Network Working Group
Request for Comments: 4169
Category: Informational

V. Torvinen
Turku Polytechnic
J. Arkko
M. Naslund
Ericsson

November 2005

Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA) Version-2

Status of This Memo

Copyright Notice

Abstract

HTTP Digest, as specified in RFC 2617, is known to be vulnerable to man-in-the-middle attacks if the client fails to authenticate the server in TLS, or if the same passwords are used for authentication in some other context without TLS. This is a general problem that exists not just with HTTP Digest, but also with other IETF protocols that use tunneled authentication. This document specifies version 2 of the HTTP Digest AKA algorithm (RFC 3310). This algorithm can be implemented in a way that it is resistant to the man-in-the-middle attack.

Table of Contents

1. Introduction 2
1.1. Terminology 4
2. HTTP Digest AKAv2 5
2.1. Password generation 6
2.2. Session keys 6
3. Example Digest AKAv2 Operation 7
4. Security Considerations 7
4.1. Multiple Authentication Schemes and Algorithms 7
4.2. Session Protection 7
4.3. Man-in-the-middle attacks 8
4.4. Entropy 9
5. IANA Considerations 10
5.1. Registration Information 10
6. References 11
6.1. Normative References 11
6.2. Informative References 11

1. Introduction

The Hypertext Transfer Protocol (HTTP) Digest Authentication, described in [4], has been extended in [6] to support the Authentication and Key Agreement (AKA) mechanism [7]. The AKA mechanism performs authentication and session key agreement in Universal Mobile Telecommunications System (UMTS) networks. HTTP Digest AKA enables the usage of AKA as a one-time password generation mechanism for Digest authentication.

HTTP Digest is known to be vulnerable to man-in-the-middle attacks, even when run inside TLS, if the same HTTP Digest authentication credentials are used in some other context without TLS. The attacker may initiate a TLS session with a server, and when the server challenges the attacker with HTTP Digest, the attacker masquerades the server to the victim. If the victim responds to the challenge, the attacker is able to use this response towards the server in HTTP Digest. Note that this attack is an instance of a general attack that affects a number of IETF protocols, such as PIC. The general problem is discussed in [8] and [9].

Because of the vulnerability described above, the use of HTTP Digest "AKAv1" should be limited to the situations in which the client is able to demonstrate that, in addition to the AKA response, it possesses the AKA session keys. This is possible, for example, if the underlying security protocol uses the AKA-generated session keys to protect the authentication response. This is the case, for example, in the 3GPP IP Multimedia Core Network Subsystem (IMS), where HTTP Digest "AKAv1" is currently applied. However, HTTP Digest

"AKAv1" should not be used with tunnelled security protocols that do not utilize the AKA session keys. For example, the use of HTTP Digest "AKAv1" is not necessarily secure with TLS if the server side is authenticated using certificates and the client side is authenticated using HTTP Digest AKA.

There are at least four potential solutions to the problem:

1. The use of the authentication credentials is limited to one application only. In general, this approach is good and can be recommended from the security point of view. However, this will increase the total number of authentication credentials for an end-user, and may cause scalability problems in the server side.

2. The keys used in the underlying security protocols are somehow bound to the keys used in the tunneled authentication protocol. However, this would cause problems with the current implementations of underlying security protocols. For example, it is not possible to use the session keys from TLS at the application layer. Furthermore, this solution would only solve the problem when HTTP Digest is used over one hop, and would leave the problem of using HTTP Digest via multiple hops (e.g., via proxy servers) unsolved.

3. Authentication credentials are used in a cryptographically different way for each media and/or access network. However, it may be difficult to know which underlying media is used below the application.

4. Authentication credentials are used in a cryptographically different way for each application.

This document specifies a new algorithm version for HTTP Digest AKA (i.e., "AKAv2"). "AKAv2" specifies a cryptographically different way to use AKA credentials in use cases that are based on either HTTP Digest authentication or UMTS authentication (cf. approach 4 above). The only difference to "AKAv1" is that, in addition to an AKA response RES, the AKA related session keys, IK and CK, are also used as the password for HTTP Digest. AKAv2 is immune to the man-in-the-middle attack described above. However, if AKAv2 is used in some environment, both with and without some underlying security, such as TLS, the problem still exists.

New HTTP Digest AKA algorithm versions can be registered with IANA, based on Expert Review. Documentation of new algorithm versions is not mandated as RFCs. However, "AKAv2" is documented as an RFC because the use of different AKA algorithm versions includes security implications of which the implementors should be aware. The

extension version and security implications are presented in this document.

1.1. Terminology

This chapter explains the terminology used in this document.

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 [3].

AKA

Authentication and Key Agreement.

AKA is a challenge-response based mechanism that uses symmetric cryptography. AKA can be run in a UMTS IM Services Identity Module (ISIM) or in UMTS Subscriber Identity Module (USIM), which reside on a smart-card-like device that also provides tamper resistant storage of shared secrets.

CK

Cipher Key. An AKA session key for encryption.

CK'

Cipher Key. HTTP Digest AKAv2 session key for encryption. CK' is derived from CK using a pseudo-random function.

IK

Integrity Key. An AKA session key for integrity check.

IK'

Integrity Key. HTTP Digest AKAv2 session key for integrity check. IK' is derived from IK using a pseudo-random function.

ISIM

IP Multimedia Services Identity Module. Sometimes ISIM is implemented using USIM.

RES

Authentication Response. Generated by the ISIM.

PRF

Pseudo-random function that is used to construct the AKAv2 password and related session keys IK' and CK'. In this document, PRF is presented in the format KD(secret, data), denoting a keyed digest algorithm (KD) performed to the data ("data") with the secret ("secret").

SIM

Subscriber Identity Module. GSM counter part for ISIM and USIM.

UMTS

Universal Mobile Telecommunications System.

USIM

UMTS Subscriber Identity Module. UMTS counter part for ISIM and SIM.

XRES

Expected Authentication Response. In a successful authentication, this is equal to RES.

2. HTTP Digest AKAv2

In general, the Digest AKAv2 operation is identical to the Digest AKAv1 operation described in [6]. This chapter specifies the parts in which Digest AKAv2 is different from Digest AKAv1 operation. The notation used in the Augmented BNF definitions for the new and modified syntax elements in this section is as used in SIP [5], and any elements not defined in this section are as defined in [6].

In order to direct the client into using AKAv2 for authentication instead of other AKA versions or other HTTP Digest algorithms, the AKA version directive of [6] shall have the following new value:

```
aka-version         =  "AKAv2"
```

The AKA version directive is used as a part of the algorithm field as defined in [6].

Example: algorithm=AKAv2-MD5

2.1. Password Generation

The client shall use base64 encoded [1] parameters PRF(RES||IK||CK, "http-digest-akav2-password") as a "password" when calculating the HTTP Digest response directive for AKAv2.

The server shall use base64 encoded [1] parameters PRF(XRES||IK||CK, "http-digest-akav2-password") as a "password" when checking the HTTP Digest response or when calculating the "response-auth" of the "Authentication-Info" header.

The pseudo-random function (PRF) used to construct the HTTP Digest password is equal to HMAC [2] using the hash algorithm that is used in producing the digest and the checksum. For example, if the algorithm is AKAv2-MD5, then the PRF is HMAC_MD5.

The string "http-digest-akav2-password" included in the key derivation is case sensitive.

2.2. Session keys

Even though the HTTP Digest AKA framework does not specify the use of the session keys IK and CK for confidentiality and integrity protection, the keys may be used for creating additional security within HTTP authentication or some other security mechanism. However, the original session keys IK and CK MUST NOT be directly re-used for such additional security in "AKAv2". Instead, session keys IK' and CK' are derived from the original keys IK and CK in the following way:

IK' = PRF(IK, "http-digest-akav2-integritykey")

CK' = PRF(CK, "http-digest-akav2-cipherkey")

Any application using the HTTP authentication framework is allowed to use these masked session keys. The unmasked session keys MAY also be re-used in some other context if application-specific strings other than "http-digest-akav2-integritykey" or "http-digest-akav2-cipherkey" are used to mask the original session keys.

The pseudo-random function (PRF) used to construct the HTTP Digest session keys is equal to HMAC [2] using the hash algorithm that is used in producing the digest and the checksum. For example, if the algorithm is AKAv2-MD5, then the PRF is HMAC_MD5. The algorithm MUST be used in the HMAC format, as defined in [2].

The strings "http-digest-akav2-integritykey" and "http-digest-akav2-cipherkey" included in the key derivation are case sensitive.

3. Example Digest AKAv2 Operation

This document does not introduce any changes to the operations of HTTP Digest or HTTP Digest AKA. Examples defined in [6] apply directly to AKAv2 with the following two exceptions:

1. The algorithm directive has a prefix "AKAv2" instead of "AKAv1".

2. The HTTP Digest password is derived from base64 encoded PRF(RES||IK||CK, "http-digest-akav2-password") or PRF(XRES||IK||CK, "http-digest-akav2-password") instead of (RES) or (XRES) respectively.

3. The optional session keys are derived from PRF(IK, "http-digest-akav2-integritykey") and PRF(CK, "http-digest-akav2-cipherkey") instead of IK and CK respectively.

Note that the password in "AKAv1" is in binary format. The "AKAv2" password is base64 encoded [1].

4. Security Considerations

4.1. Multiple Authentication Schemes and Algorithms

The rules for a user agent for choosing among multiple authentication schemes and algorithms are as defined in [6], except that the user agent MUST choose "AKAv2" if both "AKAv1" and "AKAv2" are present.

Since HTTP Digest is known to be vulnerable for bidding-down attacks in environments where multiple authentication schemes and/or algorithms are used, the system implementors should pay special attention to scenarios in which both "AKAv1" and "AKAv2" are used. The use of both AKA algorithm versions should be avoided, especially if the AKA generated sessions keys or some other additional security measures to authenticate the clients (e.g., client certificates) are not used.

4.2. Session Protection

Even though "AKAv2" uses the additional integrity (IK) and confidentiality (CK) keys as a part of the HTTP Digest AKA password, these session keys may still be used for creating additional security within HTTP authentication or some other security mechanism. This recommendation is based on the assumption that algorithms used in HTTP Digest, such as MD5, are sufficiently strong one-way functions, and, consequently, HTTP Digest responses leak no or very little

computational information about IK and CK. Furthermore, the session keys are masked into IK' and CK' before they can be used for session protection.

4.3. Man-in-the-Middle Attacks

Reference [8] describes a "man-in-the-middle" attack related to tunnelled authentication protocols. The attack can occur in an EAP context or any similar contexts where tunnelled authentication is used and where the same authentication credentials are used without protection in some other context or the client fails to authenticate the server.

For example, the use of TLS with HTTP Digest authentication (i.e., TLS for server authentication, and subsequent use of HTTP Digest for client authentication) is an instance of such scenario. HTTP challenges and responses can be fetched from and to different TLS tunnels without noticing their origin. The attack is especially easy to perform if the client fails to authenticate the server. If the same HTTP credentials are used with an unsecured connection, the attack is also easy to perform.

This is how the "man-in-the-middle" attack works with HTTP Digest and TLS if the victim (i.e., the client) fails to authenticate the server:

1. The victim contacts the attacker using TLS. If the attacker has a valid server certificate, the client may continue talking to the attacker and use some HTTP authentication compatible protocol, such as the Session Initiation Protocol (SIP).

2. The attacker contacts a real proxy/server also using TLS and an HTTP-authentication-compatible protocol. The proxy/server responds to the attacker with the HTTP Authentication challenge.

3. The attacker forwards the HTTP Authentication challenge from the proxy/server to the victim. If the victim is not careful, and does not check whether the identity in the server certificate in TLS matches the realm in the HTTP authentication challenge, it may send a new request that carries a valid response to the HTTP Authentication challenge.

4. The attacker may use the response with the victims HTTP Digest username and password to authenticate itself to the proxy/server.

The man-in-the-middle attack is not possible if the client compares the identities in the TLS server certificate and the HTTP Digest authentication challenge. Note that with HTTP Basic, the client would send the password to the attacker.

Another variant of the "man-in-the-middle" attack is the so-called "interleaving attack". This attack is possible if the HTTP Digest authentication credentials are used in several contexts, and in one of them without protection.

This is how the attack could proceed:

1. The attacker establishes a TLS tunnel to the proxy/server using one-way server authentication. The attacker sends a request to the proxy/server.

2. The proxy/server challenges the attacker with the HTTP Digest challenge.

3. The attacker challenges the victim in some other context using the challenge carried in the HTTP Digest challenge. The HTTP Digest challenge needs to be modified to the format used in the protocol of this other context.

4. The victim responds with a response.

5. The attacker uses the response from the other context for authentication in HTTP Digest.

6. The proxy/server accepts the response, and delivers the service to the attacker.

In some circumstances, HTTP Digest AKAv1 may be vulnerable for the interleaving attack. In particular, if ISIM is implemented using USIM, the HTTP Digest AKAv1 should not be used with tunneled security protocols unless the AKA-related session keys, IK and CK, are somehow used with the solution.

HTTP Digest AKAv2 is not vulnerable to this interleaving attack, and it can be used with tunneled security protocols without using the related AKA session keys.

4.4. Entropy

AKAv1 passwords should only be used as one-time passwords if the entropy of the used RES value is limited (e.g., only 32 bits). For this reason, the re-use of the same RES value in authenticating subsequent requests and responses is not recommended. Furthermore,

algorithms such as "MD5-sess", which limit the amount of material hashed with a single key by producing a session key for authentication, should not be used with AKAv1.

Passwords generated using AKAv2 can more securely be used for authenticating subsequent requests and responses because the concatenation of AKA credentials (i.e., RES||IK||CK) makes the passwords significantly longer, and the pseudo-random function heuristically provides an entropy equal to the length of this string, or the length of the PRF output, whichever is the shortest. The user agent does not need to assume that AKAv2 passwords are limited to one-time use only, and it may try to re-use the AKAv2 passwords with the server. However, note that AKAv2 passwords cannot be re-used with the HTTP Digest AKAv2 algorithm because such an authentication challenge will automatically generate a fresh password. AKAv2 passwords can be used with other HTTP Digest algorithms, such as "MD5".

The underlying AKA protocol (e.g., UMTS AKA) has been designed to keep CK and IK confidential, but will typically send RES in the clear. We note that, even if (by some unfortunate misuse of AKA) RES values were revealed, the inclusion of RES in PRF(RES||IK||CK) is still beneficial, as it makes pre-calculated dictionaries of IK||CK values rather useless (though such dictionaries are infeasible for typical sizes of IK and CK).

5. IANA Considerations

This document specifies a new aka-version, "AKAv2", to the aka-version namespace maintained by IANA. The procedure for allocation of new aka-versions is defined in [6].

5.1. Registration Information

To: ietf-digest-aka@iana.org

Subject: Registration of a new AKA version

Version identifier: "AKAv2"

Contacts for further information: Vesa.Torvinen@turkuamk.fi, jari.arkko@ericsson.com, or mats.naslund@ericsson.com

6. References

6.1. Normative References

[1] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[2] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[3] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[4] 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.

[5] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[6] Niemi, A., Arkko, J., and V. Torvinen, "Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)", RFC 3310, September 2002.

6.2. Informative References

[7] 3rd Generation Partnership Project, "Security Architecture (Release 4)", TS 33.102, December 2001.

[8] Asokan, N., Niemi, V., and K. Nyberg, "Man-in-the-Middle in Tunnelled Authentication Protocols", Cryptology ePrint Archive, http://eprint.iacr.org Report 2002/163, October 2002.

[9] Puthenkulam, J., Lortz, V., Palekar, A., and D. Simon, "The Compound Authentication Binding Problem", Work in Progress, March 2003.

Authors' Addresses

Vesa Torvinen
Turku Polytechnic
Ylhaistentie 2
Salo FIN 24130
Finland

Phone: +358 10 5536210
EMail: vesa.torvinen@turkuamk.fi

Jari Arkko
Ericsson
Hirsalantie 1
Jorvas FIN 02420
Finland

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

Mats Naeslund
Ericsson
Torshamnsgatan 23
Stockholm SE 16480
Sweden

Phone: +46 8 58533739
EMail: mats.naslund@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgement

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