3GPP TSG SA WG3 Security — S3#21bis IMS ad hoc **S3z020037**

Antwerp, Belgium

31 Jan – 1 Feb, 2002

Source: **Siemens**

Title: **Comments and questions on the revised anti-replay protection scheme for the SIP level integrity solution in TD S3z020030**

Document for: **Discussion**

Agenda Item: **6.1**

Abstract

TD S3-010664, submitted to S3#21, identified a shortcoming of the anti-replay protection scheme for the SIP level integrity solution in Annex C of TS 33.203, v070 that may lead to an avoidable loss of calls. This shortcoming carried over into TS 33.203, v100. TD S3z020030, submitted to S3#21bis, proposes a revised version of this anti-replay protection scheme to address the identified shortcoming. While the revised scheme seems to be capable of mitigating the particular shortcoming identified in TD S3-010664, TD S3z020030 still specifies replay protection only by way of example, and does not provide sufficient detail to allow a full evaluation of the revised scheme. This contribution points to required clarifications and open issues.

The comments below do not necessarily imply that there are problems with the approach sketched in TD S3z020030, which could not be solved. However, they show that quite some more work is needed to get the specification finalized.
The question and comments are largely in the order determined by the text in S3z020030. They are not ordered according to importance. In an appendix to this contribution, the example message flow from S3z020030 is given for the convenience of the reader.

From S3z020030: The implications of supporting the Digest client-server model, then, are that both the UE and the P-CSCF must: 1) be able to issue Digest challenges, which includes issuing nonces; and 2) maintain its own counter for the 'nonce-count' directive for use when operating in the client role.

Comment 2): it is also required for the server to keep track of nonce-count to check incoming requests.

From S3z020030: When either the UE or P-CSCF receives a SIP request (i.e. is acting as Digest server), it expects the sending entity (acting as client) to use in the computation of the message digest a nonce that it (the server) has previously issued. If an unrecognized nonce appears in the Digest response, the receiving entity will deem the message to have failed the integrity check.

Question: how many nonces (with the associated nonce-counts) does the server need to keep in storage to be able to check whether the nonce was previously issued or the nonce-count is too low?

From S3z020030: In this way [i.e. by using nonces, the author] the Digest framework mitigates "reflection attacks" (attacks in which a Man-in-the-Middle reflects a genuine message from an entity back to its sender).

Question: what if both sides, the UE and the P-CSCF, happen to issue the same nonce? Is there a requirement for randomness and a certain nonce length so that it is unlikely that nonces coincide?

From S3z020030: Finally, the P-CSCF adds an Integrity header field to all SIP requests sent toward the UE.

Comment: the integrity header is a new concept which is not part of rfc2617 "HTTP Authentication: Basic and Digest Access Authentication". It is rather loosely specified in TS 33.203 as part of the example information flow (see appendix to this contribution).

The example flow seems to indicate that in each 180-response to an INVITE request, a new "nextnonce" is issued by the server, to be used by the client in the next INVITE request.

Questions: what are the requirements on "nextnonce"? Does it have to be different from previous values of "nextnonce", or can it be a sequence number increased by 1 each time, or even a constant throughout a registration, or does it not matter which?

It may happen that the message carrying "nextnonce" gets lost.

Question: What does the client use as a nonce then? Conjecture: it uses an old nonce with an increased nonce-count nc. This should be made explicit.

"cnonce" appears it the example messages. This comes from rfc2617.

Question: What is the purpose of cnonce in an IMS context? Is it needed? What should its value be?

The example flow seems to be inconsistent with the principles laid out in the textual description of the enhanced digest scheme in TD S3z020030 in the following way: the text demands that the client uses a nonce in a message digest which was previously sent to him by the server as "nextnonce". But in message 6 the P-CSCF, acting as a client, uses nonce3 in a terminating INVITE request, and nonce3 was issued by the P-CSCF itself, not the UE.

Question: in which messages is the UE to send "nextnonce" values to be used by the P-CSCF? A solution also has to cover the case when the terminating INVITE request immediately follows the completion of the registration procedure. In this case, "nextnonce" would have to be carried in a REGISTER message. Is this possible?

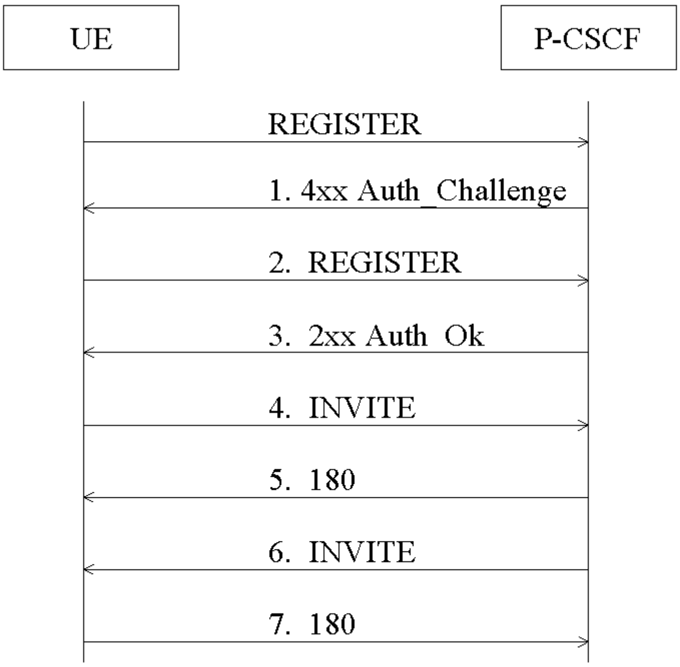
More generally:

Comment: the use of the three values nonce, cnonce and nonce-count as well as the sending of nextnonce appear unnecessary for the purposes of IMS replay protection. One counter for each direction, to be maintained by both, the UE and the P-CSCF, appears sufficient in the IMS context for the purpose of anti-replay protection. It is recognized that the enhanced digest scheme may be designed in such a way that it is sufficiently flexible to be used in other scenarios as well. However, the scheme should be also be such that its use in the IMS does not create any unnecessary overhead, e.g in managing the replay protection scheme. If a profiling of the scheme for IMS purposes is possible and advisable it should be stated in the specification how this profiling is done.

Appendix:

The example information flow in TD S3z020030 is as follows:

initiated (4-5) and one UE terminated (6-7).



1. **4xx response – this carries both the IMS AKA challenge (from the registrar) and the Digest challenge for integrity protection (from the P-CSCF):**

   SIP/2.0 4xx Auth_Challenge

   WWW-Authenticate: EAP <RAND AUTN>

   Proxy-Authenticate: Digest realm=3GPP-IMS nonce=<nonce1> algorithm=MD5 qop=extended-int

2. **Integrity protection is turned on with the next REGISTER – the integrity credentials are placed in the Digest response:**

   REGISTER sip: ... SIP/2.0

   Authorization: EAP <RES>

   Proxy-Authorization: Digest username=ims-user, realm=3GPP-IMS, nonce=<nonce1>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=extended-int

3. **The 2xx response is also integrity protected – the P-CSCF adds the Authentication-Info header to carry the message digest:**

   SIP/2.0 2xx Auth_Ok

   Authentication-Info: nextnonce=<nonce2>, qop=extended-int, rspauth=<message-digest>, nc=1, cnonce=<value>

4. **A subsequent INVITE request must also be integrity protected – the UE pre-emptively adds the Proxy-Authorization header:**

   INVITE sip: … SIP/2.0

   Proxy-Authorization: Digest username=ims-user, realm=3GPP-IMS, nonce=<nonce2>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=extended-int

5. **The 180 is integrity protected in the same fashion was the 2xx response (message #3):**

   SIP/2.0 180 Ringing

   Authentication-Info: nextnonce=<nonce3>, qop=extended-int, rspauth=<message-digest>, nc=1, cnonce=<value>

6. **An incoming INVITE must also be integrity protected – the P-CSCF adds the Integrity header, which has the same syntax as Proxy-Authorization:**

   INVITE sip: … SIP/2.0

   Integrity: Digest username=ims-user, realm=3GPP-IMS, nonce=<nonce3>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=extended-int

7. **The UE protects the 180 response by adding Authentication-Info:**

   SIP/2.0 180 Ringing

   Authentication-Info: nextnonce=<nonce4>, qop=extended-int, rspauth=<message-digest>, nc=1, cnonce=<value>