**3GPP TSG SA WG3 Security — SA3#33** **S3-040337**
**10-14 May 2004**
**Beijing, China**

**Source:** **Nokia**

**Title:** **Shared key TLS usage within Ua interface**

**Document for:** **Discussion and decision**

**Agenda Item:** **GAA**

# 1 Introduction

Shared key TLS has been discussed in several contributions in SA3 (S3-030555, S3-030721, S3-030732). These contributions discussed a version of shared key TLS internet draft [1] authored by Peter Gutmann. Since then, Nokia and Siemens have authored a new shared key TLS internet-draft called "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)" [2], published February 6, 2004. The difference between these two drafts is that Gutmann's version re-uses (or overloads) the TLS session cache and session resumption functionality, and the Nokia/Siemens version defines new ciphersuites and uses those. The main advantage of the new ciphersuite approach is that it better fits GAA usage scenarios, and solves the usage problems that were present with Gutmann's shared key TLS approach.

This contribution discusses the Nokia/Siemens shared key TLS internet draft and shows how it is usable within GAA. Throughout this document the document Gutmann's version of shared key TLS is referred as **"shared-key TLS",** and the version by Nokia and Siemens **"psk TLS"** ("psk" stands for Pre-Shared Key).

# 2 Psk TLS usage in GAA Ua interface

## 2.1 Successful TLS session establishment

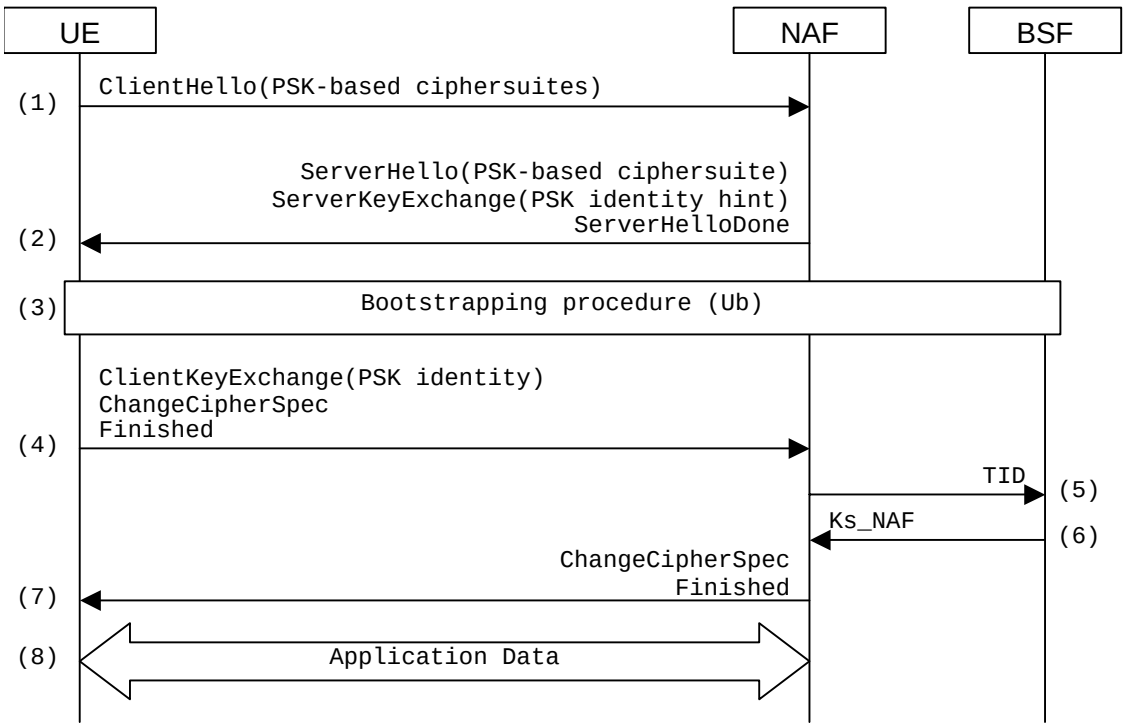


**Figure 1. psk TLS session establishment in GAA.**

1. UE sends ClientHello message to NAF. In order to indicate that UE is capable of PSK-based authentication it includes the PSK-based ciphersuites to the list of acceptable ciphersuites list.

2. If NAF wants to use PSK-based authentication, it selects one of the acceptable PSK-based ciphersuites, places the selected ciphersuite in the ServerHello message, and includes an appropriate ServerKeyExchange message. NAF may help UE in selecting the correct PSK identity by providing a hint in ServerKeyExchange message. That hint could be, for example, "3GPP_bootstrapping@bsf.operator.com."

3. UE performs the bootstrapping procedure to produce TID and Ks_NAF. If bootstrapping procedure has been done recently, UE may use the TID and Ks_NAF produced from that procedure.

4. UE sets the TID as the PSK identity, and Ks_NAF as the PSK. UE then sends ClientKeyExchange containing the TID, ChangeCipherSpec, and Finished messages to NAF. The TLS premaster secret is derived from Ks_NAF as specified in the psk TLS draft [2].

5. NAF extracts the TID from received ClientKeyExchange message and requests the Ks_NAF from BSF.

6. BSF returns Ks_NAF that corresponds to the TID.

7. NAF validates the Finished message sent by UE, and sends ChangeCipherSpec, and Finished messages to the UE.

8. UE and NAF initiate application data transfer in the TLS session.

## 2.2 Key expiration and error cases

**Key lifetime expiration:** During TLS handshake, NAF indicates to the UE that the key lifetime has expired by sending handshake_failure error message in step 7. This should trigger UE to bootstrap again.

During usage of TLS session, NAF indicates to the UE that the key lifetime has expired by first closing the connection using close_notify alert message, and when and if UE tries to resume the old TLS session (i.e., sends CientHello message with the old session ID in step 1) NAF will refuse it by generating a new session ID, and continue with normal TLS handshake (i.e., trigger UE to bootstrap again) as described in section 2.1.

**UE or NAF does not support psk TLS:** If UE does not support psk TLS, it won't add the PSK-based ciphersuites to ClientHello message (in step 1). If NAF does not support psk TLS, it won't recognize the PSK-based ciphersuites, and therefore won't set any of them as the selected ciphersuite in ServerHello message (in step 2).

If neither NAF nor UE support psk TLS, they should establish normal server authenticated TLS tunnel, and use for example HTTP Digest based authentication inside the TLS tunnel.

It should be noted, that UE should always offer also non-psk ciphersuites to the NAF, otherwise the TLS session establishment will fail, if NAF does not support psk TLS.

**Mismatched PSK identity:** Since psk TLS may be used in different authentication domains, UE may contain several shared keys, each key applicable in different authentication domain. In ClientKeyExchange message (in step 4) UE may insert a PSK identity that belongs to authentication domain other than GAA. This would mean that NAF would receive a lookup failure message from BSF because BSF does not have knowledge of the PSK identity used. If NAF does not recognize the PSK identity (i.e., TID), it should respond with a "decrypt_error" message as is specified in psk TLS draft [2].

However, NAF may indicate in ServerKeyExchange message (in step 2) to the UE in that GBA based PSK identity (i.e., TID) should be used. For example, PSK identity hint could be "3GPP_bootstrapping@bsf.operator.com", as is done in HTTP Digest case where the "realm" attribute is set to above value to indicate to the UE that bootstrapping shared secret should be used for authentication (see TS 33.220, annex A and TS 24.109, clause 5).

**Other error situations** are related to existing TLS functionality and are described in TLS [4] and TLS extensions [3] specifications.

# 3 Discussion & analysis

## 3.1 Status of psk TLS Internet draft

Authors of the psk TLS internet draft consider the draft to be ready. If 3GPP adds this draft to the IETF dependency list, it is assumed that the psk TLS shall obtain RFC status quickly.

## 3.2 Differences between HTTP Digest and psk TLS

This chapter lists the differences in usage of HTTP Digest and psk TLS. When HTTP Digest is used:

- UE cannot indicate that it supports GBA-based authentication;
- integrity protection of protocol messages is provided by either HTTP Digest's quality-of-protection scheme "auth-int", or using HTTP Digest inside a TLS tunnel;
- confidentiality protection of protocol messages cannot be provided.

In current implementations of HTTP Digest (e.g., Apache), the "auth-int" quality-of-protection scheme is typically not implemented because it is generally assumed, that if integrity protection is needed, it is provided using TLS (as specified in IETF RFC 2818 [5]).

Psk TLS has the following advantages when compared to HTTP Digest authentication:

- psk TLS is more generic than HTTP Digest since non-HTTP-based services can use it;
- in psk TLS, authentication is done only in one level;
- psk TLS supports the discovery of capabilities of both UE and NAF (e.g., whether if UE or NAF does not support psk TLS, this can be indicated during TLS handshake);
- if confidentiality is required, TLS needs to be used with HTTP Digest anyway;
- extra step of tying HTTP Digest authentication and server authenticated TLS tunnel together is not needed;

## 3.3 Differences between shared-key and psk TLS versions

Annex A of the psk TLS Internet draft [2] explains the differences between shared-key and psk versions. The main difference is that shared-key TLS re-uses the TLS session cache and session resumption functionality while the psk TLS defines new ciphersuites for pre-shared keys and uses those. Also, the psk TLS is more elegant and better in line with the design principles of TLS than the shared-key TLS.

Shortcomings of shared-key TLS compared to psk TLS are also discussed in the next chapter.

## 3.4 Concerns raised in S3-030721

Ericsson contribution to Munich meeting (S3-030721) discussed the challenges and standardization gaps related shared-key TLS [1]. This section addresses the concerns raised in S3-030721 in relation to psk TLS [2]. Paragraphs in italic font are direct quotations from the Ericsson contribution, and the reference [Shared-key-TLS] refers to shared-key TLS Internet draft.

### 3.4.1 TLS implementations

*"Shared-secret TLS should be seen as an optimization for full standard TLS. This optimization may benefit some use situation, however, it does not help implementation in general. For example, Web-browsers that are used to access services in the open Internet must anyhow have full TLS implementation or otherwise the end-user is not able to use some common services, such as banking services, in the Internet securely.*

*SA3 should adopt a working assumption that shared-secret TLS can only be defined as an optimization for TLS. Both the client and the NAF must always have full TLS implementation."*

The psk TLS is an *extension* to full standard TLS, and is not intended to replace the existing TLS implementations. The intention of psk TLS is to enable the usage of GBA directly in TLS implementation.

## 3.4.2 Shared-secret TLS capabilities

*"In order to use shared-secret TLS, both the client and the NAF needs to know that the other end is able to use the mechanism. Procedure where the UE just decided to use the shared-secret TLS without somehow knowing that one particular NAF really supports it is not appropriate. Firstly, it will cause some additional load in the UE side if the shared-secret TLS is not support in the NAF. Secondly, it will lower the flexibility of future development of the network because some UEs will always assume that shared-secret TLS will be used.*

*There may be several ways to overcome this problem. For example, it may require the use of Handshake Protocol at the beginning of shared-secret TLS session. However, note that this kind of procedure is not part of [Shared-secret-TLS] draft.*

*On the other hand, if the use of shared-secret TLS is limited to Authentication Proxy, and if SA3 decides to use "forwarding proxy" approach, the information about TLS capabilities of the proxy could be uploaded to the UE using OMA Device Management mechanisms. Alternatively, suitability of DNS could also be further studied.*

*SA3 should adopt a working assumption that capability negotiation is required if shared-secret TLS is adopted as a solution to access some NAF securely."*

The psk TLS supports the negotiation of capabilities during TLS handshake:

- UE is able to indicate that it supports psk TLS (in ClientHello message), and
- NAF is able to select whether it wants to use psk TLS.

Thus, it is possible to deal gracefully with the situation that either UE or NAF does not support psk TLS. If UE does not support psk TLS, it simply will not include the PSK-based ciphersuites to ClientHello message. If NAF does not support psk TLS, it will simply ignore the unrecognized ciphersuites (i.e., PSK-based ciphersuites).

## 3.4.3 Negotiation of security parameters

*"Shared-secret TLS is based on the idea of re-using session caching mechanism (i.e. resumed sessions) directly with symmetric keys in order to avoid public-key operations. However, the caching mechanism does not include any negotiation mechanism for TLS security parameters, such as encryption, MAC algorithms or pseudo-random functions (PRF). In order to negotiate these parameters, SA3 must specify how to extend the shared-secret-TLS for 3GPP use, e.g. by re-useing some parts of TLS Handshake Protocol. This requires breaking some general TLS related rule for not using the Handshake Protocols with resumed sessions. The use of Handshake Protocol may not be a problem, however, this kind of protocol misuse increases the possibility that some security and/or implementation problems may appear in the future.*

*If the Handshake Protocol is not used, it is also possible that shared-secret TLS is restricted for certain security parameters, and consequently the use of this mechanism is limited to time when these security parameters are known to be secure.*

*SA3 should be aware that shared-secret-TLS draft does not provide complete solution for 3GPP. SA3 can not avoid defining extensions to the draft in order to solve the negotiation problem. "*

The psk TLS includes negotiation of security parameters such as encryption, and MAC algorithms, thus SA3 does not need to extend the psk TLS for 3GPP use.

Note: TLS always uses one fixed pseudo random function (PRF), and thus PRF cannot be negotiated in TLS.

## 3.4.4 Key agreement between UE and NAF

*"In order to use shared-key TLS, both the UE and NAF need to agree on session ID and key. [Shared-secret-TLS] does not specify how the agreement is done. Instead, the key agreement is seen as an application specific problem. Generally speaking, it is assumed that the client will know the session ID and the key before contacting NAF with TLS.*

*There are at least two ways of implementing this in GBA. Assuming that UE knows that a NAF supports shared-secret TLS, it could in theory start using TLS directly without using application layer protocols first. However, this solution may need changes to the TLS in the NAF side since the NAF needs to have access to key identifier before contacting BSF. Alternatively, the key agreement could be done at application layer, as already proposed in [S3-030576].*

*It is proposed that SA3 adopts a working assumption that key agreement for shared-secret TLS is always done at application layer in order to avoid additional changes to TLS in NAF."*

The psk TLS includes session ID and key agreement, thus SA3 does not need to extend the psk TLS for 3GPP use.

### 3.4.5 Shared-secret TLS with authentication proxy

*"SA3 is currently discussing two different models for authentication proxy (AP), i.e. the so-called "forwarding" and "reverse" proxy. The "forwarding" proxy requires some additional configuration mechanism for UE to know the IP-address and services behind the proxy. On the other hand, the "reverse" proxy is transparent to the UE, and no configuration is needed. Instead, standard DNS procedures are used.*

*Depending on the decisions related to the other issues presented in this document, e.g. whether the configuration mechanism is also needed to negotiate on shared-secret TLS capabilities, SA3 should consider the suitability of shared-secret TLS with alternative proxy models. It may mean that shared-secret TLS fits better with the "forwarding" proxy than with "reverse" proxy.*"

The psk TLS fits for both "forwarding" and "reverse" proxy scenarios:

**In forwarding proxy scenario**, the TLS session should be terminated in the authentication proxy, which is not typical for forwarding proxy scenarios, where TLS traffic transparently flows through the forwarding proxy. In this case, UE must be configured to contact the forwarding proxy using TLS, and know that the proxy is terminating the TLS connection, i.e., the Ks_NAF must be derived using proxy's NAF_ID, not the application server's NAF_ID.

**In reverse proxy scenario**, the TLS session is terminated in the reverse proxy. This is a typical configuration option application server in the Internet today. In this case, UE does not need any additional configuration related to proxy since the reverse proxy is transparent to the UE. However, the reverse proxy must be able distinguish between different connections intended for different application servers behind the reverse proxy, because it needs to discover the NAF_ID before fetching the Ks_NAF from BSF over Zn interface. This can be done by using TLS Extensions [3], where ClientHello message is extended to include ServerNameList which can be used to indicate to the reverse proxy what is the NAF_ID (i.e., fully qualified domain name, FQDN) of the intended application server behind the reverse proxy.

# 4 Conclusion & Proposal

This contribution described the usage of pre-shared-key (psk) TLS [2] usage in GAA, and compared psk TLS against shared-key TLS [1], and HTTP Digest authentication usage within GAA. We also addressed concerns that were listed in Ericsson contribution S3-030721. The psk TLS internet draft can be used as it is defined in IETF, and 3GPP/SA3 does not need to specify any extensions to it in order to be able to use psk TLS in GAA. Thus, psk TLS is a complete solution for 3GPP on how a TLS session can be established between UE and NAF using GBA.

Therefore, we ask SA3 to:

- add psk TLS internet draft to the 3GPP IETF dependency list;
- endorse the usage of psk TLS within GAA; and
- add descriptions of psk TLS usage into relevant specifications in both SA3 (stage 2) and CN1 (stage 3) domain.

A pseudo-CR is in annex of the present contribution implementing this method to TS 33.222.

# 5 References


[1] IETF Internet-Draft: "Use of Shared Key in the TLS Protocol", October 2003, URL: http://www.ietf.org/internet-drafts/draft-ietf-tls-sharedkeys-02.txt

[2] IETF Internet-Draft: "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", February 6, 2004, URL: http://www.ietf.org/internet-drafts/draft-eronen-tls-psk-00.txt

[3] IETF RFC 3546: "Transport Layer Security (TLS) Extensions", June 2003.

[4] IETF RFC 2246: "The TLS protocol Version 1.0", January 1999.

[5] IETF RFC 2818: “HTTP over TLS”, May 2000.

# ANNEX: Pseudo CR to TS 33.222 V1.0.0:

**===== BEGIN CHANGE =====**

## 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*.

[1] 3GPP TS 23.002: “Network architecture”.

[2] 3GPP TS 22.250: “IP Multimedia Subsystem (IMS) group management”; Stage 1”.

[3] 3GPP TS 33.220: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture".

[4] 3GPP TR 33.919: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Generic Authentication Architecture (GAA); System description".

[5] 3GPP TS 33.141: "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Presence Service; Security"

[6] IETF RFC 2246 (1999): "The TLS Protocol Version 1".

[7] IETF RFC 3268 (2002): "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)".

[8] IETF RFC 3546 (2003): "Transport Layer Security (TLS) Extensions".

[9] IETF RFC 2818 (2000): “HTTP Over TLS ”.

[10] IETF RFC 2617 (1999): "HTTP Authentication: Basic and Digest Access Authentication"

[11] IETF RFC 3310 (2002): "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)"

[X] IETF Internet-Draft: “Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)”, February 6, 2004, URL: http://www.ietf.org/internet-drafts/draft-eronen-tls-psk-00.txt

**===== BEGIN NEXT CHANGE =====**

## 5.4 Shared key-based mutual authentication between UE and NAF

Editor's note: If the "Pre-Shared Key Ciphersuites for TLS" Internet Draft [X] does not reach the RFC status by the time when Release 6 is frozen, this subclause shall be removed and the support for the Pre-Shared Key TLS is postponed to Release 7.

The HTTP client and server may authenticate each other based on shared-key generated during the bootstrapping procedure. The shared-key shall be used as master key to generate TLS session keys, and also be used as the proof of secret key posession thus the authentication function. The exact procedure is specified in Pre-Shared Key Ciphersuites for Transport Layer Security (TLS) [X].

Editor's note: The exact procedure of "Pre-Shared Key Ciphersuites for TLS" is under inspection in IETF. When the procedure is ready in IETF, the description how it used in GAA should be added to TS 24.109, and this subclause should refer to it. The following gives general guidelines how the TLS handshake may be accomplished using GBA-based shared secret. The exact definitions of the message fields are left to stage 3 specifications.

This section explains how GBA-based shared secret that is established between the UE and the BSF as specified in 3GPP TS 33.220 [3] is used with Pre-Shared Key (PSK) Ciphersuites for TLS as specified in IETF Internet-Draft [X].

1. When an UE contacts a NAF, it may indicate to the NAF that it supports PSK-based TLS by adding one or more PSK-based ciphersuites to the ClientHello message. The UE shall include other than PSK-based ciphersuites to the ClientHello message.

2. If the NAF is willing to establish TLS tunnel using PSK-based ciphersuite, it shall select one of the PSK-based ciphersuites offered by the UE, and send the selected ciphersuite to the UE in the ServerHello message. The NAF shall send the ServerKeyExchange message with a PSK-identity that shall contain a constant string "3GPP-bootstrapping" to indicate the GBA as the required authentication method. The NAF finishes the reply to the UE by sending ServerHelloDone message.

NOTE: If the NAF does not wish to establish TLS tunnel using PSK-based ciphersuite, it shall select a non-PSK-based ciphersuite and continue TLS tunnel establishment based on the procedure described either in subclause 5.3 or subclause 5.5.

3. The UE shall use GBA-based shared secret for PSK TLS, if the NAF has sent ServerHello message containing a PSK-based ciphersuite, and a ServerKeyExchange message containg a constant string "3GPP-bootstrapping" as the PSK identity hint. If the UE does not have valid GBA-based shared secret it shall obtain one by running the bootstrapping procedure with the BSF over Ub interface as specified in 3GPP TS 33.220 [3].

   The UE derives the TLS premaster secret from the NAF specific key (Ks_NAF) as specified in IETF Internet Draft [X].

   The UE shall send a ClientKeyExchange message with the B-TID as the PSK identity. The UE concludes the TLS handshake by sending the ChangeCipherSuite and Finished messages to the NAF.

4. When the NAF receives the B-TID in the ClientKeyExchange messages it fetches the NAF specific shared secret (Ks_NAF) from the BSF using the B-TID.

   The NAF derives the TLS premaster secret from the NAF specific key (Ks_NAF) as specified in IETF Internet Draft [X].

   The NAF conludes the TLS handshake by sending the ChangeCipherSuite and Finished messages to the UE.

The UE and the NAF have established a TLS tunnel using GBA-based shared secret, and the may start to use the application level communication through this tunnel.

**===== END CHANGE =====**