3GPP TSG SA WG3 Security — S3#32 **S3-040010**
9 – 13 February 2004, Edinburgh, UK

3GPP TS 33.222 V0.2.0 (2004-01)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Services and System Aspects Generic Authentication Architecture (GAA); Access to Network Application Functions using HTTPS (Release 6)

Keywords

<keyword[, keyword]>

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

Introduction

This clause is optional. If it exists, it is always the second unnumbered clause.

1 Scope

This clause shall start on a new page.

The present document …

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[<seq>] <doctype> <#>[([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

[1] 3GPP TR 41.001: "GSM Release specifications".

[2] 3GPP TR 21 912 (V3.1.0): "Example 2, using fixed text".

3 Definitions, symbols and abbreviations

Delete from the above heading those words which are not applicable.

Subclause numbering depends on applicability and should be renumbered accordingly.

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

Definition format

***<defined term>**: <definition>.*

example: text used to clarify abstract rules by applying them literally.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format

<symbol> <Explanation>

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviation format

<ACRONYM> <Explanation>

4 Authentication Schemes

4.1 Requirements and principles

This document is based on the architecture specified in [TS33.220]. All notions not explained here can be found in [TS33.220].

Editor's note: care must be taken that this specification is in line with TS 33.141 on presence security. SA3 has yet to decide the split between the two documents.

4.2 Shared key-based UE authentication with certificate-based NAF authentication

This section explains how the procedures specified in [TS33.220] have to be enhanced when HTTPS is used between a UE and a NAF. The only enhancement required is the need to specify how the set up of a TLS tunnel is included in the general procedures specified in [TS33.220].

Editor's note: The sequence of events needs to be updated to reflect the initiation of bootstrapping as described in TS 33.220, section 4.3.1.

When the UE accesses a NAF, with which it does not yet share a key, then the sequence of events is as follows:

1. the UE runs http digest aka [rfc3310] with the BSF over the Ub interface.

2. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

After the completion of step 1), the UE and the BSF share a secret key. This shared key is identified by a transaction identifier supplied by the BSF to the UE over the Ub interface key, cf. [TS 33.220, section 4.3.1].

3. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

Editor's note: TLS needs to be profiled in an appropriate section of this specification.

4. The UE sends an http request to the NAF.

5. The NAF invokes http digest [rfc 2617] with the UE over the Ua interface in order to perform client authentication using the shared key agreed in step 1), as specified in [TS 33.220, Annex A].

6. While executing step 5), the NAF fetches the shared key from the BSF over the Zn interface, as specified in [TS 33.220, Annex A and section 4.3.2].

6)After the completion of step 4), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

When the UE accesses a NAF, with which it already shares a key, steps 1), 5) and 6) may be omitted, as specified in [TS 33.220].

Editor's note: the above procedure is generally applicable and conforms to [TS 33.220]. For the case of a co-located BSF and NAF an optimisation is possible which is currently located in the informative Annex Z. SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.

4.3 Shared key-based mutual authentication between UE and NAF

4.4 Certificate based mutual authentication between UE and NAF

5 Use of authentication proxy

5.1 Requirements and principles

The authentication proxy may reside between the UE and the NAF as depicted in Figure y [tba to section 5.2]. The usefulness of an Authentication Proxy may be to reduce the consumption of authentication vectors and/or to minimize SQN synchronization failures.

The following requirements apply for the use of an Authentication Proxy:

- Authentication proxy shall be able to authenticate the UE using the means of Generic Bootstrapping Architecture, as specified in [Ts33.220].
- Authentication proxy shall send the authenticated identity of the UE to the application server belonging to the trust domain at the beginning of new HTTP session.
- Authentication proxy may not reveal the authenticated identity of the UE to the application server not belonging to the trust domain if required.
- The authenticated identity management mechanism shall not prevent the application server to use an appropriate session management mechanisms with the client.
- The UE shall be able to create multiple parallel HTTP sessions via the authentication proxy towards different application servers.

NOTE1: The used session management mechanism is out of the scope of 3GPP specifications.

- Implementation of check of asserted user identity in the AS is optional.
- Activation of transfer of asserted user identity shall be configurable in the AP on a per AS base.

The use of an authentication proxy should be such that there is no need to manage the authentication proxy configuration in the UE.

NOTE2: This requirement implies that the authentication proxy should be a reverse proxy in the following sense: A reverse proxy is a web server system that is capable of serving web pages sourced from other web servers - in addition to web pages on disk or generated dynamically by CGI - making these pages look like they originated at the reverse proxy.

[Editors' note: The above requirements may be revisited after the following issues are fully studied:

- feasibility of shared-key TLS;
- terminal configurability]

5.2 Authentication proxy architecture

<include figure y here>

The use of an authentication proxy (AP) is fully compatible with the architecture specified in [TS33.220] and in section 4 of this specification. When an AP is used in this architecture, the AP takes the role of a NAF. When an https request is destined towards an application server behind an authentication proxy (AP), the AP terminates the TLS tunnel and performs UE authentication. The AP proxies the http request to the application server.

Annex A contains further guidance on technical solutions for authentication proxies.

5.3 Interfaces

5.4 Management of UE identity

Annex A (informative): Technical Solutions for Access to Application Servers via Authentication Proxy and HTTPS

This annex gives some guidance on the technical solution for authentication proxies so as to help avoid misconfigurations. An authentication proxy acts as reverse proxy which serves web pages (and other content) sourced from other web servers (AS) making these pages look like they originated at the proxy.

To access different hosts with different DNS names on one server (in this case the proxy) the concept of virtual hosts was created.

One solution when running HTTPS is to associate each host name with a different IP address (IP based virtual hosts). This can be achieved by the machine having several physical network connections, or by use of virtual interfaces which are supported by most modern operating systems (frequently called "ip aliases"). This solution uses up one IP address per AS and it does not allow the notion of “one TLS tunnel from UE to AP-NAF” for all applications behind a NAF together.

If it is desired to use one IP address only or if “one TLS tunnel for all” is required, only the concept of name-based virtual hosts is applicable. Together with HTTPS, however, this creates problems, necessitating workarounds which may deviate from standard behaviour of proxies and/or browsers. Workarounds, which affect the UE and are not generally supported by browsers, may cause interoperability problems. Other workarounds may impose restrictions on the attached application servers.

Editors' note: The text in this informative annex may need to be revisited if changes in the main body of the text are made.

Annex B (informative): Optimised Sequence of Events for Access to co-located BSF and NAF via HTTPS

Editor’s note: SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.

Editor’s note: the material in this annex is based on the information flow in S3-030371, Annex A.

Editor’s note: The impact on implementation when co-locating BSF and NAF is for further study.

Editor’s note: The sequence of events needs to be updated to reflect the initiation of bootstrapping as described in TS 33.220, section 4.3.1.

When the UE accesses a NAF, and the NAF is co-located with the BSF, then the optimised sequence of events is as follows:

1. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

Editor's note: TLS needs to be profiled in an appropriate section of this specification.

2. If the UE does not share a key with the NAF, the UE sends an http request to a NAF, containing the UE's identity.

3. If the NAF receives an http request from the UE without an Authorization header, or with an Authorization header it does not accept, the NAF contacts the (co-located) BSF to obtain a challenge and a password, computed from an AKA authentication vector according to [draft-torvinen-http-digest-aka-v2].

4. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

5. The NAF replies to the UE by sending a 401 "unauthorized" message with a WWW-Authenticate header according to [draft-torvinen-http-digest-aka-v2].

6. The UE sends an http request to the NAF with an Authorization header according to [draft-torvinen-http-digest-aka-v2].

7. The NAF verifies the Authorization header.

After the completion of step 7), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

8. The NAF replies to the http request returning the requested information to the UE, if any.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

Editor's note: the transport of of key derivation information from NAF/BSF to UE needs further study.

Note on co-location of BSF and NAF: a BSF and a NAF may be combined on one machine in such a way that the BSF is accessed through http, not using TLS, and the NAF is accessed through https. From a functional point of view, this case is identical to the general case described in section 4.2. It is even possible to functionally duplicate the BSF on one machine in such a way that the BSF is accessed through http, when TLS is not required, and accessed through https, when access to the NAF requires TLS.

Editor's note on carrying identities: the first http request after TLS set-up needs to contain the identity of the UE. The reason is that for http digest the server can issue a challenge without knowing the client's identity, whereas for http digest aka the challenge is specific to a particular client. There seem to be at least two solutions for this:

a) use a specially formed http GET request, as described for the Ub interface in [TS33.220].

b) use an Authorization header with dummy values (to be defined). The server will not accept the credentials, and will reply with a 401 "unauthorised". For maximum harmonisation, the UE identity, which needs to be included by the UE at the start of the http digest aka protocol run, should be carried in the same way in the general and the optimised case.

Note on tunnelled authentication and the use of http digest aka:

In this annex and in section 4.2 respectively, different versions of http digest aka are used. This prevents man-in-the-middle attacks with tunnelled authentication. Version 1 of http digest aka [rfc 3310] is used between the UE and the BSF when http digest aka is NOT used to authenticate the client endpoint of a TLS tunnel extending between UE and BSF. Version 1 may be run inside or outside a TLS tunnel, as long as it is not used for client authentication. Version 2 [draft-torvinen-http-digest-aka-v2] is used when http digest aka IS used to authenticate the client endpoint of a TLS tunnel. Version 2 is always run inside a TLS tunnel.

[Editor'Note on tunnelled authentication and the use of http digest aka:

Instead of using different versions of http digest aka to distinguish whether http digest aka is used for client authentication of a TLS tunnel or not, this distinction could be provided by different means. Possibilities suggested on the SA3 mailing list include to extend the specification of http digest akav2 to include a "situation" (or "context") parameter in the computation of the password, then always use http digest akav2, but with different values for the "situation" parameter for the two different uses.]

Note on transaction identifiers: the general approach, as specified in section 4, which is based on [TS 33.220], requires the use of a transaction identifier over the interfaces Ua, Ub and Zn. The use of such a transaction identifier is neither possible nor necessary in the optimised case described in this annex

Annex C (informative):
Change history

It is usual to include an annex (usually the final annex of the document) for specifications under TSG change control which details the change history of the specification using a table as follows:

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2003-10	SA3 #30	S3-030646			First Draft TS: Generic Authentication Architecture; Access to Network Application Function using HTTPS (Release 6), table of contents added		0.1.0
2003-10					Updated based on editorial comments on the SA3 e-mail list	0.1.0	0.1.1
2004-01	SA3 #31	S3-030744 S3-030745 S3-030746 S3-030749			Updated based on agreements at SA3 #31	0.1.1	0.2.0

3GPP TSG SA WG3 Security — S3#32 **S3-040010_att**
9 – 13 February 2004, Edinburgh, UK

3GPP TS 33.222~~ab.cde~~ V0.2.0~~1.1~~ (2004~~3~~-

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects
Generic Authentication Architecture (GAA);
Access to Network Application Functions using HTTPS
(Release 6)

Remove GSM logo from the cover page for pure 3rd Generation documents.

Select keywords from list provided in specs database.

Keywords

<keyword[, keyword]>

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

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

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

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

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

Introduction

This clause is optional. If it exists, it is always the second unnumbered clause.

1 Scope

This clause shall start on a new page.

The present document …

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[<seq>] <doctype> <#>[([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

[1] 3GPP TR 41.001: "GSM Release specifications".

[2] 3GPP TR 21 912 (V3.1.0): "Example 2, using fixed text".

3 Definitions, symbols and abbreviations

Delete from the above heading those words which are not applicable.

Subclause numbering depends on applicability and should be renumbered accordingly.

3.1 Definitions

For the purposes of the present document, the [following] terms and definitions [given in ... and the following] apply.

Definition format

<defined term>*: <definition>.*

example: text used to clarify abstract rules by applying them literally.

3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format

<symbol> <Explanation>

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviation format

<ACRONYM> <Explanation>

4 Authentication Schemes

4.1 Requirements and principles

This document is based on the architecture specified in [TS33.220]. All notions not explained here can be found in [TS33.220].

Editor's note: care must be taken that this specification is in line with TS 33.141 on presence security. SA3 has yet to decide the split between the two documents.

4.2 4.2 Shared key-based UE authentication with certificate-based NAF authentication

This section explains how the procedures specified in [TS33.220] have to be enhanced when HTTPS is used between a UE and a NAF. The only enhancement required is the need to specify how the set up of a TLS tunnel is included in the general procedures specified in [TS33.220].

Editor's note: The sequence of events needs to be updated to reflect the initiation of bootstrapping as described in TS 33.220, section 4.3.1.

When the UE accesses a NAF, with which it does not yet share a key, then the sequence of events is as follows:

1. the UE runs http digest aka [rfc3310] with the BSF over the Ub interface.

2. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

After the completion of step 1), the UE and the BSF share a secret key. This shared key is identified by a transaction identifier supplied by the BSF to the UE over the Ub interface key, cf. [TS 33.220, section 4.3.1].

3. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

Editor's note: TLS needs to be profiled in an appropriate section of this specification.

4. The UE sends an http request to the NAF.

5. The NAF invokes http digest [rfc 2617] with the UE over the Ua interface in order to perform client authentication using the shared key agreed in step 1), as specified in [TS 33.220, Annex A].

6. While executing step 5), the NAF fetches the shared key from the BSF over the Zn interface, as specified in [TS 33.220, Annex A and section 4.3.2].

6)After the completion of step 4), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

When the UE accesses a NAF, with which it already shares a key, steps 1), 5) and 6) may be omitted, as specified in [TS 33.220].

Editor's note: the above procedure is generally applicable and conforms to [TS 33.220]. For the case of a co-located BSF and NAF an optimisation is possible which is currently located in the informative Annex Z. SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.

4.3 Shared key-based mutual authentication between UE and NAF

4.4 Certificate based mutual authentication between UE and NAF

5 Use of authentication proxy

5.1 5.1 Requirements and principles

The authentication proxy may reside between the UE and the NAF as depicted in Figure y [tba to section 5.2]. The usefulness of an Authentication Proxy may be to reduce the consumption of authentication vectors and/or to minimize SQN synchronization failures.

The following requirements apply for the use of an Authentication Proxy:

- Authentication proxy shall be able to authenticate the UE using the means of Generic Bootstrapping Architecture, as specified in [Ts33.220].
- Authentication proxy shall send the authenticated identity of the UE to the application server belonging to the trust domain at the beginning of new HTTP session.
- Authentication proxy may not reveal the authenticated identity of the UE to the application server not belonging to the trust domain if required.
- The authenticated identity management mechanism shall not prevent the application server to use an appropriate session management mechanisms with the client.
- The UE shall be able to create multiple parallel HTTP sessions via the authentication proxy towards different application servers.

NOTE1: The used session management mechanism is out of the scope of 3GPP specifications.

- Implementation of check of asserted user identity in the AS is optional.
- Activation of transfer of asserted user identity shall be configurable in the AP on a per AS base.

The use of an authentication proxy should be such that there is no need to manage the authentication proxy configuration in the UE.

NOTE2: This requirement implies that the authentication proxy should be a reverse proxy in the following sense: A reverse proxy is a web server system that is capable of serving web pages sourced from other web servers - in addition to web pages on disk or generated dynamically by CGI - making these pages look like they originated at the reverse proxy.

[Editors' note: The above requirements may be revisited after the following issues are fully studied:

- feasibility of shared-key TLS;
- terminal configurability]

5.2 5.2 Authentication proxy architecture

<include figure y here>

The use of an authentication proxy (AP) is fully compatible with the architecture specified in [TS33.220] and in section 4 of this specification. When an AP is used in this architecture, the AP takes the role of a NAF. When an https request is destined towards an application server behind an authentication proxy (AP), the AP terminates the TLS tunnel and performs UE authentication. The AP proxies the http request to the application server.

Annex A contains further guidance on technical solutions for authentication proxies.

5.3 Interfaces

5.4 Management of UE identity

Annex A (informative): Technical Solutions for Access to Application Servers via Authentication Proxy and HTTPS

This annex gives some guidance on the technical solution for authentication proxies so as to help avoid misconfigurations. An authentication proxy acts as reverse proxy which serves web pages (and other content) sourced from other web servers (AS) making these pages look like they originated at the proxy.

To access different hosts with different DNS names on one server (in this case the proxy) the concept of virtual hosts was created.

One solution when running HTTPS is to associate each host name with a different IP address (IP based virtual hosts). This can be achieved by the machine having several physical network connections, or by use of virtual interfaces which are supported by most modern operating systems (frequently called "ip aliases"). This solution uses up one IP address per AS and it does not allow the notion of “one TLS tunnel from UE to AP-NAF” for all applications behind a NAF together.

If it is desired to use one IP address only or if “one TLS tunnel for all” is required, only the concept of name-based virtual hosts is applicable. Together with HTTPS, however, this creates problems, necessitating workarounds which may deviate from standard behaviour of proxies and/or browsers. Workarounds, which affect the UE and are not generally supported by browsers, may cause interoperability problems. Other workarounds may impose restrictions on the attached application servers.

Editors' note: The text in this informative annex may need to be revisited if changes in the main body of the text are made.

Annex B (informative): Optimised Sequence of Events for Access to co-located BSF and NAF via HTTPS

Editor’s note: SA3 still needs to decide whether the material in the annex should be moved to the main body, or remain in an informative or normative annex, or be deleted.

Editor’s note: the material in this annex is based on the information flow in S3-030371, Annex A.

Editor’s note: The impact on implementation when co-locating BSF and NAF is for further study.

Editor’s note: The sequence of events needs to be updated to reflect the initiation of bootstrapping as described in TS 33.220, section 4.3.1.

When the UE accesses a NAF, and the NAF is co-located with the BSF, then the optimised sequence of events is as follows:

1. The UE establishes a TLS tunnel with the NAF. The NAF is authenticated to the UE by means of a public key certificate.

Editor's note: TLS needs to be profiled in an appropriate section of this specification.

2. If the UE does not share a key with the NAF, the UE sends an http request to a NAF, containing the UE's identity.

3. If the NAF receives an http request from the UE without an Authorization header, or with an Authorization header it does not accept, the NAF contacts the (co-located) BSF to obtain a challenge and a password, computed from an AKA authentication vector according to [draft-torvinen-http-digest-aka-v2].

4. If the BSF has no authentication vectors for the UE it fetches authentication vectors from the HSS over the Zh interface.

5. The NAF replies to the UE by sending a 401 "unauthorized" message with a WWW-Authenticate header according to [draft-torvinen-http-digest-aka-v2].

6. The UE sends an http request to the NAF with an Authorization header according to [draft-torvinen-http-digest-aka-v2].

7. The NAF verifies the Authorization header.

After the completion of step 7), UE and NAF are mutually authenticated as the TLS tunnel endpoints.

8. The NAF replies to the http request returning the requested information to the UE, if any.

The UE may now run an appropriate application protocol with the NAF through the authenticated tunnel.

Editor's note: the transport of of key derivation information from NAF/BSF to UE needs further study.

Note on co-location of BSF and NAF: a BSF and a NAF may be combined on one machine in such a way that the BSF is accessed through http, not using TLS, and the NAF is accessed through https. From a functional point of view, this case is identical to the general case described in section 4.2. It is even possible to functionally duplicate the BSF on one machine in such a way that the BSF is accessed through http, when TLS is not required, and accessed through https, when access to the NAF requires TLS.

Editor's note on carrying identities: the first http request after TLS set-up needs to contain the identity of the UE. The reason is that for http digest the server can issue a challenge without knowing the client's identity, whereas for http digest aka the challenge is specific to a particular client. There seem to be at least two solutions for this:

a) use a specially formed http GET request, as described for the Ub interface in [TS33.220].

b) use an Authorization header with dummy values (to be defined). The server will not accept the credentials, and will reply with a 401 "unauthorised". For maximum harmonisation, the UE identity, which needs to be included by the UE at the start of the http digest aka protocol run, should be carried in the same way in the general and the optimised case.

Note on tunnelled authentication and the use of http digest aka:

In this annex and in section 4.2 respectively, different versions of http digest aka are used. This prevents man-in-the-middle attacks with tunnelled authentication. Version 1 of http digest aka [rfc 3310] is used between the UE and the BSF when http digest aka is NOT used to authenticate the client endpoint of a TLS tunnel extending between UE and BSF. Version 1 may be run inside or outside a TLS tunnel, as long as it is not used for client authentication. Version 2 [draft-torvinen-http-digest-aka-v2] is used when http digest aka IS used to authenticate the client endpoint of a TLS tunnel. Version 2 is always run inside a TLS tunnel.

[Editor'Note on tunnelled authentication and the use of http digest aka:

Instead of using different versions of http digest aka to distinguish whether http digest aka is used for client authentication of a TLS tunnel or not, this distinction could be provided by different means. Possibilities suggested on the SA3 mailing list include to extend the specification of http digest akav2 to include a "situation" (or "context") parameter in the computation of the password, then always use http digest akav2, but with different values for the "situation" parameter for the two different uses.]

Note on transaction identifiers: the general approach, as specified in section 4, which is based on [TS 33.220], requires the use of a transaction identifier over the interfaces Ua, Ub and Zn. The use of such a transaction identifier is neither possible nor necessary in the optimised case described in this annex

Annex <X> C (informative): Change history

It is usual to include an annex (usually the final annex of the document) for specifications under TSG change control which details the change history of the specification using a table as follows:

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2003-10	SA3 #30	S3-030646			First Draft TS: Generic Authentication Architecture; Access to Network Application Function using HTTPS (Release 6), table of contents added		0.1.0
2003-10					Updated based on editorial comments on the SA3 e-mail list	0.1.0	0.1.1
2004-01	SA3 #31	S3-030744 S3-030745 S3-030746 S3-030749			Updated based on agreements at SA3 #31	0.1.1	0.2.0