3GPP TSG WG3 Security - S3#19 Tdoc S3-010330
Newbury, Great Britain
4-6 July, 2001

Agenda Item: 7.3

Source: Ericsson

Title: EAP Extensions - status report

Document for: Information

1 Scope and objectives

The scope of this document is to briefly describe the current status of 3GPP and IETF standardization efforts related to the use of Extensible Authentication Protocol (EAP) and UMTS Authentication and Key Agreement (AKA) for SIP authentication. The pieces of work discussed in this document are:

1) HTTP authentication using EAP
2) EAP in Diameter
3) EAP AKA
4) Introduction of EAP in 3GPP Specifications

2 Background

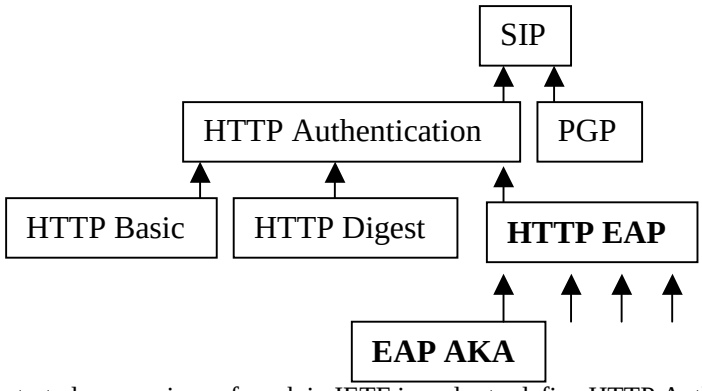
A working assumption in SA3 has been that AKA defined in R'99 shall be reused for Session Initiation Protocol (SIP). However, currently within IETF SIP AKA has not been defined. The current working assumption is that AKA would be included into the SIP protocol by extending the protocol. However, Ericsson contributions to the Madrid [S3z010036] and Phoenix [S3-010263] meetings stated that it would be beneficial to use a more generic authentication framework for various reasons, e.g.:

- Modification (and addition) of authentication schemes should be as flexible as possible. For this reason, the number of interdependencies between organisations (3GPP vs. IETF) and standards (SIP vs. authentication schemes) should be minimised.
- The used protocols and protocol extensions should be usable unchangeably on other access types, promoting access independence.
- Existing AAA transport attributes should be reused directly, without having to standardise special ones for UMTS.

Ericsson has proposed Extensible Authentication Protocol (EAP) [EAP] as a solution for the problems above. This document describes briefly the current status of 3GPP and IETF standardisation work related to the solution.

3 HTTP Authentication using EAP

Default authentication methods for SIP are HTTP Authentication and PGP [SIP]. Standardisation path for creating flexible AKA integration for SIP goes through HTTP Authentication (see figure below). Firstly, HTTP authentication scheme using EAP packets is defined. Secondly, EAP protocol extension using AKA is defined (see chapter 5 for status of this piece of work). When SIP recognises HTTP EAP scheme, no changes in SIP is required while maintaining / extending authentication methods. EAP will also open up several new authentication schemes for SIP. Furthermore, also other Internet protocols, which apply HTTP authentication, can utilise EAP authentication when needed.



Ericsson and Nokia have started a new piece of work in IETF in order to define HTTP Authentication with EAP [HTTPEAP]. The first version of this draft will be submitted to IETF mailing list for preliminary review during this week. The draft is also attached to this contribution.

4 DIAMETER EAP Extensions

Presently, the 3GPP is designing 3GPP-specific extension to the DIAMETER protocol to carry authentication information from home proxies to the HSS and back. These involve both new messages and new data attributes, to carry the AKA parameters. However, an alternative standardisation path has been proposed in which EAP (and EAP AKA) is used. In this scenario, existing AAA protocols are exploited in a greater extent.

The use of EAP in DIAMETER is defined by a group of expert in IETF AAA working group. The group has identified many problematic issues on their original specification. New versions have been written and the identified problems have been corrected [DIAMACC]. It seems that the specification has reached a stable stage.

In the next phase, the specification will enter to the “Last Call” phase very soon, probably during the next week. Last Call announces the intention of the IETF steering group to consider the specification as RFC, and it will solicit final comments from the IETF within a period of two weeks.

[DIAMACC] contains messages and AVPs sufficient to carry EAP authentication to a home authentication server. However, 3GPP work in the Cx interface should look into the possibility of reusing these existing parts. Furthermore, 3GPP should still define its own extensions to the Cx interface because Cx interface does more than traditional IETF AAA NASREQ interfaces (i.e. this draft should include definition of how authentication data information (AVs) is requested and downloaded between S-CSCF and HSS using the EAP extensions in DIAMETER).

5 EAP AKA

Ericsson and Nokia have continued their work around EAP AKA specification [EAPAKA]. The current version of the draft was already presented in Phoenix. Draft has been brought to the attention of the PPPEXT working group where some technical discussion has occurred. A standards track status for this work item has been requested, but as of now there is no feedback on this part of the issue. The discussion will continue on the PPPEXT mailing list. (This group is not meeting in the next general IETF meeting, but by IETF rules mailing list discussion should be sufficient. This was also indicated to us when we asked about the progression of this draft.) Some 3GPP2 members have also indicated their interests on the specification.

3GPP comments together with IETF discussion will probably result in an updated version of the specification in the near future.

6 Introduction of EAP into 3GPP Specifications

The use of EAP within 3GPP shall be primarily specified in S3 TS 33.203. This document will include a general definition of EAP and how it shall be applied for the IMS-AKA mechanism. References to corresponding IETF drafts/RFCs, in which the use of EAP into SIP and DIAMETER and the use of IMS-AKA into EAP are defined, are also given.

This activity has been already initiated and the working assumption on the use of EAP appears in the latest version of TS 33.203 (v.0.4.0) presented to this meeting.

The following step would be to introduce the use of EAP into stage 3 specifications at 3GPP WGs CN1 and CN4 (TSs 24.228 and 24.229 might be affected at CN1 while TSs 29.228 and 29.229 might be affected at CN4). These groups have been already informed of S3's working assumption to use EAP but they shall be asked to evaluate the impact on their specs and further request corresponding information from S3 in order to complete the required changes.

7 Conclusions

IETF standardisation efforts related to SIP authentication has proceed as follows:

- A new work item around HTTP authentication using EAP has been started.
- Work around EAP AKA has proceeded based on valuable input from 3GPP and IETF. New version of the draft will probably be available after the summer.
- Work related to DIAMTER EAP is approaching to its final stage. 3GPP should still define its own extensions for the Cx-AuthenticationDataRequest procedure.

Regarding the work related to the introduction of EAP into 3GPP specifications:

- The working assumption of the use of EAP has been already included in latest version of S3 TS 33.203. The information there shall be completed if required and agreed upon.
- Impacts on CN1 (24.228, 24.229) and CN4 (29.228 and 29.229) specifications shall be identified and corresponding updates included in affected specs.

References

[DIAMACC] P. R. Calhoun, W. Bulley, A. C. Rubens, J. Haag & G. Zorn, *Diameter NASREQ Application*, IETF, Internet Draft, document: draft-ietf-aaa-diameter-nasreq-06.txt, June 2001.

[EAP] L. Blunk & J.Vollbrecht.: *Extensible Authentication Protocol (EAP);* IETF RFC 2284, March, 1998.

[EAPAKA] J. Arkko & H. Haverinen, *EAP AKA Authentication*, IETF, document: draft-arkko-pppext-eap-aka-00.txt, May 2001.

[HTTPEAP] V. Torvinen, J. Arkko & A. Niemi, HTTP Authentication with EAP, IETF, Internet Draft, document: draft-torvinen-http-eap-00.txt, June 2001.

[SIP] SIP: Session Initiation Protocol, Internet Draft, IETF, November 24, 2000.

[S3z010036] Different Issues on aSIP (SASL)

[S3-010263] Proposal to Use a Generic Authentication Scheme for SIP

Internet Draft
Document: draft-torvinen-http-eap-00.txt
Expires: January, 2002

V. Torvinen
J. Arkko
Ericsson
A. Niemi
Nokia

June 2001

HTTP Authentication with EAP

Status of this Memo

Abstract

This document describes HTTP authentication scheme using PPP Extensible Authentication Protocol (EAP).

HTTP EAP authentication enables HTTP connections to be authenticated using any of the authentication schemes supported through EAP. EAP performs the authentication without sending the password in the clear text format (which is the biggest weakness of the Basic HTTP authentication scheme, for example). It is useful for HTTP protocol because it opens up several new authentication schemes without additional specification work. The same benefits can be reached by any other protocol, which apply the HTTP authentication scheme, such as Session Initiation Protocol (SIP).

Table of Contents

Conventions 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 RFC-2119 [1].

1 Introduction

HTTP Authentication framework includes two authentication schemes: Basic and Digest [2]. In the Basic scheme, the client authenticates itself with a user-ID and a password for each realm. Basic scheme is perceived as insecure since the user credentials are transformed across the public network in a cleartext format. The Digest scheme is based on cryptographic hashes and it is perceived consequently as more secure authentication scheme than Basic, but is limited to the use of passwords. See [2] for detailed information about the general HTTP authentication protocol.

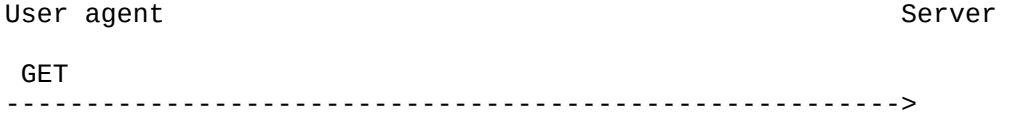
The PPP Extensible Authentication Protocol (EAP) is a general protocol for PPP authentication [3]. Even though EAP was originally developed as a link layer protocol, it can be applied at application layer, too. EAP supports multiple authentication mechanism (e.g. smart cards, Kerberos, Public Key, One Time Passwords, and others) and it can, by definition, be easily extended to support new authentication mechanisms [see e.g. 4, 5, 6, 7]. EAP packets are defined in a binary format, and their contents depend highly on the used authentication scheme.

HTTP EAP Authentication Scheme supplements HTTP Authentication with EAP functionality. This opens up several new authentication schemes for HTTP Authentication without additional specification work.

2 HTTP EAP Authentication Scheme

HTTP EAP Authentication Scheme delivers base64 encoded EAP packets within HTTP Authentication headers (e.g. Authorization Request headers and WWW-Authenticate Response headers). EAP packets include all relevant information about the required authentication scheme, e.g. authentication scheme, packet type (request, response, success or failure) and/or challenge. The content of these packets is up to the chosen EAP authentication scheme.

The progression of an authentication procedure depends also on the chosen authentication mechanism. Typically, the authenticator sends an initial Identity Request followed by one or more Requests for authentication information. The peer sends a Response packet in reply to each Request. As with the Request packet, the Response packet contains a type field, which corresponds to the type field of the Request. The authenticator ends the authentication phase with a Success or Failure packet. See Figure 1.



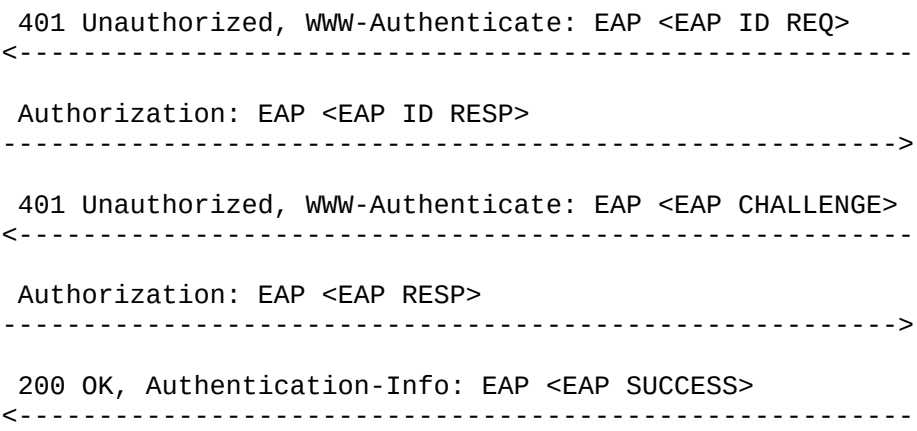


Figure 1. HTTP EAP Authentication message flow

This message flow above represents only the typical situation. Variations of the flow are also possible in the following situations:

- The chosen authentication mechanism requires more than the single challenge-response message pair shown. Any number of message exchanges are allowed here.
- Error situations result in terminating the flow from the server's side with an error response. This response could be one of 401 Unauthorized, 403 Forbidden, or 407 Proxy Authentication Required. For 401 and 407, the client distinguishes the error situation from the continuation of the EAP exchange by the existence of EAP FAILURE payload, or the lack of any EAP payload.
- Error situations from the client's side result in terminating the communications with the server.
- Certain EAP authentication mechanisms such as [7] allow an optimized flow where identity request does not need to be sent. In these cases, if the client knows it will be demanded EAP authentication, it can include an unsolicited EAP ID RESP already in the GET message. This would enable the server to start the actual authentication exchange immediately.

- EAP authentication was shown to be run towards the server which responds with 401 Unauthorized responses. It is also possible to run towards a proxy, which responds with 407 Proxy Authentication Required responses.

In this document, we define three new header types for HTTP authentication framework. These headers, WWW-Authenticate Response Header, Authorization Request Header and Authentication-Info Response Header, are needed for making EAP as an independent HTTP authentication scheme.

2.1 The WWW-Authenticate Response Header

The general HTTP authentication framework uses an extensible, case-insensitive token to identify the authentication scheme. Authentication scheme identifier is followed by a comma-separated list of attribute-value pairs, which carry the parameters necessary for achieving authentication via that scheme.

```
auth-scheme     = token
auth-param      = token "=" ( token | quoted-string )
```

If a server receives a request for an access-protected object without acceptable Authorization header, the server responds with a "401 Unauthorized" status code, a WWW-Authenticate header and at least one challenge applicable to the requested resource. Proxy acts in the same way but it uses "407 Proxy Authentication Required" status code instead.

```
challenge   = auth-scheme 1*SP 1#auth-param
```

The authentication parameter realm is defined for all authentication schemes:

```
realm       = "realm" "=" realm-value
realm-value = quoted-string
```

The realm value and the canonical root URL of the server being accessed define the protection space. The realm value (case-sensitive) is a string, which may have additional semantics specific to the authentication scheme.

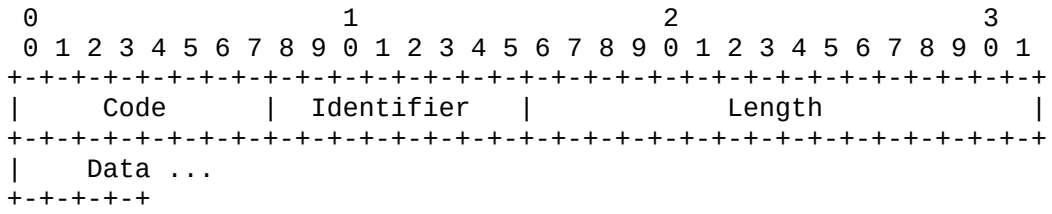
For HTTP EAP Authentication, the framework above is utilized as follows:


```
challenge     = "eap" eap-challenge

eap-challenge = 1#([realm] | eap-param)
realm         = "realm" "=" realm-value
realm-value   = quoted-string
eap-param     = "eap-p" "=" eap-packet
eap-packet    = <base64 encoded eap-packet, except
                   not limited to 76 char/line>
```

Realm is made optional because EAP notification mechanism can be used as well [3]. If realm value is present, it should be considered as an opaque string, which can only be compared for equality with other realms on that server. The server will service the request only if it can validate the user credentials for the protection space of the Request-URI.

EAP packets have a general structure consisting of four basic fields: code, identifier, length and data. Code field is one octet and it identifies the type of EAP packet. Packet type is a request, response, success, or failure. Identifier field is also one octet and it is used for matching responses with corresponding requests. Length field is two octets and it indicates the whole EAP packet including code, identifier, length and data fields. Data field is zero or more octets and its format depends on the content of code field. Example below demonstrates the general structure of EAP packets.



All these fields (Code, Identifier, Length, and Data) are included in the eap-packet in base64 form. Note that since the packets are self-identifying and self-delimiting it is allowed to include multiple EAP packets within one eap-packet, should some EAP mechanism be able to benefit from this.

Example below demonstrates how WWW-Authenticate Response Header using EAP authentication would look like:

```
WWW-Authenticate: eap realm="BollyWorld",
eap-p=QWxh4ZGRpb2jpvcGVuNlctZQ==
```


where "BollyWorld" is the string assigned by the server to identify the protection space of the Request-URI.

A proxy may respond with the same challenge using the Proxy-Authenticate header field.

2.2 The Authorization Request Header

In the general HTTP authentication framework, a user agent that wishes to authenticate itself with an origin server or a proxy MAY do so by including an Authorization header or a Proxy- Authorization header field with the request. The authorization field value(s) consists of credentials containing the authentication information of the client for the realm of the resource being requested. The user agent must apply the strongest authentication scheme it understands and request credentials from the user based upon corresponding challenge.

```
credentials = auth-scheme #auth-param
```

For HTTP EAP Authentication, the framework above is utilized as follows:

```
credentials = "eap" eap-packet

eap-packet = <base64 encoded eap-packet, except
              not limited to 76 char/line>
```

Example below demonstrates how the Authorization Request Header using EAP authentication would look like:

```
Authorization: eap QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

Rules for handling potential user identifiers, passwords, challenges and so on, are defined in EAP protocol [3].

2.3 Authentication-Info Response Header

The Authentication-Info header is used by the server to communicate information back to the client. This can be either the successful authentication in the response, or the continuation of the EAP mechanism.


```
auth-info          = #auth-param
```

For HTTP EAP authentication the framework above is utilized as follows:

```
auth-info          = eap-packet
eap-packet         = <base64 encoded eap-packet, except
                      not limited to 76 char/line>
```

Example below demonstrates how the Authentication-Info Response Header using EAP authentication would look like:

```
Authentication-Info: QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

The semantics of Proxy-Authentication-Info follow those of Authentication-Info. Proxy-Authentication-Info is used by proxy servers in conjunction with the "407 Proxy Authentication Required" response, and the consequent client authorization request.

3 Security Considerations

Very little about the security of HTTP EAP Authentication can be stated without knowing the chosen EAP authentication scheme.

Generally speaking, depending on the chosen EAP authentication scheme, HTTP EAP is subject to the same security threats as HTTP Authentication. However, there are some general aspects, which SHOULD be considered when analyzing the security of HTTP EAP Authentication:

1) Authentication of clients: All EAP mechanisms authenticate the client, using a method dependent on the mechanism.
2) Authentication of servers: Some EAP mechanisms perform also mutual authentication.
3) Using the strongest authentication mechanism: Servers and clients accepting multiple authentication mechanisms should be aware of the possibility of 'bidding-down' attacks where a man-in-the-middle modifies authentication offers until the peers agree on an easily breakable mechanism. In general, we expect HTTP EAP _based servers to require a predefined authentication mechanism from a particular client in any case, which avoids this problem. For instance, the user data base at a server indicates that user A has a particular public key. The server should then insist on using the EAP TLS [4] mechanism to authenticate the user.
4) Confidentiality: Each EAP mechanism offers its specific protection schemes for the exchanged credentials. For instance,

the EAP AKA [7] mechanism sends secure cryptographic hashes rather than cleartext passwords like HTTP Basic Authentication does, even if both are based on the concept of a shared secret. As in EAP in general, HTTP EAP does not protect against revealing the identity of the client since the EAP ID RESP packets are not encrypted. Confidentiality and integrity of the HTTP requests themselves beyond on the authentication parameters is not within the scope of HTTP EAP, but is discussed below under item 7.
5) Replay protection: Each EAP mechanism offers its specific protection schemes for preventing the replay of the credentials. For instance, the EAP AKA mechanism uses a cryptographically strong sequence number scheme. This is in contrast to the replay possibilities that exist for the HTTP Basic Authentication, and is similar to the use of nonces in the HTTP Digest Authentication.
6) Integrity protection: Again, each EAP mechanism offers its specific protection against a man-in-the-middle modifying the authentication credentials. Mechanisms based on secure hashes prevent any modifications to the authentication parameters themselves. Again, integrity of the HTTP requests themselves beyond the authentication parameters is a separate issue and is discussed below.
7) Integrity and confidentiality protection of the HTTP request itself is also an important issue. Without such protection, it is possible for a man-in-the-middle to read and modify the actual contents of the request, regardless of any

authentication that was performed

Currently, there is no such authentication scheme in HTTP authentication, which would fully protect the integrity of HTTP messages. HTTP Basic Authentication scheme provides no integrity protection. HTTP Digest Authentication provides only limited (and optional) protection. Most header fields and their values could be modified as a part of a man-in-the-middle attack. It should also be noted that HTTP EAP does not inherently provide the integrity protection qualities present in Digest, namely the protection of Request-URI and request-method (and possibly the payload).

Even though HTTP EAP Authentication scheme does not include a protection mechanism, it can be used for setting up one. Chosen EAP authentication scheme may be used to generate session keys, which together with some additional security protocol can provide e.g. integrity protection.

However, such protection should include the protection of original HTTP requests as well. This is not trivial because session protection keys are generated during the authentication, which takes place after submitting the request. In practice, full protection is only possible if the request is repeated at the end of the authentication procedure. This is, however, already the behavior in many typical usage situations. For instance, when authenticating a SIP REGISTER message, the authentication procedure takes a few message rounds, and on each round the REGISTER message is repeated until the session keys are available and the procedure is completed. The last such message can then use integrity protection. Servers that want to avoid man-in-the-middle attacks MUST NOT act on requests until both the authentication procedure has completed and the messages have been received under integrity protection.

References

1 RFC 2119 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 Stewart, L. "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

3 Blunk, L. and Vollbrecht, J. "PPP Extensible Authentication Protocol (EAP)" RFC 2284, March 1998.

4 Aboba, B. and Simon, D. "PPP EAP TLS Authentication Protocol" RFC 2716, October 1999.

5 Aboba, B. "EAP GSS Authentication Protocol" Internet Draft, draft-aboba-pppext-eapgss-03.txt, February 2001.

6 Carlson, J. "PPP EAP SRP-SHA1 Authentication Protocol" Internet Draft, draft-ietf-pppext-eap-srp-01.txt, May 2001.

7 Arkko, J. and Haverinen, H. "EAP AKA Authentication" Internet Draft, draft-arkko-pppext-eap-aka-00.txt, May 2001.

Acknowledgements

The authors wish to thank Henry Haverinen and Bernard Aboba for interesting discussions in this problem space.

Author’s Addresses

Jari Arkko
Ericsson
02420 Jorvas Phone: +358 40 5079256
Finland Email: jari.arkko@ericsson.fi

Vesa Torvinen
Ericsson
02420 Jorvas
Finland
Phone: +358 40 7230822
Email: vesa.torvinen@ericsson.fi

Aki Niemi
Nokia Networks
P.O. Box 301
00045 Nokia Group
Finland
Phone: +358 50 3891644
E-mail: aki.niemi@nokia.com