Technical Specification Group Services and System Aspects
Meeting #16, Marco Island, Florida, 10-13 June 2002

TSGS#16(02)0354

Source: **SA WG3**

Title: **CR to 33.203: Remove Annexes that describes Extended HTTP Digest solution (Rel-5)**

Document for: **Approval**

Agenda Item: **7.3.3**

SA doc#	Spec	CR	R	Phase	Subject	Cat	Current Version	WI	SA WG3 doc#
SP-020354	33.203	011		Rel-5	Remove Annexes that describes Extended HTTP Digest solution	D	5.2.0	IMS-ASEC	S3-020224

3GPP TSG SA WG3 Security — S3#23 S3-020224

14 - 17 May 2002, Victoria, Canada

CR-Form-v5

CHANGE REQUEST

⌘ 33.203 CR 011 ⌘ rev - ⌘ Current version: 5.1.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☐ Radio Access Network ☐ Core Network ☒

Title:	⌘	Remove Annexes that describes Extended HTTP Digest solution
Source:	⌘	SA WG3
Work item code:	⌘	IMS-ASEC
Date:	⌘	06/05/2002
Category:	⌘	D
Release:	⌘	Rel-5

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Take away redundant information
Summary of change:	⌘	Removes the solutions and discussions around Extended HTTP Digest
Consequences if not approved:	⌘	Information is kept in TS33.203 which is not required for IMS and has no use for IMS in Release 5

Clauses affected:	⌘	Annex C, Annex E, Annex F
Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

********** FIRST CHANGED SECTION **********

Annex C (informative): Mechanisms for SIP-level solution

[Editors Note: If the SIP-level solution is finally chosen the chapters below shall be moved into the main body of this TS in the corresponding sections.]

C.1 [6.2] Confidentiality mechanisms

[Editor's note: This section shall deal with cipher algorithms]

C.2 [6.3] Integrity mechanisms

[Editors note: There seems to be an unexpected shortcoming in the way SIP provides integrity protection on messages between UE and Proxies. In current SIP, HTTP Digest can be used to partially integrity protect the messages originated by an UE. However, SIP fails to provide integrity for Proxy to UE communication, i.e. for terminating INVITEs, for example. Proxies are not able to add Authorization headers on these messages, thus leaving the messages unprotected.

For the reason above, the headers and field names used in this section may not be final. However, the found inconsistency will probably make it easier for 3GPP to discuss about new SIP level integrity protection schemes with IETF.]

HTTP Digest shall provide integrity protection of SIP signalling between the UE and the P-CSCF, protecting all SIP signalling messages at the SIP level.

C.2.1 [6.3.1] Security Association Setup

The SA that is required for Digest integrity protection shall use the 128-bit integrity key IK generated through IMS AKA, as specified in section 6.1. The integrity algorithm and key are identical for integrity protection applied to messages travelling in either direction. Negotiation of the integrity algorithm to use occurs in the following way: The UE communicates the set of integrity algorithms that it supports to the P-CSCF through the Security-setup header field of the REGISTER message, as described in section 7.2. The P-CSCF selects an algorithm to use from the set of algorithm capabilities common to both the UE and the P-CSCF. The P-CSCF indicates the algorithm to use in the "algorithm" directive of the Digest challenge that is subsequently issued to the UE.

C.2.2 [6.3.2] Scope of Integrity Protection

Digest supports integrity protection of the SIP message body (not the headers) when the "qop-options" directive within the Digest challenge is set to the value "auth-int".

Digest supports integrity protection of the SIP message body plus a named list of headers when the "qop-options" directive is set to the value "auth-hdr-int".

Digest supports integrity protection of the entire SIP message when the "qop-options" directive within the Digest challenge is set to the value "auth-extd-int".

To provide for protection of the entire SIP message, the P-CSCF shall issue a Digest challenge to the UE specifying the value "auth-extd-int" for the "qop-options" directive.

C.2.3 [6.3.3] Computation of Integrity Protection Credential

The message 'digest', or message authentication code, is conveyed in the "response" directive of the Digest response. The rules for computing "response" are as described in [1] with the following consideration: if the UE receives a Digest challenge with "realm" directive including a 3GPP specific key word (e.g. "ik."), then the UE substitutes IK for the "password" component of A1 when computing "response=" in the Digest response. The UE saves the content of the whole realm directive from the Proxy-Authentication header to be used as a key identifier for subsequent messages. At this stage UE can not be sure whether the proxy identified in the realm really knows the IK, however the Proxy-Authentication-Info header will be used for final verification.

The UE sets the "username" component of A1 to some user identifier, e.g. the IMPI. When sending messages to the UE that are to be integrity protected, the P-CSCF applies the same rules when computing "response". Within these terminating messages, the rules for the content of 'realm' and 'username' parameters are opposite than for originating messages: the "realm" directive will include the same user identifier as above, e.g. the IMPI, and the "username" the identifier of the P-CSCF (including the 3GPP specific key word, e.g. "ik."). In this manner, the whole SIP message is always protected.

Note that terminating messages arriving to the P-CSCF from the home network will probably not include IMPI. For these messages, P-CSCF must use some other identifier (e.g. Request-URI) to find the IMPI and the IK needed for the integrity protection.

C.2.4 [6.3.4] Anti-Replay Protection

The Digest framework specifies that a server-initiated nonce is to be used by the client as a random number input to the production of the message digest. This nonce, along with a counter ('nonce-count') that is incremented by the client when sending each SIP request that is to be protected, facilitate anti-replay protection. The anti-replay protection feature of the integrity protection mechanism is as described in [12] with the following considerations. Per [12], the role of the server is to issue the nonce and to detect replays (through validation of 'nonce-count'), and the client must increment 'nonce-count' when computing the digest for each new SIP request that is to be integrity protected. In the one-hop environment that exists for the UE and the P-CSCF in the IMS, both the UE and the P-CSCF may fill either the client or server role in particular operational situations. When the UE sends an INVITE or other request towards the P-CSCF, the UE is the client and the P-CSCF is the server. When the P-CSCF sends (or re-submits) an INVITE towards the UE, the P-CSCF acts as the client and the UE acts as the server. 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.

New nonce values are communicated by the server to the client in two ways: 1) through the 'nonce' directive that is an obligatory part of the Digest challenge; and 2) through the 'nextnonce' directive that is an obligatory part of the Digest authentication of SIP responses (e.g., Authentication-Info header). Nonce values themselves are selected entirely by the server implementation – counter-based, clock- or other random number-based, and hybrid implementations are all possible. It is also a matter of server implementation how frequently new nonces are to be issued. To minimize the number of "stale" authentication attempts (generation of credentials by the client using an older nonce), the server should maintain a list of reasonable size of previously issued nonce values.

Expected behaviour of the UE and P-CSCF in relation to anti-replay protection is illustrated in the example information flow that follows in this section.

C.2.5 [6.3.5] Mitigation of 'Reflection Attacks'

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. In this way 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). It is possible that in the course of generating random nonces the UE and P-CSCF, while operating in the server role, happen to issue identical nonces for use; by making the nonces of sufficient length, the chance of such an occurrence is minimized.

C.2.6 [6.3.6] Digest Operation and Syntax in SIP

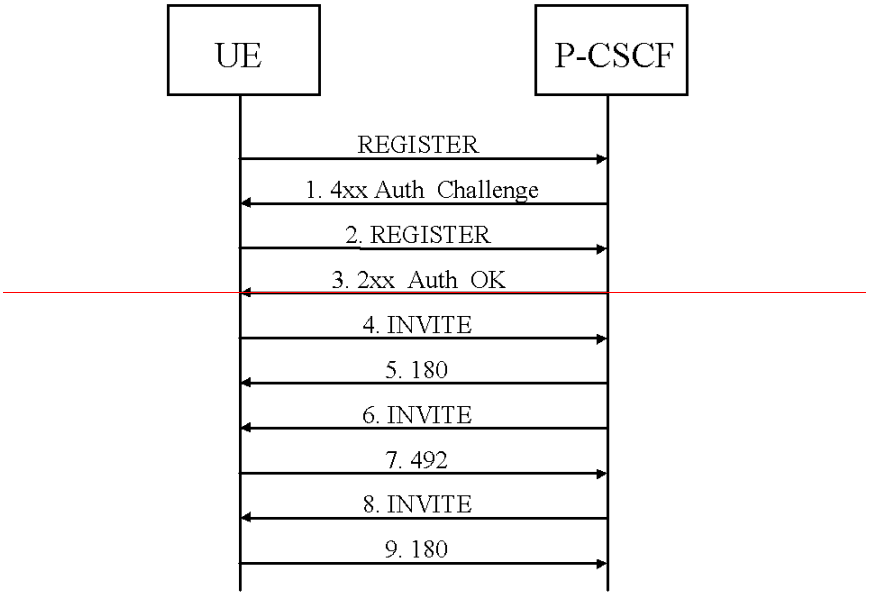
In the 3GPP IMS, then, normal operation of the Digest challenge-response mechanism for integrity protection is as follows:

The Digest challenge-related directives are carried in the WWW-Authenticate, Proxy-Authenticate or UAS-Authenticate header fields. The P-CSCF adds a Proxy-Authenticate header field to the 4xx Auth_Challenge that is sent by the S-CSCF (SIP registrar) toward the UE; the Proxy-Authenticate contains the Digest challenge that has been constructed by the P-CSCF.

The Digest response-related directives are carried in the Authorization, Proxy-Authorization or UAS-Authorization header fields, depending upon which header field carried the corresponding Digest challenge. These directives contain the credentials for the message integrity check. In the IMS context, the UE responds to the initial Digest challenge by adding a Proxy-Authorization header field to the REGISTER toward the S-CSCF (registrar). The UE pre-emptively adds a Proxy-Authorization header field to all subsequent UE-initiated SIP requests. The P-CSCF adds the Proxy-Authentication-Info header to all SIP responses. **The P-CSCF adds an UAS-Authorization header field to all SIP requests sent toward the UE. Finally, the UE adds the UAS-Authentication-Info header to all SIP responses.**

C.2.7 [6.3.7] Example Information Flow

The simplified message flow shown below illustrates the relevant header fields and contents for the SIP-level integrity protection mechanism. Please note that the message flow contains three cases: a registration (1-3), and two SIP sessions: one UE initiated (4-5) and one UE terminated (6-9).



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: <RAND AUTN>

   Proxy-Authenticate: Digest realm=ik.p-cscf@operator2.com nonce=<P-nonce1> algorithm=MD5 qop=auth-extd-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: <RES>

— Proxy-Authorization: Digest username=IMPI, realm= ik.p-cscf@operator2.com, nonce=<P-nonce1>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=auth-extd-int

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

— SIP/2.0 2xx Auth_Ok

— Proxy-Authentication-Info: Digest realm=ik.p-cscf@operator2.com, nextnonce=<P-nonce2>, qop=auth-extd-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=IMPI, realm= ik.p-cscf@operator2.com, nonce=<P-nonce2>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=auth-extd-int

NOTE: The client (UE) may re-use the previously issued nonce (i.e. set "nonce" to <P-nonce1> and "nc" to 2), but the Digest specification recommends against this. If the 2xx message containing 'nextnonce' were lost and not received by the UE, the UE would then use <P-nonce1> in the computation of the credential.

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

— SIP/2.0 180 Ringing

— Proxy-Authentication-Info: Digest realm=ik.p-cscf@operator2.com, nextnonce=<P-nonce3>, qop=auth-extd-int, rspauth=<message-digest>, nc=1, cnonce=<value>

2. An incoming INVITE must also be integrity protected – the first terminating SIP request, however, must be sent without the integrity credential (this permits the UE to issue a Digest challenge containing its own server-provided nonce).

3. The UE issues a 492 response containing a Digest challenge:

— SIP/2.0 492 Proxies Unauthorized

— UAS-Authenticate: Digest realm=IMPI, nonce=<UE-nonce1>, algorithm=MD5, qop=auth-extd-int, target=ik.p-cscf@operator2.com

4. The P-CSCF adds the UAS-Authorization header, which has similar syntax to Proxy-Authorization:

— INVITE sip: … SIP/2.0

— UAS-Authorization: Digest username=ik.p-cscf@operator2.com, realm=IMPI, nonce=<UE-nonce1>, uri=<SIP-URI>, response=<message-digest>, cnonce=<value>, nc=1, qop=auth-extd-int, responder= ik.p-cscf@operator2.com

5. The UE protects the 180 response by adding UAS-Authentication-Info:

— SIP/2.0 180 Ringing

— UAS-Authentication-Info: Digest realm=ik.p-cscf@operator2.com, nextnonce=<UE-nonce2>, qop=auth-extd-int, rspauth=<message-digest>, nc=1, cnonce=<value>

[Editors Note: A description of the security mode setup headers shall be included in this Annex. Furthermore the message flows need to be enhanced.]

[Editors Note: It is FFS how to optimize the profiling of HTTP Digest such that the extra roundtrip can be avoided for the first terminating INVITE. It is also FFS the exact profiling of the nonces]

~~[Editors note: There might be a need for IMS specific rules on how the error situations are handled with HTTP Digest. HTTP Digest includes a mechanism for a server/proxy to communicate some information about the status of the username, password or nonce to the client. If a server/proxy adds a 'stale=true' parameter in an authentication challenge, the client will try using the same password (i.e. integrity key) with the delivered new nonce value. If the 'stale=false' or anything else, or if it is missing, the client must ask for a new password from the end-user. In IMS, stale values can be used to deal with different error situations related to the key update. For example, P-CSCF could ask the client to perform re-registration if it sent a "stale=false" parameter. The potential error situations are for further study.]~~

~~*[Editors note: it is not so nice to test or try which SA is correct if the P-CSCF has two under certain situations. A better approach might be to add a counter in e.g. realm that not only indicates that IK should be used but also which IK. This could be e.g. a 2 bit field or similar. This is FFS]*~~

********** NEXT CHANGED SECTION **********

~~Annex E (informative):~~
~~Set-up procedures for SIP level based solution~~

~~[Editors Note: If the SIP level solution is chosen the chapters below shall be moved into the main body of this TS in the corresponding sections.] This chapter is based on chapter 7 and provides additional specification for the support of SIP level integrity protection].~~

~~For each incoming message the SIP application must verify that the correct inbound SA associated with the public ID (IMPU) given in the SIP message has been used.~~

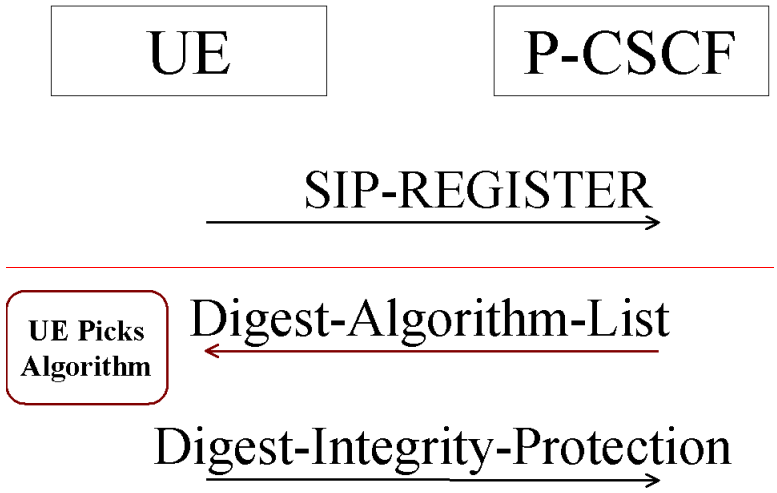
********** NEXT CHANGED SECTION **********

~~Annex F (informative):~~
~~Bidding down protection~~

~~This annex contains the Bidding Down Protection mechanism which is an extension to HTTP Digest i.e. [12]. The purpose with this Annex is to keep track on the development of the Bidding Down Protection and to have it as a fallback solution if Security Mode Setup is not available in time from IETF.~~

~~[Editors note: This text is FFS but it has to be further developed describing the mechanism in more detail. It is also FFS how to ensure that the UE picks the strongest algorithm and what algorithms should be mandatory.]~~

~~The extended HTTP Digest can negotiate what integrity algorithm to use. The general scheme is described in the figure below.~~



~~This security mode set-up looks different to the current requirements defined in clause 7 where the P-CSCF chooses the algorithm. A proposed mechanism for bidding down protection is to utilise a nonce, which will have a meaning for the client. The nonce-value in this case is not longer only a random number it will include the integrity algorithm and quality of protection along with the traditional nonce value. The nonce in this case could look like:~~

~~Nonce = base64 encoding (auth-algorithms, auth-extd-int, time-stamp || Hash(time-stamp, Request URI, private-key))~~

~~The server (in the IMS profile the server will be the P-CSCF) issues a list of supported mechanisms like e.g. MD5 and SHA-1. The client (in the IMS profile the client is the UE) picks the strongest algorithm it supports i.e. SHA-1 and protects the following messages with this algorithm. A man in the middle could not degrade the proposed list since the client shall repeat the nonce value which in this case includes the proposed list of algorithms as suggested above. The server or the P-CSCF can check that the list is correct but it does not have to store the suggested list.~~