3GPP TSG SA WG3 Security **S3-020093**

February 25 – February 28, 2002

Bristol, UK

Agenda Item: 7.3

Source: Ericsson

Title: A security framework for IMS utilising HTTP Digest

Document for: Discussion and decision

1. Scope and objectives

The scope of this document is to propose a framework based on Digest for:

1. Authenticating an IMS subscriber
2. Bidding Down protection
3. SIP signalling protection between the UE and the P-CSCF

The work in IETF for all these three mechanisms are either maturing or have been suggested by IETF as the way to go forward. Therefore taking the time constraints into account SA3 is encouraged to adopt bullet point number one and three as a working assumption and include it in the main body in [TS33203]. It is also proposed that SA3 adopts bullet point number two as a working assumption for a fallback solution and include a description of the mechanism in Annex E in [TS33203].

2 Introduction

SIP utilises HTTP Digest for authenticating SIP requests but due to several shortcomings of the existing HTTP Digest i.e. [RFC2617] IETF mandated James Undery from Ubiquity to drive a small team and define a better solution. A draft has been published at IETF, cf. [Undery], which tries to solve the known shortcomings.

The known shortcomings are e.g. that HTTP Digest

- fails to protect To, From, Call ID etc
- can not be used by the SIP UA to distinguish the source of authentication info headers in forking scenarios
- can not be used for proxy to UAS authentication

The extensions to HTTP Digest address these issues. The threats that the draft is currently addressing are:

- Replay attacks
- Man in the middle attacks i.e. trying to alter the sent message and bidding down attacks
- Masquerade attacks

Other threats like e.g. Privacy and Denial of service are not covered.

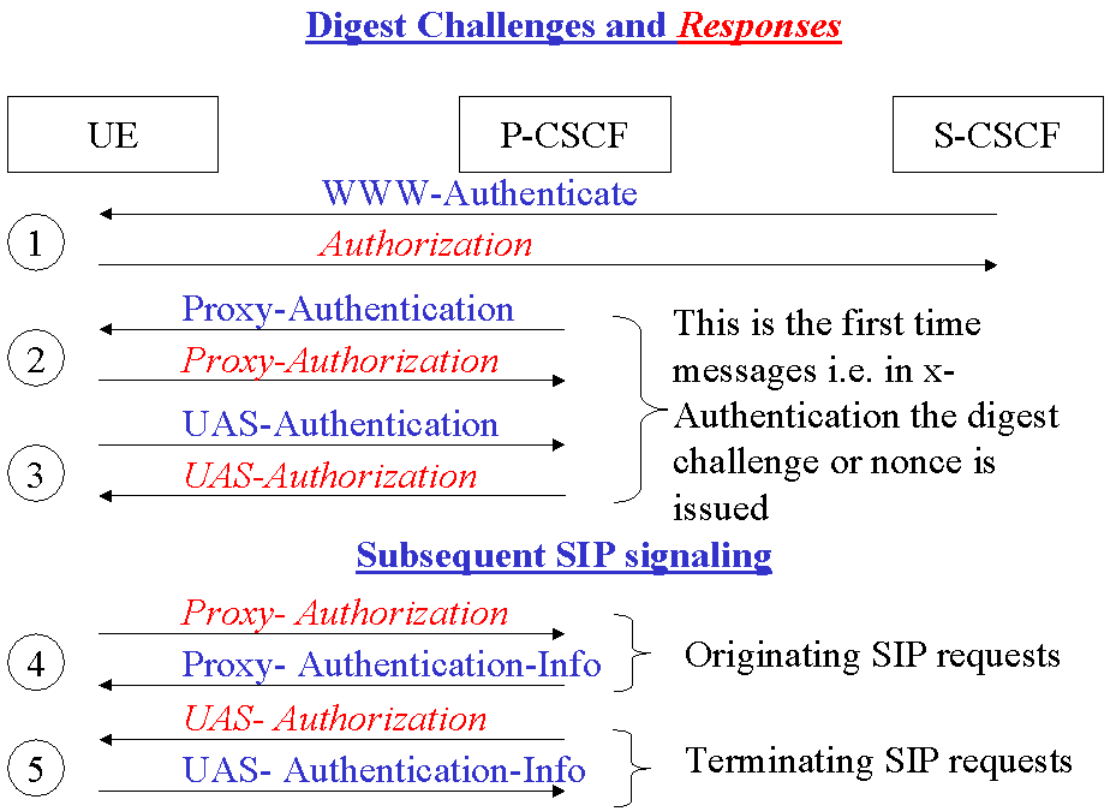
In order to utilise HTTP Digest both for originating and terminating requests a number of new headers have been defined in addition to the existing headers. All headers that applies are defined below:

- 401 WWW-Authenticate/407 Proxy-Authenticate/492 UAS-Authenticate (new)
- Authorization/Proxy-Authorization/UAS-Authorization (new)
- Authentication-Info/Proxy-Authentication-Info/UAS-Authentication-Info (new)

The qop-directive i.e. Quality of Protection indicates what part of the message is protection. There are now including the extensions four types of qop-directives:

- auth – authentication
- auth-int – authentication with message body integrity protection
- auth-extd-int – authentication with complete message integrity protection (*New*)
- auth-hdr-int – authentication and integrity protection of the message body and chosen headers (*New*)

It is assumed below that *the IMS profile* of the extended HTTP Digest need to use only auth-extd-int qop in Release 5 i.e. the whole SIP-message is protected.



The picture above shows at the highest level how the headers are utilised in HTTP Digest.

Case 1 and 2:

AKA challenge will be transported to the UE in WWW-Authenticated header and the AKA response (RES) to the S-CSCF in Authorization header. P-CSCF will add the Proxy-Authenticate and a Digest challenge (the so-called nonce) to the bypassing message. The UE will calculate the RES and add this into Authorization header. Furthermore the UE will calculate a MAC over the SIP-message and add that into Proxy-Authorization header to the P-CSCF.

Case 3:

Assume that the P-CSCF has sent an unprotected terminating INVITE towards the UE. It is unprotected since this is the first terminating request for the UE (i.e. SIP UAS). At this stage the UE sends a 492 Proxy Unauthorized response towards the P-CSCF and also includes a challenge i.e. a new nonce for the P-CSCF in the UAS-Authentication header. The P-CSCF takes the nonce and calculates a MAC and puts this into UAS-Authorization header and sends it back to the UE. The UE will check the MAC and if successful respond to the original INVITE message.

Case 4 and 5:

For these cases the SAs in both directions have been activated through 1,2 and 3. When the UE e.g. sends an originating INVITE a MAC in Proxy-Authorization header will protect it. The response to the INVITE is protected by the P-CSCF utilising Proxy-Authentication-Info. For a terminating INVITE the P-CSCF adds UAS-Authorization header along with a MAC which integrity protect the SIP message. And when the UE responds to the request, it adds UAS-Authentication-Info header and a MAC for this message. Note that the P-CSCF and the UE may issue fresh challenges (i.e. nonces) in Proxy-Authentication-Info and UAS-Authentication-Info headers.

3 Nonces, cnonces and nc

HTTP Digest was designed based on the client server model. However in SIP the UE may take both roles depending on whether it sends originating requests or terminating requests. In the originating scenario the UE acts as a User Agent Client (UAC), and in the terminating case it acts as a User Agent Server (UAS). This section discusses how SIP signalling can be protected between the UE and the P-CSCF with HTTP Digest.

A nonce is always generated by a UAS (e.g. UAS in UE or P-CSCF). As depicted in the figure in section 2 the nonce is carried in any of the Authenticate headers (i.e. WWW-Authenticate, Proxy-Authenticate or UAS-Authenticate) respectively. The nonce shall ideally be uniquely generated for each 401, 407 and 492 response. An attacker should not be able to predict this value. The HTTP Digest draft [RFC2617] proposes the following format for nonce even though the content of nonce is purely an implementation issue:

> Nonce = base64 encoding (time-stamp || Hash(time-stamp, Request URI, private-key))

The server generates the time-stamp and the server only knows the private key. By using the time stamp it is possible to limit the validity of the nonce and also to build stateless proxies. When the client responds to this it has to include the nonce in the MAC checksum.

The client shall also include a nonce-count (nc) to the response. Nc is a hexadecimal count of the number of requests that the client has sent using a particular nonce. As stated above, the nonce is updated for each 401, 407 and 492 response. If the nonce construction exemplified above is utilised along with the nc it will give replay protection. As already mentioned this is only an example and work has to be done to profile the exact mechanism for IMS.

When the first nonce value i.e. the digest challenge is sent towards the UAC, i.e. case 2 and 3 in the figure in section 2, a man in the middle could easily modify the nonce to a value of his choice i.e. a chosen plaintext attack. This attack could potentially reveal information about the session key to the attacker especially if the password is found from a dictionary. In order to resist this kind of attack the client shall utilise the cnonce. The cnonce is a value chosen as a random number by the client, which will affect the MAC value. The issued cnonce will be copied into the Authentication-Info credential by the UAS and sent with integrity protection to the client.

The value of having cnonce would then be, when the SA is active, that the client does not have to rely on how the nonce is generated by the server and if it is random enough or not. One alternative for IMS profiling would then be to keep the cnonce constant and only update it at e.g. 401 responses and when a next-nonce is chosen. This means that the client has to trust how the nonce is generated. The generation of nonces could be specified by SA3. The exact rule for this needs to be investigated further. It is envisioned that this work shall start as soon as possible if the extended HTTP Digest is accepted by SA3 as the working assumption. This work should be progressed by utilising change requests to [TS33203] when under change control.

4 IMS profile for Digest nonces

By IMS profile for Digest nonces, we mean potential 3GPP recommendations on how the IMS implementations should apply nonce, nc, cnonce and next-nonce parameters in HTTP Digest. HTTP Digest, [RFC 2617], is quite flexible in terms of content of these parameters even though some basic rules must be fulfilled in order to provide interoperability. The basic principal is that the content of nonce and cnonce are up to the implementation. This is possible because these parameters have meaning only to the entities, which issued them. Nonce is issued by UAS, and consequently it only has a meaning for UAS. Similarly, cnonce is issued by UAC, and it only has a meaning for itself. The rest of the rules are related to how these opaque parameters are handled by the other entity in the following way:

- UAS must include the cnonce to its response in *-Authenticate-Info headers.
- UAC must include the nonce or the next-nonce to its next request in *-Authorization headers.
- If UAC reuse an old nonce, it must increment nonce-count (nc) by one.

Note: Even though a nonce has only meaning to the server/proxy, there have been some new proposals on how the nonce could have a meaning to the client. Examples for such exceptions are the bidding down protection solution discussed in [Undery] and in chapter 5, and the Digest AKA solution discussed in [Niemi] and in chapter 5. These potential exceptions should be kept in mind when designing IMS profiles for Digest nonces.

SIP has some special features, which has not been taken into account in HTTP authentication. For example, [RFC2617] does not recognize the fact that the same SIP entity in fact includes both a UAC and UAS, and could consequently share some information for authentication. This is because HTTP entity is always a client or a server – but never both of them at the same time. Unfortunately, SIP as a standard has not specified any additional rule on how HTTP authentication should be optimized in the situations in which the client and the server are in the same entity. In practice, this means that two SIP entities sharing a security association through HTTP authentication should maintain two nonce values, two cnonce values and two nonce-counters (nc) for each security

association. However, this is not effective and efficient way to maintain a security association because of the various nonce related variables. Because HTTP Digest is flexible on the content of nonce values, it is beneficial to define more optimized profile for them. This profiling is possible to define in 3GPP, however, it is also possible the SIP working group in IETF would give some guidance on the issue.

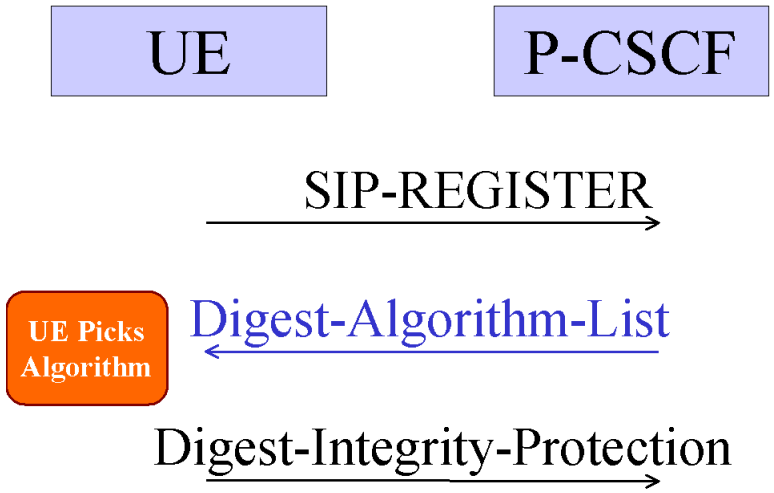
A preliminary list of design requirements for the IMS profile for Digest nonces is as follows:

- IMS profile for Digest nonces should optimized nonce related rules for communication between a UE and a proxy/server which share a security association for both originating and terminating messages.
- IMS profile for Digest nonces must not assume that all security associations are bi-directional. There may exist use scenarios in which the security association is unidirectional, cf. [RFC2617].
- The UE must be able to use the same IMS profile with all standard SIP implementations. In particular, the solution must not assume that UE includes two different rules for generating nonces; one for IMS and another for non-3GPP networks.
- IMS Digest profile must not cause any additional security risks when used with non-3GPP specific networks.
- UAS should be able to distinguish the freshness of the nonce it has issued.
- UAC must not trust that a nonce received in a digest challenge is secure. UAC must use cnonce to ‘challenge’ the UAS.
- It is not necessary for UAC to remember all those cnonces, which are already used. UAC must be able to remember those cnonces, which have not been replied by the UAS.
- UAC may keep the cnonce constant if the same nonce value is used with incremented nonce-counter. UAC must change the cnonce every time it receives a new nonce or next-nonce value.

5 Digest AKA and Bidding Down attack

In Digest AKA the WWW-Authenticate will include the RAND and the AUTN as defined by IMS AKA. The UE by using the RAND derives the session keys and the RES, which in HTTP Digest terminology could act as a password. If the received SQN is not within the acceptable range a synchronisation failure will occur. In that case the UE shall use the new authentication parameter auts and include AUTS in this field in the Authorisation header, cf. [Niemi].

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 working assumption in [TS33203] 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, cf. [Undery]. 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 as exemplified in section 2. 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. This gives another advantage to the existing working assumption in [TS33203] that the P-CSCF becomes stateless.

SA3 should investigate if it could be possible to accept that the UE decides on which algorithm to use.

Conclusions

This contribution has described the principles behind the extensions to HTTP Digest and the use of nonce, cnonce and nonce counter. Some potential profiling of HTTP Digest has been identified, which could apply to the IMS specific architecture. A brief description of Digest AKA was given along with the mechanism for resisting bidding down attacks in HTTP Digest.

It is proposed that SA3 adopts the framework described in this contribution as a working assumption i.e. adopts that Digest be utilised for:

- Authenticating an IMS Subscriber i.e. Digest AKA as described in [Niemi]
- Bidding down protection as a fallback solution for security mode set-up and include the current solution in [Undery] in Annex E of [TS33203].
- Integrity protection of SIP signalling

If this is accepted by SA3 Ericsson volunteers to work on profiling HTTP Digest for the IMS architecture. Ericsson also hopes that SA3 and the supporting companies of the work item would contribute on profiling HTTP Digest for IMS. Ericsson also suggests that the requirements in Section 4 be adopted as current working assumption for profiling Digest nonces in IMS.

It is also proposed that SA3 investigates if it could be acceptable to allow the UE to select algorithm(s) in general. Ericsson believes that such a construction could be equally secure as the current working assumption.

If this is acceptable to SA3 Ericsson will continue investigate the progress and the feasibility of Bidding Down protection as being specified in [Undery].

References

[Niemi] IETF Networking Group, *HTTP Digest Authentication Using AKA*, draft-niemi-sip-digest-aka-00, February 2002

[Undery] IETF *SIP Digest Authentication: Extensions to HTTP Digest Authentication*, draft-undery-sip-auth-00, January 2002

[RFC2617] IETF Network Working Group *HTTP Authentication: Basic and Digest Access Authentication*, RFC 2617, June 1999.

[TS33203] 3G TS 33.203: "3rd Generation Partnership Project (3GPP); Technical Specification Group (TSG) SA3; Access Security for IP-based services".