3GPP TSG SA WG3 Security — S3#28 **S3-030224**

6 – 9 May 2003

Berlin, Germany

Source:	**Siemens**
Title:	**Security protocols for the use of HTTP at the Mt reference point in the IMS**
Document for:	**Discussion and decision**
Agenda Item:	**7.19 Presence, (7.1 IMS)**

Abstract

In SA3#27, several contributions (S3-030056, 60, 69, 84) discussed possible solutions for security for the use of HTTP at the Mt reference point. This contribution proposes protocols to provide mutual authentication, confidentiality and integrity over the Mt reference point. Server authentication, confidentiality and integrity is to be provided by TLS, client authentication is to be provided by HTTP digest. The problem of a man-in-the-middle attack is also addressed in this contribution. A companion contribution addresses the key management problem for HTTP digest.

1. Introduction

A scenario currently under discussion in several 3GPP groups is the use of HTTP communication over the Mt reference point between a UE and an IMS-based application server (AS). An example of the use of HTTP over the Mt reference point is a means for a mobile user subscribed to the IMS, to manage his or her data on the application server, e.g.

- Access lists on the presence server;
- Buddy lists for chat (IMS messaging)
- IMS Group Management
- Conference Settings: Creation, data, type, participants, …

It is obvious that the communication over the Mt reference point needs to be adequately secured. In SA3#27, several contributions (S3-030056, 60, 69, 84) addressed this issue. This contribution proposes protocols to provide mutual authentication, confidentiality and integrity over the Mt reference point. A companion contribution by Siemens ("Key management for the use of HTTP at the Mt reference point in the IMS") addresses a key management problem pertaining to the proposal in this contribution.

The proposal for security protocols over the Mt reference point is motivated by the following considerations:

- Re-use of existing protocols and implementation as much as possible;
- Key management based on IMS registration, no additional authentication runs with the HSS.

2. Confidentiality, integrity, and server authentication

In accordance to SA3's current working assumption it is proposed here to use TLS for integrity and confidentiality protection for the use of HTTP over the Mt reference point. In addition, we propose that AS to UE authentication is also performed using TLS. This implies that the AS presents a certificate on its public key to the UE. In other words, we proposed to use HTTPS for integrity, confidentiality and server authentication. The rationale for this proposal is that HTTPS implementations are readily available.

3. Client authentication

Once a secure TCP connection has been established and the first HTTP request is sent by the UE to the AS, the AS replies with a *401 unauthorized* message and includes a WWW-Authenticate header containing a nonce to start a run of the HTTP digest protocol [rfc2617].

HTTP digest assumes the use of a password shared between client and server. In our proposal, the password shared by the UE and the AS is a key DKi which is derived during the IMS registration of the UE preceding the current communication between UE and AS (cf. companion contribution). The derived key is pushed to the AS by the S-CSCF as part of the registration procedure. The derived key is specific for the UE and the AS. It shall not be used for any other purpose than with the HTTP digest protocol for UE to AS authentication, as specified in this contribution.

After receiving a *401 unauthorized* message with a WWW-Authenticate header, the UE proceeds, according to [rfc2617], by sending an Authorization header. This header contains a message digest computed over the nonce received from the AS and some other data. The AS responds with a Authentication-Info header indicating the success or failure of the UE authentication.

Selection of options in the HTTP digest protocol:

Algorithms: it is suggested to use MD5 (default in HTTP digest), as this is supported by all HTTP digest implementations.

qop-options: in the WWW-Authenticate header, this indicates the "quality of protection" values supported by the server. The value "auth", indicating authentication, shall be used in our setting. The same qop values shall then be used in the Authorization header and the Authentication-Info header. No use is seen for integrity of the message body, which would be provided if the option was set to “auth-int”.

response digest: the optional response digest in the "response-auth" directive in the Authentication-Info header supports mutual authentication -- the server proves that it knows the user's secret. It is proposed to omit the response digest as server authentication is already provided by TLS using server certificates. Furthermore, the use of server authentication in HTTP digest provides no additional protection against man-in-the-middle attacks discussed in the next section, in the case server authentication by TLS was compromised.

4. Considerations on man-in-the-middle attacks

[Aso] and [Put] showed that security solutions where a client authentication protocol is run through a server-authenticated tunnel is susceptible to certain man-in-the-middle attacks. In principle, also the proposal presented in this contribution falls into the class of security solutions which may be susceptible to the desribed type of attacks.

We quote from [Put]: *“This section describes how man-in-the-middle vulnerabilities can be exploited, as well as discussing the underlying causes of the attacks. ... The major scenario for the attack is a one-way authenticated tunnel encapsulating subsequent authentication methods. In this scenario, the client and server first establish a tunnel, then include within the tunnel one or more authentication method(s). The attacker first poses as a valid client to the server and establishes a tunnel that is authenticated only on the server end, obtaining tunnel keys. ... Once the attacker has established a tunnel to the server, it seeks to induce clients to connect to it. In the third step, the client connects to the [attacker], and the attacker tunnels the authentication method between the client and server. In the last step, the attacker obtains access to the server, using the successfully tunnelled authentication and the tunnel keys.”* For details see [Aso] and [Put].

[Put] also lists four causal conditions CC-A to CC-D which all have to be met for the attack to be successful. But for our proposal, causal condition CC-A is not met, hence the attack does not apply. We quote CC-A from [Put]:

“[CC-A] Client and Authentication server policy allowing client credentials to be used both within one-way server-authenticated tunnels and outside them.”

In our proposal, the key shared between UE and AS is specific to UE and AS (as the identities of the user and of the AS implicitly or explicitly enter the key derivation process), and shall only be used with the combination of security protocols as proposed in this contribution. In other words, the derived key shared between UE and AS must only be used for authentication of the UE to the AS by means of http digest run through a server-authentication TLS tunnel between UE and AS. Our proposal is in line with solution S2 of [Put]. We quote:

"[S2] Guarantee that the same peer credential is never usable inside and outside a tunnel using server and client policy. This prevents condition CC-A.

This solution actually works for all methods, but is sometimes hard to deploy, due to legacy deployments, and since clients and servers need to be synchronised for proper policy enforcement. An additional problem with this solution is the manageability issues due to the multiple credentials that have to be managed by the same client and server."

We address the concerns listed with solution S2 in [Put].

a) there is no legacy deployment which could conflict with the use of the derived key DKi as we specifically introduce the derived key for this particular use.

b) The policies of UE and AS are indeed synchronised as they both have to adhere to the 3GPP specifications.

c) There are no multiple credentials to be managed by the same UE and AS as the secure communication between UE and AS as specified by 3GPP uses only the derived key DKi. 3GPP-specific credentials need to be defined by 3GPP anyhow, as there are no credentials available from outside 3GPP which 3GPP specifications could use.

Of the other three solutions listed in [Put], S1 and S4 only work for key-deriving client authentication methods, and the method proposed here, namely HTTP digest, does not derive a key. Solution S3 (which consists in preventing that server entities can be used for man-in-the-middle attacks, e.g. preventing the use of false base stations) is called impractical and too expensive in real-world settings, according to [Put]. So, only solution S2 applies to our setting anyhow.

A note of caution: our statement that man-in-the-middle attacks as described in [Aso] and [Put] do not apply in our setting only holds if server authentication by TLS (with server certificates) can be relied upon. This assumption, however, should not be taken for granted as the sometimes sloppy handling of server certificates issuing and verification on the World Wide Web shows. But careful certificate handling can avoid these risks. For a discussion of related issues we refer to [S3-030030].

5. How are the conditions set in SA3's working assumptions met?

In their last meeting #27, SA3 adopted the following **working assumptions**:

1. Transport Layer Security (TLS) will be taken as a priority mechanism for protection of HTTP, such as integrity and confidentiality, but it will be further studied along with other mechanisms;
2. The authentication method should use the AKA authentication;
3. The approach to be adopted shall mitigate the interleaving attack in tunnelled authentication protocols.

We will show in this section that the proposed solution is in line with these working assumptions.

Re 1: it is proposed in section 3 to use TLS for integrity and confidentiality, so this condition is met.

Re 2: UE to AS authentication is based on a shared key which is derived from the key CK obtained by a run of HTTP digest aka during IMS registration (see Siemens companion contribution on "Key management for the use of HTTP at the Mt reference point in the IMS"). So, the authentication method does use the AKA authentication for user authentication. AS to UE authentication is performed via TLS, and is not based on AKA. But, although this is not entirely clear from the formulation of the working assumption, we believe that it was the intention of this working assumption to require the re-use of the 3G authentication base for user authentication. Our proposal satisfies this requirement. In case of doubt, we propose to clarify the working assumption in the above sense.

Re 3: section 4 showed that man-in-the-middle attacks are not possible because the key shared between UE and AS is specific to the use with one particular combination of protocols over the Mt reference point.

6. How are the conditions set in SA2's response LS to SA3#28 met?

Document S3-030210 is sent to SA3#28 by SA2 as a "Response to LS (S2-030445) on use of HTTP between UE and AS in the IMS". In this document, SA2 asks SA3 to take into account three agreements listed in S3-030210 in SA3's work. We show in the following that the solution proposed in this document is in accordance with these agreements.

SA2's agreement 1: *dependencies on work, which is uncertain to meet the release 6 deadline should be avoided, given that the Mt reference point and Presence are important parts of release 6. This applies to dependencies with work done inside and outside 3GPP, in particular as the latter is outside the control of 3GPP.*

No such dependencies are seen as everything to be specified is under control of 3GPP, and there is no dependency on Rel6 work items other than presence.

SA2's agreement 2: *usage of Mt does not require that the user is IMS registered (though in most cases the UE will be registered).*

The Siemens companion contribution on key management for the Mt reference point explains that the derived application keys DKi live on when a user de-registers. It is only required that a user has been registered before using Mt.

SA2's agreement 3: *the possibility to have multiple Application Servers serving the same user at the same time should be taken into account.*

This is taken into account as multiple application keys are derived, specific to the application servers, and are pushed to the application servers by means of the 3rd party registration mechanism, as explained in the Siemens companion contribution on key management for the Mt reference point.

Conclusions

This contribution shows that HTTP communication between UE and AS over the Mt reference point can be secured using TLS with server certificates for AS authentication, confidentiality and integrity, and HTTP digest for UE authentication, where the HTTP digest uses a key derived during the most recent IMS registration which is specific for this use over the Mt reference point. The contribution also shows that the proposal is in line with SA3's working assumptions, in particular, it is shown, that man-in-the-middle (or interleaving) attacks should be of no concern. The proposal is also in line with SA2's agreements as stated in their response LS to this meeting.

References

[Aso] N. Asokan, V. Niemi, K. Nyberg: "Man-in-the-Middle in Tunnelled Authentication", Draft, October 2002, http://eprint.iacr.org/2002/163/ "

[Put] "The Compound Authentication Binding Problem ", INTERNET-DRAFT, <draft-puthenkulam-eap-binding-02.txt>", www.ietf.org, March 2003.

[S3-030030] Intel, Cisco: "PKI Deployment Models for PEAP", contribution to 3GPP SA3#27, www.3gpp.org, Feb 2003.

[rfc2617] "HTTP Authentication: Basic and Digest Access Authentication", Request for Comments: 2617, www.ietf.org, June 1999.