

25 - 28 February 2002

Bristol, UK

Source: Nokia

Title: HTTP AKA Internet Draft

Document for: Discussion and decision

Agenda Item: 7.3, IP multimedia subsystem security

1. Scope and objectives

This document presents the latest submitted version of HTTP AKA Internet Draft (draft-niemi-sipping-digest-aka-00). The draft is attached.

The rationale of this work is to specify an AKA based password generation mechanism for HTTP Digest access authentication. Thus, HTTP AKA re-uses the Digest mechanism based on shared secret. Re-use guarantees that HTTP AKA is a workable solution since HTTP Digest is a proven mechanism. It will be therefore much more quickly to be approved in IETF.

The work accomplishes the conclusion from 3GPP-IETF co-ordination meeting on January 25, 2002, which is reflected in [S3z020009].

The Draft will be presented and discussed in the next IETF meeting (Minneapolis, 17th –22nd , 2002).

2. Reference

[S3z020009] Results of a conference call with IETF ADs and SIP WG chairs regarding IMS security, Ericsson. S3#21bis, Antwerp.

Network Working Group
Internet-Draft
Expires: August 22, 2002

A. Niemi
Nokia
J. Arkko
V. Torvinen
Ericsson
February 21, 2002

HTTP Digest Authentication Using AKA
draft-niemi-sipping-digest-aka-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 22, 2002.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

The Hypertext Transfer Protocol (HTTP) Authentication Framework includes two authentication schemes: Basic and Digest. Both schemes employ a shared secret based mechanism for access authentication. The Authentication and Key Agreement (AKA) mechanism performs user authentication and session key distribution in Universal Mobile Telecommunications System (UMTS) networks. AKA is a challenge-response based mechanism that uses symmetric cryptography. This memo specifies an AKA based password generation mechanism for HTTP Digest access authentication.

Table of Contents

1.	Introduction and Motivation	3
1.1	Terminology	3
1.2	Conventions	4
2.	AKA Mechanism Overview	5
3.	Specification of Digest AKA	7
3.1	Algorithm Directive	7
3.2	Creating a Challenge	7
3.3	Client Authentication	8
3.4	Synchronization Failure	9
3.5	Server Authentication	9
4.	Example Digest AKA Operation	10
5.	Security Considerations	12
5.1	Authentication of Clients using Digest AKA	12
5.2	Limited Use of Nonce Values	12
5.3	Multiple Authentication Schemes and Algorithms	13
5.4	Online Dictionary Attacks	13
5.5	Session Protection	13
5.6	Replay Protection	14
5.7	Improvements to AKA Security	14
6.	IANA Considerations	15
	References	16
	Authors' Addresses	16
A.	Acknowledgements	18
	Full Copyright Statement	19

1. Introduction and Motivation

The Hypertext Transfer Protocol (HTTP) Authentication Framework described in RFC 2617 [2] includes two authentication schemes: Basic and Digest. Both schemes employ a shared secret based mechanism for access authentication. The Basic scheme is inherently insecure in that it transmits user credentials in plain text. The Digest scheme improves security by hiding user credentials with cryptographic hashes, and additionally by providing limited message integrity.

The Authentication and Key Agreement (AKA) [8] mechanism performs authentication and session key distribution in Universal Mobile Telecommunications System (UMTS) networks. AKA is a challenge-response based mechanism that uses symmetric cryptography. AKA is typically run in a UMTS IM Services Identity Module (ISIM), which resides on a smart card like device that also provides tamper proof storage of shared secrets.

This document specifies a mapping of AKA parameters onto HTTP Digest authentication. In essence, this mapping enables the usage of AKA as a password generation mechanism for Digest authentication, thus meeting the requirements set in [6] and [5].

As the Session Initiation Protocol (SIP) [4] Authentication Framework closely follows the HTTP Authentication Framework, Digest AKA is directly applicable to SIP as well as any other embodiment of HTTP Digest.

1.1 Terminology

This chapter explains the terminology used in this document.

AKA

Authentication and Key Agreement.

AuC

Authentication Center. The network element in mobile networks that can authorize users either in GSM or in UMTS networks.

AUTN

Authentication Token. A 128 bit value generated by the AuC, which together with the RAND parameter authenticates the server to the client.

AUTS

Authentication Token. A 112 bit value generated by the client upon experiencing an SQN synchronization failure.

CK

Cipher Key. An AKA session key for encryption.

IK

Integrity Key. An AKA session key for integrity check.

ISIM

IM Services Identity Module.

PIN

Personal Identification Number. Commonly assigned passcodes for use with automatic cash machines, smart cards, etc.

RAND

Random Challenge. Generated by the AuC using the SQN.

RES

Authentication Response. Generated by the ISIM.

SIM

Subscriber Identity Module. GSM counter part for ISIM.

SQN

Sequence Number. Both AuC and ISIM maintain the value of the SQN.

UMTS

Universal Mobile Telecommunications System.

XRES

Expected Authentication Response. This is equal to RES.

1.2 Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

2. AKA Mechanism Overview

This chapter describes the AKA operation in detail:

1. A shared secret K is established beforehand between the ISIM and the Authentication Center (AuC). The secret is stored in the ISIM, which resides on a smart card like, tamper proof device.
2. The AuC of the home network produces an authentication vector AV based on the shared secret K and a sequence number SQN. The authentication vector contains a random challenge RAND, network authentication token AUTN, expected authentication result XRES, a session key for integrity check IK, and a session key for encryption CK.
3. The authentication vector is downloaded to a server. Optionally, the server can also download a batch of AVs, containing more than one authentication vector.
4. The server creates an authentication request, which contains the random challenge RAND, and the network authenticator token AUTN.
5. The authentication request is delivered to the client.
6. Using the shared secret K and the sequence number SQN, the client verifies the AUTN with the ISIM. If the verification is successful, the network has been authenticated. The client then produces an authentication response RES, using the shared secret and the random challenge RAND.
7. The authentication response is delivered to the server.
8. The server compares the authentication response RES with the expected response XRES. If the result is correct, the user has been successfully authenticated, and the session keys IK and CK can be used for protecting further communications between the client and the server.

When verifying the AUTN, the client may detect, that the sequence numbers between the client and the server have fallen out of sync. In this case, the client produces a synchronization parameter AUTS, using the shared secret K and the client SQN. The AUTS parameter is delivered to the network in the authentication response, and the authentication can be tried again based on authentication vectors generated with the synchronized sequence number.

For a specification of the AKA mechanism and the generation of the cryptographic parameters AUTN, RES, IK, CK, and AUTS, see reference

3GPP TS 33.102 [8].

3. Specification of Digest AKA

In general, Digest AKA operation is identical to Digest operation in RFC 2617 [2]. This chapter specifies the parts, in which Digest AKA extends the Digest operation.

3.1 Algorithm Directive

In order to direct the client into using AKA for authentication, instead of the standard password system, the RFC 2617 defined algorithm directive is extended:

```

algorithm          = "algorithm" EQUAL aka-namespace
aka-namespace      = aka-version "-" algorithm-value
aka-version        = "AKAv" 1*DIGIT
algorithm-value    = ( "MD5" / "MD5-sess" / token )

```

algorithm

A string indicating the algorithm used in producing the digest and the checksum. If the directive is not understood, the nonce SHOULD be ignored, and another challenge (if one is present) should be used instead. The default aka-version is "AKAv1". Further AKA versions can be specified, with version numbers assigned by IANA [9]. When the algorithm directive is not present, it is assumed to be "MD5". This indicates, that AKA is not used to produce the password.

Example:

```
algorithm=AKAv1-MD5
```

3.2 Creating a Challenge

In order to deliver the AKA authentication challenge to the client in Digest AKA, the nonce directive defined in RFC 2617 is extended:

```

nonce              = "nonce" EQUAL aka-nonce
aka-nonce          = LDQUOT aka-nonce-value RDQUOT
aka-nonce-value    = <base64 encoding of RAND + AUTN +
                    server specific data>

```

nonce

calculating the response directive of RFC 2617.

3.4 Synchronization Failure

For indicating an AKA sequence number synchronization failure, and to re-synchronize the SQN in the AuC using the AUTS token, a new directive is defined for the "digest-response" of the "Authorization" request header defined in RFC 2617:

```
auts           = "auts" EQUAL auts-param
auts-param     = LDQUOTE auts-value RDQUOTE
auts-value     = <base64 encoding of AUTS>
```

auts

A string carrying a base64 encoded AKA AUTS parameter. This directive is used to re-synchronize the server side SQN. If the directive is present, the client doesn't use any password when calculating its credentials. Instead, the client MUST calculate its credentials using an empty password (password of "").

Example:

```
auts="5ccc069c403ebaf9f0171e9517f40e41"
```

Upon receiving the "auts" parameter, the server will check the validity of the parameter value using the shared secret K. A valid AUTS parameter is used to re-synchronize the SQN in the AuC. The synchronized SQN is then used to generate a fresh authentication vector AV, with which the client is then re-challenged.

3.5 Server Authentication

Even though AKA provides inherent mutual authentication with the AKA AUTN token, mutual authentication mechanisms provided by Digest may still be useful in order to provide message integrity.

In Digest AKA, the server uses the AKA XRES parameter as "password" when calculating the "response-auth" of the "Authentication-Info" header defined in RFC 2617.

4. Example Digest AKA Operation

Figure 2 below describes a message flow describing a Digest AKA process of authenticating a SIP request, namely the SIP REGISTER request.

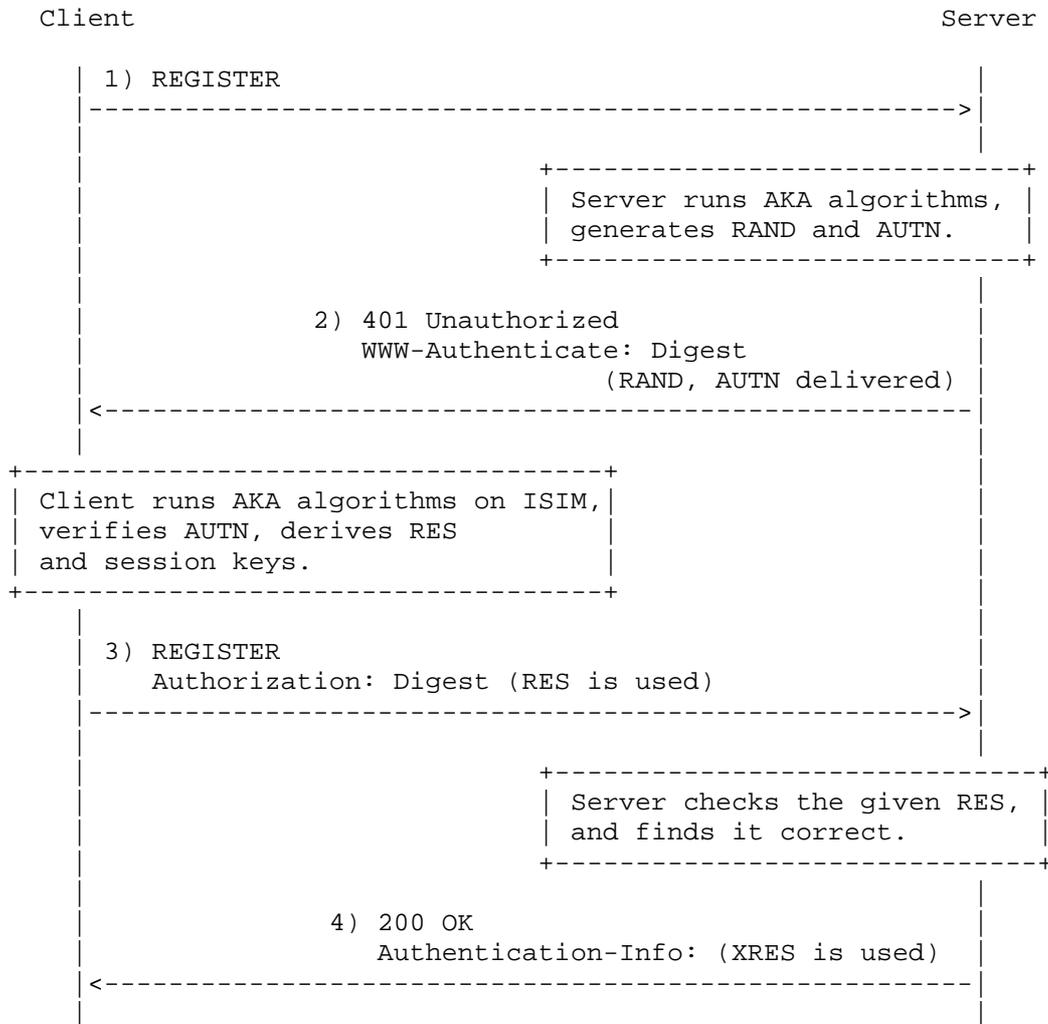


Figure 2: Message flow representing a successful authentication.

1) Initial request

REGISTER sip:home.mobile.biz SIP/2.0

2) Response containing a challenge

```
SIP/2.0 401 Unauthorized
WWW-Authenticate: Digest
    realm="RoamingUsers@mobile.biz",
    qop="auth-int",
    nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
    opaque="5ccc069c403ebaf9f0171e9517f40e41",
    algorithm=AKAv1-MD5
```

3) Request containing credentials

```
REGISTER sip:home.mobile.biz SIP/2.0
Authorization: Digest
    username="jon.dough@mobile.biz",
    realm="RoamingUsers@mobile.biz",
    nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
    uri="sip:home.mobile.biz",
    qop=auth-int,
    nc=00000001,
    cnonce="0a4f113b",
    response="6629fae49393a05397450978507c4ef1",
    opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

4) Successful response

```
SIP/2.0 200 OK
Authentication-Info:
    qop=auth-int,
    rspauth="6629fae49393a05397450978507c4ef1",
    cnonce="0a4f113b",
    nc=00000001
```

5. Security Considerations

In general, Digest AKA is vulnerable to the same security threats as HTTP authentication [2]. This chapter discusses the relevant exceptions.

5.1 Authentication of Clients using Digest AKA

AKA is typically -- though this isn't a theoretical limitation -- run on an ISIM application that usually resides in a tamper resistant smart card. Interfaces to the ISIM exist, which enable the host device to request authentication to be performed on the card. However, these interfaces do not allow access to the long-term secret outside the ISIM, and the authentication can only be performed if the device accessing the ISIM has knowledge of a PIN code, shared between the user and the ISIM. Such PIN codes are typically obtained from user input, and are usually required when the device is powered on.

The use of tamper resistant cards with secure interfaces implies that Digest AKA is typically more secure than regular Digest implementations, as neither possession of the host device nor Trojan Horses in the software give access to the long term secret. Where a PIN scheme is used, the user is also authenticated when the device is powered on. However, there may be a difference in the resulting security of Digest AKA, compared to traditional Digest implementations, depending of course on whether those implementations cache/store passwords that are received from the user.

5.2 Limited Use of Nonce Values

The Digest scheme uses server-specified nonce values to seed the generation of the request-digest value. The server is free to construct the nonce in such a way, that it may only be used from a particular client, for a particular resource, for a limited period of time or number of uses, or any other restrictions. Doing so strengthens the protection provided against, for example, replay attacks.

Digest AKA limits the applicability of a nonce value to a particular ISIM. Typically, the ISIM is accessible only to one client device at a time. However, the nonce values are strong and secure even though limited to a particular ISIM.

A server may allow each nonce value to be used only once by sending a next-nonce directive in the Authentication-Info header field of every response. However, this may cause a synchronization failure, and consequently some additional round trips in AKA, if the same SQN space is also used for other access schemes at the same time.

5.3 Multiple Authentication Schemes and Algorithms

In HTTP authentication, a user agent **MUST** choose the strongest authentication scheme it understands and request credentials from the user based upon that challenge.

In general, using passwords generated by Digest AKA with other HTTP authentication schemes is not recommended even though the realm values or protection domains would coincide. In these cases, a password should be requested from the end-user instead. Digest AKA passwords **MUST NOT** be re-used with such HTTP authentication schemes, which send the password in clear. In particular, AKA passwords **MUST NOT** be re-used with HTTP Basic.

The same principle must be applied within a scheme if several algorithms are supported. A client receiving an HTTP Digest challenge with several available algorithms **MUST** choose the strongest algorithm it understands. For example, Digest with "AKAv1-MD5" would be stronger than Digest with "MD5".

5.4 Online Dictionary Attacks

Since user-selected passwords are typically quite simple, it has been proposed that servers should not accept passwords for HTTP Digest, which are in the dictionary [2]. This potential threat does not exist in HTTP Digest AKA because the algorithm will use ISIM originated passwords. However, the end-user must still be careful with PIN codes. Even though HTTP Digest AKA password requests are never displayed to the end-user, she will be authenticated to the ISIM via a PIN code. Commonly known initial PIN codes are typically installed to the ISIM during manufacturing and if the end-users do not change them, there is a danger that an unauthorized user may be able to use the device. Naturally this requires that the unauthorized user has access to the physical device, and that the end-user has not changed the initial PIN code. For this reason, end-users are strongly encouraged to change their PIN codes when they receive an ISIM.

5.5 Session Protection

Digest AKA is able to generate additional session keys for integrity (IK) and confidentiality (CK) protection. Even though this document does not specify the use of these additional keys, they may be used for creating additional security within HTTP authentication or some other security mechanism.

5.6 Replay Protection

Optionally, AKA allows sequence numbers to be tracked for each authentication, with the SQN parameter. This allows authentications to be replay protected even if the RAND parameter happened to be the same for two authentication requests. More importantly, this offers additional protection for the case where an attacker replays an old authentication request sent by the network. The client will be able to detect that the request is old, and refuse authentication. This proves liveness of the authentication request even in the case where a MitM attacker tries to trick the client into providing an authentication response, and then replaces parts of the message with something else. In other words, a client challenged by Digest AKA is not vulnerable for chosen plain text attacks. Finally, frequent sequence number errors would reveal an attack where the tamper-resistant card has been cloned and is being used in multiple devices.

The downside of sequence number tracking is that servers must hold more information for each user than just their long-term secret, namely the current SQN value. However, this information is typically not stored in the SIP nodes, but in dedicated authentication servers instead.

5.7 Improvements to AKA Security

Even though AKA is perceived as a secure mechanism, Digest AKA is able to improve it. More specifically the AKA parameters carried between the client and the server during authentication may be protected along with other parts of the message, by using using Digest AKA. This is not possible with plain AKA.

6. IANA Considerations

This document specifies a namespace for the AKA versions in Section 3.1, with the default version being "AKAv1". Following the policies outlined in [3], versions above 1 are allocated as Expert Review.

References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [2] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
- [3] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [4] Rosenberg, J. and H. Schulzrinne, "SIP: Session Initiation Protocol", draft-ietf-sip-rfc2543bis-07 (work in progress), February 2002.
- [5] Garcia, M., "3GPP requirements on SIP", draft-garcia-sipping-3gpp-reqs-02 (work in progress), November 2001.
- [6] Arkko, J., "3GPP Requirements for SIP Authentication", draft-uusitalo-sipping-authentication-00 (work in progress), February 2002.
- [7] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [8] 3rd Generation Partnership Project, "Security Architecture (Release 4)", TS 33.102, December 2001.
- [9] <http://www.iana.org>, "Assigned Numbers", February 2002.

Authors' Addresses

Aki Niemi
Nokia
P.O. Box 301
NOKIA GROUP, FIN 00045
Finland

Phone: +358 50 389 1644
EMail: aki.niemi@nokia.com

Jari Arkko
Ericsson
Hirsalantie 1
Jorvas, FIN 02420
Finland

Phone: +358 40 5079256
EMail: jari.arkko@ericsson.com

Vesa Torvinen
Ericsson
Joukahaisenkatu 1
Turku, FIN 20520
Finland

Phone: +358 40 7230822
EMail: vesa.torvinen@ericsson.fi

Appendix A. Acknowledgements

The authors would like to thank Sanjoy Sen, Jonathan Rosenberg, Pete McCann, Tao Haukka, Ilkka Uusitalo, Henry Haverinen, John Loughney and Allison Mankin.

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

