ITU - Telecommunication Standardization Sector

Study Period 2005 - 2008

NGN GSI Rapporteur Group Meeting: Geneva, 11-21 September 2007

Question	15/13
Source:	Editors
Title:	Draft Recommendation Y.secMechanisms (NGN security mechanisms and procedures)
Purpose:	<For information / discussion / approval>

Contact:	Martin Dolly AT&T USA	Tel: +1 732 420 4574 Email: mdolly@att.com
	Takashi Egawa NEC Corporation Japan	Tel: +81 44 396 3169 Fax: +81 44 431 7644 Email: t-egawa@ct.jp.nec.com

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

]INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION SECTOR

STUDY PERIOD 2005-2008

STUDY GROUP 13

TD ~~284~~ xxx (WP 2/13)

English only

Original: English

Question(s): 15/13 Geneva, ~~16-27~~11-21 ~~April~~ September 2007

TEMPORARY DOCUMENT

Source: Editors

Title: Draft Recommendation Y.secMechanisms (NGN security mechanisms and procedures) ~~(Beijing, China, 8-12 January 2007)~~

NGN Security Mechanisms and Procedures

Summary

This text provides the implementation realization of NGN security mechanisms and procedures..

Table of Contents

1. Scope 5

2. References 5

3. Definitions 9

4. Abbreviations 9

5. Identification and Authentication 9

5.1 Credential usage in the NGN Security Architecture 10

5.1.1 Device, Subscriber, and End-User Credentials 10

5.1.2 X.509 Certificates as credentials 10

5.1.3 Shared keys as credentials 11

5.1.4 Information provisioned in SAA/TAA-FEs for each set of credentials 11

5.2 Identification and Authentication of Subscribers 13

5.2.1 General Strategy 13

5.2.2 Identification of the Subscriber through Network Source Address 14

5.2.3 Identification of the Subscriber through TLS/IPsec Security Association 14

5.2.4 Identification of the Subscriber through Challenge/Response 15

5.2.5 Generic Bootstrapping Architecture (GBA) 16

5.3 Identification and Authentication of End-Users 17

5.3.1 General Strategy 17

5.3.2 Identification of the End-User through TLS/IPsec Security Association 17

5.3.3 Identification of the End-User through Challenge/Response 18

5.4 Identification and Authentication with CPE-BE 18

5.4.1 Use of X.509 Certificates 18

5.5 Authenticator-SAA/TAA-FEs Interface 19

5.5.1 Authenticator-SAA/TAA-FEs Interface for Digest Authentication (section needs clarification and rewording) 19

5.5.2 Authenticator-SAA/TAA-FEs Interface for Credential Information Retrieval 20

6. Transport Security for Signaling and OAMP 21

6.1 TLS 22

6.1.1 Ciphersuites 22

6.1.2 TLS Use of Certificates 24

6.1.3 Connection Persistence and Re-Use 24

6.1.4 Session Caching 24

6.1.5 Session Key Management 25

6.2 IPsec in Trusted and Trusted-but-Vulnerable Domains 25

6.2.1 AH and ESP 25

6.2.2 Transport and Tunnel Mode 26

6.2.3 Replay Protection 26
6.2.4 Key Management 26
6.3 IPsec between Untrusted and Trusted-but-Vulnerable Domains 28

7. Media Security 28
7.1 SRTP 30
7.1.1 Encryption and Authentication Algorithms 30
7.1.2 Cipher Suite Negotiation and Key Generation 30
7.2 Authentication interface between NGN Network Element and Secure Token Server 31

8. Audit Trail, Trapping, and Logging Systems 32
8.1. Time Stamping and Time Source 32
8.2 Events Reported in the Security Log (preliminary) 32
8.3 Network Element interface to Syslog Server 32
8.4 Network Element Use of SNMP 32
8.5 Network Element use of Integrity Checking Agent 33
8.6 Security Patch Management 33
8.7 Tracking Changes and Rolling Back Changes 33
8.8 Audit Trail, Trapping, and Logging at CPE-BE 34

9. CPE Provisioning 34

References 42

Ed Note: guideline: select the mandatory features and identify others as optional when the work matures.

EdNote: replace 'byte' with 'octet' where appropriate.

1. Scope

Y.2701, *Security Requirements for NGN Release 1* describes NGN security requirements and architecture. The security mechanisms selected to implement these requirements will contain options; however, mismatched options should be avoided since they tend to introduce security vulnerabilities and make it more difficult to achieve interoperability.

This Recommendation describes the specific security mechanisms and suites of options that should be used to realize the requirements in Y.2701 in each NGN Network Element. Other mechanisms and suites of options which may adequately fulfill NGN security requirements are also described.

This Recommendation also specifies a minimum suite of security mechanisms and options for those implementations that address the needs of nomadic end users and end user equipments.

Editor's Note: The following need to be addressed/verified:

Need to provide linkage with the Requirements in Section 7 of this document.

Consistent use of the terms credentials & certificates

Consistent/proper use of network, operator, service provider

Either move all requirements to Section 7 or reword (as not read like a reqt.)

This section is not exhaustive and it is anticipated that contributions will identify additional M&Ps at the next meeting.

2. References

The following ITU-T Recommendations, and other references contain provisions, which, through reference in this text, constitute provisions of this Supplement. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Supplement are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below.

[Y.2012~~1~~] ITU-T Y.NGN-FRA (200x), *Functional Requirements and Architecture of the NGN.*

[Y.2701] ITU-T Y.NGN security (200x), NGN security requirements.~~.~~

[Y.NGNauth] ITU-T Y.NGNAuthentication (200x), NGN Release 1 Authentication. (latest version: TD99(WP2))

[X.1035] ITU-T Recommendation X.1035, Password-authenticated key exchange (PAK) protocol.

[RFC3268] IETF RFC 3268 (2002), Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS).

[RFC3713] IETF RFC 3713 (2004), A Description of the Camellia Encryption Algorithm. EdNote: this is an informational RFC.

[RFC 3830] IETF RFC 3830 (2004), MIKEY: Multimedia Internet KEYing. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman.

[RFC4132] IETF RFC 4132 (2005), Addition of Camellia Cipher Suites to Transport Layer Security (TLS).

[RFC4312] IETF RFC 4312 (2005), The Camellia Cipher Algorithm and Its Use with IPsec.

~~[draft-ietf-aaa-diameter-sip-app-03] Miguel Garcia-Martin, "Diameter Session Initiation Protocol (SIP) Application", April 2006.~~

[RFC 4566] SDP: Session Description Protocol. M. Handley, V. Jacobson, C. Perkins. July 2006.

[RFC 4567] Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP). J. Arkko, F. Lindholm, M. Naslund, K. Norrman, E. Carrara. July 2006.

[RFC 4568] IETF RFC 4568, Session Description Protocol (SDP) Security Descriptions for Media Streams, F. Andreasen, M. Baugher, D. Wing, July 2006.

[RFC 4648] The Base16, Base32, and Base64 Data Encodings. S. Josefsson. October 2006.

[RFC4590] IETF RFC 4590, *Radius Extension for Digest Authentication*.

[RFC4740] IETF RFC 4740, *Diameter Session Initiation Protocol (SIP) Application*.

[draft-ietf-sip-connect-reuse-xx] Rohan Mahy, "Connection Reuse in the Session Initiation Protocol (SIP)", 21-Aug-06.

~~[draft-sterman-sip-aaa-03] no relevant I-D?~~

[FIPS sp800-38a] National Institute of Standards and Technology, Special Publication 800-38: Recommendation for Block Cipher Modes of Operations . Methods and Techniques, December 2001.

[FIPS 197] Federal Information Processing Standard (FIPS) 197: Advanced Encryption Standard, November 2001.

[FIPS 198] NIST FIPS 198, The Keyed-Hash Message Authentication Code

[RFC1155] Structure and identification of management information for TCP/IP-based internets. M.T. Rose, K. McCloghrie. May 1990.

[RFC1212] Concise MIB definitions. M.T. Rose, K. McCloghrie. March 1991.

[RFC1215] Convention for defining traps for use with the SNMP. M.T. Rose. March 1991. EdNote: this is an informational RFC.

[RFC2222] Simple Authentication and Security Layer (SASL). J. Myers. October 1997.

[RFC2246] The TLS Protocol Version 1.0. T. Dierks, C. Allen. January 1999.

[RFC2263] SNMPv3 Applications. D. Levi, P. Meyer, B. Stewart. January 1998.

[RFC2264] User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen. January 1998.

[RFC2367] PF_KEY Key Management API, Version 2. D. McDonald, C. Metz, B. Phan. July 1998. EdNote: this is an informational RFC.

[RFC2401] Security Architecture for the Internet Protocol. S. Kent, R.Atkinson. November 1998.

[RFC2402] IP Authentication Header. S. Kent, R. Atkinson. November 1998. EdNote: this is obsoleted by RFC 4302 and 4305

[RFC2403] The Use of HMAC-MD5-96 within ESP and AH. C. Madson, R. Glenn. November 1998.

[RFC2404] The Use of HMAC-SHA-1-96 within ESP and AH. C. Madson, R. Glenn. November 1998.

[RFC2406] IP Encapsulating Security Payload (ESP). S. Kent, R. Atkinson. November 1998.

[RFC2407] The Internet IP Security Domain of Interpretation for ISAKMP. D. Piper. November 1998.

[RFC2409] The Internet Key Exchange (IKE). D. Harkins, D. Carrel. November 1998.

[RFC2451] The ESP CBC-Mode Cipher Algorithms. R. Pereira, R. Adams. November 1998.

[RFC2574] User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen. April 1999. EdNote: this is obsoleted by RFC 3414.

[RFC2578] Structure of Management Information Version 2 (SMIv2). K. McCloghrie, D. Perkins, J. Schoenwaelder. April 1999.

[RFC2579] Textual Conventions for SMIv2. K. McCloghrie, D. Perkins, J. Schoenwaelder. April 1999.

[RFC2580] Conformance Statements for SMIv2. K. McCloghrie, D. Perkins, J. Schoenwaelder. April 1999.

[RFC2617] HTTP Authentication: Basic and Digest Access Authentication. J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart. June 1999.

[RFC2808] The SecurID® SASL Mechanism. M. Nystrom. April 2000. EdNote: this is an informational RFC.

[RFC2865] Remote Authentication Dial In User Service (RADIUS). C. Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

[RFC3164] The BSD Syslog Protocol. C. Lonvick. August 2001. EdNote: this is an informational RFC.

[RFC3261] SIP: Session Initiation Protocol. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. June 2002.

[RFC3310] Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA). A. Niemi, J. Arkko, V. Torvinen. September 2002. EdNote: this is an informational RFC.

[RFC3410] Introduction and Applicability Statements for Internet-Standard Management Framework. J. Case, R. Mundy, D. Partain, B. Stewart. December 2002. EdNote: this is an informational RFC.

[RFC3411] An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. D. Harrington, R. Presuhn, B. Wijnen. December 2002.

[RFC3414] User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3). U. Blumenthal, B. Wijnen. December 2002.

[RFC3415] View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP). B. Wijnen, R. Presuhn, K. McCloghrie. December 2002.

[RFC3416] Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP). R. Presuhn, Ed.. December 2002.

[RFC3417] Transport Mappings for the Simple Network Management Protocol (SNMP). R. Presuhn, Ed.. December 2002.

[RFC3550] RTP: A Transport Protocol for Real-Time Applications. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. July 2003.

[RFC3588] Diameter Base Protocol. P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko. September 2003.

[RFC3602] The AES-CBC Cipher Algorithm and Its Use with Ipsec. S. Frankel, R. Glenn, S. Kelly. September 2003.

[RFC3711] The Secure Real-time Transport Protocol (SRTP). M. Baugher, D. McGrew, M. Naslund, E. Carrara, K. Norrman. March 2004.

[RFC3830] MIKEY: Multimedia Internet KEYing. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, K. Norrman. August 2004.

[RFC 4492] Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS), S.Blake-Wilson, N.Bolyard, V.Gupta, C.Hawk, B.Moeller, May 2006. EdNote: this is an informational RFC.

[RFC4567] Key Management Extensions for Session Description Protocol (SDP) and Real Time Streaming Protocol (RTSP).

[RFC4568] Session Description Protocol (SDP) Security Descriptions for Media Streams.

[X.509] ITU-T Recommendation X.509 (1997 E): Information Technology – Open Systems Interconnection – The Directory: Authentication Framework, June 1997.

[X.690] ITU-T Recommendation X.690 Information Technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: General procedures, 1992.

[ISO/IEC 15946-5-1] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves. Part 1: General

[ISO/IEC 15946-5-2] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves. Part 2: Digital Signatures (ECDA)

[ISO/IEC 15946-5-3] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves. Part 3 Key Establishment (ECDH)

[ISO/IEC 15946-5-4] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves. Part 4: Digital signatures giving message recovery

[ISO/IEC 15946-5-5] Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves. Part 5: Elliptic curve generation Cryptographic techniques based elliptic Curves.

3. Definitions

Authenticator: An Authenticator is a Network Element that facilitates identification and authentication of subscribers, devices or end-users. For example, Border Elements with B2BUA functionality or P-CSCFs can be Authenticators of subscribers for SIP-based services.

4. Abbreviations

5. Identification and Authentication

A request for an NGN service is associated with a subscriber.~~NGN service requests for service will be associated with a specific subscriber.~~ This association is determined through identification of the request with the subscriber whose service is being used to make the request. Based on the Service Level Agreement with the Subscriber, further identification and authentication of the End-User may be required.

An Authenticator is a Network Element that facilitates identification and authentication of subscribers, devices or end-users. For example, Border Elements with B2BUA functionality or P-CSCFs can be Authenticators of subscribers for SIP-based services. Identification and authentication is achieved by the exchange of credentials between an Authenticator and the CPE, allowing the NGN Carrier to correctly charge service usage to the customers.

Border Element

If the Border Element received the request from a NGN Network Element in the Trusted Domain, the identification contained in the request is believed to be accurate and not checked further.

This clause describes identification and authentication mechanisms, in particular, those concerning SIP-based services. The mechanisms concerning other services are for further study.

5.1 Credential usage in the NGN Security Architecture

Credentials are used in the NGN Security Architecture to identify and authenticate a Device, a Subscriber, and/or an End-User. These types of credentials are described in section 0. In all cases, the Authenticator uses the credentials to associate the request with a specific subscriber.

The credentials may take two different forms, either a shared key (described in section 0) or a X.509 certificate (described in section 0). The X.509 certificate is used to establish a secure transport between the CPE and the Authenticator (described in section 0). The shared key may be used either to establish a secure transport, or in generating/verifying the response to an Authenticator -initiated challenge (described in section 0).

5.1.1 Device, Subscriber, and End-User Credentials

Three distinct types of credentials are used in the NGN: Device credentials, Subscriber credentials, and End-User credentials~~Three distinct types of credentials are defined by the NGN Security Architecture: Device credentials, Subscriber credentials, and End-User credentials~~.

Device credentials maybe supplied by the manufacturer with the device. For example, during the manufacture of the device, it may have a credentials “burned in” from the manufacturer, which includes such information as the device serial number, the manufacturer, etc. Device credentials identify the device to the NGN Carrier network. The NGN Carrier may associate device credentials with a particular subscriber’s service to alleviate the need for subscriber credentials; in such cases all calls made from the device are charged to the account of the subscriber.

Subscriber credentials are used for association of the originator of an NGN request with a particular account~~Subscriber credentials associate the originator of a NGN request with a particular account~~. Subscriber credentials are entered (via download, SIM, etc.) in the devices capable of accepting such credentials. Subscriber credentials installed on a device associate the subscriber with that device. All calls made from the device will be charged to the subscriber whose credentials are installed on the device. Multi-line devices may install multiple sets of credentials on the same device, in which case the device provides means to segregate among calls associated with each subscriber. Information needed to charge subscribers is provided in or associated with the subscriber credentials.

Ed Note: subscriber not person, give illustrative example linking to individual

End-User credentials are used to identify and authenticate specific users to the network. For example, a SIM card can identify the person; when the person places their SIM card into the phone it becomes associated with (and all calls are identified as being from) that person. Another example is a Secure Token – a request originating from a NGN device will be identified and authenticated as being from the person assigned that Secure Token. End-users are people that use the service associated with a subscriber, and calls originated by the end-user are charged to the subscriber. The subscriber and end-user may be the same, or there may be many end-users for a single subscriber. End users can identify and authenticate themselves with the network to take advantage of personal services. Individual transport layer security associations may be established, using end user credentials, between the CPE and NGN Network (Authenticators). The NGN Carrier associates the end user credentials with a particular subscriber service for billing purposes.

5.1.2 X.509 Certificates as credentials

Certificates [X509] may be used by the NGN Network elements in establishing security associations with other network elements, and provide the basis for mutual identification and authentication. They can also be used between CPE and the Authenticator for the same purposes.

Contents of CPE certificates are given in sections x. A certificate that is used to identify a device is described in section y, a certificate that is used to identify a subscriber is described in section z, and a certificate that is used to identify an end-user is described in section zz.

For a subscriber or end-user certificate, the <Subscriber Account Identifier> is used by the Authenticator to obtain further information about the credentials from the SAA/TAA-FEs. For a device certificate, the Device Manufacturer and Device Serial Number are used by the Authenticator to determine the associated <Subscriber Account Identifier> (valid only if the device has been associated with a subscriber), and then the <Subscriber Account Identifier> is used as above.

End-user, Service and Device certificates may be used in creating TLS connections between the device and the Authenticator (section 0), or may be used in creating IPsec connections through IKE Authentication (section 0).

5.1.3 Shared keys as credentials

Every key must have a name, and the key name must be unique. One copy of the shared key is given to the subscriber or end-user, and one copy is stored in the SAA/TAA-FEs.

When using pre-shared keys, the strength of the system is dependent upon the strength of the shared secret. The goal is to keep the shared secret from being the weak link in the chain of security. This implies that the shared secret needs to contain as much entropy (randomness) as the cipher being used. In other words, the shared secret should have at least 128-160 bits of entropy. ~~This means if the shared secret is just a string of random 8-bit bytes, then the key can be 16-20 bytes. If the shared secret is derived from a passphrase that is a string of random alphanumeric, then it should be at least 22-27 characters. This is because there are only 64 characters (6 bits) instead of 256 characters (8 bits) per 8-bit byte. Both random 8-bit bytes and random 6-bit bytes assume truly random numbers. If there is any structure in the password/passphrase, like deriving from English, then even longer passphrases are necessary. A passphrase composed of English would need on the order of 60-100 characters, depending on the mixing of case. Using English passphrases (or any language, for that matter) creates the problem that, if an attacker knows the language of the passphrase then they have less space to search. It is less random. This implies fewer bits of entropy per character, so a longer passphrase is required to maintain the same level of entropy.~~

~~Text from Martin E., contribution to July~~

The key name is used by the Authenticator to obtain further information about the credentials from the Service Authentication and Authorization-FE (SAA-FE) or Transport Authentication and Authorization-FE (TAA-FE).

5.1.4 Information provisioned in SAA/TAA-FEs for each set of credentials

The SAA/TAA-FEs are the repositories for all device, subscriber, and end-user credentials in the NGN infrastructure. It will typically be implemented as an integral part of the Authenticator in order to optimize authentication requests. However, to support mobility, it may be necessary for the Authenticator to consult a remote SAA/TAA-FEs server to obtain information about credentials. The Subscriber Account Identifier, or the Key Name, is used to fetch this information from the SAA/TAA-FEs.

The following security-related information must be provisioned in the SAA/TAA-FEs and associated with each set of credentials: (1) the key name or the subscriber account identifier, (2) whether end-user identification and authentication is required by this subscriber, (3) whether these credentials describe a subscriber or an end-user, (4) allowable values of the “From” header in requests, and (5) the proper setting of the P-Asserted-Identity header value for requests from this subscriber/end-user.

Following are several examples of the information stored in the SAA/TAA-FEs.

For a CPE NGN device that handles four POTS lines, with numbers 212-555-1111-1113 and 1151:

Subscriber Account: 123-456789
From headers: sip:212-555-111[1-3]@NGN .ngn.com
| sip:212-555-1151@NGN .ngn.com
Identity string: sip:212-555-1111@NGN .ngn.com
Type of credentials: subscriber
End-User ID required: no

For a subscriber certificate assigned to the John Doe family:

Subscriber Account: Doe-family
From headers: sip:*Doe@NGN .ngn.com
Identity string: sip:Doe@NGN .ngn.com
Type of credentials: subscriber
End-User ID required: no

For a pre-shared key assigned to the John Doe family:

Key name: JohnDoe
Key: dfe56131d1958046689d83306477ecc
From headers: sip:*Doe@NGN .ngn.com
Identity string: sip:Doe@NGN .ngn.com
Type of credentials: subscriber
End-User ID required: no

For a CPE-BE serving the Acme Widget Company:

Subscriber Account: Acme Widget Company
From headers: sip:*@acme.com
Identity string: sip:acme.com
Type of credentials: subscriber
End-User ID required: no

For an end-user at the Acme Widget Company:

Subscriber Account: Acme Widget Company
From headers: sip:bob@acme.com
Identity string: sip:bob@acme.com

Type of credentials: end-user

5.2 Identification and Authentication of Subscribers

5.2.1 General Strategy

The originator's identity in SIP is generally contained in the "From" header. However, identification of the subscriber through the use of the "From" header in a SIP request is susceptible to spoofing attacks and is therefore not used. Instead, the value of the "From" header is compared against the subscriber identity obtained by other means.

In order to minimize the effect on call setup delay, the identification and authentication of the subscriber is derived from the Network Source Address or the transport security association whenever possible. When these techniques do not produce an identification consistent with the "From" header in the SIP request, then a challenge is issued to the originator; if the response contains proper credentials then the request will proceed. Further details of these procedures are described in the following sections.

Based on the Network Source Address, and the procedures described in section 0, the Authenticator determines (1) the Subscriber can't be determined by this method, (2) a Subscriber is determined and it matches the "From" header in the request, or (3) a Subscriber is determined but it is different than the "From " header in the request.

Based on the Transport Security association, and the procedures described in section **Error! Reference source not found.**, the Authenticator determines (1) the Subscriber can't be determined by this method, (2) the Subscriber is determined and it matches the "From" header in the request, or (3) the Subscriber is determined but it is different than the "From" header in the request.

The actions then taken by the Authenticator are given in the following table:

Source Address determination of Subscriber	Transport Security determination of Subscriber	Authenticator Actions
N/A	N/A	Use Challenge/Response
N/A	Match	OK
N/A	Different	Use Challenge/Response
Match	N/A	OK
Match	Match	OK
Match	Different	Use Subscriber identity from Network Source Address
Different	N/A	Use Challenge/Response
Different	Match	Use Subscriber identity from Transport Security Association
Different	Different	Use Challenge/Response

If the resulting action is to use a challenge/response, the procedures of section 0 are followed.

The authentication strategy described above is a typical example, and each NGN provider may use other strategies (e.g., using just one procedure described in the following clauses.)

5.2.2 Identification of the Subscriber through Network Source Address

This is the simplest form of subscriber identification, based solely on the source address provided with the IP packets. The Authenticator may consult a pre-provisioned mapping of IP address ranges to <Subscriber Account Identifier>, and if the source address of the request is within one of these ranges, the Authenticator considers the request to be originated from that subscriber. The <Subscriber Account Identifier> is then used to fetch the subscriber credentials from the SAA/TAA-FEs and check the consistency of the value of the "From" header.

If the value of the "From" header is consistent with the subscriber, then it is considered a "Match"; if the value of the "From" header is not consistent with the subscriber, then it is considered "Different"; if the source IP address is not contained in any of the pre-provisioned address ranges, then it is considered "N/A".

The strength of this method of subscriber identification depends on providing Source-Address Assurance. The Source-Address Assurance means that the IP address can be used only by the legitimate subscriber to whom the address is assigned. To achieve this, the following two mechanisms are necessary for transport-processing or transport-control FEs, and they must be properly coordinated: 1) strict management of a mapping between a subscriber and his/her assigned address, and 2) prevention of address spoofing based on this managed information. See Appendix I for examples of the above mechanisms and their coordination.

~~Editors note: add additional clarity.~~

5.2.3 Identification of the Subscriber through TLS/IPsec Security Association

If a secure TLS transport was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE-BE certificate (see section x), the Authenticator may check that the "From" header and "P-Asserted-Identity" header (which was added by the CPE-BE) are consistent with the allowed values for the subscriber identified in the <Subscriber Account Identifier> contained in the certificate. If the "From" header value is consistent with those allowed by the subscriber, then the "P-Asserted-Identity" header added by the CPE-BE is retained. If no "P-Asserted-Identity" header was added by the CPE-BE, then this method of identification is considered "N/A", and the Authenticator will depend on other subscriber identification methods.

If a secure transport (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE NGN device certificate (see section x) then the Authenticator utilizes the Device Manufacturer and Device Serial Number to determine the associated <Subscriber Account Identifier> (valid only if the device has been associated with a subscriber). The <Subscriber Account Identifier> is then used to fetch the subscriber credentials from the SAA/TAA-FEs and check the consistency of the value of the "From" header.

If a secure transport (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE NGN subscriber certificate (see section x) then the Authenticator utilizes the <Subscriber Account Identifier> to fetch the subscriber credentials from the SAA/TAA-FEs and check the consistency of the value of the "From" header.

If a secure transport (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE NGN End-User certificate (see section x), then the Authenticator utilizes the <Subscriber Account Identifier> to fetch the subscriber credentials from the SAA/TAA-FEs and check the consistency of the value of the "From" header.

If a secure transport (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a pre-shared key (see section 0), the Authenticator utilizes the Key Name to fetch the subscriber credentials from the SAA/TAA-FEs, and check that the "From" header is consistent with the allowed values.

If a secure transport was not used between the originating device and the Authenticator, or an "anonymous client" TLS connection was used, then this method is "N/A".

5.2.4 Identification of the Subscriber through Challenge/Response

Challenge/response is a more secure version of the old style userid/password scheme (i.e. sending of a user identification and password as part of a request for service, and the problem being that it was easily replayed to obtain fraudulent service later). In a challenge/response scheme, the server sends a challenge to the client, asking the client to perform some encryption task using a shared key. The result of that calculation is included in the response, which is then verified by the server. If the exchange were intercepted by others, it cannot be replayed as long as the server never reuses an old challenge.

There is one important type of the challenge-response methods that combines convenience of the password-based authentication methods and security of the methods that are based on the challenge-response scheme. The Password-authenticated key Exchange protocol (PAK), presents this type. The PAK protocol ensures mutual authentication of both parties in the act of establishing a symmetric cryptographic key via Diffie-Hellman exchange. The use of Diffie-Hellman exchange ensures the *Perfect Forward Secrecy* – a property of a key establishment protocol that guarantees that compromise of a session key or long-term private key after a given session does not cause the compromise of any earlier session**.** With the PAK authentication method, the exchange, in addition, is protected from the *man-in-the-middle* attack. The authentication relies on a pre-shared secret (e.g., password), which is protected (i.e., remains unrevealed) to an eavesdropper preventing an off-line dictionary attack. Thus, the protocol can be used in a wide variety of applications where pre-shared secrets based on the possibly weak password exist. The PAK protocol is specified in the ITU-T Recommendation X.1035, Password-authenticated key exchange (PAK) protocol.~~A Recommendation specifying PAK protocol is being developed by SG 17. (Note: This statement should be changed after the Recommendation X.pak has been approved).~~ A similar specification of the PAK protocol has been developed by 3GPP2 and standardized in TIA Standard TIA-683-D (2005), *Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems.*

A challenge/response involves an additional message exchange between the Authenticator and the originating endpoint, and a calculation done by the originating endpoint. It therefore has an impact on the delay perceived by the user. It is the goal of the NGN Security Architecture to use a challenge/response only when absolutely necessary to achieve the necessary level of identification and authentication.

If a secure transport connection (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and a previous request within a configurable time period with the same "From" header contents was successfully authenticated by the Authenticator, then the authentication is considered successful and the request is accepted. Since the typical first request over a new connection is a "Register", this challenge/response will be done at a time that will not affect call setup delay.

Since authentication requests may be very computationally-intensive, it is essential that the Authenticator limit the queries to the SAA/TAA-FEs. The limits defined in this paragraph may be followed whether the SAA/TAA-FEs is an integral part of the Authenticator or if it is a separate element. A very simple Denial-of-Service attack is for an endpoint to simply flood the

Authenticator with incorrect requests – if each requires a cryptographic calculation in the SAA/TAA-FEs then service is essentially halted for all valid requests. The Authenticator may locally reject a request if there is a pending authorization request from the same endpoint. The Authenticator may locally reject a request if there have been at least XXX total requests within the past YYY seconds (both XXX and YYY values to be configurable in the Authenticator). The Authenticator may deliberately respond slowly to a failed authorization request; by deliberately waiting a configurable period of time; this prevents various kinds of “password cracking” attacks.

5.2.4.1 Challenge/Response with SIP signaling from originating device

If the originating device is using the SIP signaling protocol, then the Proxy-Authenticate mechanisms defined in [RFC3261] may be used to implement a challenge/response. See Section 22.2 of [RFC3261], section 3 of [RFC2617], and section 3 of [RFC3310].

The Authenticator responds to the SIP request with a 407 (Proxy-Authentication-Required) response. In this response it may include a Proxy-Authenticate header with: Authentication Scheme of “Digest”, Realm of “NGN .ngn.net”, Qop of “auth”, Nonce of a cryptographically random 16-byte value (in hex), optionally a value of the “Opaque” parameter, and Algorithm of “MD5” or “AKAv1-MD5” depending on service agreement with the Customer.

An example of a Proxy-Authenticate header in a 407 response is:

> Proxy-Authenticate: Digest realm=“NGN .ngn.com", qop="auth", nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="", stale=FALSE, algorithm=MD5

The originating device responds to the 407 with a regenerated request, containing a Proxy-Authentication header. This header may be verified to contain the following information: Authentication scheme of “Digest”, Realm identical to that in the 407 response, Nonce identical to that in the 407 response, and Opaque identical to that in the 407 response. In addition, the Proxy-Authentication header includes a “Username” parameter giving the key name, a “Uri” parameter matching the Request-URI of the request, and a “Response” parameter being the hash as specified in [RFC2617] or [RFC3310].

An example of a Proxy-Authorization header in a re-issued request is:

> Proxy-Authorization: Digest username="bob", realm=“NGN .ngn.com“, nonce="ea9c8e88df84f1cec4341ae6cbe5a359", opaque="", uri="sip:5551212@ngn.com", response="dfe56131d1958046689d83306477ecc"

The “User-to-User Authentication” mechanisms defined in [RFC3261] may also be used to implement a challenge/response. See Section 22.2 of [RFC3261], section 3 of [RFC2617], and section 3 of [RFC3310] for details.

5.2.4.2 Challenge/Response with signaling other than SIP from originating device

If the originating device is using a signaling protocol other than SIP, then the challenge/response is considered to have failed. The request is rejected.

5.2.5 Generic Bootstrapping Architecture (GBA)

[EdNote: this section is a placeholder for further discussion. Main body is in Appendix I. Contributions to describe generic bootstrapping mechanisms are invited that will include 3GPP/3GPP2 GBA as an option.]

Generic Bootstrapping Architecture is an optional mechanism for mutual authentication of the End-User and an application server (NAF).

5.3 Identification and Authentication of End-Users

5.3.1 General Strategy

While identification of the subscriber is absolutely required for the NGN infrastructure, identification of the end-user is an optional service that may be requested by the subscriber. Typically this would be to provide additional services, e.g. personal mobility and presence, where the identity of the requesting user is required to enable the service. If a subscriber desires this additional level of identification, it is necessary that all the endpoint devices support the ability to enter additional end-user credentials or to use an end-user instead of a subscriber certificate.

End user credentials may be used to create a TLS association or to register an end-user on top of a TLS association created by device or subscriber credentials with or without any association with the end-user credentials. Calls made by the end-user will be charged to the service to which the end-user belongs.

Two methods exist for the Authenticator to identify and authenticate the end-user. The first is through the transport level security association used for the signaling exchange – if that security association was established with an end-user certificate (or a pre-shared key associated with a single end-user), then the end-user identification is complete. The second method is through a challenge/response, where the key name given in the response is associated with a single end-user. These two methods are described further in the following sections.

Advanced NGN devices may have multiple identities, e.g. a subscriber certificate and also one or more end-user certificates for the person(s) currently using the device. Such a device would create multiple TLS connections to the Authenticator, one separate connection for each certificate. The device would then send requests to the Authenticator over the appropriate signaling connection based on the desired identity for the call.

There is a concern about credentials for a single user being valid long after the user has “left”. If the transport security association was based on an end-user certificate, the subscriber may require continuous activity to maintain the validity of the authentication. Without such activity, the Authenticator closes the secure transport connection, and requires the originating device to re-establish it with the current end-user certificate (or subscriber certificate or device certificate if no end-user certificate is available). The actual requirements for this behavior of Authenticator are given in section 0 and 0, and are based on two timers: one that limits the absolute amount of time for which user credentials can be valid for a security association, and the second that limits the idle time between successive requests. Timeout values may be provisioned per subscriber or end-user, but have to be limited to maximum values.

5.3.2 Identification of the End-User through TLS/IPsec Security Association

If a secure TLS transport was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE-BE certificate (see section x), the Authenticator verifies that the “From” header is consistent with the allowed values for the subscriber identified in the <Subscriber Account Identifier> contained in the certificate. If the “From” header value is consistent with those allowed by the subscriber, then the “P-Asserted-Identity” header, containing the end-user identification and added by the CPE-BE, is retained. If no “P-Asserted-Identity” header is present, then a challenge/response is needed to identify the end-user.

If a secure transport (either IPsec or TLS) was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a X.509 CPE NGN End-User certificate (see section x), then the Authenticator utilizes the <Subscriber Account Identifier> to fetch the subscriber credentials from the SAA/TAA-FEs and checkes the

consistency of the value of the “From” header. If consistent, the value of the “P-Asserted-Identity” header is taken from the <End-User Identifier> contained in the certificate.

If a secure IPsec transport was established for the signaling traffic between the originating device and the Authenticator, and that secure transport was authenticated with a pre-shared key (see section 0), the Authenticator utilizes the Key Name to fetch the subscriber credentials from the SAA/TAA-FEs, and checks that the “From” header is consistent with the allowed values. If the retrieved credentials indicate that the key is associated with an end-user, then the “P-Asserted-Identity” header is taken from the information stored in the SAA/TAA-FEs about the key.

Otherwise a challenge/response is needed to identify the end-user.

5.3.3 Identification of the End-User through Challenge/Response

The challenge/response procedures for identification of an end-user are identical to those used to identify the subscriber, as given in section 0.

The only extension is that the Authenticator checks the information retrieved from the SAA/TAA-FEs for the key name for an indication that the key is associated with an end-user. If so, then the end-user identification is successful.

If the Authenticator had already done a challenge/response to identify the subscriber, and the named key returned in the response did not identify an end-user, then the end-user identification fails. If a challenge/response was not needed to identify the subscriber, a challenge is issued now.

5.4 Identification and Authentication with CPE-BE

The identification and authentication procedures performed by a CPE-BE are identical to those performed by an Authenticator with only three differences:

The CPE-BE may be provisioned with all the credentials needed to identify and authenticate the subscriber(s) and end-users that it serves, since it has no access to the distributed SAA/TAA-FEs function available to an Authenticator,

If identification of the subscriber fails (visiting end-user), the request is passed to the Authenticator without a P-Asserted-Identity header, rather than rejecting the request, and

The request re-issued in response to a challenge from the Authenticator, containing the “Proxy-Authorization” header, is passed to the Authenticator rather than processed at the CPE-BE.

5.4.1 Use of X.509 Certificates

There is a security association between every CPE-BE and at least one network BE, established with the X.509 certificate issued to the CPE-BE. Requests received at the BE follow the identification and authentication procedures given in section **Error! Reference source not found.**, which result in minimal verification of the identification performed by the CPE-BE. When a challenge/response is needed (e.g. for a “roaming” user), the exchange will be between the originating endpoint and the BE, and transparently passed through the CPE-BE.

A secure transport between the end-point and the CPE-BE is optional. It is anticipated that the network source address will adequately identify most requests.

End-points register to network BE via CPE-BE.

5.5 Authenticator-SAA/TAA-FEs Interface

5.5.1 Authenticator-SAA/TAA-FEs Interface for Digest Authentication (section needs clarification and rewording)

The SAA/TAA-FEs contains the decision point and is the repository for all end-user and device credentials in the NGN infrastructure. Some SAA/TAA-FEs functions like authentication, may be distributed among the Authenticator in order to optimize authentication request performance.

Ed Note: When Authenticator does not have the credentials for authenticating a user (due to endpoint nomadity, loadsharing, etc.) there is a need for a protocol to be defined between the Authenticator and the SAA/TAA-FEs for the Authenticator to obtain authentication information. Need to clarify the need

There are two competing choices for Authenticator to SAA/TAA-FEs communication. RADIUS [RFC2865] (well-known and well-supported) and ~~DIAMETER~~ Diameter [RFC3588] (newly defined to fix several deficiencies of RADIUS). It is the eventual goal of the NGN infrastructure to migrate to ~~DIAMETER~~Diameter; however we recognize that current implementation of servers are based on RADIUS, and that numerous extensions of the basic RADIUS protocol have been developed to meet the needs of this authentication function. While this release of this document is based on RADIUS with the extension described in [RFC4590~~draft-sterman-sip-aaa-03~~], a future release will likely change this interface to be based on ~~DIAMETER~~ Diameter with the extension described in [RFC4740~~draft-ietf-aaa-diameter-sip-app-03~~].

The Authenticator may implement a RADIUS client, and the SAA/TAA-FEs server may implement a RADIUS server, as defined in [RFC2865]. Both may implement the extensions for SIP Digest Authentication, as given in [~~draft-sterman-sip-aaa-03~~RFC4590]. The connection between the Authenticator and SAA/TAA-FEs server may be secured with IPsec with mutual authentication. ~~(note: need to update references)~~

With this extension, the Authenticator makes a RADIUS request with the parameters from the Proxy-Authentication header; the RADIUS server calculates the expected response and returns it to the Authenticator. The Authenticator then validates the request by comparing the actual response from the endpoint to the expected response.

An example of the message sent from the Authenticator to the SAA/TAA-FEs server is:

Code = 1 (Access-Request)

Identifier = 1

Length = 164

Authenticator = 56 7b e6 9a 8e 43 cf b6 fb a6 c0 f0 9a 92 6f 0e

Attributes:

NAS-IP-Address = d5 89 45 26 (213.137.69.38)

NAS-Port-Type = 5 (Virtual)

User-Name = “bob"

Digest-Response (206) = "2ae133421cda65d67dc50d13ba0eb9bc"

Digest-Attributes (207) = [Realm (1) = “NGN .ngn.com"]

Digest-Attributes (207) = [Nonce (2) = " ea9c8e88df84f1cec4341ae6cbe5a359 "]

Digest-Attributes (207) = [Method (3) = "INVITE"]

Digest-Attributes (207) = [URI (4) = " sip:5551212@ngn.com "]

Digest-Attributes (207) = [Algorithm (5) = "md5"]

Digest-Attributes (207) = [User-Name (10) = “bob"]

An example of the response sent from the SAA/TAA-FEs server to the Authenticator is:

Code = 2 (Access-Accept)

Identifier = 1

Length = 20

Authenticator = 6d 76 53 ce aa 07 9a f7 ac b4 b0 e2 96 2f c4 0d

Attributes:

Digest-Response (206) = "dfe56131d1958046689d83306477ecc"

5.5.2 Authenticator-SAA/TAA-FEs Interface for Credential Information Retrieval

When an X.509 Certificate is used in the establishment of the transport signaling security association, the SAA/TAA-FEs stores (indexed by the <Subscriber Account Identifier>) the set of acceptable “From” headers that may appear in requests from that source, which will be matched against the “From” header provided in the request. The SAA/TAA-FEs also stores an identity string for the use of the Authenticator in the P-Asserted-Identity header if the “From” header matches.

If a pre-shared key is used in the establishment of the transport signaling security association (e.g. similar peering service provider, or CNGN customer), then the SAA/TAA-FEs stores (indexed by the Key name) the set of acceptable “From” headers that may appear in requests from that source, which will be matched against the “From” header provided in the request. The SAA/TAA-FEs also stores an identity string for the use of the Authenticator in the P-Asserted-Identity header if the “From” header matches.

The protocol for the Authenticator to access this database is specified elsewhere.(provide reference or delete).

5.6 Identification and Authentication of flow of bearer traffic

EdNote: Rationale of this mechanism vs. IPsec necessary

EdNote: It must be clarified that t is a nonce, and it is a timestamp, it must be noted that the time in the network is assumed to be synchronized up to a defined skew (left to the operators)

EdNote: The size of the integer t+i must be specified. (Probably left to the operators?) Need to make a recommendation at least for the minimum value.

In NGN, a flow of the bearer traffic can be identified by a quintuple that contains

source IP address,

destination IP address,

source port,

destination port, and

protocol number.

The identification mechanism described in this section uses this identifier for authentication of every packet. The mechanism is based on a shared secret and the use of the cryptographic hash functions, such as keyed-Hash Message Authentication Code (HMAC) [HMAC-FIPS].

The entities involved in the process of authentication – the End-User Function and the Access Node FE – are described in [Y.NGN FRA] and depicted in Figure 1.

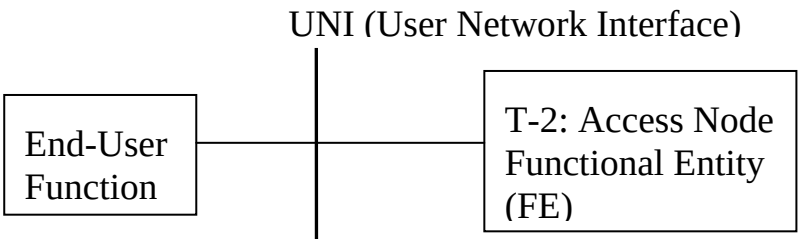


Figure 1 NGN Entities involved in authentication procedure.

The description of the mechanism uses the following conventions:

- *F* is a flow identifier (quintuple)
- *K* is a shared secret that both, the End-User Function and the Access Node FE possess
- *P* is a packet that the End-User Function intends to send to the Access Node FE
- *i* is a sequence number of a packet that is being incremented by both communicating parties
- *t* is a time stamp
- *(P', Q)* is a packet that the Access Node FE has received

When the End-User Function intends to send a packet *P* to the Access Node FE, it first computes a quantity *H(F, t+i, K)*, which is a hash function of a concatenation of *F*, *t+i*, and *K*, and then attaches this quantity to the packet *P*. So, what the End-User Function really sends to the Access Node FE is a packet *[P, H(F, t+i, K)]*. When the Access Node FE receives a packet *(P', Q)*, it computes the quantity *H(F, t+i, K)* and checks whether it is equal to *Q*. If it is not, then the Access Node FE drops the packet. Otherwise, the Access Node FE is assured that an entity that knows the shared secret *K* has sent the packet (in this case *P* and *P'* are the same) and admits it into NGN.

The use of the specified authentication mechanism prevents the denial-of-service attack that is described in the contribution *Identification and authentication of bearer traffic*. [Note: the reference to this contribution should be replaced by a reference to the draft Recommendation Y.NGN Authentication when the contribution is accepted.]

The mechanism also allows for authentication of the user-generated traffic without revealing the user's identity.

The use of this mechanism is a subject of a network operator's security policy. There are other mechanisms that can be used for authenticating the flows (e.g., IPsec, TLS).

6. Transport Security for Signaling and OAMP

Transport security is used in the NGN infrastructure to achieve confidentiality and integrity guarantees of the signaling data and the operations administration and maintenance messages. This section specifies the profile of TLS and IPsec to be used by the NGN infrastructure network

elements as two of the important mechanisms. The list of mechanisms are not exhaustive and other implementations may be adopted depending on the NGN provider policies.

Within the Trusted Domain and Trusted-but-Vulnerable Domain, VPN tunnel (e.g., IPsec or TLS)is required for securing the OAMP. Section 0 gives the profile of IPsec to be used in these cases. Between the CPE -BE and OAMP-SE (i.e. between the Untrusted Domain and Trusted-but-Vulnerable Domain), IPsec is used for creating a VPN tunnel. Section 0 gives the profile of IPsec to be used in these cases.

While media security is not required within the NGN infrastructure, some Border elements implement media security for service to specific endpoints. For these elements, a later section contains a profile of media security protocols.

6.1 TLS

In the NGN infrastructure, TLS may be used to secure various(generic) types of signaling traffic (e.g. SIP, COPS, TRIP, HTTP, etc) between network elements within the trusted domain. It is also supported in Border Elements that might receive encrypted signaling from customer endpoints, and by the CPE-BE for communicating to a NBE. Specific requirements for each type of network element are given in Section x.

The TLS protocol is defined in [RFC2246]. It provides privacy and data integrity over a reliable transport layer protocol such as TCP or SCTP. The protocol is comprised of two layers: the TLS Record Protocol and the TLS Handshake Protocol. The TLS Record Protocol is used to securely encapsulate upper layer protocols, while the TLS Handshake Protocol provides the key management functionality required to establish TLS sessions.

Editor’s note: do we need ‘record protocol’ and handshake protocol here?

Unless specified otherwise in this section, NGN infrastructure elements requiring TLS may be compliant with the TLS specification [RFC2246] and any requirements specified in [RFC3261] relating to its usage in SIP. While TLS supports the negotiation and use of compression methods, compression may NOT be used within the NGN infrastructure, die to performance degradation.

6.1.1 Ciphersuites

The ciphersuite includes the authenticated key agreement method used in the TLS handshake, as well as encryption and authentication ciphers used to secure the record layer. Ciphersuites are negotiated with the TLS client presenting a list of supported ciphersuites in the Client Hello message, and the server responding with the selected ciphersuite in the Server Hello message.

Editor’s note: to make this section in table format to make it more concise and easier to understand.

The following list is not exhaustive:

The ciphersuite TLS_RSA_WITH_AES_128_CBC_SHA, defined in [RFC 3268], should be supported in all network elements. This ciphersuite includes authenticated key agreement based on RSA, encryption based on AES-128 in CBC mode, and authentication using the SHA-1 function.

The ciphersuite TLS_DHE_RSA_WITH_AES_128_CBC_SHA, defined in [RFC 3268] should be supported in all network elements. This ciphersuite includes authenticated key agreement based on ephemeral Diffie-Hellman with RSA signatures, encryption based on AES-128 in CBC mode, and authentication using the SHA-1 function.

The ciphersuite TLS_RSA_WITH_RC4_128_SHA ciphersuite, defined in [RFC2246], may be supported in all nework elements. This ciphersuite includes authenticated key management based on RSA, encryption based on RC4-128 and authentication using the SHA-1 function.

The ciphersuite TLS_RSA_WITH_3DES_EDE_CBC_SHA, defined in [RFC2246], may be supported in network elements. This ciphersuite includes authenticated key agreement based on RSA, encryption based on 3DES in CBC mode, and authentication using the SHA-1 function.

The ciphersuite TLS_DHE_WITH_3DES_EDE_CBC_SHA, defined in [RFC2246], may be supported in network elements. This ciphersuite includes authenticated key agreement based on ephemeral Diffie-Hellman with RSA signatures, encryption based on 3DES in CBC mode, and authentication using the SHA-1 function.

The ciphersuite TLS_RSA_WITH_CAMELLIA_128_CBC_SHA, defined in [Camellia02], may be supported in all network elements. This ciphersuite includes authenticated key agreement based on RSA, encryption based on CAMELLIA-128 in CBC mode, and authentication using the SHA-1 function.

The ciphersuite TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA, defined in [Camellia02], may be supported in all network elements. This ciphersuite includes authenticated key agreement based on ephemeral Diffie-Hellman with RSA signatures, encryption based on AES-128 in CBC mode, and authentication using the SHA-1 function.

[Editor's note: contributions on other options are invited to compare various features of encryption algorithms.]

[Editor's note: do we need backup algorithms such as Camellia? This comment is an global one.]

The ciphersuits below [RFC 3268, RFC 4132] MAY be used by any NEs.

TLS_DH_DSS_WITH_AES_128_CBC_SHA,

TLS_DH_RSA_WITH_AES_128_CBC_SHA,

TLS_DHE_DSS_WITH_AES_128_CBC_SHA,

TLS_DHE_RSA_WITH_AES_128_CBC_SHA,

TLS_RSA_WITH_AES_256_CBC_SHA,

TLS_DH_DSS_WITH_AES_256_CBC_SHA,

TLS_DH_RSA_WITH_AES_256_CBC_SHA,

TLS_DHE_DSS_WITH_AES_256_CBC_SHA,

TLS_DHE_RSA_WITH_AES_256_CBC_SHA,

TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA,

TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA,

TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA,

TLS_RSA_WITH_CAMELLIA_256_CBC_SHA,

TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA,

TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA,

TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA,

TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA,

[EdNote: The paragraph on ECC below was added with the understanding that follow-up contribution to describe two items (a) and (b) are necessary.

a) The use of ECC in the existing IETF-standardized profiles for TLS or IPsec (or both)

b) The clear advantages (or disadvantages) of the use of ECC vis. RSA. El Gamal, and other PK algorithms in the NGN context.

RFC 4492 is an informational RFC, so description that does not mention the RFC is also necessary.

]

RFC 4492 Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) describes new key exchange algorithms based on Elliptic Curve Cryptography (ECC) for the Transport Layer Security (TLS) protocol. In particular, it specifies the use of Elliptic Curve Diffie-Hellman (ECDH) key agreement in a TLS handshake and the use of Elliptic Curve Digital Signature Algorithm (ECDSA) as a new authentication mechanism.

6.1.2 TLS Use of Certificates

TLS is a client-server based protocol with optional client authentication. However, within the Trusted Domain of the NGN infrastructure, and between the Trusted Domain and the Trusted-but-Vulnerable Domain, mutual authentication may be used. The TLS server may send a Certificate Request to the Client. If a client in the Trusted Domain or Trusted-but-Vulnerable Domain does not provide a client certificate, then the connection request may be rejected by the server. Both the TLS client and server certificates may conform to the NGN infrastructure certification specifications given in section x. Certificates may be verified as specified in section x. Before continuing with a TLS connection, the TLS server or client may validate the remote system matches its certificate.

Between the Trusted-but-Vulnerable Domain and the Untrusted Domain, the TLS server may send a Certificate Request to the Client. If the client has no certificate, it responds with an empty Client Certificate message, and the session proceeds as an anonymous client.

When a Border Element accepts an authenticated connection with an endpoint based on a NGN end-user certificate (see section 0), then the BE may implement two timers on the connection. The first timer, T1, is started when the connection is established. The second timer, T2, is started when the connection is established and is reset to zero every time a request is received at the BE over the connection. Whenever either timer reaches its limit value (which may depend on values contained in the certificate), the connection is reset by the BE and will be re-established by the endpoint to refresh the NGN end-user certificate.

6.1.3 Connection Persistence and Re-Use

TLS session establishment requires a TCP 3-way handshake, several round-trips to establish TLS, expensive asymmetric authentication and key generation algorithms, and certificate verification. This can be quite costly both in terms of performance and network latency, and is therefore not suited for on-demand SIP signaling. As such, Clients may re-establish the TLS connection with the server after an interruption in communication with the server. SIP connection re-use, as defined in [draft-ietf-sip-connect-reuse-xx] may be supported.

6.1.4 Session Caching

TLS makes it possible to resume a previous session if it has been cached on both the TLS client and server. Resuming sessions drastically speeds up the session establishment, as fewer messages are exchanged and authentication is based on symmetric key cryptography. TLS session caching SHOULD be supported. A TLS client initiating a TLS session may attempt to resume a cached session if it has retained a session for the remote server. A TLS client or server SHOULD retain a cached session for at least a configurable period of time.

6.1.5 Session Key Management

TLS sessions between NGN infrastructure network elements are expected to be long-lasting, e.g. years. It is therefore important that the session keys be changed periodically. Session keys for TLS sessions may be changed after a configurable period of time.

6.2 IPsec in Trusted and Trusted-but-Vulnerable Domains

In the NGN infrastructure IPsec may be used to secure various types of traffic (e.g. SNMP, RADIUS, etc) between network elements within the trusted domain~~IPsec is used in the NGN infrastructure may be used to secure various types of traffic (e.g. SNMP, RADIUS, etc) between network elements within the trusted domain~~. Specific requirements for each type of network element are given in Section x.

As described generally in [RFC2401], IPsec is composed of a number of different pieces. These can be used to provide confidentiality, integrity, and replay protection. Some of these can be configured manually, but in general a key management component is used. Additionally, the decision on the use of IPsec is typically controlled by a policy database~~Additionally, the decision about whether and how to use IPsec is typically controlled by a policy database of some sort~~. This section describes the mandatory-to-implement subset of the components of IPsec.

Network elements that use IPsec SHOULD ensure that TLS-secured connections are not run over IPsec. Network elements that use IPsec may ensure that media streams secured with SRTP or RC4 are not run over IPsec. This is to ensure that no double-encryption is done, which would be wasteful of NGN ~~Architecture~~ resources. It should be noted that tunnelling of encryption may occur from the end user.

6.2.1 AH and ESP

The Authentication Header (AH), described in [RFC4302, RFC4305], and the Encapsulating Security Protocol (ESP), described in [RFC2406], are the two choices of over-the-wire security protocols. Both optionally provide replay protection. ESP typically is used to provide confidentially, integrity, and authentication of traffic. ESP also can provide integrity and authentication without confidentiality. ESP can also be used to provide confidentiality alone. AH protects portions of the preceding IP header, including the source and destination address. AH can also protect those IP options that need to be seen by intermediate routers, but must be intact and authentic when delivered to the receiving system, though use of such IP options is extremely rare.

NGN infrastructure network elements may support the Encapsulating Security Protocol (ESP), as defined in [RFC2406]. ESP_DES (both 40 and 56 bits), ESP_3DES, ESP_AES [RFC3602], and ESP_CAMELLIA [RFC4312] may be supported in Cipher Block Chaining (CBC) mode. Network Elements that support ESP_NULL may NOT use ESP_NULL when communicating with another NGN infrastructure network element. The actual encryption algorithm used within ESP is negotiated during key management.

All implementations of ESP are required by [RFC2401] to support the concept of Security Associations (SAs), and [RFC2401] provides a general model for processing IP traffic relative to SAs. Although particular IPsec implementations need not follow the details of this general model, the external behavior of any IPsec implementation may match the external behavior of the general model. This ensures that components do not accept traffic from unknown addresses and do not send or accept traffic without security (when security is required). NGN infrastructure network elements that implement IPsec may provide behavior that matches the general model described in [RFC2401].

6.2.2 Transport and Tunnel Mode

Both AH and ESP can be used in either transport mode or tunnel mode. In tunnel mode, the IPsec header is followed by an inner IP header. This is the normal usage for Virtual Private networks (VPNs), and is generally required when either end of the IPsec-protected path is not the ultimate destination, e.g. when IPsec is implemented in a firewall, router, etc. Transport mode is preferred for point-to-point communication.

NGN infrastructure network elements may support IPsec in Transport Mode.

6.2.3 Replay Protection

IPsec provides an optional replay-protection service (anti-replay service). Within NGN infrastructure network elements, the IPsec anti-replay service may be turned on at all times. An IPsec sequence number outside of the current anti-replay window is flagged as a replay and the packet is rejected. When the anti-replay service is turned on, an IPsec sequence number cannot overflow and roll over to 0. Before that happens, a new Security Association may be created as specified in [RFC2406].

6.2.4 Key Management

All cryptographic systems require key management. While IPsec provides for both manual and automatic key management schemes, manual schemes don't scale very well and do not offer replay protection. Key management schemes also provide authentication. NGN infrastructure network elements may implement one of the automated key exchange mechanisms described in this section.

Note: need to add IMS aka procedure.

When IKE is not used for key management, an alternative key management protocol needs an interface to the IPsec layer in order to create/update/delete IPsec Security Associations. IPsec Security Associations may be automatically established or re-established as required. This implies that the IPsec layer also needs a way to signal a key management application when a new Security Association needs to be set up (e.g. the old SA is about to expire or there is no SA on a particular interface). In addition, some Border Elements may be required to run multiple key management protocols (e.g. IKE for securing connections for OAMP, and PKINIT for securing connections). In these cases the PF_KEY [RC2367] interface SHOULD be used.

6.2.4.1 Transform Identifiers

The IPsec Transform Identifier is used by the key management procedures to negotiate an encryption algorithm that is used by ESP in IPsec. The transform identifier is also used by IKE to secure its phase-1 and phase-2 messages. A list of available IPsec transform identifiers is given in [RFC2407]. Within the NGN infrastructure, the transform IDs ESP_3DES (value 0x03, with key size of 192 bits, CBC mode) and ESP_CAMELLIA (value 0x16, with 128 bits key, CBC mode) [RFC4312] may be supported. The Transform ID ESP_AES (value 0x0C, with 128 bit key, CBC mode) SHOULD be supported. IKE allows negotiation of the encryption key size, so if in the future it is desired to increase the key size for one of the above algorithms, IKE will use this built-in function.

For all of these transforms the CBC Initialization Vector (IV) is carried in the clear inside each ESP packet payload [RFC2451]. AES-128, defined in [FIPS197, RFC3602] may be used in CBC mode with a 128-bit block size and a randomly generated Initialization Vector. AES-128 requires 10 rounds of cryptographic operations [RFC3602]. Camellia-128, defined in [RFC3713 and RFC4312] may be used in CBC mode with a 128-bit block size and a randomly generated Initialization Vector. It requires 18 rounds of cryptographic operations [RFC3713]

6.2.4.2 Authentication Algorithms

The IPsec Authentication Algorithm is used by the key management procedures to negotiate a packet-authentication algorithm that is used. A list of available IPsec authentication algorithms is given in [RFC2407]. Within the NGN infrastructure, the authentication algorithms HMAC-MD5-96 (value 0x01, key size of 128 bits, defined in [RFC2403]) and HMAC-SHA-1-96 (value 0x02, key size 160 bits, defined in [RFC2404]) may be supported.

6.2.4.3 Internet Key Exchange (IKE)

One automated key exchange mechanism is described in [RFC2409], and is known as IKE. IKE key management is completely asynchronous to data messages and does not contribute to any delays during communications setup. The only exception would be some unexpected error, where the Security Association is unexpectedly lost by one of the endpoints.

IKE is a peer-to-peer key management protocol. It consists of two phases. In the first phase, a shared secret is negotiated via a Diffie-Hellman key exchange. It is then used to authenticate the second IKE phase. The second phase negotiates another secret, used to derive keys for the IPsec ESP protocol.

6.2.4.3.1 First IKE Phase

Three different modes are defined for authentication during the first IKE phase. IKE authentication with Public-Key Encryption SHALL NOT be used in the NGN infrastructure, as this requires the initiator to already know the responder's public key. IKE Authentication with Signatures and IKE Authentication with Pre-Shared Keys may be supported.

IKE defines specific sets of Diffie-Hellman parameters (i.e. prime and generator) that may be used for the phase 1 IKE exchange. The first group may be supported in NGN infrastructure network elements, and the remaining groups SHOULD be supported.

6.2.4.3.1.1 IKE Authentication with Signatures

If IKE Authentication with signatures is used, both client and server may exchange X.509 certificates (see section x). Certificates may be verified as specified in section x.

When a Border Element accepts an authenticated connection with an endpoint based on a NGN end-user certificate (see section 0), then the BE may implement two timers on the connection. The first timer, T1, is started when the connection is established. The second timer, T2, is started when the connection is established and is reset to zero every time a request is received at the BE over the connection. Whenever either timer reaches its limit value (which may depend on values contained in the certificate), the connection is reset by the BE and will be re-established by the endpoint to refresh the NGN end-user certificate.

IKE Authentication with Pre-Shared Keys

If IKE Authentication with Pre-Shared keys is used, a key derived by some out-of-band (e.g. manual) mechanism is used to authenticate the exchange. Implementations may allow a pre-shared key of at least 128 bytes. Verification of the requirements for the pre-shared keys is not required in the network elements. Implementations may support Aggressive Mode, defined in Section 5.4 of [RFC2409], and use the key name as the identity of the initiator/responder.

When using pre-shared keys, the strength of the system is dependent upon the strength of the shared secret. The goal is to keep the shared secret from being the weak link in the chain of security. This implies that the shared secret needs to contain as much entropy (randomness) as the cipher being used. In other words, the shared secret should have at least 128-160 bits of entropy. ~~This means if the shared secret is just a string of random 8-bit bytes, then the key can be 16-20 bytes. If the~~

~~shared secret is derived from a passphrase that is a string of random alphanumeric, then it should be at least 22-27 characters. This is because there are only 64 characters (6 bits) instead of 256 characters (8 bits) per 8-bit byte. Both random 8-bit bytes and random 6-bit bytes assume truly random numbers. If there is any structure in the password/passphrase, like deriving from English, then even longer passphrases are necessary. A passphrase composed of English would need on the order of 60-100 characters, depending on the mixing of case. Using English passphrases (or any language, for that matter) creates the problem that, if an attacker knows the language of the passphrase then they have less space to search. It is less random. This implies fewer bits of entropy per character, so a longer passphrase is required to maintain the same level of entropy.~~

6.2.4.3.2 Second IKE Phase

In the second IKE phase, an IPsec ESP Security Association is established, including the ESP keys and ciphersuites. First, a shared second-phase secret is established, and then all the IPsec keying material is derived from it using the one-way function specified in [RFC2409]. The second-phase secret is built from encrypted nonces that are exchanged by the two parties. Another Diffie-Hellman exchange is allowed by [RFC2409] in addition to the encrypted nonces, but may NOT be used in NGN infrastructure network elements. This is to avoid the associated performance penalties.

6.2.4.4 PKINIT (Update with SG9 references)

TBD

6.3 IPsec between Untrusted and Trusted-but-Vulnerable Domains

The CPE-BE is a NGN Network Element that resides in the Untrusted Domain. However, it is still managed by the NGN Carrier and needs access to the OAMP systems located within the Trusted Domain. Therefore there is a OAMP-SE that resides in the Trusted-but-Vulnerable Domain that acts as a relay point for the OAMP messages.

The CPE-BE may ensure that TLS-secured connections are not run over the IPsec VPN Tunnel. The CPE-BE may ensure that media streams that are secured with SRTP media Security are not run over the IPsec VPN Tunnel.

The IPsec VPN Tunnel may use IPsec ESP [RFC 2406] in Tunnel mode [RFC2401].

The IPsec anti-replay service may be enabled at all times.

The IPsec VPN Tunnel may support Transform Identifiers ESP_3DES (with key size of 192 bits, in CBC mode) and ESP_CAMELLIA (with 128-bit key and CBC mode) [RFC4312]. The IPsec VPN Tunnel SHOULD support Transform Identifier ESP_AES (with 128-bit key and CBC mode).

The IPsec VPN Tunnel may support Authentication Algorithms HMAC-MD5-96 (key size of 128 bits), and HMAC-SHA-1-96 (key size 160 bits).

Key generation and management for the IPsec VPN Tunnel may be done with IKE [RFC2409], using IKE Authentication with Digital Signatures, or IKE authentication with a pre-shared key. If IKE authentication with Digital Signatures is used, both client and server may exchange X.509 certificates, and certificates may be verified.

7. Media Security

Media encryption is not required within the NGN infrastructure, but it may be required to be supported for customers that desire its use. Such support may include two media encryption protocols, SRTP [RFC3711]. In the rest of this section Border Elements are assumed to implement encryption/decryption although it is possible to do the same in a separate platform shared among

BEs. In either case, the encryption and decryption must be collocated with other media processing capabilities such as DTMF detection and transcoding.

With the requirement to connect subscribers desiring media encryption on their access link with those that do not (or don't support it), there are five separate cases that need to be considered.

The first and simplest case is where neither endpoint desires encryption. The media will flow from source to destination, through the border elements, without any encryption on any of the links. Neither Border Element#1 (serving the originator) nor Border Element#2 (serving the destination) does any encryption or decryption.

The second case occurs if the originator desires an encrypted media stream but the destination doesn't, Border Element#1 acts as a encryption/decryption relay point. Border Element#1 receives the encrypted stream from the originator, decrypts it and passes it through the NGN infrastructure to Border Element#2, who passes it (still unencrypted) to the destination. In the reverse direction Border Element#1 receives unencrypted media through the NGN infrastructure and encrypts it before sending it to the originator. Thus the media over leg#1 (from the originator to Border Element#1) is encrypted, leg#2 (between Border Element#1 and Border Element#2) is not, and leg#3 (between Border Element#2 and the destination) is not.

The third case occurs if the destination desires an encrypted media stream but the originator doesn't. Border Element#2 acts as an encryption/decryption relay point. Border Element#1 receives unencrypted media from the originator and passes it (still unencrypted) through the NGN infrastructure to Border Element#2. Border Element#2 encrypts it and passes it to the destination. In the reverse direction Border Element#2 receives the encrypted media stream from the destination endpoint and decrypts it before forwarding through the NGN infrastructure. Border Element#1 passes the unencrypted media to the originator. Thus the media over legs #1 and #2 are unencrypted, and the media over leg#3 is encrypted.

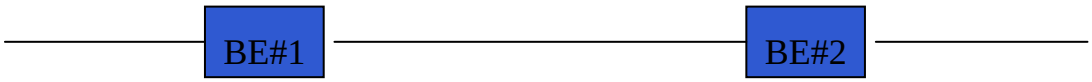
The fourth case occurs if the originator and destination both desire encrypted media, but either they do not support compatible encryption schemes or there is some enhanced service being provided by the NGN infrastructure (such as DTMF detection for calling card applications). Both Border Element#1 and Border Element#2 act as encryption/decryption relay points. Border Element#1 receives the encrypted stream from the originator, decrypts it and passes it through the NGN infrastructure to Border Element#2. Border Element#2 encrypts it and passes it to the destination. In the reverse direction, Border Element#2 receives encrypted media from the destination endpoint and decrypts it before forwarding through the NGN infrastructure. Border Element#1 receives unencrypted media and encrypts it before sending it to the originator. Thus the media over legs #1 and #3 are encrypted, and the media through the NGN infrastructure (leg #2) is not.

The fifth case occurs if the originator and destination both desire encrypted media, support compatible encryption schemes, and there is no enhanced service being provided by the NGN infrastructure. Border Element#1 receives encrypted media from the originator and passes it unchanged through the NGN infrastructure to Border Element#2, who passes it unchanged to the destination. In the reverse direction Border Element#2 receives encrypted media from the destination and passes it unchanged through the NGN infrastructure to Border Element#1 who passes it unchanged to the originator. Thus the media over all three legs is encrypted. The signaling needed to achieve this case is beyond the scope of this document.

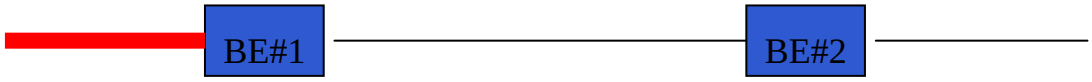
These five cases are shown in the following diagram:

Ed Note: redraw figure and add figure number

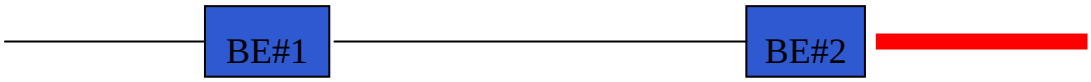
Case 1: Neither originator nor destination desires media encryption,



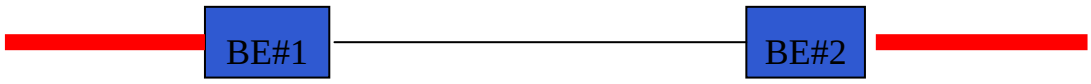
Case 2: Originator desires encryption, destination doesn't
NBE#1 encrypts/decrypts the media to/from the originator



Case 3: Destination desires encryption, originator doesn't



Case 4: Originator and destination desire encryption, but no common capabilities



Case 5: Originator and destination desire encryption, with common capabilities

Media encryption, by either of the two techniques described in this section, provide authentication, confidentiality, and message integrity.

7.1 SRTP

Secure RTP is described in [RFC3711], and is defined as a profile of RTP [RFC3550]. It is intended to be implemented between the RTP application and the transport layer in the protocol stack – intercepting an RTP packet and forwarding an equivalent SRTP packet on the transmit side, and intercepting SRTP packet and passing equivalent RTP packet up the stack on the receiving side. It basically encrypts the payload of the RTP packet and adds a message integrity check (MIC) to the end of the packet on the transmit side, and verifies the MIC and decrypts the payload on the receive side.

7.1.1 Encryption and Authentication Algorithms

A Border Element supporting SRTP may support AES in Counter Mode [RFC3711, sp800-38a]. The BE may support HMAC-SHA1 for message integrity check generation, with tag length of 80 bits.

7.1.2 Cipher Suite Negotiation and Key Generation

EdNote: Several mechanisms are described here, but all of them are optional. Further contributions are needed on proposing specific mandatory options.

EdNote: Further contributions are requested to explain the need for the selection of a specific IETF solution (like key distribution mechanism)

EdNote: 3GPP SA3 is working on the same problem, but does not yet have a solution. It is desirable that the ITU-T and 3GPP directions on that are harmonized.

EdNote: ETSI TISPAN and 3GPP SA3 ruled out option 4 for IMS-based NGN.

Key generation for SRTP can be ~~either~~ done in several ways: (1) via provisioning (via CPE Provisioning Element), ~~or~~ (2) by using key material generated by the endpoint device and included in SDP [RFC4566] in the INVITE requests, (3) key material is exchanged using separate key management protocol and piggybacked with SDP, (4) establish a key using key material without using SDP~~key material is exchanged using key management protocol along media path without using SDP~~.

For each subscriber, the BE may obtain from the SAA/TAA-FEs the SRTP Master Key, and from this derive preliminary encryption and authentication session keys. An SRTP Master Key of length 128 may be supported. The key derivation algorithm described in [RFC3711] may be supported. The preliminary encryption key length may be 128 bits, the preliminary session salt key length may be 112 bits, and the preliminary authentication key may be 160 bits. When a new SRTP Master key is issued to a subscriber, the BE may be able to use it immediately.

If the SDP contained in the INVITE request has “RTP/SAVP” as the media protocol value in the “m=” line, and no key value in a “k=” line, and no “a=crypto” attribute, then the BE may use the preliminary keys generated from the provisioning system as the actual keys for the session. The cipher suite is not negotiable in this case.

If the SDP contained in the INVITE request has “RTP/SAVP” as the media protocol value in the “m=” line, and no “a=crypto” attribute, and a key value in a “k=” line, then the BE may use the key contained in the “k=” line as the SRTP Master Key and generate the session and authentication keys from it. The cipher suite is not negotiable in this case.

If the SDP contained in the INVITE request has “RTP/SAVP” as the media protocol value in the “m=” line, and a “a=crypto” attribute, then the BE may follow the requirements of [RFC 4568] to generate the session and authentication keys. For example, the SDP entry “a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:PS1uQCVeeCFCanVmcjkpPywjNWhcYD0mXXtxaVBR|2^20|1:4” indicates the cipher-suite is AES_CM_128_HMAC_SHA1_80, and the key_param is defined by the text starting with “inline:”. Within the key_param, the first field is the master key appended with the master salt, concatenated and then base64 encoded. The list of valid cipher suites is given in section 5.2 of [RFC 4568], from which one is chosen as part of the SDP offer/answer exchange.

If the SDP contained in the INVITE request has “RTP/SAVP” as the media protocol value in the “m=” line, and a “a=key-mgmt” attribute, then BE may follow the requirements of [RFC4567] to generate keys and security parameters. For example, “a=key-mgmt:mikey AQAFgM0XflABAAAAAAAAAAAA...” indicates that the key management protocol is mikey [RFC3830], and the remainder text is the key management data which is base64 [RFC4648] encoded.

7.2 Authentication interface between NGN Network Element and Secure Token Server

NGN Network Elements may implement SASL [RFC2222] protecting their OAMP functions. The SASL layer may include an authentication check based on Secure Token, as defined in [RFC2808]. This is identified with the SASL key “Secure Token”. The user desiring OAMP access provides (1) an authorization identity (which allows system administrators to log in with a different user identity; if empty it defaults to the authentication identity), (2) an authentication identity (an identity whose passcode will be used), (3) the pin value of the user and 6-digit passcode on the Secure Token.

The NGN Network Element may implement an SAA/TAA-FE compliant client as part of the SASL handling of Secure Token. The NGN Network Element collects the presented user credentials and

then sends them to the Secure Tokenserver. Collected fields include the username, pin code, and currently-displayed Secure Token value. The Network Element receives back an Accept/Deny/Retry status message. If successful, the SASL enables the user to access the OAMP functions, based on the level of access associated with that username.

8. Audit Trail, Trapping, and Logging Systems

An audit trail is kept showing all OAMP access attempts (whether successful or not), all OAMP changes made, and all OAMP signoffs. In addition significant events, as described in previous sections, are logged.

In this section some mechanisms on important features are described. They are not exhaustive and other implementations may be adopted depending on the NGN provider policies.

8.1. Time Stamping and Time Source

8.2 Events Reported in the Security Log (preliminary)

NGN network elements shall be capable of reporting events in a Security Log. Refer to ITU SG 4 for requirements.

8.3 Network Element interface to Syslog Server

The Network Elements should send their logging information to a remote log host. Such elements that utilize the Syslog protocol [RFC3164] to achieve this function may follow the requirements of this section.

Network Elements that utilize the Syslog protocol may include a timestamp, with the time based on the value received via SNTP/NTP from a trusted time source, and may give the timestamp in Zulu Time (UTC). Elements may include their Hostname (if one has been provisioned) or their IP address in the syslog message header.

8.4 Network Element Use of SNMP

It is essential that the NGN Network Elements be able to be managed from a remote platform. SNMP is the industry standard mechanism to do this. While SNMPv3 [RFC2263, RFC2264, RFC3414, RFC3415] solves many of the security faults present in SNMPv2, it is becoming increasingly widely available.

The Network Elements should send their logging information to a remote log host. Such elements may utilize the SNMP protocol to achieve this function, noting the caveats elsewhere in this document relating to SNMP v3

SNMP is defined by an overall architecture [RFC3411], the mechanism for naming objects and events (MIBs) [RFC1155, RFC1212, RFC1215, RFC2578, RFC2579, and RFC2580], and protocol operations [RFC3416, RFC3417]. For a more detailed overview of the documents that describe the current Internet-Standard Management Framework, see section 7 of [RFC3410].

Each NGN Network Element may implement an SNMP client. If SNMP v1 or v2 are used, and if required by security policy of the NGN provider, they must use UDP over IPSec as the transport. Each instance of a message may be encoded using the Basic Encoding Rules of ASN.1 [ITU-T X.690, available at http://www.itu.int/ITU-T/publications/index.html, also can ref ISO8825] into a single UDP datagram. The client may listen on port 161 for Command Responder Applications, and may listen on port 162 for Notification Receiver Applications.

NGN Network Elements SHALL implement all necessary MIBs for reporting security events and audit trails.

Ed Note: list the MIBs

~~8.5 Network Element use of Integrity Checking Agent~~

~~Ed Note: Integrity Checking Agent requires definition~~

~~All network elements shall install an Integrity Checking Agent, which reports security events to the SATM system. An agent such may be used for this purpose. Same agent also reports results to a remote OAMP server on the OAMP network.~~

~~Ed Note: make consistent with previous changes.~~

8.~~6~~ 5 Security Patch Management

Regular installation of maintenance and security patches on NGN network elements and servers minimizes their volnurabities to attacks and unintentional failures. A comprehensive patch management strategy must be deployed including Installation and Verification processes and platforms. The Verification element may include Security Management Clients (SMC) [need to define] and a single logical Verification Manager. SMCs may be installed and run at all times on every NGN NE. SMCs may communicate patch verification data to the Verification Manager. Verification Manager generates patch installation reports including the compliance status of the servers. A standard interface between the SMCs and the Verification Manager is highly recommended. OAMP traffic security requirements apply equally to the patch management traffic.

Ed Note: verify whether may should be a may or shall.

8.6~~7~~ Tracking Changes and Rolling Back Changes

Network Elements configurations and changes must be backed up. The primary goal of system backup is to allow system recovery in the event of hardware or software troubles that result in corruption of a software load and/or the associated system data. The following types of information may be included in a system backup load:

Customer data and logic

Network traffic connectivity such as facilities and trunks

The NGN Carrier and vendor-provided application software

Operating system

Hardware configuration

An on-going record of provisioning work must be maintained so that any Network Element (NE) can be brought up to date with provisioning actions that have occurred since a backup image was taken.

The Provisioning Platform may provide the following capabilities.

A journal of provisioning activities for each of the Network Elements (NEs) that directly provisions.

At least one week's worth of Provisioning activities for each NE.

Provisioning Platform may allow users to manually review the stored provisioning activities for each NE. The activity description provided to the user SHALL summarize the size, number, and types of transactions for a given time interval.

Provisioning Platform may provide a utility to allow reprovisioning of a designated NE by reentering data into a specified NE. This utility would be user controlled and allow selection of begin and end dates/times for the data to be reprovisioned. Based on the specified begin and end dates/times, Provisioning Platform would automatically reenter all of the intervening data into the specified NE.

8.7~~8~~ Audit Trail, Trapping, and Logging at CPE-BE

All of the Audit Trail, Trapping, and Logging requirements for NGN Network Elements apply to the CPE-BE.

The CPE-BE is connected to the OAMP systems through a VPN tunnel. It therefore sends its SYSLOG messages, receives SNMP requests and sends SNMP responses through this VPN. The CPE-BE may NOT accept any OAMP requests on any other interface.

The requirements for the VPN tunnel are given in section 0.

9. CPE Provisioning

All customer premise equipments are configured by the CPE Provisioning Element. CPE Provisioning Element resides in the trusted domain and may only communicate with the CPEs via the Device Configuration & Bootstrap Border Element (DCB BE) as shown in Figure 1. A CPE may authenticate and establish a security association (SA) with the DCB BE before it can obtain configuration file from CPE Provisioning Element. DCB BE may support both TLS and Ipsec for establishing SA with the CPEs (including CPE-BE). Refer to sections 0 and 0 for more detail.

Border Element

CPE Provisioning Element includes the address of a Border Element in the configuration data downloaded to the authenticated device. CPE Provisioning Element may also include a certificate that is used to authenticate the subscriber with the border element as described in section 0.

A CPE device will request provisioning from NGN service provider. The Border Element will receive this request and authenticate the CPE with the SAA/TAA-FEs. When the device is authenticated, the Border Element will forward the provisioning request to CPE Provisioning Element. CPE Provisioning Element then downloads the configuration and/or firmware to the CPE. If the CPE cannot be authenticated, the failure is logged.

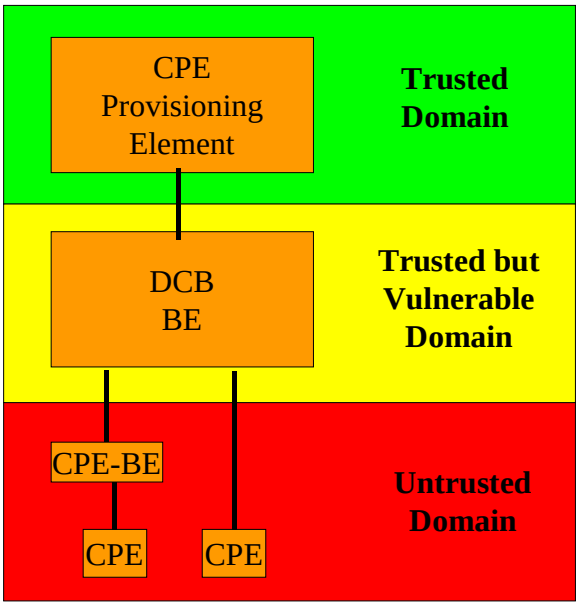


Figure 1 – Application of the security trust model to CPE provisioning

Appendix I. Examples of Source-Address Assurance and its application to the mechanism of subscriber identification and authentication

This Appendix provides concrete examples of the Source-Address Assurance mechanisms and its application to the subscriber identification and authentication through Network Source Address described in Clause 5.2.2.

I.1. Subscriber identification and authentication linked to access-line authentication

This clause provides an example of the subscriber identification and authentication, in which an IP address is assigned as the result of access-line authentication. In this example, each subscriber is statically associated with his/her access line. Hence, the mechanism described in this example is applicable only to non-nomadic (i.e. fixed) services.

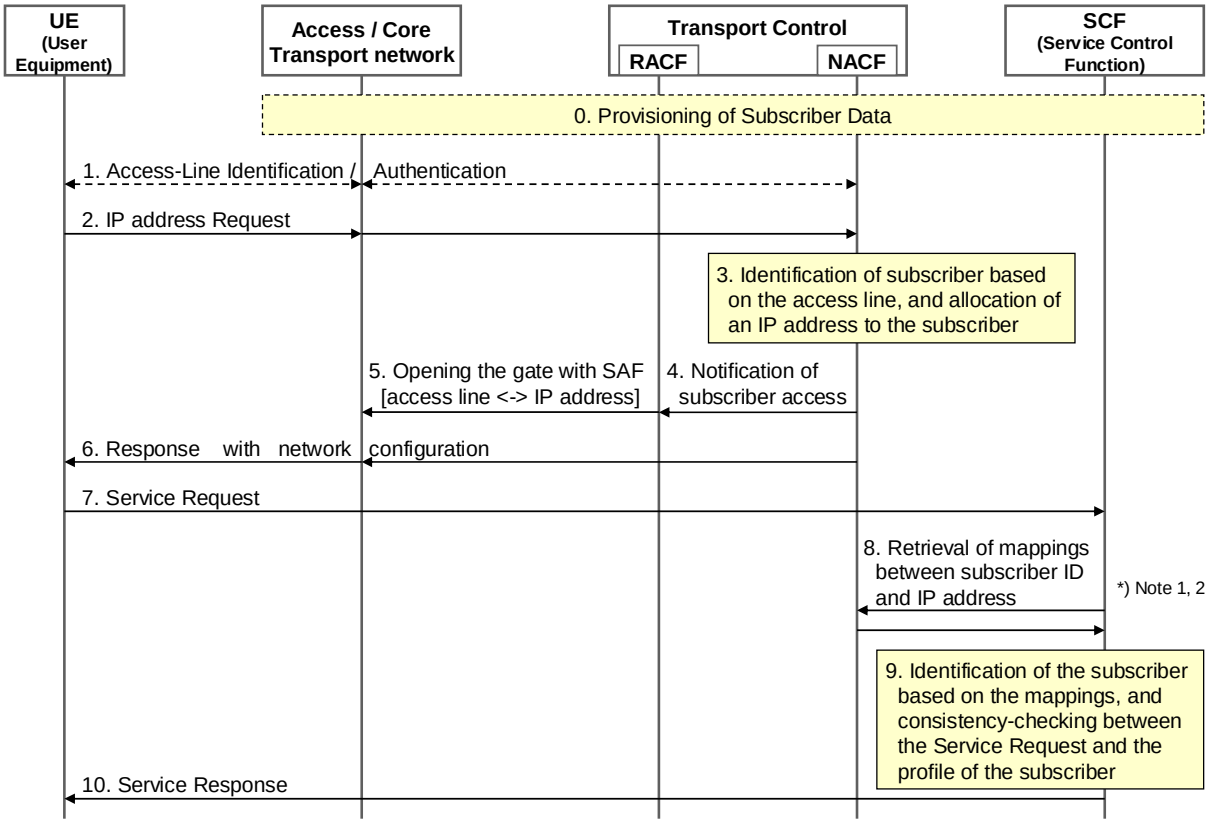


Figure I.1: High-level message flows of example 1

NOTE 1 – The mapping information between the IP address and the Subscriber ID may be provided from the NACF to the SCF at the time of address allocation by the NACF.

NOTE 2 – The NACF may provide the mapping between the IP address and the location information (e.g. Line Identifier) instead of the mapping between the IP address and the subscriber ID. In that case, the SCF is required to maintain the mappings between subscriber IDs and locations and derive the subscriber ID from the location information sent from the NACF.

Descriptions

0. The subscriber profiles are preconfigured to the corresponding FEs (e.g. TUP-FE, SUP-FE) in the NACF or the SCF.

   The most important setting issues in this scenario are: 1) The NACF (typically TUP-FE) maintains the mappings between subscriber IDs (Subscriber Account Identifiers) and logical/physical access-line IDs (e.g. VLAN ID or access port), and 2) The SCF (typically SUP-FE) maintains the mappings between subscriber IDs and the attributes or profiles of the corresponding subscribers (e.g. values of “From” header or “P-Asserted-ID” header in case of SIP-based services). In cases where the name space of subscriber IDs in the SCF is different from that in the NACF, the SCF should also maintain the mappings between these IDs.

   Alternatively, the NACF does not have to maintain the mappings between subscriber IDs and access-line IDs. In such scenarios, the SCF should maintain the mappings between subscriber IDs and access-line IDs, so the SCF can retrieve a corresponding subscriber ID from an access-line ID.

   On the gateways in the Access/Core Transport, all gates for subscriber's access lines are initially configured to be closed so that any incoming IP packets, except for the packets necessary for UE to attach the network (e.g. sending address requests or authentication requests), are dropped.

1. A UE attaches to the Access Network through its access line to get IP connectivity to the NGN. This example assumes that access authentication by the NACF is implicit and is executed at step 3. However, the NACF may alternatively employ an explicit access authentication method (e.g. IEEE 802.1X). In that case, network access authentication is executed in this phase, i.e. before IP address assignment.

2. The UE requests allocation of an IP address. This is typically performed by sending DHCP Discover and Request, and these messages are relayed to the NACF by the gateways.

3. In this example, Access Network authenticates the access line and provides the authenticated access-line ID (e.g. VLAN ID or access port) to the NACF. Hence, the NACF can identify the subscriber ID of the UE based on the access-line ID, which the IP address request is sent through. Then, the NACF allocates an IP address to the requesting UE and stores the mapping between the subscriber ID and the allocated IP address.

   This mapping information may be pushed from the NACF to the SCF and be stored (cached) in the SCF. In that case, the 8th step below can be skipped.

4. The NACF notifies the RACF that the subscriber has been connected. This notification includes the subscriber ID, the access-line ID (physical/logical), the allocated IP address, and QoS profiles.

5. The RACF makes a policy decision on network resource allocation to the subscriber and orders the gateways to open the gate for the access line with packet-filtering rules, which are defined to accept and forward incoming IP packets whose source address is the IP address assigned to the subscriber, and to drop other incoming packets.

   The enforcement of source IP address filtering coordinated with access-line authentication by the NACF, which is described above, ensures that an IP address can be used only by the subscriber to whom the address is assigned.

6. The NACF returns the allocated IP address to the UE with other network configuration parameters (e.g. the addresses of DNS servers and P-CSCF). This is typically done by sending DHCP Offer and Response messages.

7. After getting IP-connectivity, the UE sends a Service Request (e.g., REGISTER signal in case of SIP-based services) to the SCF. The Service Request is passed by the gateways (firewalls with source-address filtering) to the SCF only if the source address of the Request is one assigned by the NACF.

8. The SCF retrieves the mapping information (i.e., the subscriber ID and its assigned IP address) corresponding to the source address of the Service Request, from the NACF.

9. The SCF considers the Service Request to be originated from the subscriber who assigned the subscriber ID contained in the retrieved mapping information. In cases where the name space of subscriber IDs in the SCF is different from that in the NACF, the retrieved subscriber ID must be translated into the subscriber ID in the name space used by the SCF based on the mappings between these IDs.

   The SCF extracts the value of attributes regarding the subscriber's identity (e.g. the value of the "From" header in case of SIP-based services) from the Service Request and checks consistency between those values and the corresponding subscriber's profile.

10. If the authentication and the authorization succeed, the SCF returns the normal reply to offer the requested service (e.g. "200 OK" in case of SIP-based services).

I.2. Subscriber identification and authentication linked to explicit access authentication at IP Connectivity Establishment

This clause provides an example of the subscriber identification and authentication, in which an IP address is assigned as the result of explicit access authentication at the establishment of IP connectivity. In this example, each subscriber is dynamically associated to a L2 session, which is established at the time of access authentication. Hence, the mechanism described in this example is applicable to both nomadic and non-nomadic services.

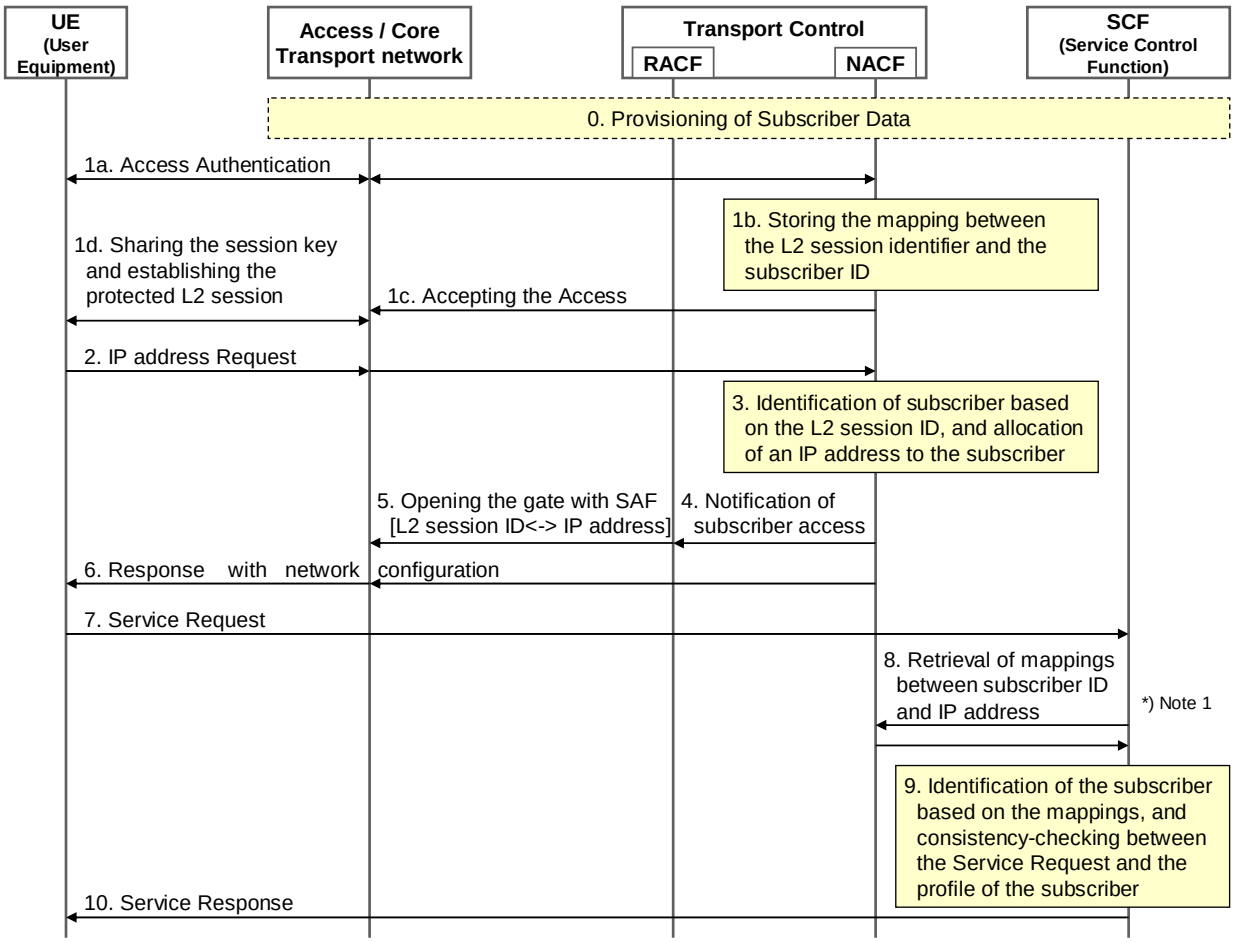


Figure I.2: High-level message flows of example 2

NOTE 1 – The mapping information between the IP address and the Subscriber ID may be provided from the NACF to the SCF at the time of address allocation by the NACF.

Descriptions

0. The subscriber profiles are preconfigured to the corresponding FEs (e.g. TUP-FE, SUP-FE) in the NACF or the SCF. In contrast with the previous example, the NACF does not need to maintain the mappings between subscriber IDs and access-line IDs.

   On the gateways in the Access/Core Transport, all gates for L2 access sessions with UEs are initially configured to be closed so that any incoming IP packets, except for the packets necessary for UE to attach the network (e.g. sending address requests or authentication requests), are dropped.

1a. When a UE requests connectivity to the NGN, the Access Network dynamically creates an L2 session with the UE, and an Access Authentication procedure is performed between the UE and the NACF based on the subscriber's credential (typically with an explicit authentication method such as IEEE 802.1X and RADIUS/Diameter). The signalling messages for authentication are forwarded by the gateways.

1b. During the authentication procedure, the identifier of the L2 session (e.g. VLAN-ID, L2 address of the UE, etc) assigned to the UE is sent to the NACF. When the authentication succeeds, the NACF stores this L2 session identifier with the authenticated subscriber ID.

1c. The NACF notifies the Access Network that the UE has been successfully authenticated and the access to the network has been authorized (e.g. an ACCESS ACCEPT message in case of RADIUS protocol).

1d. Upon receiving the notification of successful authentication of the subscriber from the NACF, the Access Network establishes a security association (SA) with the UE to protect the integrity and confidentiality of the L2 session. Typically, these are achieved by the session-keys derivation mechanisms defined in IEEE 802.1X and the protection procedure defined for each L2 technology (e.g. TKIP/CCMP defined in IEEE 802.11i for 802.11 Wireless-LAN.)

The security mechanisms described above protect the L2 session from being used by other subscribers and provides necessary grounds for the prevention of IP-address spoofing.

2. The UE requests allocation of an IP address. This is typically performed by sending DHCP Discover and Request, and these messages are relayed to the NACF by the gateways.

3. The NACF identifies the subscriber ID of the UE based on the identifier of the L2 session, which the request is sent through. Then, the NACF allocates an IP address to the requesting UE and stores the mapping between the subscriber ID and the allocated IP address.

This mapping information may be pushed from the NACF to the SCF and stored (cached) in the SCF. In that case, the 8th step below can be skipped.

4. The NACF notifies the RACF that the subscriber has been connected. This notification includes the subscriber ID, the L2 session ID (physical/logical), the allocated IP address, and QoS profiles.

5. The RACF makes a policy decision on network resource allocation to the subscriber and orders the gateways to open the gate for the L2 session with packet-filtering rules, which are defined to accept and forward incoming IP packets whose source address is the IP address assigned to the subscriber, and to drop other incoming packets.

The enforcement of source IP address filtering coordinated with access authentication the by NACF, which is described above, ensures that an IP address can be used only by the subscriber to whom the address is assigned.

Steps 6 - 10 are identical to those explained in the previous example described in Clause I.1.

Appendix II: Living List 1 on GBA

[EdNote: This is a living list to catch the contents of C-137 for the further discussion on GBA section, or section].

The Generic Bootstrapping Architecture (GBA) specifies access-independent bootstrapping procedure. The architecture has been standardized by 3GPP [1] and 3GPP2 [2]. Both solutions are very similar and are based on common main principles. Both specifications use the common terminology that is also used in this document.

The GBA is an authentication system that includes three parties:

- An end-user who is trying to obtain network services using User Equipment (UE)
- Application server (called Network Application Function or NAF)
- A trusted entity (called Bootstrapping Server Function or BSF), which is involved in authentication and key exchange between two other entities.

The GBA enables mutual authentication of the End-User, who is using UE, and an application server (NAF) without revealing the End-User's long-term credentials and secrets to the NAF by using a trusted entity BSF. Indeed, in the GBA framework the NAF does not need to know what kind of credentials (e.g. smart card) were used in the exchange between the UE and the BSF.

The basics of the GBA authentication process are illustrated and described below.

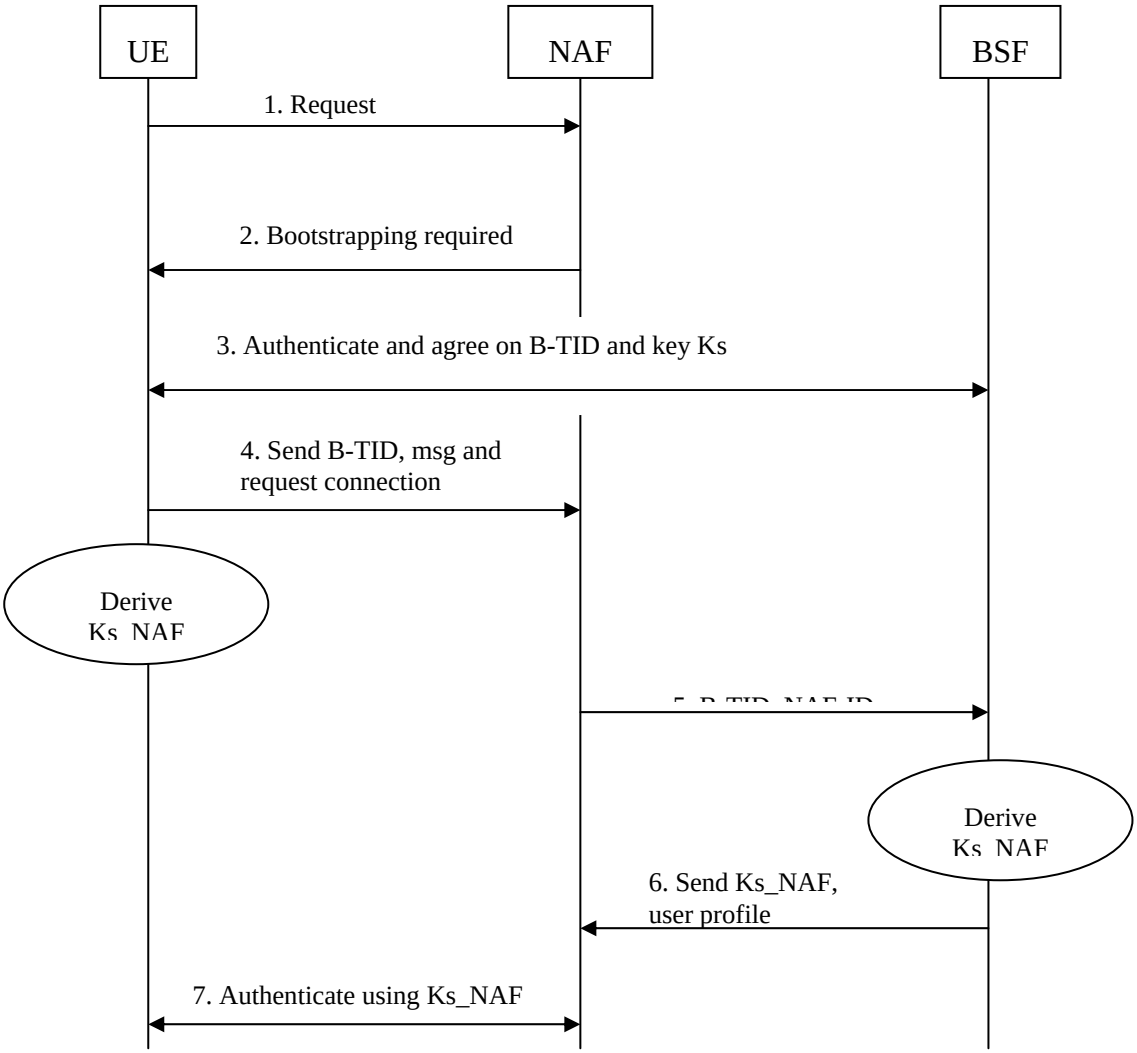


Figure 1 – The GBA authentication process

1. User initiates communication with NAF by sending request from UE without any GBA-related parameters.
2. The NAF replies with a bootstrapping initiation message.
3. The UE contacts the Bootstrapping Server Function (BSF) and both parties proceed with authenticating each other using long-term credentials and the End-User's profile. This authentication process involves communication between BSF and the back-end authentication databases, such as Service Authentication and Authorization Functional Entity (SAA-FE) in NGN. In GBA specified by 3GPP the Home Subscriber System (HSS) provides all authentication information to BSF, which finds the HSS that contains the required data by querying Subscriber Location Function (SLF). In the 3GPP2 architecture three entities are involved in providing this information to BSF - Home Location Register (HLR), HSS, and Authentication, Authorization, and Accounting server (AAA). As a result of a successful authentication procedure, UE and BSF share a secret key (Ks) and a Bootstrapping Transaction Identifier (B-TID). Such procedures are specified by the 3GPP and 3GPP2 GBA architectures. (The NGN shall support all mechanisms for mutual authentication and key distribution between UE and BSF that are specified in both architectures).
4. The UE contacts NAF sending the application request that includes B-TID and msg (msg denotes here the application-specific data).
5. The NAF sends B-TID to the BSF along with its own ID (NAF-ID).
6. In the authentication answer BSF sends to NAF a key Ks_NAF, which it had derived from Ks, and the application-specific user security settings.
7. Finally, UE and NAF authenticate each other using Ks_NAF, which both entities had derived from Ks.

References

[1] ETSI TS 133 220, 03/2006, Generic Authentication Architecture (GAA); Generic bootstrapping architecture.

[2] TIA PN 3-0238, 12/2005, Generic Bootstrapping Architecture (GBA) Framework.

_____