keying material between the EAC and the SP. It has a lifetime that is also set by the EAC. When the SP queries the authentication status of a service subscriber from the EAC, its IAC-ID is used to establish its identity.

3.2.11 interim service request identifier (ISR-ID): A temporary identifier for an SS assigned by the EAC during the initial entity authentication of a service subscriber with the EAC. When an SS requests service from a service provider, its ISR-ID is used to establish the SS's identity. It can also act as an index of shared keying material between the EAC and the SS.

3.2.12 policy: A set of rules defined locally by a network element or other network management authority.

NOTE – There could be various policies regarding the enforcement or implementation of this Recommendation.

3.2.13 private entity identifier (PID): A permanent identifier that represents the real identity of an SS. It should be defined to be unique and can be used by the network operator to authenticate, account and manage the SS. The private information of the SS can only be obtained by the EAC and the ESD. An example of a PID is the international mobile subscriber identity (IMSI) in a 3rd Generation Partnership Project (3GPP) network.

3.2.14 public entity identifier (UID): The identifier of an SP that is used for public access of the SP by the EAC and the SS.

3.2.15 service authentication proxy (SAP): A network element that is deployed to transmit the authentication information between the visited SP and the SS's home EAC when an SS uses the service provided by an SP in the visited network which has had a subscription relationship with the SS's home network.

3.2.16 service entity: A service subscriber or a service provider.

3.2.17 service provider (SP): A network entity that is capable of providing services for mobile users. It may be an application server in an operator network or in open network. It may also be a mobile user that has the facilities to provide services to other mobile users.

3.2.18 service subscriber (SS): A network entity that is capable of requesting subscribed services from an SP. Generally, the SS is a mobile user but it may also be an application server that acts as a retailer to obtain service resources for mobile users.

3.2.19 security degree: An identifier (e.g., number) that represents a set of security parameters including at least one authentication mechanism, the crypto algorithms and related parameters to reflect the security requirement of a certain service. It is defined to profile the security requirement of each service.

3.2.20 service type: The particular kind or category of service provided by the SP and consumed by the SS. There are several kinds of service and each service belongs to one service type, e.g., in 3GPP, there are service types of unspecific service, PKI-Portal, presence and MBMS, etc.

3.2.21 shared keying material: Key data which is generated during mutual authentication procedure of the SS (or SP) and the EAC, and which is used to protect the security communication of the reference point connecting the EAC and the SS (or SP). The shared keying material between the SS and the EAC is denoted by *Ks* and the shared keying material between the SP and the EAC is denoted by *Kp*. Its key length, generation algorithm and lifetime are set according to some other parameters, such as service type and security degree.

3.2.22 subscription information: The information that reflects the subscribing relationship among an SS, an SP and the relying mobile network. Subscription information between a service subscriber and its home network contains the subscriber's private entity identifier (e.g., PID), the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (i.e., whether it is allowed to request a specific service), the supported authentication mechanism(s) (e.g., HTTP authentication and key agreement, Diffie-Hellman based authentication mechanisms, a biometric authentication mechanism, etc.), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc. Subscription information between a service provider and a mobile network contains the provider's identity information and public entity identifier (e.g., UID) according to the service, optionally the basic key material (e.g., a shared secret or a public-key certificate) and its lifetime, entity service permission flag (e.g., whether it is allowed to provide a specific service), the supported authentication

mechanisms (e.g., certificate-based TLS authentication mechanism, PSK-TLS, IPSec), and the authentication inquiring and key generation mechanism (e.g., GBA, Kerberos, Mediation), etc.

3.2.23 user universal identification module (UUIM): The logical entity in the mobile terminal containing the permanent keys and the functions required to perform the mobile user authentication to the mobile network. For example, UUIM can be one or more IMSI/USIM applications in 3GPP. UUIM is technically equivalent to UICC in 3GPP specifications or UIM in 3GPP2 specifications.

3.2.24 visited network: A mobile network, to which an SS does not subscribe, which has a roaming relationship with the SS's home network. A visited network can also provide services for the SS.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AAA	Authentication, Authorization and Accounting
AKA	Authentication and Key Agreement
AS	Application Server
CAVE	Cellular Authentication and Voice Encryption
CK	Cipher Key
CRL	Certificate Revocation List
EAC	Entity Authentication Centre
ESD	Entity Subscription Database
GBA	Generic Bootstrapping Architecture
HTTP	HyperText Transfer Protocol
IAC-ID	Interim Authentication Check Identifier
IMSI	International Mobile Subscriber Identity
IK	Integrity Key
IKE	Internet Key Exchange
IMPI	Internet Protocol Multimedia Private Identity
IPSec	Internet Protocol Security
ISIM	Internet Protocol Multimedia Services Identity Module
ISR-ID	Interim Service Request Identifier
Ks	shared Key between SS and EAC
Ksp	shared Key between SS and SP
Kt	denote Key for session communication
MBMS	Multimedia Broadcast/Multicast Service
MN	Mobile Node
MTK	MBMS Traffic Key
NAF	Network Application Function

NAI	Network Access Identifier
PID	Private Entity Identifier
PKI	Public-Key Infrastructure
PSK-TLS	Pre-Shared Key Ciphersuites for Transport Layer Security
RES	a 3GPP authentication parameter
SAP	Service Authentication Proxy
SGT	Service Granted Ticket
SIM	Subscriber Identity Module
SP	Service Provider
SS	Service Subscriber
TLS	Transport Layer Security
TLS-Cert	Transport Layer Security based on Certificate
TLS-PSK	Transport Layer Security based on Pre-Shared Key
TTP	Trusted Third Party
UE	User Equipment
UICC	Universal Integrated Circuit Card
UID	Public Entity Identifier
UMTS	Universal Mobile Telecommunication System
USIM	Universal Subscriber Identity Module
UUIM	User Universal Identification Module
XRES	a 3GPP authentication parameter

5 Conventions

None.

6 Overview

6.1 Use case description

When providing certain services to a mobile user, most of the application servers must first establish a mutual trust relationship with the user (e.g., the relationship between a mobile user and an authentication proxy, a mobile user and a PKI system, or a mobile user and an application server). It is through mutual authentication that the trust relationship between a mobile user and an application server is established. With the development of the mobile network, there are more and more types of service, and the service provider can be not only a network operator, but also a third-party service provider, or even a mobile user. That is, some mobile users cannot only use the services provided by a mobile operator, but can also provide some services for other users.

This Recommendation defines three types of service provider (SP): mobile operators; third-party service providers; and mobile users. The service subscriber (SS) can be a mobile user or a third party application server (AS).

In order to establish a mutual trust relationship between various types of service entity it is necessary to have a common authentication architecture that is applicable to different standards for mobile networks. This Recommendation defines a common architecture to meet this requirement.

NOTE – The service entities may be located in the same network or different networks, e.g., one service entity may be in 3GPP network, and another in 3GPP2 network.

6.2 Security considerations

In the authentication architecture, the following security threats are considered, which are derived from the security threats defined in [ITU-T X.805]:

– Eavesdropping: Intruders obtain transmitted data without authorization by monitoring the transmission media. This is the most common type of interception attack method. In mobile communication, this threat results from the open nature of the wireless interface. Intruders can acquire authentication information and identity information by intercepting the radio signals and decoding the transmissions.

– Masquerade: The pretence by an entity to be a different entity. Intruders impersonate a legitimate user to access a resource to obtain a benefit or for other unauthorized purposes. Masquerading as a legitimate user is easier in mobile communication than in wired communication. With a masquerade attack, an intruder fools the system into believing he/she is a legitimate user, thereby gaining access to system services and confidential information.

– Unauthorized message manipulation: Intruders may modify, insert, replay or delete authentication data on radio interface, including both accidental and deliberate manipulation.

– Replay attack: Intruders can copy or relay intercepted messages but may not be able to decrypt them. The re-transmission of such messages can be used to gain unauthorized access to a network or resources.

– Denial of service attack: The prevention of authorized access to resources or the delaying of time-critical operations. For example, an intruder may prevent user authentication data from being transmitted on the radio interface by physical means or by inducing protocol failures.

In this architecture the following security requirements are considered:

– Entity identity confidentiality: The property that the permanent identity (e.g., international mobile subscriber identity, i.e., IMSI in 3GPP) of a user to whom a service is delivered cannot be compromised by eavesdropping on the radio access link. To achieve these objectives, the service entity (i.e., an SS or SP) is normally identified by a temporary identity allocated by the network. To avoid entity traceability, which may lead to compromise of entity identity confidentiality, the entity should not be identified for a long period by means of the same temporary identity.

– Entity authentication: The entity authentication process contains service entity authentication and EAC authentication, i.e., the EAC validates the identity of the entity and the service entity validates that the EAC is legitimate.

In order to achieve these objectives, before obtaining service, the service subscriber must perform an entity authentication procedure that includes the following two steps. The first step is authentication between the service entity and the EAC, i.e., the service entity and the EAC validate identities mutually. The second step is authentication between entities: the service entities validate identities mutually. The authentication procedure between the service entity and the EAC is specified in clauses 8.2 and 8.3. The authentication between service entities is specified in clauses 8.4 and 8.5.

The service entity and the EAC make an agreement on their shared key and securely negotiate the derived key generation algorithm used subsequently.

The negotiation of the shared keying material is accomplished in an authentication procedure between the entity and the EAC (see clause 8.2). The negotiation of the derived key generation algorithm depends on the security requirement of the service requested by the entity (see clause 8.4).

7 Security architecture

7.1 Authentication model

Figure 1 shows an authentication reference model for mobile end-to-end data communication.

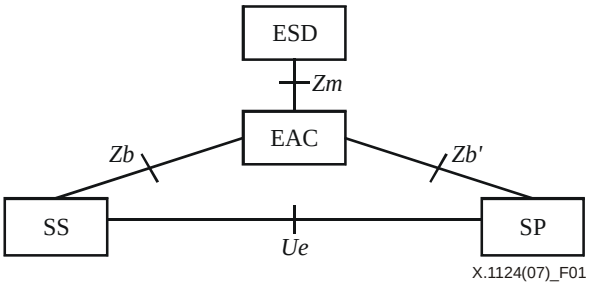

Figure 1 – Authentication model in mobile end-to-end data communication

Figure 1 shows the relationship between service entities (SS and SP), the entity subscription database (ESD) and the entity authentication centre (EAC) in the operator's network.

If the SS wants to use the service provided by the SP in the visited network, which has a subscription relationship with the SS's home network, then a service authentication proxy (SAP) must transmit the authentication information between the visited SP and the SS's home EAC. Figure 2 shows an authentication reference model for mobile end-to-end data communication.

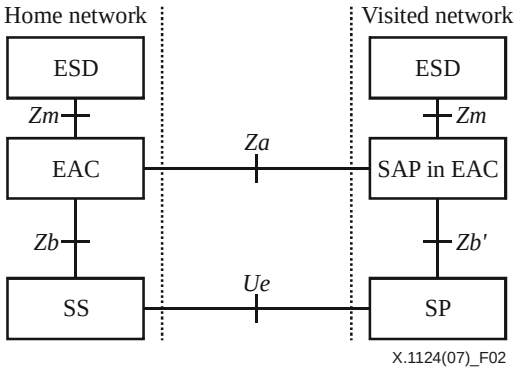

Figure 2 – Authentication model for accessing service in visited network

When the visited network has an EAC, the SAP may be co-located with the EAC. When the visited network has not deployed an EAC, it needs at least an SAP which is a proxy of the visited EAC. The SP can communicate with the home EAC via the SAP.

Requirements of the authentication model for accessing services in a visited network (also called the inter-network case) are as follows:

1) The inter-network service subscriber can perform the authentication and make a service request through the EAC in the home network.

2) The home network can control whether the service subscriber is authorized to use the services in the visited network.

The inter-network SS always negotiates the authentication mode with the home EAC and performs the authentication procedure. The home EAC should send the authentication mode that has been negotiated to the SAP in the visited EAC, and initiate authentication negotiation and authentication between the SAP in the visited EAC and the visited SP. The home EAC must generate trustworthy proof for the inter-network SS and the visited SP, then it must secure transport to the visited SP over the *Za* interface and the SAP in the visited EAC.

7.2 Network elements

7.2.1 Entity subscription database (ESD)

Every entity must have a subscription relationship with network operators. The subscription information is stored in the ESD, where the authentication information is stored binding with the corresponding entity identity.

The requirements for ESD include:

– The ESD must store users' subscription information securely over a long period of time.

– The ESD must store different entities' authentication information according to different authentication mechanisms as specified in the subscription information. For example, if the authentication mechanism is a symmetrical key mechanism, it should store the shared secret information between entities and network; if the authentication is a public key certificate mechanism, it should store the certificate repository and certificate revocation list (see [ITU-T X.509]).

– The ESD must update subscription information in a timely manner.

– The ESD must send the response to initiate an update or repeal of subscription information to the service entity.

– The ESD must accomplish the calculation function needed in an authentication procedure if it has the authentication centre function. Otherwise, it can obtain the relative authentication data from the authentication centre.

7.2.2 Entity authentication centre (EAC)

When each service entity subscribes or provides service, it should first contact the EAC to negotiate the authentication mechanism and accomplish the authentication procedure. The authentication procedure must generate a shared key *Ks*. If authentication is successful, the EAC must assign a temporary identifier for each entity (ISR-ID or IAC-ID).

Then, the EAC must store *Ks/Kp* and the temporary identifiers in a local database, and set a lifetime for them.

The requirements for the EAC include:

– The EAC must obtain subscription information from the ESD according to the user's identifier.

– The EAC must judge whether to accept an entity's authentication request according to the subscription information.

– The EAC must choose a proper authentication mechanism according to the local policy, and can accomplish authentication procedure with entities.

– If the authentication is successful, the EAC must assign a temporary entity identifier and generate shared keying material *Ks/Kp* for each entity, and set a lifetime for it.
– If *Ks/Kp* or a temporary entity identifier expires, the EAC must instruct the entity to initiate a re-authentication procedure.
– The EAC must collocate and manage whether several service communications can use the shared keying material generated from one authentication.

The EAC allows the *Ks/Kp* generated from one authentication procedure to be used by several services. However, it is unreasonable to use the same *Ks* to derive keys for several services (the derived key derived from the *Ks/Kp* is a shared key between the service subscriber and the service provider). For example, it is not reasonable or secure to share the same authentication result for the mobile operator or the third party AS, and mobile terminals acting as a service provider. An operator's AS, having a very high security requirement, should not share a *Ks/Kp* created from authentication with other entities.

Entities having the same security requirements and background can be allotted into one group, which can share an authentication result. When the group member first comes to request an ISR-ID, the EAC must link the ISR-ID with the group. Then if any group member requires authentication information, it can receive a correct response. If the entity does not belong to this group, the EAC must inform it that the ISR-ID is invalid, and ask it to request the service subscriber to initiate re-authentication with the EAC.

7.2.3 Service authentication proxy (SAP)

If an SS sends a service request to an SP that is in a mobile network other than the home mobile network, a service authentication proxy (SAP) of the SP's network must be used for this visited SP to communicate with SS's home EAC. The SAP is a sub-function of the EAC.

NOTE – The SAP may be a separate network element, or may be a part of any network element in the visited network that it can implement authentication proxy functionality (an example of such a network element is the EAC of the network to which the visited SP belongs, or an AAA server).

The requirements for the SAP include:

– The SAP must be able to authenticate the visited SP.
– The SAP must be able to locate an SS's home EAC and communicate with it over a secure channel.
– The SAP must be able to transmit the authentication information, keying material and other relative information between the visited SP and the home EAC.
– The SAP may generate charging information in an inter-network situation and send it to relevant charging servers.

7.2.4 Service subscriber (SS)

The service subscriber can only apply for services. Generally, an SS is simply a mobile user.

The requirements for the SS include:

– The SS must support one or several authentication mechanisms, can initiate the authentication procedure with the EAC, and can identify itself with its identifier in the authentication procedure.
– When receiving a re-authentication request from the EAC/SP, the SS can initiate the re-authentication procedure and can identify itself with a temporary identifier. When the SS finds that keying material has expired, it must initiate a re-authentication request to the EAC.

– The SS must make use of the derived key to accomplish further authentication with the service provider, and then start a secure communication.
– The SS's function can be performed in mobile equipment or an integrated circuit card that is contained in mobile equipment.

7.2.5 Service provider (SP)

The service provider may be an application server in an operator network or other network.

The requirements for an SP include:

– The SP must support one or several authentication mechanisms, can perform initial authentication procedure with the EAC, and can identify itself by user identifier.
– When receiving a re-authentication request from the EAC, the SP can initiate a re-authentication procedure, and identify itself with a temporary identifier. When the SP finds that keying material has expired, it must initiate a re-authentication request to the EAC.
– The SP can inquire about a service subscriber's authentication status.
– The SP can make use of the derived key to accomplish further authentication with a service subscriber, and then start secure communication.
– The SP can initiate a subscription procedure to the ESD by the EAC.

7.3 Reference points

7.3.1 *Zm* reference point

Zm is the interface between the EAC and the ESD where the EAC can obtain an entity's subscription information from the ESD. The subscription information includes the user's identity information, the subscribed service type and the security degree requirement of the service.

The requirements for a *Zm* reference point include:

– The communication at a *Zm* reference point must be secure.
– All of the communications at a *Zm* reference point must be initiated by the EAC.

7.3.2 *Zb* reference point

Zb and *Zb'* are interfaces between a service entity and the EAC. *Zb* contains two parts, the common part, which is shared with the *Zb'* reference point (see *Zb'* below) and the separate part, which is not shared with the *Zb'* reference point.

The common part of *Zb* is the interface for authentication mechanism negotiation and mutual authentication between any service entity and the EAC.

Over the common part of *Zb*, the EAC can choose a proper authentication mechanism according to the local policies and the authentication mechanism supported by the network and entities. Then the EAC can complete an authentication with entities.

Over the separate part of *Zb*, the SS can initiate a service request, which may be transmitted to the SP by the EAC.

The requirements for the *Zb* reference point include:

– The EAC can identify service entities.
– Shared keying material must be generated during the mutual authentication between the EAC and service entities to secure the communication over the *Zb* reference point.
– The EAC must set the lifetime of the shared keying material and temporary identifiers for the SS. The SS may be able to apply a service granted ticket (SGT) towards EAC, see also clause 8.2.

– The SS can initiate a service request which is carried to the SP by the EAC.

7.3.3 *Zb'* reference point

Zb' is the interface between the SP and the EAC (or SAP, if the SP is in another operator's network). The security of the interface may be secured by IPSec protocol. Like *Zb*, *Zb'* also contains two parts, the common part, which is shared with the *Zb* reference point, and the separate part, which is not shared with the *Zb* reference point.

The common part of *Zb'* is the interface for authentication mechanism negotiation and mutual authentication between all types of service entities and the EAC.

Over the common part of *Zb'*, the EAC can choose a proper authentication mechanism according to the local policies and the authentication mechanism supported by the network and the entities. Then the EAC can authenticate the entities.

Over the separate part of *Zb'*, the SP can enquire about the service subscriber's authentication status through *Zb*.

Requirements for the *Zb'* reference point include:

– The EAC must be able to identify service entities.

– During the mutual authentication procedure between the EAC and service entities, shared keying material must be generated to secure the communication over the *Zb'* reference point.

– The SP may be able to enquire about the SS's authentication status from the EAC.

– The EAC must set the lifetime of the shared keying material and any temporary identifiers for service entities.

– The SP may be able to retrieve part or all of the SS's subscription information from the EAC.

7.3.4 *Ue* reference point

Ue is the interface between service subscribers and service providers. Certain kinds of application protocol can be run on the *Ue* interface. Communication on this reference point is protected by the derived key *Ksp* negotiated between the EAC and the service subscriber.

NOTE – For example, the *Ksp* may be mapped to Ks_ext_NAF, Ks_int_NAF in 3GPP GBA.

7.3.5 *Za* reference point

Za is the interface between the SAP in the visited network and the home EAC. The *Za* reference point between the SAP and the home EAC should be secured (e.g., by using TLS protocols).

NOTE – The mapping of the reference points to those in 3GPP/3GPP2 is provided in Appendix IV.

7.4 Requirements for authentication information

7.4.1 Requirements for security of authentication information

The requirements for security of authentication information include:

– The secret information in an authentication exchange (e.g., shared keying material between the SS/SP and the EAC, a shared derived key between the SS and the SP, a temporary entity identifier of the service entity, lifetime, etc.) must be stored in the secure medium of the SS, the SP and the EAC.

– Subscription information must be securely stored at the ESD. Only the EAC can inquire about subscription information in the ESD.

7.4.2 Requirements for ISR-ID

– The ISR-ID must be globally unique. It can contain the identifier of the SS's home network and home EAC. In order to achieve global uniqueness of the ISR-ID, it is necessary to have a specific field in the ISR-ID which is assigned to a specific mobile operator. This is out of the scope of this Recommendation.

– The ISR-ID must have a specified lifetime. The lifetime may be defined by the service type and the security degree.

– The service provider must be able to detect the home network and the service subscriber's EAC according to the ISR-ID.

7.4.3 Requirements for the IAC-ID

– The IAC-ID must be globally unique. It can contain the identifier of the SP's home network and the identifier of the home EAC. In order to achieve global uniqueness of the IAC-ID, it is necessary to have a specific field in the ISR-ID which is assigned to a specific mobile operator.

– The IAC-ID must have a specified lifetime. The lifetime may be determined by the service type and the security degree.

– The EAC must be able to detect the EAC of the home network and of the service provider based on the IAC-ID.

7.4.4 Requirements for PID

– The PID must be globally unique. It can contain the home network identifier of the SS/SP, for example: IMSI of 3GPP mobile terminal, IMPI of IMS' user, NAI of WLAN's user.

– The PID must be permanent.

– The EAC can distinguish different types of subscribers' PID (e.g., mobile users' PID or PID of the third party AS).

7.4.5 Requirements for UID

– The UID must be globally unique.

– The UID must be permanent.

– The same service entity must have different UIDs corresponding to different services, that is, different services can be distinguished by their UID.

7.5 Key structure

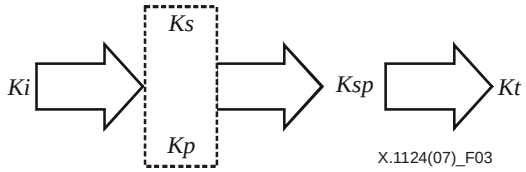

Figure 3 – Key material structure

The keying materials associated with this end-to-end authentication architecture are divided into four layers.

The first layer keying material is the basic keying material shared between a service entity and the network or public key pair of the service entity and the EAC, which may be permanent or updated periodically. Using the first layer keying material, the mutual authentication between the service entity and the EAC is performed and the second layer material is generated.

The second layer keying material is the shared keying material generated during the authentication procedure between the service entity and the EAC. A temporary entity identifier ISR-ID/IAC-ID is assigned to the service entity as an index of the shared keying material *Ks/Kp*. The shared keying material may be a symmetrical key and some other material (e.g., a shared cipher algorithm, a data compress algorithm, a security association). The shared keying material and the temporary identifier have a common lifetime. When they are about to expire or have expired, a re-authentication procedure must be performed, and new shared keying material and a new temporary identifier must be generated.

The third layer keying material is the derived key *Ksp* shared between the SS and the SP during the service request and response procedure. Usually (but not always) it is derived from *Ks* and some identity information, and then transmitted to the SP in cipher. The *Ksp* should be stored with the SP's UID and the SS's ISR-ID together in both sides. The *Ksp* may be reused during its lifetime, and it should expire when the *Ks* expires, if derived from the *Ks*.

The fourth layer keying material (e.g., *Kt* in Appendix II) may be derived from the *Ksp* for protecting the service communication between the SS and the SP. Appendix II gives an example for the generation of this kind of keying material.

8 Authentication procedures

8.1 Authentication procedures overview

In a mobile network, before an SS is ready to use a service provided by an SP, it should have a subscription relationship with the service. Since the SS is a user of the mobile network, the SS and the SP that provides a service to the SS must be authenticated and controlled by the mobile network (represented by the EAC) for security considerations.

The overall authentication procedure is specified as follows, i.e., phases a-e:

a) Authentication preparation

Before a service entity requests or provides a service to other entities, the service entity must first have a subscription relationship with the mobile network operator, and the subscription information must be stored in the ESD. The subscription mechanism may be one of two kinds of mechanisms: offline-subscribe and self-subscribe.

The offline-subscribe mechanism: A mobile user can purchase a smart card and then establish a subscription relationship with the mobile network; a third application server can contract with the mobile network and also establish a subscription. The related subscription information will be stored in the entity subscription database (ESD).

The self-subscribe mechanism: The AS sends a *SubscriptionRequest* message to the mobile network. The message contains the AS's identity and the type of service that the AS will provide. After receiving the message, the mobile network confirms the AS's authentication status (i.e., whether it has been authenticated or not) according to the service type. Then it performs the authentication procedure with the AS, using the authentication mechanism according to the service type. If successful, the mobile network generates subscription information for the AS and sends the subscription information to the entity subscription database (ESD). When the ESD receives the subscription information, it stores it and sends a subscription confirmed response to the AS directly or via the mobile network.

b) Authentication Mode (AM) negotiation and initial entity authentication

Authentication mode negotiation

Once the subscription relationship has been established, if the service entity needs to request or provide services to other entities, it must first negotiate the authentication mode with the EAC. The authentication mode contains: the authentication mechanism between the SS and the EAC; the

authentication mechanism between the SP and the EAC; the authentication enquiring and derived key generation mechanism; and the mutual authentication mechanism between the SS and the SP. The specific description of each authentication mechanism is as follows:

The authentication mechanism between the SS and the EAC: The SS can use many authentication methods or protocols according to the service type, the requirements (e.g., security requirements) of the requested SP or services, or the SS's capability to perform authentication with the EAC.

The authentication mechanism between the SP and the EAC: The SP can use many authentication methods or protocols that will be negotiated by the SS and the EAC to perform the authentication with the EAC.

The authentication enquiring and derived key generation mechanism: Based on three basic types of optional model: the generic bootstrapping architecture (GBA) model, the Kerberos model (see [IETF RFC 4120]) and the Mediation model. The specific procedure for authentication enquiring and derived key generation can be illustrated respectively in the following figures:

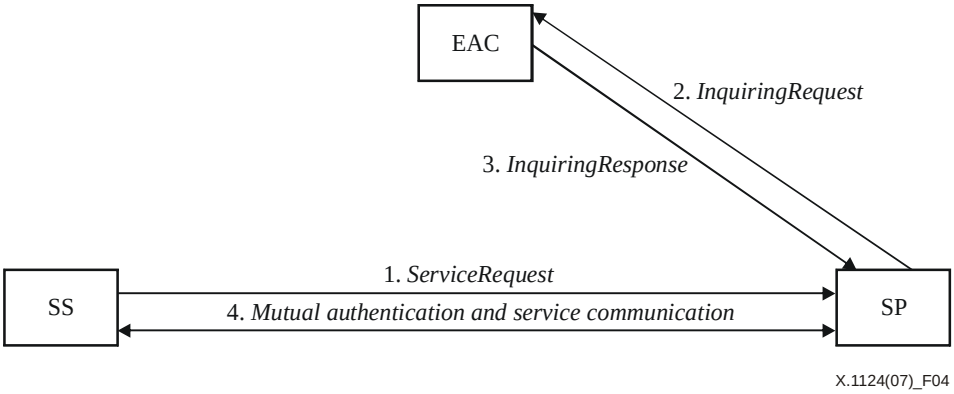

Figure 4 – Authentication enquiring and derived key generation mechanism based on GBA

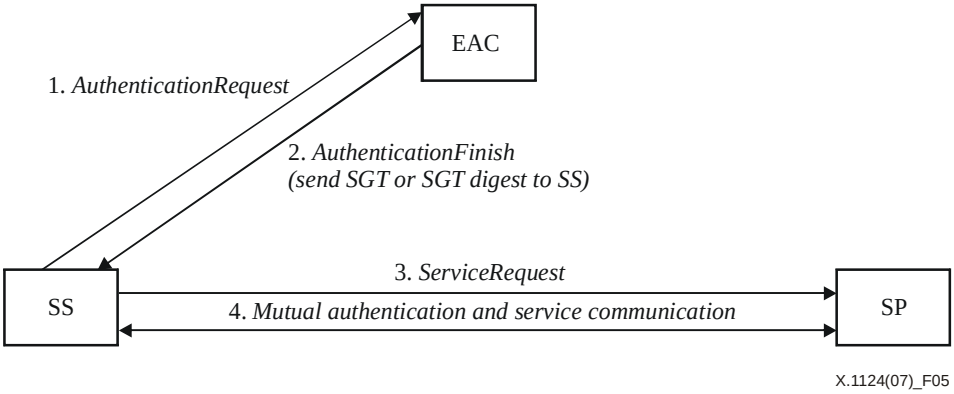

Figure 5 – Authentication inquiring and derived key transportation mechanism based on Kerberos

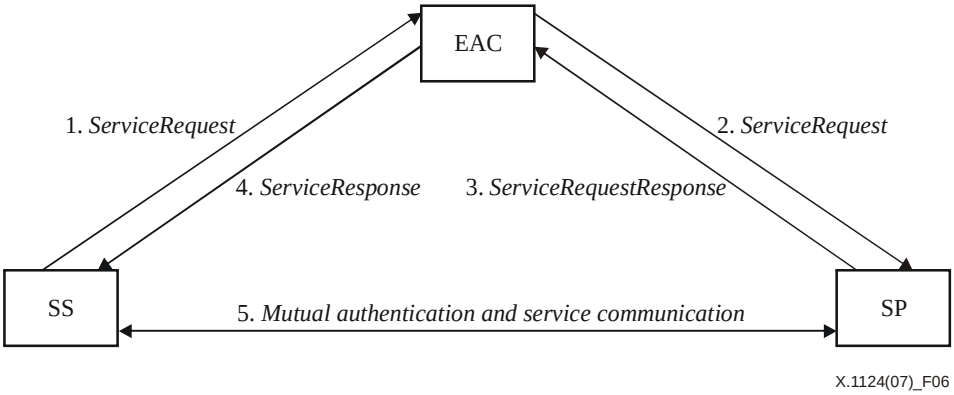

Figure 6 – Authentication inquiring and derived key agreement mechanism based on Mediation

Mutual authentication between the SS and the SP: When the SS and the SP have shared a derived key *Ksp*, they can mutually authenticate using the *Ksp* and generate the session key which can protect the current communication.

In this Recommendation, four kinds of essential authentication mode are specified: E2E_3G_GAA; E2E_KERBEROS; E2E_Mediation; and E2E_TLS. The specific description of each authentication mode is shown as follows.

The characteristics of E2E_3G_GAA are:

– The authentication mechanism between the SS and the EAC should be: SIM, AKA, CAVE, MN-AAA key, TLS-PSK or TLS-Cert.

– The authentication mechanism between the SP and the EAC should be: TLS, IKE or to be determined later.

– The authentication enquiring and derived key generation mechanism should be: GBA.

– The mutual authentication between the SS and the SP should be: TLS-PSK or TLS-Cert.

The characteristics of E2E_KERBEROS are:

– The authentication mechanism between the SS and the EAC should be: The same as E2E_3G_GAA and after authentication has succeeded; the EAC will generate a ticket SGT and send it to the service entity.

– The authentication mechanism between the SP and the EAC should be: NULL, TLS, IKE.

– The authentication enquiring and derived key generation mechanism should be: Kerberos.

– The mutual authentication between the SS and the SP should be: NULL, TLS-KBR5.

The characteristics of E2E_Mediation are:

– The authentication mechanism between the SS and the EAC should be: The same as E2E_3G_GAA and also may be IKE.

– The authentication mechanism between the SP and the EAC should be: The same as E2E_3G_GAA.

– The authentication enquiring and derived key generation mechanism should be: Mediation.

– The mutual authentication between the SS and the SP should be: TLS-PSK.

The characteristics of E2E_TLS are:

– The authentication mechanism between the SS and the EAC should be: NULL.

– The authentication mechanism between the SP and the EAC should be: NULL.

– The authentication enquiring and derived key generation mechanism should be: NULL.

– The mutual authentication between SS and SP should be: TLS-Cert, TLS-PSK.

Initial entity authentication

After the SS and the EAC have negotiated the authentication mode, the SS/SP will perform the initial entity authentication, according to the authentication mechanisms negotiated or agreed in the authentication mode. If the authentication succeeds, then an ISR-ID/IAC-ID will be assigned to the entity and keying material (*Ks/Kp*) to be shared with the EAC, and the service entity will be generated, as specified in clause 8.2.

c) Entity re-authentication: When the initial entity authentication fails or the key is expired, the service entity or the EAC can initiate re-authentication, as specified in clause 8.3. Other reasons for re-authentication are specified in clause 8.3.

d) Authentication enquiring and key exchange: After the SS/SP is mutually authenticated by the EAC, as described in the previous step, when the SS requests a service from the SP, an authentication enquiring and key generation procedure must be performed. For the authentication enquiring and key generation procedure, this Recommendation provides three kinds of optional mechanism: authentication inquiring and key generation model – based on GBA; authentication enquiring and key transport model – based on Kerberos; and authentication enquiring and key agreement model – based on Mediation.

   When the service entity sends an enquire request to the EAC, the EAC can check the validity of the SS's and the SP's temporary identifier and query the corresponding authentication information to confirm the right of the entity to use the service. If the service entity is legitimate, i.e., if the authentication enquiring procedure is successful, the SS and the SP will share a derived key *Ksp*, as specified in clause 8.4.

e) Mutual authentication between entities: The derived key *Ksp* shared between the SS and the SP can be used for several types of service request during its lifetime. It will be used as pre-shared key material by the SS and the SP to do further mutual authentication and to generate a new session key for the protection of each time service communication, as specified in clause 8.5.

8.2 Entity initial authentication procedure

Before communicating with other service entities, every service entity (SS or SP) should have completed, at least once, an initial authentication procedure for each type of requested service. The other case for the start of the procedure is the finding of an invalid temporary entity identifier or *Ks* or *Kp* of the service entity. The specific procedure is as follows, see Figure 7.

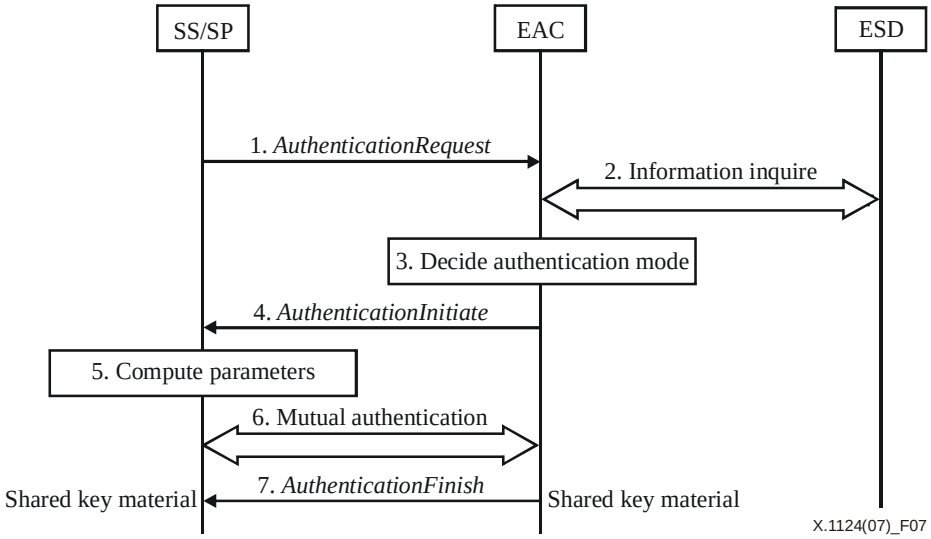

Figure 7 – Initial authentication procedure between a service entity and EAC

1) The service entity (SS or SP) sends an *AuthenticationRequest* message to the EAC, including its private entity identifier (i.e., PID), authentication capability information such as the authentication mechanisms supported by the service entity, a service security degree to reflect the security requirement of the specific service, the public entity identifier (i.e., UID) of the SP providing the requested service if the service entity is an SS, and optionally the SS's *ServiceRequest* message, if the service entity is an SS.

   NOTE 1 – The security-related information for services, which may be called the security degree list for services, must be stored in the ESD on the network side and could be also stored in the service entities.

   NOTE 2 – In the *AuthenticationRequest*, the PID may also be encrypted by a particular algorithm. When the EAC receives the encrypted PID, it will send it to the ESD which can then decrypt this encrypted PID. Finally, the ESD sends the PID to the EAC. This scheme protects the PID by not sending it in clear text.

2) After receiving the message *AuthenticationRequest*, the EAC must fetch the subscription information of the service entity from the ESD according to the entity's PID. After enquiring, the EAC will know whether the service entity may request or provide the service.

   If the entity is the SP, this step could be omitted.

   NOTE 3 – This Recommendation assumes that an SP can only perform the entity initial authentication passively, which means being triggered by a particular service request from an SS. Therefore, the EAC knows the related information about the SP already.

   NOTE 4 – When the service entity supports only one authentication mode that is also supported by the EAC, there is no need for the authentication mode negotiation process and they will simply perform the mutual authentication using this mode. The mode is called the agreed or default authentication mode. In this case, the necessary authentication parameters may be also exchanged here.

3) If the service entity is an SS and there is no agreed default value, the EAC will consider the authentication capabilities of the SS, the SP and the network, the requested service type and the service security degree, and then use its local policy to decide the authentication mode and its related parameters.

   The authentication mechanism between the SS and the EAC in authentication mode and the related parameters are determined as follows: the EAC first acquires the SP's authentication capability information from the requested SP's UID, then combines the factor of the authentication capability of the SS and the network, and finally EAC selects various

authentication mechanisms, their crypto algorithms and other parameters according to the service security degree parameter carried in the message *AuthenticationRequest*. The EAC confirms the security degree of the authentication mode and parameter which, by the local policy, should not lower the requirement of the service security degree.

NOTE 5 – If the service entity is an SP, the EAC will perform the following steps of the initial authentication procedure using the authentication mode information (including the authentication mode and the related information, e.g., encryption parameters) that has been previously negotiated with an SS that has previously initiated the SP's initial entity authentication procedure.

NOTE 6 – If the EAC cannot choose a proper authentication mode that can satisfy the requirement of the service security degree, it will return a failure notice to the SS and the procedure will end.

4) The EAC sends back an initial authentication message *AuthenticationInitiate*. The message may include the authentication mode information decided by the EAC.

If, according to the authentication mode information, the authentication mechanism between the SS/SP and the EAC requires that the first authentication message of the mechanism is sent by the EAC, then the message *AuthenticationInitiate* should also contain the content of the first authentication message required by the mechanism.

5) After receiving the message from the EAC, the service entity will know what authentication mode and related authentication parameters should be used afterwards. In particular, it knows the authentication mechanism between itself and the EAC. Then the service entity will compute the initial authentication parameters if the authentication mechanism is initiated by itself. If the authentication mechanism is initiated by the EAC, then the service entity has received the initial authentication parameters in the *AuthenticationInitiate* and it will compute the corresponding authentication response parameters.

6) According to the negotiated or agreed authentication mechanism, the service entity and the EAC will perform the following mutual authentication steps specified by the authentication mechanism protocol. When the authentication succeeds, the entity and the EAC will share some kind of keying material (i.e., *Ks* if the entity is an SS, or *Kp* if the entity is an SP).

NOTE 7 – If the authentication fails, the initial entity authentication procedure ends here.

7) If the mutual authentication succeeds, the EAC will assign a temporary identifier (i.e., ISR-ID or IAC-ID) and a lifetime of the shared keying material (*Ks* or *Kp*) for the service entity, and then send them to the service entity in an authentication finish message *AuthenticationFinish* which may also carry the very last message of the mutual authentication.

After sending and receiving the message, both the EAC and the service entity will store the temporary identifier, the shared keying materials, their lifetime, the selected authentication mode and all other related information about the service entity. The state of the temporary identifier and the keying materials are also stored.

NOTE 8 – If the authentication mode was determined to be E2E_KERBEROS, then before sending the message *AuthenticationFinish*, the EAC will also generate a service granted ticket (SGT) for the SS (note that the service entity must not be an SP). The SGT will also be carried in the message *AuthenticationFinish*. The SGT contains at least {*Ksp*: ISR-ID of the SS: UID of the SP: lifetime: timestamp}, where the *Ksp* is a derived key generated by EAC shared between the SS and the SP for protection of the communication between them, and the lifetime is the valid time period of the *Ksp* which is assigned by the EAC. The generation parameters of the *Ksp* generally include the *Ks*, the SP's UID and the SS's PID, etc. Note that:

– If the SP has been authenticated by the EAC, the contents of the SGT can be encrypted using the *Kp* shared between the SP and the EAC. The EAC will then transmit the SS's service request message (carrying the SGT), which is included or implied in the *AuthenticationRequest* message from the SS to EAC, to the SP. In this case, the EAC can send the SGT to the SS in the *AuthenticationFinish* message encrypted by *Kp*,

which is a traditional distribution mechanism of tickets for Kerberos's authentication protocol; the SGT also can be directly pushed to the SP by the EAC, in which case, just the SGT digest is sent by the EAC in the *AuthenticationFinish* message. This is an advanced mechanism of the ticket whereby the SGT is not directly transported. This increases the security. The EAC will confirm the mechanism in authentication mode, according to the network environment and the operator's policy.

– If the SP has never been authenticated by the EAC, the action of the transmission by the EAC will be done immediately after the success of the SP's entity initial authentication with the EAC.

NOTE 9 – If the service entity is an SS, it may trigger the initial authentication procedure between the EAC and the corresponding SP through the EAC when the SP has not been authenticated by the EAC, as specified in this clause, or it may trigger the authentication inquiring and key generation procedure when the SP has been authenticated by the EAC, as specified in clause 8.4.

NOTE 10 – Some examples for entity authentication procedure are illustrated in Appendix I.

8.3 Entity re-authentication procedure

At any time when a temporary identifier (i.e., ISR-ID/IAC-ID) or its associated shared keying material (i.e., *Ks/Kp*) is to be used, the service entity or EAC must first determine if its lifetime has expired. If it has expired, an entity re-authentication procedure between the related service entity and the EAC must be initiated before any further processing. The entity re-authentication procedure can also be initiated when a failure notice is received with the failure reason value "inactive identifier or key".

If the temporary identifier (i.e., ISR-ID/IAC-ID) or its associated shared keying material (i.e., *Ks/Kp*) is found invalid, the initial entity authentication procedure, as described in the previous clause 8.2, will be started between the EAC and the corresponding service entity.

The entity re-authentication procedure may be initiated by a service entity in either of the following two situations:

– The lifetime of the temporary identifier or shared key of the service entity is found to be expired by the EAC, then the EAC will send a notice in the next message to it indicating the need for re-authentication.

– The lifetime of the temporary identifier or shared key has expired or the security corresponding to the shared key is found fit for the service requirement by the service entity itself. If the service entity with an inactive identifier or key is an SS and it is found by an SP, then the SP will send a notice in the next message to the SS indicating the need for re-authentication.

The specific procedure is shown in Figure 8.

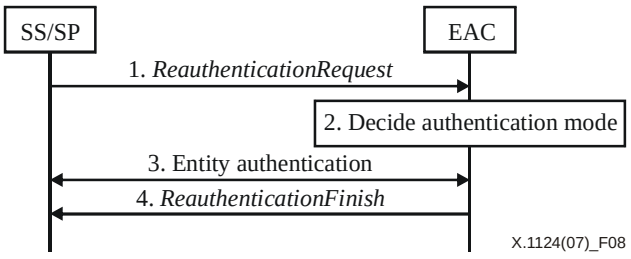

Figure 8 – Re-authentication procedure between service entity and EAC

1) The service entity (SS or SP) sends a re-authentication request message *ReauthenticationRequest* to EAC. The message contains:

   – The entity's temporary identifier (ISR-ID or IAC-ID).

– If the service entity is an SS, the public entity identifier (i.e., UID) of the SP providing the requested service.
– If there has been any change since the last *AuthenticationRequest* message sent to the EAC, the changed part of the authentication capability information of the service entity and the service security degree.
– And, if there has been no change except the temporary entity identifier and the shared keying material, the authentication mode information and, optionally, the first few authentication parameters according to the authentication mechanism between the service entity and the EAC specified in the authentication mode information.

If there has been any change in the authentication capability information of the service entity and the service security degree since the last *AuthenticationRequest* message, the following steps of the re-authentication procedure are the same as in the initial authentication procedure.

2) Through the temporary identifier ISR-ID or IAC-ID, the EAC finds out the PID of the service entity and then compares the received authentication mode information with the stored information.

3) If they are the same, the service entity and the EAC must perform the mutual authentication, using the authentication mechanism specified in the authentication mode information. The mutual authentication should be performed so as to reuse as much information as specified in the re-authentication part of the protocol of the underlying authentication mechanism.

   If not, the re-authentication procedure ceases and an initial entity authentication procedure will be initiated immediately by a failure notice from the EAC.

4) This step is the same as step 7 of the initial authentication procedure.

   NOTE 1 – After being refreshed, the old ISR-ID/IAC-ID, the *Ks/Kp* and their lifetime are replaced by the new ones.

   NOTE 2 – In the re-authentication procedure between the service entity and the EAC, the *Ki* can be a parameter used to perform authentication and generate the new *Ks/Kp*.

8.4 Authentication inquiring procedure with key generation

8.4.1 General

After the SS has performed the entity authentication procedure with the EAC, it can initiate the authentication inquiring and key generation procedure by sending a service request message to the SP (or possibly to the EAC) at any time when it needs a service from the SP. The procedure is performed according to the negotiated or agreed authentication mode information. After the procedure has been performed, the SS and the SP will have the same shared derived key *Ksp* between them. The *Ksp* will be used both in their mutual authentication and in their secure service communications.

8.4.2 Authentication inquiring and key generation procedure based on GBA

In the authentication inquiring and key generation procedure based on GBA, when a service subscriber sends a service request to a service provider, the service provider must ask the EAC for the authentication status of the SS. The specific procedure is shown in Figure 9.

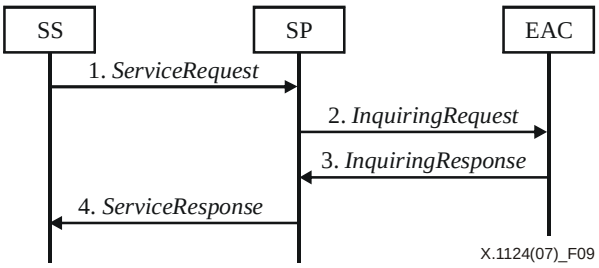

Figure 9 – Authentication inquiring and key generation procedure based on GBA

1) The SS sends a *ServiceRequest* to the SP, the request message includes the ISR-ID of the applicant (SS) and the public entity identifier (UID) of the SP, the negotiated or agreed authentication mode information (specifying the authentication inquiring and key generation mechanism to be "authentication inquiring and key generation procedure based on the GBA", and the mechanism of the mutual authentication mechanism between the SS and the SP, etc.), and the service security degree of the requested service.

2) After receiving the service request, the SP tries to find the corresponding derived key *Ksp* associated with the ISR-ID (note that if this is the case, the service security degree has also been associated with the ISR-ID). If the *Ksp* is found, then the SS and the SP can start to securely communicate immediately. If not, the SP validates the authentication mode information (e.g., the authentication inquiring and key generation mechanism and the mutual authentication mechanism with the SS). Then the SP sends an *InquiringRequest* message to EAC to enquire about the SS's authentication status. The message *InquiringRequest* includes the ISR-ID of the applicant SS, the UID and the IAC-ID of the enquiring SP and the service security degree.

3) After receiving the *InquiringRequest* message from the SP, the EAC first finds the related information according to the ISR-ID and IAC-ID, and validates the conformance of the authentication mode information with the service security degree.

   If the validation fails, the EAC returns an *InquiringResponse* error message to the SP. The SP will also respond with an error message to the SS.

   If the validation succeeds, the EAC will generate a derived key *Ksp*. For the calculation, the EAC first looks up the derived key generation method *KSx* (x = 1, 2, 3) and the related parameters that meet the requirement of the service security degree (mainly about key generation algorithm and key length, etc., which has been stored as the authentication mode information locally in the EAC). The *Ksp* is then computed using the SS's shared key *Ks*, the SS's PID, the SP's UID and other necessary parameters (i.e., *Ksp* = *KSx* (*Ks*, SP's UID, SS' PID, etc.)). The EAC then responds to the SP with the *Ksp* and its assigned lifetime in a message *InquiringResponse* which is encrypted using the shared key *Kp* with the cryptological algorithm negotiated or agreed in the authentication mode information.

4) The SP decrypts the received message using the *Kp* and retrieves derived key *Ksp*, and stores it locally with its lifetime value, the SS's ISR-ID, the SP's UID, the requested service security degree, and other relative information together. The SP then sends a service response message *ServiceResponse* back to the SS. The message *ServiceResponse* is encrypted with the key *Ksp* and the specific crypto-algorithm type, and the related parameters can be acquired in the related authentication mode information.

   After receiving a successful response, the SS computes the *Ksp* according to the service security degree using the same algorithm and parameters with the EAC. Using the calculated *Ksp*, the SS decrypts the received message from the SP to acquire the necessary application service information and then they can start the service communication securely.

NOTE – The *Ksp* calculation operation could be saved by the SS in the case that some failure occurs in the previous steps of authentication inquiring procedure. However, the *Ksp* may be also computed by the SS before the first step of the procedure, e.g., which is the case in 3GPP GBA.

8.4.3 Authentication inquiring and key transport procedure based on Kerberos

If the EAC has the Kerberos server function, in the authentication inquiring and key transport procedure based on Kerberos, the service subscriber has had a SGT or SGT digest. When the SS sends a service request message (carrying the SGT or SGT digest) to a service provider, the service provider must verify the authentication status of the SS. The specific procedure is shown in Figure 10.

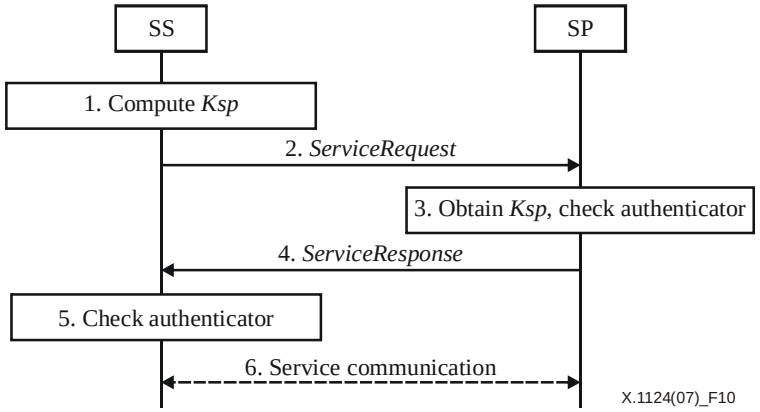

Figure 10 – Authentication inquiring and key transport procedure based on Kerberos

1) The SS generates the same derived key *Ksp* as the EAC by using the same algorithm and parameters as the EAC, and then generates a nonce for use in the next message sent to a target SP.

2) The SS sends a *ServiceRequest* to the SP, the request message includes ISR-ID of the applicant (SS) and public entity identifier (UID) of the SP, the negotiated or agreed authentication mode information (specifying the authentication inquiring and key generation mechanism to be "authentication inquiring and key generation procedure based on Kerberos", and the mechanism of the mutual authentication mechanism between the SS and the SP, etc.), the service security degree of the requested service and the service granted ticket (SGT) or the SGT digest. The message *ServiceRequest* also carries an authenticator which is Encrypt_*Ksp*{SS's ISR-ID: nonce and its lifetime} (for example, the nonce can be a serial number that is used to resist replay attack). The lifetime of the nonce should be short in the sense that every time the SS requests a service, the nonce will be generated again. If the service request message has been transferred to the SP by the EAC, as described in Note 5 in clause 8.2, then the message *ServiceRequest* only needs to carry the authenticator.

3) After receiving the *ServiceRequest*, the SP will verify the authentication inquiring and key generation mechanism and other related authentication mode information.

   – If the received *ServiceReques*t contains an SGT, the SP will decrypt it using the shared key *Kp*, check the validity of the SGT through verifying the SGT's contents, and obtain the *Ksp*. Then, using the *Ksp*, it will decrypt and check the validity of the authenticator by comparing the ISR-ID with the one achieved in the SGT.

   – If the received *ServiceReques*t contains an SGT digest, then the SP will look up the corresponding SGT and *Ksp* in its local memory (note that the SGT has been directly pushed to the SP by the EAC. The SP has decrypted the SGT to get the *Ksp* and other related information items using its shared key *Kp* and checked the SGT's validity. After

the validity check, the SP will store the *Ksp* and the SGT together with the corresponding ISR-ID). Then it checks the validity of the SGT digest and authenticator. If they are valid, then the SS is a valid user and can be provided the requested service. Otherwise, the SP sends an error message to the SS and the procedure ends.

4) The SP sends a *ServiceResponse* message back to the SS. In the message, the authenticator with the updated nonce is carried.

5) After receiving the *ServiceResponse* message, the SS will decrypt it using the shared key *Ksp*, which is computed in the first step, and check if the information items included are correct. If it is the case, then the SS can be sure that the SP is valid.

6) The SS and the SP begin their service communication.

8.4.4 Authentication inquiring and key agreement procedure based on Mediation model

If the EAC has the function of trusted third party (TTP) in the authentication and key agreement procedure, the communication messages between the SS and the SP in the procedure are transmitted via the EAC. The specific procedure is shown in Figure 11.

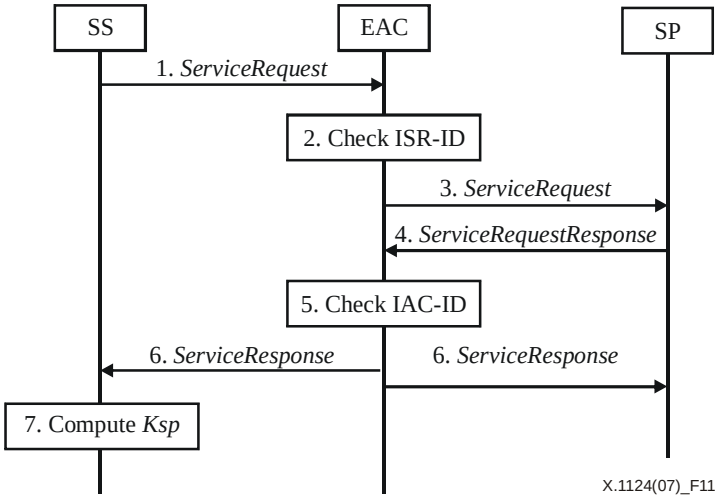

Figure 11 – Authentication and key agreement procedure based on Mediation model

1) When an SS needs to request a service provided by an SP, it must first send a *ServiceRequest* message to the EAC. The message contains the SS's ISR-ID, the SP's UID, and the encrypted (with encryption key *Ks*) information of the ISR-ID, the UID, the negotiated or agreed authentication mode information (specifying the authentication inquiring and key generation mechanism to be "authentication inquiring and key agreement based on Mediation", and the mechanism of the mutual authentication mechanism between the SS and the SP, etc.), and the service security degree of the requested service.

2) The EAC checks the validity of the ISR-ID by definition and looks up the subscription information of the SS to confirm that the SS can use the requested service.

   If successful, then the EAC decrypts the message and obtains the ISR-ID. After that, the EAC compares the decrypted ISR-ID with the one in the clear text. If coincident, it confirms that the ISR-ID has not been tampered with. Otherwise, the EAC sends an error message to the SS and the procedure ends.

3) The EAC transfers the contents of the message *ServiceRequest* in a new *ServiceRequest* message to the requested SP, but this time the message is encrypted by the key *Kp*.

4) The SP decrypts the message, retrieves the authentication mode information (e.g., the authentication inquiring and key generation mechanism and mutual authentication mechanism with the SS, etc.), and verifies the related information. Then, the SP sends a

ServiceRequestResponse message to the EAC carrying its IAC-ID, the SS's ISR-ID and the encrypted IAC-ID, ISR-ID with the key *Kp*.

5) The EAC decrypts the message and checks the validity of the IAC-ID. If successful, it will generate a derived key *Ksp* for the SS and the SP using the negotiated or agreed crypto algorithm in the authentication mode information. If not successful, it will send an error indication to the SP and the procedure ends.

6) The EAC sends a *ServiceResponse* message to the SP carrying the encrypted derived key *Ksp* using the key *Kp*.

   The EAC also sends a *ServiceResponse* message to the SS carrying the SP's UID to confirm the success of the SS's service request in the first step.

7) After receiving the *ServiceResponse*, the SS decrypts and checks if SP's UID is correct, and then computes the same derived key *Ksp* using the cipher algorithm and parameters negotiated or agreed in the authentication mode information.

   Now the SS and the SP can begin their mutual authentication or service communication procedure, which is based on the derived key *Ksp*.

8.5 Mutual authentication procedure between SS and SP

When the SS and the SP have the shared derived key *Ksp*, they can perform the proprietary mutual authentication procedure using the derived key. For every service session, a new session key *Kt* for the protection of their communication must be generated. The specific method of session key generation has been negotiated or agreed in their authentication mode information. In Appendix II, some examples of mutual authentication between the SS and the SP are illustrated.

9 Overall authentication procedures

Before an SS is ready to use a specific service provided by an SP, it must first decide whether it has ever been authenticated to the EAC, by checking whether or not it has a valid (active or inactive) ISR-ID. If the ISR-ID is valid, the SS can find out the negotiated or agreed authentication mode which specifies the following authentication methods and related parameters. If the SS cannot find them, then it must perform the procedure for initial authentication, as specified in clause 8.2. After the procedure, the shared keying material *Ks* and the associated ISR-ID are generated.

Then, the SS must decide whether it has performed the authentication inquiring and key generation procedure, by checking whether it has the relevant derived key *Ksp* stored locally. If it has the *Ksp*, it will send a service request to the SP to start a SS/SP specific mutual authentication procedure specified in the agreed authentication mode. Then they will generate a session key for the protection of their current service communication. Otherwise, the SS must perform the authentication inquiring and key generation procedure specified in the agreed authentication mode, as specified in clause 8.4. In the service request message, the mutual authentication method between the SP and the EAC is also carried which has been specified in the authentication mode, as specified in clause 8.4. After the procedure, the derived key *Ksp* is generated.

After receiving a service request from the SS, the SP must first decide whether it has ever been authenticated to the EAC, by checking whether or not it has the relevant valid *Kp* and IAC-ID stored in its local memory. If no *Kp* or IAC-ID is found, it must perform the procedure for initial authentication, as specified in clause 8.2. After the procedure, the shared keying material *Kp* and the associated IAC-ID are generated. After that, the SP must decide whether it has performed the authentication inquiring and key generation procedure relevant to the SS, by checking whether it has the relevant derived key *Ksp*. If it has no *Ksp* stored locally, it must perform the authentication inquiring and key generation procedure in the agreed authentication mode, as specified in clause 8.4. Otherwise, it will send a service response to the SS to start the service communication using *Ksp* as key material for the generation of session keys.

Appendix I

Some examples of entity authentication procedure

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

I.1 HTTP digest AKA used in 3GPP

AKA is the only authentication mechanism supported by the UE and network in 3GPP, so the authentication mechanism negotiation can be omitted. The mutual authentication between UE and EAC is performed as follows in the context of this Recommendation.

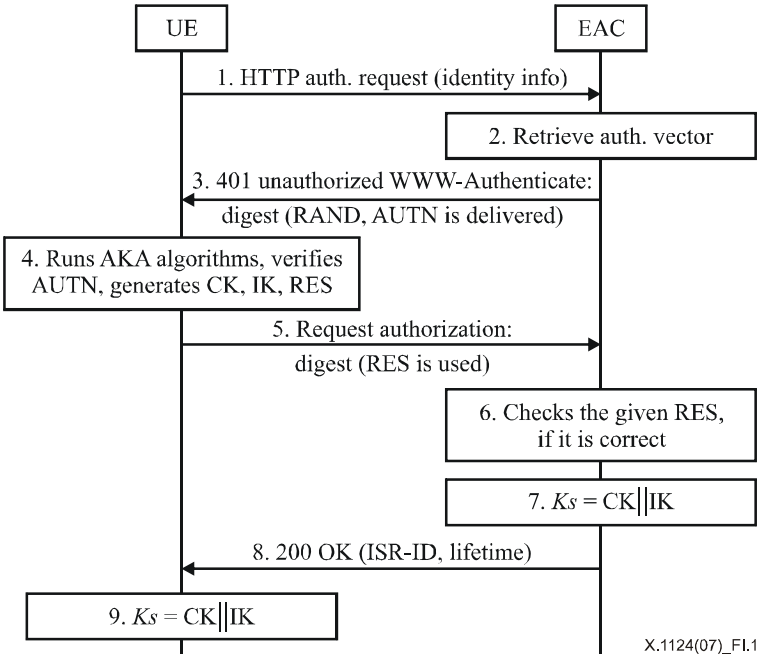

Figure I.1 – HTTP digest AKA used in 3GPP

1) UE sends HTTP digest authentication request to the EAC, containing its entity identifier.
2) EAC retrieves one authentication vector (AV, AV = RAND||AUTN||XRES||CK||IK) from the ESD.
3) Then EAC forwards the RAND and AUTN to the UE in the 401 message (without the CK, IK and XRES). This is to request the UE to authenticate itself.
4) The UE checks AUTN to verify that the challenge is from an authorized network; the UE also calculates CK, IK and RES. This will result in session keys IK and CK in both EAC and UE.
5) The UE sends another HTTP request, containing the digest AKA response (calculated using RES), to the EAC.
6) The EAC authenticates the UE by verifying the digest AKA response.
7) The EAC generates key material *Ks* by concatenating CK and IK and the ISR-ID which is the same with bootstrapping transaction identifier.
8) The EAC must send a 200 OK message, including an ISR-ID, to the UE to indicate the success of the authentication. In addition, in the 200 OK message, the EAC must supply the lifetime of the key *Ks*.
9) The key material *Ks* is generated in UE by concatenating CK and IK, which then must be stored with the lifetime and ISR-ID together.

I.2 HTTP digest AKA used in 3GPP2

There are three authentication mechanisms in 3GPP2: AKA, authentication based on CAVE, and authentication based on MN-AAA key. Each of the authentication mechanisms requires a corresponding identity. If an IMPI is sent as the corresponding identity, then AKA-based authentication must use the ISIM associated with that IMPI. Otherwise, it will be based on the CDMA 2000 application. The authentication negotiation and mutual authentication procedure is performed as follows.

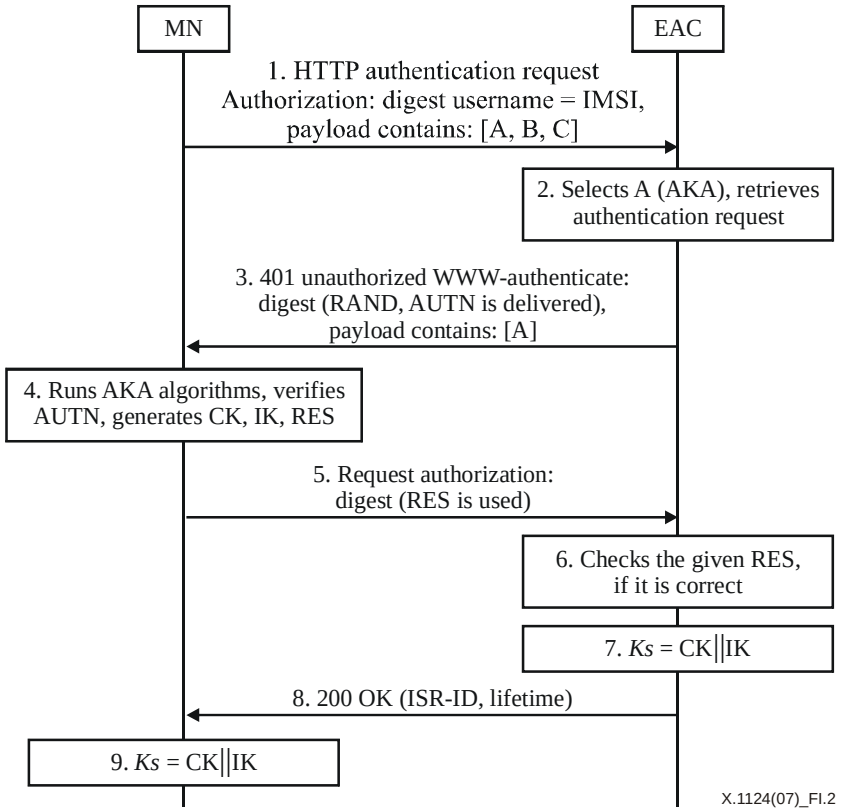

Figure I.2 – HTTP digest AKA used in 3GPP2

1) MN sends HTTP digest authentication request to the EAC, containing international mobile subscriber identity as its entity identifier and the authentication mechanism (A – AKA, B – authentication based on CAVE, C – authentication based on MN-AAA key) supported by it.

2) AKA is selected according to the entity identifier brought in the authentication request and then retrieves one authentication vector (AV, AV = RAND||AUTN||XRES||CK||IK) from the ESD.

3) Then EAC forwards the RAND and AUTN to the MN in the 401 message (without the CK, IK and XRES). This is to demand the MN to authenticate itself. In addition, the payload will include an indication of the selected mechanism (in this case, A), together with the corresponding identity.

4) The MN checks AUTN to verify that the challenge is from an authorized network; the MN also calculates CK, IK and RES. This will result in session keys IK and CK in both EAC and MN.

5) The MN sends another HTTP request, containing the digest AKA response (calculated using RES) to the EAC.

6) The EAC authenticates the MN by verifying the digest AKA response.

7) The EAC generates key material *Ks* by concatenating CK and IK and the ISR-ID which is the same with bootstrapping transaction identifier.

8) The EAC must send a 200 OK message, including an ISR-ID, to the MN to indicate the success of the authentication. In addition, in the 200 OK message, the EAC must supply the lifetime of the key *Ks*.

9) The key material *Ks* is generated in MN by concatenating CK and IK, which then must be stored with the lifetime and ISR-ID together.

I.3 TLS-Cert based authentication mechanism

If the SP is a bank and it wants to provide a mobile telephone bank service, it must first perform an authentication procedure and generate the shared keying material with the EAC. The following shows the procedure of authentication negotiation and mutual authentication between the SP and EAC, in which case that TLS based on certificate authentication mechanism is selected.

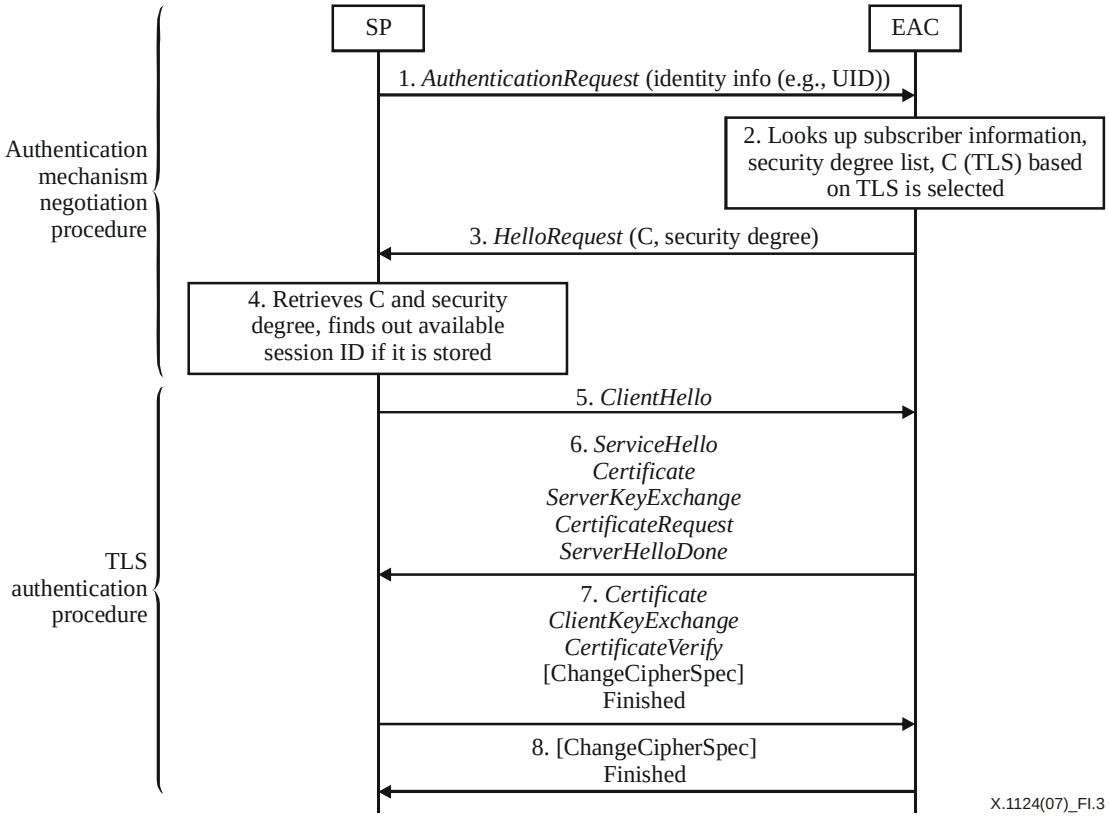

Figure I.3 – TLS based on certificate authentication mechanism use

NOTE – In Figure I.3, steps 1, 2, 3 and 4 are the authentication mechanism negotiation procedure. Steps 5, 6, 7 and 8 are the TLS authentication procedure.

1) SP sends *AuthenticationRequest* to EAC, containing its UID.

2) EAC looks up the subscription information of the SP according to UID and validates its privilege for providing this kind of service, then retrieves the authentication capability information (e.g., certificate-based, TLS based on certificate, PSK-TLS, etc.)

   After that, the EAC looks up the service security degree list to find the security degree corresponding to the mobile telephone service, then it looks up the authentication security list and finds that HTTP digest AKA and TLS based on certificate are the two authentication mechanism according to the security degree and supported by the network. After matching the authentication mechanism supported by the SP, the TLS based on certificate is selected to be performed between the SP and EAC.

3) EAC sends *HelloRequest* to SP and brings the authentication mechanism identifier C (TLS based on certificate) and security degree identifier together.

4) SP retrieves authentication mechanism identifier C (TLS based on certificate) and security degree identifier, and finds out if the available session ID (IAC-ID) has been stored (the ID of a session the client wishes to use for this connection).

5) SP sends *ClientHello* to EAC. The session ID field should be empty if no session_id is available or the client wishes to generate new security parameters, otherwise it contains the available session_id.

6) After receiving *ClientHello*, the EAC checks if the session ID field is empty. If the *ClientHello* session_id is non-empty, the EAC will look in its session cache for a match. If a match is found and the EAC is willing to establish the new connection using the specified session state, the server will respond with the same value as was supplied by the SP. This indicates a resumed session and dictates that the parties must proceed directly to the finished messages. Otherwise, this field will contain a different value identifying the new session. The EAC may return an empty session_id to indicate that the session will not be cached and therefore cannot be resumed. The EAC will send a server certificate, *ServerKeyExchange* (optional), *CertificateRequest* orderly.

   At last, EAC sends *ServerHelloDone* to indicate the end of the server hello and associated messages.

7) After receiving the *ServerHelloDone* message, the SP sends *Certificate* message and *ClientKeyExchange* message. Then the shared secret parameter is retrieved on both sides.

   Then *CertifiicateVerify* message is sent to provide explicit verification of a client certificate.

   At last, *Finished* message is sent immediately after a change cipher spec message to verify that the key exchange and authentication processes were successful.

8) After receiving the finished message, the EAC verifies that the contents are correct, and then sends *Finished* message back.

   The SP also verifies that the contents are correct. If correct, the authentication and key exchange procedure is finished successfully.

I.4 Authentication procedure based on public key certificate authentication mechanism

The central preconditions of this procedure are shown as follows.

1) EAC may be able to receive a certificate from certificate repository, download the certificate revocation list, and verify the certificate's validity.

2) ESD may contain certificate repository.

3) Some service entities, especially the mobile terminal, may just store the certificate URLs in order to minimize the amount of memory required.

4) Elliptic curves cryptography may be used.

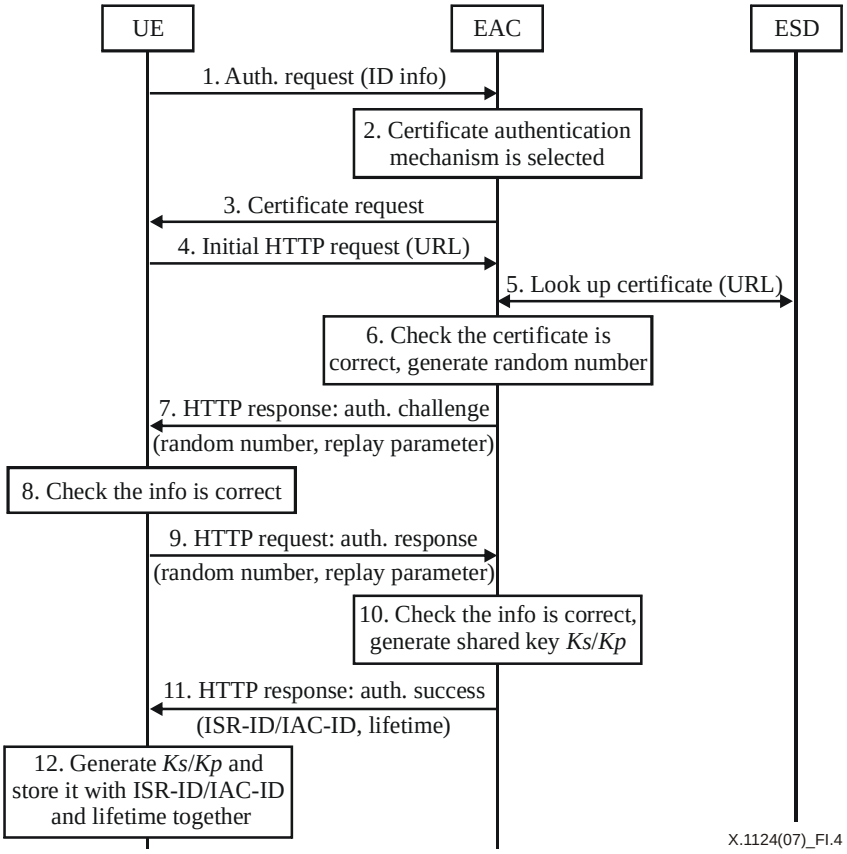

Figure I.4 – Authentication mechanism based on certificate use

1) The UE sends authentication request to the EAC, containing its identity information.

2) The EAC looks up subscription information corresponding to the identity, and selects the certificate authentication mechanism for the mutual authentication between them by using the operator's local policy.

3) The EAC sends certificate request to the UE containing certificate authentication mechanism identifier.

4) The UE sends HTTP request to the EAC containing its certificate URL.

5) The EAC looks up the certificate of UE from the ESD by using the URL.

6) The EAC checks the certificate's validity and generates a random number.

7) The EAC responds with authentication challenge, containing a random number generated by it and its domain name, which are signed by using the private key of EAC and encrypted by using the public key of the UE, and a replay parameter encrypted by using the shared secret between the EAC and UE, which is used to avoid replay attacks.

8) The UE first checks the replay parameter is correct by using the shared secret, and then checks the information signed and encrypted is correct by using its private key and EAC's public key, after that the random number domain name of EAC and replay parameter are stored together.

9) The UE sends HTTP request with authentication response message, containing the replay parameter updated encryption by using the shared secret, the random number and its entity identifier, which are signed by using the private key of UE and encrypted by using the public key of EAC.

10) The EAC first checks the replay parameter is correct by using the shared secret, and then checks the information signed and encrypted is correct by using its private key and the UE's

public key. After that, EAC generates the shared key *Ks/Kp* by using random number, replay parameter, PID of the UE and the shared secret as input parameters.

11) The EAC responds with authentication success response, containing temporary entity identifier ISR-ID/IAC-ID assigned to the UE and the lifetime of the shared key, encrypted by using the shared key *Ks/Kp*.

12) The UE generates *Ks/Kp* by using the same input parameters and cipher algorithm, and stores it with the lifetime and ISR-ID/IAC-ID together.

I.5 Authentication procedure based on a biometric authentication mechanism

During the authentication between the service entity and EAC, shown in clause 8.2, the entity and EAC should first negotiate an authentication mechanism. If choosing biometric authentication, the service entity and EAC separately compute a shared key *Kb* using the parameters: user's private entity identifier, biometric authentication mechanism identifier (e.g., fingerprint authentication mechanism, iris authentication mechanism, etc.).

The mutual authentication procedure between the service entity and the EAC based on a biometric authentication mechanism is shown in Figure I.5.

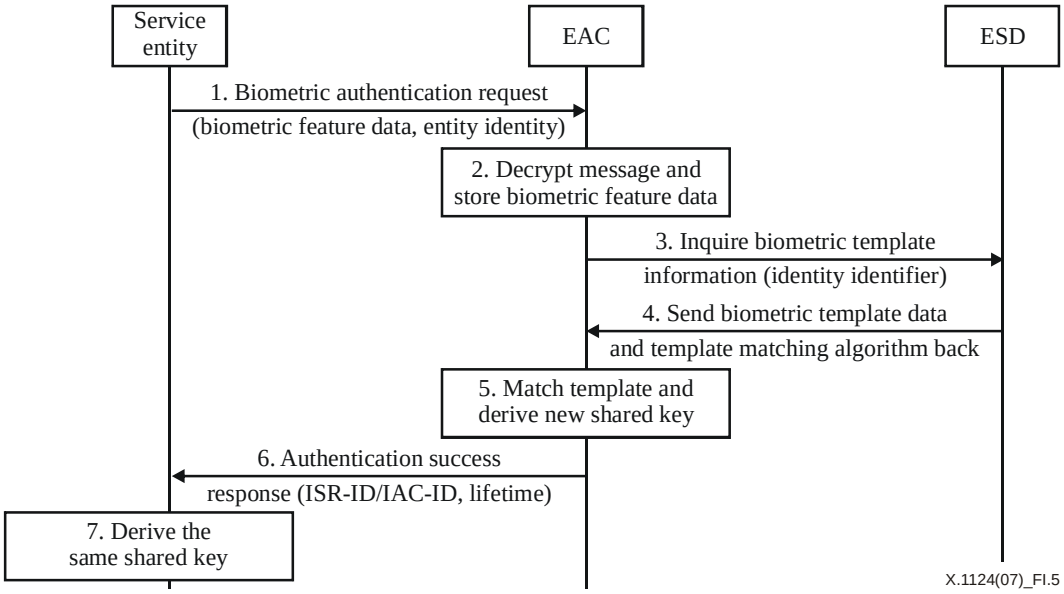

Figure I.5 – Authentication procedure between a service entity and the EAC based on biometric authentication mechanism

1) The service entity sends a biometric authentication request to the EAC, including biometric feature data and its identifier. The request message is encrypted with the shared key *Kb*.

2) Receiving the request message, the EAC decrypts it and stores biometric data.

3) The EAC inquires about the entity's biometric template information from the ESD with its identifier.

4) The ESD sends the entity's biometric template, biometric match algorithm and some environment information parameters.

5) The EAC authenticates the user's identity using its biometric feature data. If authentication succeeds, the EAC will derive a new shared key *Ks* using biometric feature data.

6) The EAC sends an authentication success message encrypted with the shared key. The message includes the temporary entity identifier, the identifier and the shared key's lifetime.

7) Receiving the authentication success message, the service entity derives the same shared key *Ks* using its biometric feature data, and stores the key binding with the temporary identifier and their lifetime.

Appendix II

Examples of mutual authentication between SS and SP

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

II.1 Standardized cases

In 3GPP MBMS, the fourth layer key *Kt* is actually the application key MTK which can be generated and used according to the corresponding specification (see [b-3GPP TS 33.246]).

II.2 Other possible cases

If sharing the derived key *Ksp*, SS and SP must use it to authenticate each other and derive a session key for this service communication. The procedure is shown in Figure II.1.

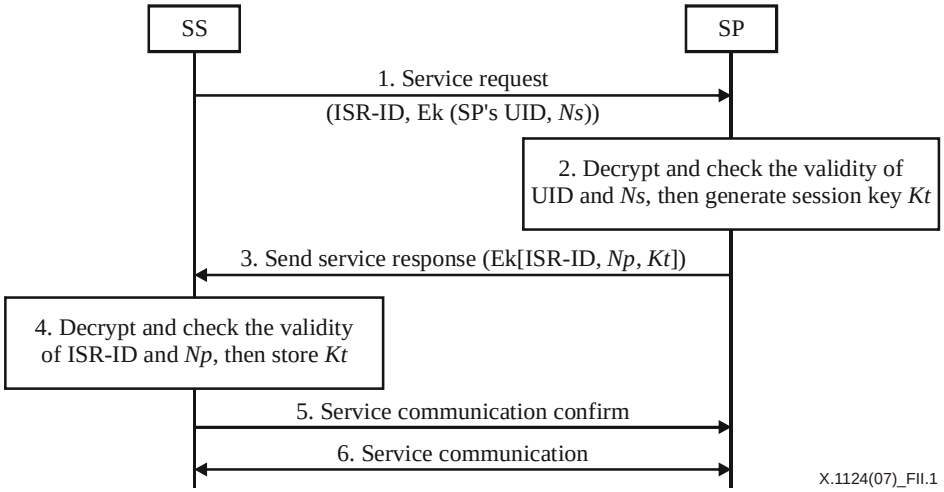

Figure II.1 – Authentication and session key generation procedure between SS and SP

1) SS sends service request or authentication request to SP, including ISR-ID and the encrypted UID of SP and a sequence number *Ns* (called nonce).

   The sequence number *Ns* shows the time of the SS request for the same service towards the SP by using the same derived key, which is stored with the derived key, temporary entity identifier, service type, etc., together, and the sequence number *Ns* increased with the request time one by one.

2) After receiving the service request, if SP can find the derived key and some other information banded with the temporary entity identifier, then it can decrypt and retrieve its UID and then *Ns*, and check the validity of them.

   A sequence number *Np*, showing the time of the SS request for the same service towards the SP by using the same derived key, is also stored in the SP side. Whenever receiving a service request, the SP must match the *Ns* brought in the message and the *Np*. Whether the *Ns* and *Np* match or not can be judged by some policy (e.g., 2>*Ns*-*Np*>0).

   If they match, then the *Ns* must be stored as a new SP with other key material together. Then the SP must generate a session key *Kt* for this service communication.

   The session key *Kt* is derived from the derived key and another parameter is randomly generated by the SP, which means that each service communication has a different session key.

   If the SP fails to decrypt, then it must not generate a session key, but send a service failure response back to the SS.

3) If the authentication for SS succeeds, the SP sends a service response. The message contains ISR-ID, new *Np* (called response) and session key *Kt*, encrypted by the derived key .

   If the authentication for SS does not succeed, the SP sends a failure response, which gives a clear indication of failure reason. The SS must decide whether to initiate the re-authentication procedure with the EAC.

4) After receiving the service response, the SS must decrypt the message, and check the validity of ISR-ID and *Np*. If all are correct, then the *Kt* must be stored with other key material together.

   If the *Np* is the same as the *Ns* brought in the service request, it is correct.

5) The SS sends a service communication confirm back to SP.

6) They begin to communicate by using the session key *Kt*.

Appendix III

Key lifetime

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

The temporary entity identifiers and related keys (*Ks*, *Kp*, *Ksp*) have lifetime assigned by EAC (for *Ks* and *Kp*) or derived from other lifetimes (for *Ksp*, the lifetime of which is before both of the deriving *Ks'* and *Kp*'s).

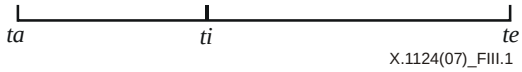

Figure III.1 – Lifetime of a key

ISR-ID and *Ks* have the same lifetime interval. IAC-ID and *Kp* have the same lifetime interval. During their lifetime, they are in either of the two states: ACTIVE (i.e., during the first part of its lifetime, i.e., from the beginning time instance *ta* to an interim time instance *ti*) or INACTIVE (i.e., during the second part of its lifetime or from *ti* to the end time instance *te*).

In the state of "ACTIVE", it can be used in any circumstance.

In the state of "INACTIVE", it must only be used in two cases: in the procedure to produce a new identifier and the corresponding shared keying material, or in the processing of received messages (e.g., decryption operation). An inactive key must not be used to encrypt a new message that is not for the purpose of re-authentication. After a new identifier and a new *Ks/Kp* are generated, the old ones will be replaced and their states are set to "INVALID". However, they are not yet destroyed until the end of their lifetime.

Ksp only has one state, i.e., "ACTIVE" during its lifetime.

An identifier (and its related keys) becomes INVALID if it has run out of its lifetime. An invalid identifier or key will not be used in any other cases except that it is related to inquiring of its status. The reason for an identifier or a key entering the state of "INVALID" may be the end of its lifetime or the revocation or elimination by EAC for any reason that could pose a hazard to security (e.g., key leakiness).

Appendix IV

Mapping of the reference points to those in 3GPP/3GPP2

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

The following Table IV.1 shows the mapping relationship between the above-defined reference points and those defined in 3GPP/3GPP2.

Table IV.1 – Mapping of the reference points to those in 3GPP/3GPP2

Reference point in this Recommendation	Reference point in 3GPP	Reference point in 3GPP2
Zm	*Zh*	*Zh*
Zb	*Ub*	*Ub*
Zb'	*Zn, Zn'*	*Zn*
Ue	*Ua*	*Ua*
Za	*Zn'*	–

Bibliography

[b-ITU-T Q.1701] Recommendation ITU-T Q.1701 (1999), *Framework for IMT-2000 networks*.

[b-ITU-T Q.1711] Recommendation ITU-T Q.1711 (1999), *Network functional model for IMT-2000*.

[b-ITU-T X.811] Recommendation ITU-T X.811 (1995) | ISO/IEC 10181-2:1996, *Information technology – Open Systems Interconnection – Security frameworks for open systems: Authentication framework.*

[b-3GPP TR 22.934] 3GPP TR 22.934 (2003), *Feasibility study on 3GPP system to Wireless Local Area Network (WLAN) interworking (Release 6).*
<http://www.3gpp.org/ftp/specs/html-info/22934.htm>

[b-3GPP TR 33.919] 3GPP TR 33.919 (2005), *3G security; Generic Authentication Architecture (GAA); System description (Release 6).*
<http://www.3gpp.org/ftp/Specs/html-info/33919.htm>

[b-3GPP TS 33.102] 3GPP TS 33.102 (in force), *3G security; Security architecture.*
<http://www.3gpp.org/ftp/specs/html-info/33102.htm>

[b-3GPP TS 33.210] 3GPP TS 33.210 (in force), *3G security; Network Domain Security (NDS); IP network layer security.*
<http://www.3gpp.org/ftp/Specs/html-info/33210.htm>

[b-3GPP TS 33.220] 3GPP TS 33.220 (2007), *Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (Release 6).*
<http://www.3gpp.org/FTP/Specs/html-info/33220.htm>

[b-3GPP TS 33.221] 3GPP TS 33.221 (2007), *Generic Authentication Architecture (GAA); Support for Subscriber Certificates (Release 6).*
<http://www.3gpp.org/FTP/Specs/html-info/33221.htm>

[b-3GPP TS 33.222] 3GPP TS 33.222 (2007), *Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS) (Release 7).*
<http://www.3gpp.org/FTP/Specs/html-info/33222.htm>

[b-3GPP TS 33.246] 3GPP TS 33.246 (in force), *3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS).*
<http://www.3gpp.org/FTP/Specs/html-info/33246.htm>

[b-3GPP TS 33.310] 3GPP TS 33.310 (in force), *Network Domain Security (NDS); Authentication Framework (AF).*
<http://www.3gpp.org/FTP/Specs/html-info/33310.htm>

[b-IETF RFC 3310] IETF RFC 3310 (2002), *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA).*
<http://www.ietf.org/rfc/rfc3310.txt>

[b-IETF RFC 4279] IETF RFC 4279 (2005), *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS).*
<http://www.ietf.org/rfc/rfc4279.txt>

[b-IETF RFC 4346] IETF RFC 4346 (2006), *The Transport Layer Security (TLS) Protocol Version1.1.*
<http://www.ietf.org/rfc/rfc4346.txt>

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	Telephone transmission quality, telephone installations, local line networks
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, 2008