Internet Engineering Task Force
INTERNET-DRAFT
draft-salgarelli-pppext-eap-ske-01-pre02.txt
4/02
Expires: Oct. 31, 2002

L. Salgarelli,
M. Buddhikot, J. Garay,
S. Miller, U. Blumenthal,
S. Patel, P. Dahl,
D. Stanley, C. Carroll

# EAP SKE authentication and key exchange protocol

## Status of this memo

Abstract

This note describes EAP Shared Key Exchange (SKE), a method for authentication of Mobile Nodes (MN) and generation of a per session, per node EAP Master Secret. The method applies to scenarios where a Mobile Node (MN) is in a foreign network such as public 802.3 network that uses Home-AAA and Foreign-AAA services. The method assumes that a pre-deployed cryptographically secure shared key is present on the MN and on its Home-AAA server, and use of the 802.1x standard [1], Extensible Authentication Protocol (EAP) [2] messages, and RADIUS [3] authentication servers. The protocol can easily be extended to support the migration from RADIUS to DIAMETER [4].

Contents

## 1 Introduction

In this document, we describe a new EAP authentication and key exchange method, EAP-Shared Key Exchange (EAP-SKE), which (1) supports mutual authentication between the MN and a Home-AAA (H-AAA); (2) provides for the generation of the EAP Master Secret necessary to derive per-user, per-session EAP Master Session Keys [5, 6]; and (3) efficiently supports roaming across multiple network provider networks by significantly reducing the number of messages required to perform steps (1) and (2) above. This is particularly important in roaming scenarios, where the delay involved in exchanging messages with a distant H-AAA has a critical impact on the latency of the overall authentication procedure. Finally, (4) the protocol realizes the goals above without requiring state to be kept at the AAA's in between sessions.

## 2 Terminology

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 [7].

### 2.1 Notation

The following notation is used throughout this document:

| | |
|---|---|
| A_(A,B) | A security association between nodes A and B, as defined in [8], which includes a shared secret key known to A and B, and allows the secure exchange of information between nodes A and B. |
| AAA | Authentication, Authorization and Accounting (server).<br>Same as the Authentication Server in the IEEE 802.1x terminology. For simplicity, we will use the term AAA server throughout this document.<br>H-AAA: Home AAA; F-AAA: Foreign AAA. |

| | |
|---|---|
| Authenticator System (AS) | An IEEE 802 LAN entity that uses port based network access control. For example, in case of an IEEE 802.11 LAN, an Access Point (AP) may be an authenticator system. An AS consists of (1) a Port Access Entity (PAE) that plays the role of an authenticator to authenticate a user, and (2) an entity that provides the network access service offered by the AS. In the remaining document, we will refer to the PAE entity as AS-PAE. |
| AS-PAE | See Authenticator System above. |
| K_(A,B) | A shared secret key known to nodes A and B. |
| K_EMS | The negotiated EAP Master Secret as defined in [5, 6]. K_EMS can then be used to derive Master Session Keys as specified in section 3.5 of [9]. |
| MAC(K,·) | Message Authentication Code (or integrity check function), which is applied to a piece of information for authentication using a key K. Examples include keyed cryptographic hash functions (e.g., keyed-MD5 [10], keyed-SHA-1 [11, 12], HMAC [13, 14], etc.), and block ciphers (e.g., AES in CBC-MAC mode [15]). |
| MN | Mobile Node. |
| NSP | Network Service Provider. |
| N_1/2/3 | A nonce, in this case a freshly generated (unstructured) random number. Nonces are typically implemented as pseudo-random bit strings of length 64-128. |
| PRF(K,·) | A pseudo-random function with key K. Pseudo-random functions [16] are characterized by the pseudo-randomness of their output, namely, each bit in the output of the function is unpredictable if K is unknown. We use pseudo-random functions for the derivation of the EAP master secret given the shared key K. In practice, pseudo-random functions are realized using AES in CBC-MAC mode (and other block ciphers), or keyed one-way hash functions (see examples of MAC functions above). |

Supplicant A 802 LAN port that wishes to obtain (network) services offered by the AS. The mobile node (MN) in an 802.1x network will contain an entity that serves as a supplicant. For brevity, we refer to such an entity as MN-SUP.

## 3 EAP-SKE authentication and dynamic key exchange

It is desirable that when a MN roams into a foreign 802 network, its supplicant should be able to establish credentials with the NSP of the foreign network to obtain network access. One example of this is as follows: user John Doe who has an account with, say, carrier.com roams into a public network in a mall or airport run by an NSP such as (hypothetical) airport-mall.net. It is desirable that John be able to present his credentials with carrier.com to airport-mall.net to authenticate himself and obtain network access. The access charge for this service is later posted to John's monthly access bill with his carrier via a revenue settlement agreement between the two NSPs. This requires (a) that the AAA services employed by carrier.com and airport-mall.net peer with each other using pre-established secure channels and (b) that a database of AAA services be exchanged among the providers. Such a scenario already exists between providers which provide network access to roaming dial-up customers.

### 3.1 Assumptions

We assume that as a part of a service contract with a network provider (say carrier.com), the supplicant has (1) a pre-configured network access identifier (NAI) (e.g., john.doe@carrier.com), and (2) a pre-configured security association with its Home AAA server (H-AAA), which includes a sufficiently long (say, 128 bit) key K_(MN,H-AAA), as shown in Figure 1. At the same time, we assume that each AS-PAE in the foreign domain has a pre-configured security association A_(AS-PAE,F-AAA) with the F-AAA, which allows the F-AAA and the AS-PAE to authenticate and encrypt the messages that they exchange. The mechanism by which these security associations are setup is outside the scope of this document.

We also assume that a security association A_(F-AAA,H-AAA) exists between the F-AAA and H-AAA, which allows the F-AAA and H-AAA to

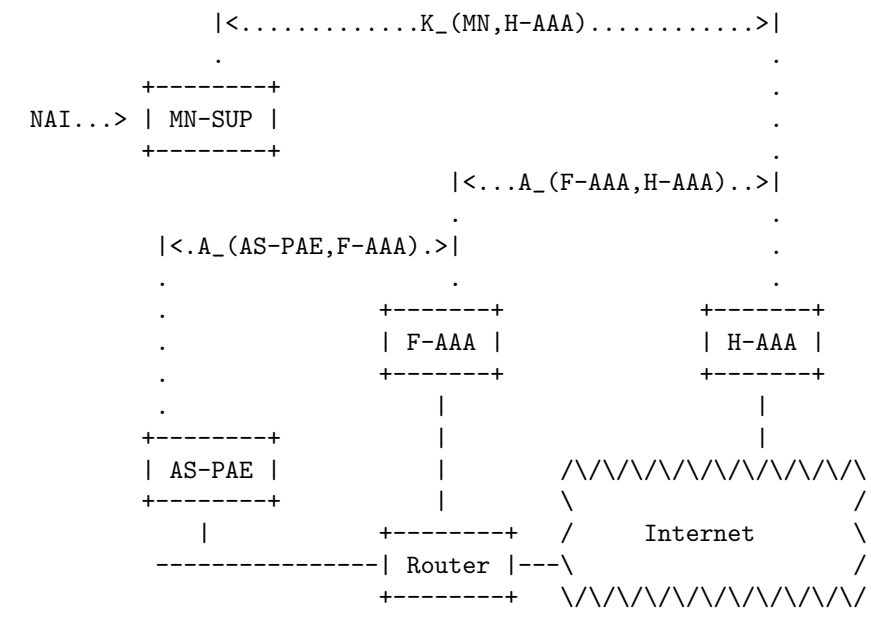



Figure 1: Entities in the proposed 802.1x architecture

authenticate and encrypt each other's messages. This association is setup as part of the roaming agreement between the foreign and home domains, and allows the home domain to trust the AAA servers and the AS-PAEs of the foreign domain. Also in this case, the mechanism by which this security association is setup is outside the scope of this document.

Furthermore, we assume that the F-AAA and the AS-PAE are trusted network elements, and that they will not deviate from the execution of the protocol.

Note that a number of proxy AAA servers MAY be present in the path between the AS-PAE and the F-AAA and between the F-AAA and the H-AAA. In this case we assume that a pre-set security association exists between any pair of adjacent nodes between the AS-PAE and the H-AAA.

Note also that Figure 1 only represents logical network elements. In particular, in actual implementations the F-AAA and H-AAA might be a single server, with a chain of trusted AAA proxy servers between it and the AS-PAE. More details on this will be given in

section 5.

### 3.2 Protocol requirements

Given the assumptions outlined in section 3.1, our protocol must meet the following objectives:

**Network Efficiency:** Keep both the number of messages exchanged between the parties and the computational overhead to a minimum. This means also that the latency of the authentication process must be kept to a minimum. Ideally, only one message exchange should take place between the Foreign network and the H-AAA to perform authentication and Session Master Key distribution.

**Secrecy and Authenticity:** Guarantee the participating entities that only the intended parties learn the Master Session Keys exchanged, and that these keys are fresh, random and unique. Specifically, the scheme should support the following

- Requirement 1 (Authenticate MN-SUP): Allow H-AAA to authenticate and authorize that the MN-SUP has rights to establish a security association with, and receive service from the AS in a foreign domain with which the home domain has a roaming agreement.

- Requirement 2 (Authenticate H-AAA): Allow the MN-SUP to establish that it is authenticating to a trusted H-AAA that is in possession of K_(MN-SUP,H-AAA);

- Requirement 3 (Master Session Key Establishment): generate the EAP Master Secret K_EMS necessary to derive the EAP Master Session Keys [6]. Guarantee both MN supplicant and H-AAA that K_EMS is fresh, random and unique.

**Forward Secrecy:** When used in this document forward secrecy refers to the notion that compromise of the Master Session Keys will permit access only to data protected by those keys. In other words, even if an attacker is eventually able to derive the Master Session Keys for one session, future (and past) session keys (and, of course, the pre-shared key K_(MN-SUP,H-AAA)) are not compromised. EAP-SKE only defines

the mechanism by which the EAP Master Secret K_EMS is derived, and relies on the mechanism specified in section 3.5 of [9] to derive the Master Session Keys. Therefore, forward secrecy of the Master Session Keys derives from forward secrecy of K_EMS, assuming that the key explosion mechanism specified in [9] preserves this property. See, e.g., [17, 18, 19] for more general notions of forward secrecy.

**Statelessness:** The scheme must not rely on state that needs to be kept at AAA servers in between sessions.

**Simplicity:** The scheme must be amenable to analysis and formal security proof.

### 3.3 Protocol description

Figure 2 describes a successful EAP-SKE authentication and key exchange procedure, involving a client (MN-SUP), an 802.1x Port Access Entity (AS-PAE), a Foreign and a Home AAA (F-AAA and H-AAA).

The EAP-SKE exchange proceeds as follows:

Phase 1: F-AAA obtains identity and realm of user. The AS-PAE issues an EAP Request ID frame. The MN responds with an EAP Response ID message that includes its NAI. The AS-PAE forwards the EAP Response in a RADIUS Access-Request message to the F-AAA.

Phase 2: F-AAA challenges MN and obtains MN's own challenge and authenticator AUTH1. The F-AAA generates a random challenge N_1. The F-AAA then issues an EAP Request SKE-AS-Challenge message with N_1, and sends it to the AS-PAE encapsulated in a RADIUS Access-Challenge message. The AS-PAE forwards the EAP request to MN. Reception of this packet signals to the MN-SUP that the F-AAA is requesting authentication scheme EAP-SKE. In the event MN-SUP does not support the scheme, it will send EAP Response of type NAK. Otherwise, MN-SUP also generates a nonce N_2 and computes the authenticator

AUTH1 = MAC(K_(MN-SUP,H-AAA), N_1 | N_2 | NAI).

The MN-SUP sends N_2 with the authenticator AUTH1 back to the AS-PAE in an EAP Response SKE-MN-Challenge packet. The EAP

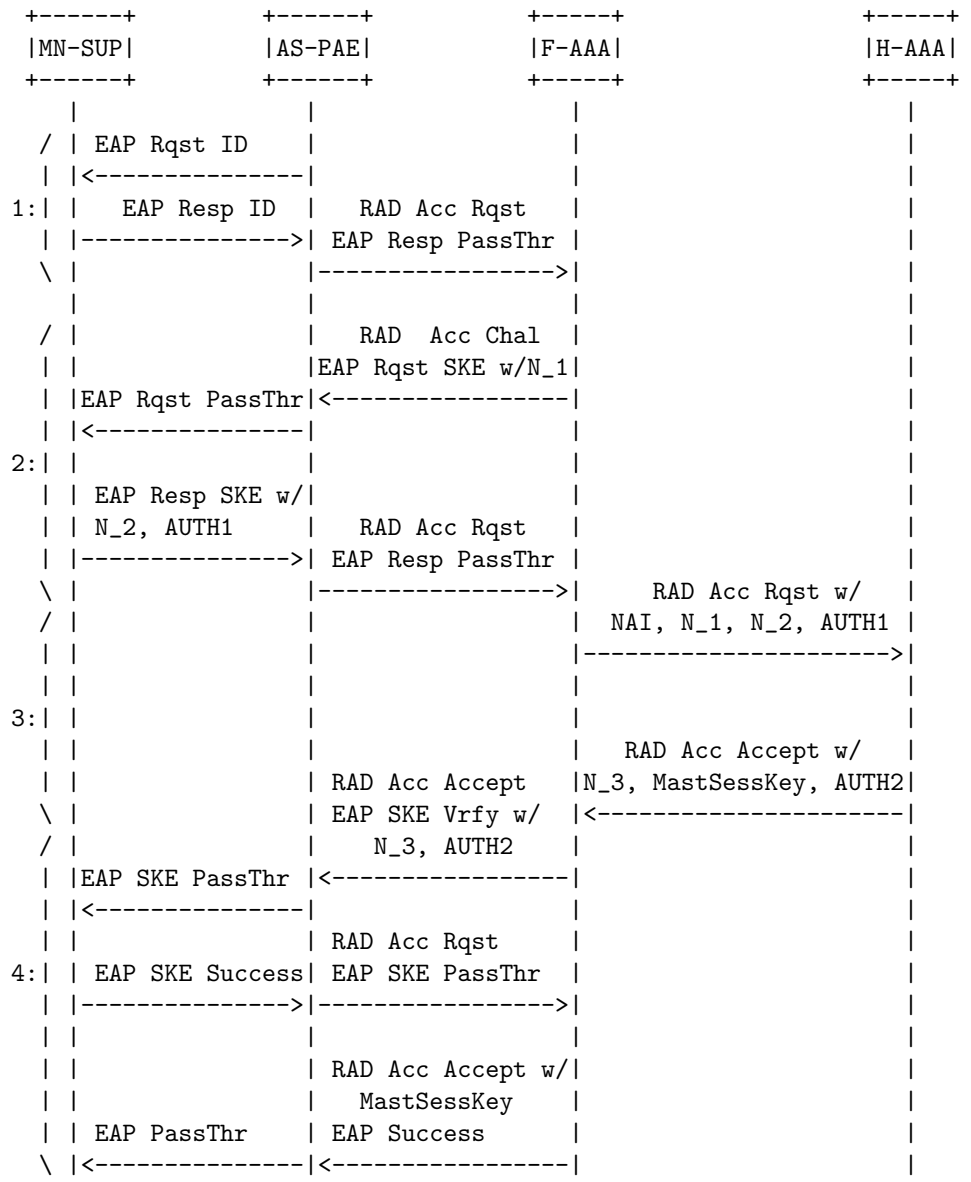



Figure 2: Successful SKE authentication and key exchange

SKE Response is forwarded by the AS-PAE encapsulated in a RADIUS Access-Request message to the F-AAA.

Phase 3: H-AAA authenticates the user, generates its own authentication information and generates the Master Session Keys. The F-AAA forwards the user's NAI, N_1, N_2 and AUTH1 to H-AAA in a RADIUS Access-Request message.
On receipt of the Access-Request, the H-AAA server does the following:

1. Looks up K_(MN-SUP,H-AAA), the key for user 'NAI';

2. calculate the Authenticator
   AUTH1' = MAC(K_(MN-SUP,H-AAA), N_1 | N_2 | NAI)

   as explained in Phase 2 above.

3. Compares AUTH1' with the received AUTH1. If the two do not match, authentication fails and the H-AAA server will send a RADIUS Access-Reject. If the authentication succeeds, the H-AAA undertakes the following steps:

4. Computes the authenticator

   AUTH2 = MAC(K_(MN-SUP,H-AAA), N_2 | N_1 | NAI).

   (Note change in the order of arguments with respect to AUTH1.)

5. Generates nonce N_3.

6. Computes the EAP Master Secret K_EMS as
   K_EMS = PRF(K_(MN-SUP,H-AAA), N_3 | AUTH2).

7. Derives the Master Session Keys from K_EMS as specified in section 3.5 of [9].

8. Forwards AUTH2, N_3 and the Master Session Keys to F-AAA in a RADIUS Access-Accept.

Phase 4: MN authenticates H-AAA, AS-PAE receives session keys.

Upon receipt of the RADIUS Access-Accept message, the F-AAA sends AUTH2 and N_3 in an EAP Request SKE-AS-Verify message encapsulated in a RADIUS Access-Accept message to the AS-PAE. The AS-PAE relays the EAP SKE-AS-Verify message to the MN. The MN first uses N_1 and N_2 to compute AUTH2' as in Phase 3 above and compares it with the supplied AUTH2. If the two match, it concludes that its request was processed by a valid H-AAA, and concludes the SKE exchange with an EAP Response SKE-Success message. If the two do not match, the MN MUST send a EAP Response SKE-Failure message. After receiving the EAP Response SKE-Success the F-AAA issues an EAP Success message and the Master Session Keys in a RADIUS Access-Accept message. Then, using K_(MN-SUP,H-AAA), N_3 and AUTH2, the MN generates K_EMS and the Master Session Keys following the exact same procedure used at the H-AAA server. Note that the Master Session Keys are not transmitted from the AS-PAE to the MN but are locally computed.

## 4 Security Considerations

Let us recall the assumptions we made in section 3.1. In particular, we assume that AS-PAE and F-AAA are network elements which are trusted by the H-AAA, by means of the existence of A_(H-AAA,F-AAA) and A_(AS-PAE,F-AAA), and do not misbehave.

Consider authenticators AUTH1 and AUTH2 in Phases 2 and 3, respectively. The nonces N_1 and N_2 in the authenticators act as challenges to H-AAA and MN to "prove" to each other the possession of the pre-shared key K_(MN-SUP,H-AAA). Moreover, including N_1 (respectively, N_2) assures the H-AAA (resp., the MN) that the authenticator is fresh for every session. The fact that N_1 is generated by the F-AAA and not by the H-AAA does not invalidate this claim, since the F-AAA is trusted by the H-AAA by virtue of A_(H-AAA,F-AAA). The included identities (i.e., the username and realm parts of the NAI) serve to reassure the parties of the correct binding between the shared key and their identities.

The authenticity, freshness and randomness of the EAP Master Secret follow from the authenticity and freshness of AUTH2, and the properties of pseudo-random functions; specifically, the value PRF(K_(MN-SUP,H-AAA), N_3 | AUTH2) is (computationally) independent of any other value output by the function. Thus, the protocol reveals no information to an adversary on the value of the EAP Master Secret K_EMS, as well as the Master Session Keys

subsequently derived from it (forward secrecy), assuming that the mechanism used to explode K_EMS into the Master Session Keys (Section 3.5, [9]) preserves forward secrecy.

Replay attacks by illegitimate network elements are detected by the MN and the H-AAA by the application of MAC functions to N_1 and N_2, given that both nonces are freshly generated every time by the F-AAA and MN. Denial Of Service (DOS) attacks are alleviated, because if they are mounted by replaying these authentication messages, they would be detected as described above.

## 4.1 Considerations on misbehaving nodes in foreign access networks

A full adversarial model can also be considered, where the trusted AS-PAEs and F-AAAs might misbehave. In particular, since both N_1 and N_2 are sent in clear between the MN-SUP and the AS-PAE, a misbehaving AS-PAE could replay an access-request with the same N_1 and N_2 as a previous request. The objectives of this attack, and their effects on EAP-SKE are twofold:

- A misbehaving node might replay a successful registration to get access to the Master Session Keys that a legitimate user has obtained. In this case, the inclusion of a fresh value for N_3 (Phase 3, Section 3.3) would counteract the replay, by guaranteeing that each generated EAP Master Secret and subsequently Master Session Keys are different even in cases where N_1 and N_2 are replayed.

- A misbehaving node might replay a successful registration to make the H-AAA believe that a legitimate user is initiating a session with the foreign network. In this case EAP-SKE does not offer any direct protection. However, a brief analysis of the possible motivations behind this attack should clarify that such protection is indeed not necessary. Infact the effects of such attack would be that the Foreign Network could overcharge the Home Network, or could otherwise apply malicious charging schemes where it would permit the use of its infrastructure to unauthorized clients at the expense of the Home Network. In this case, given the trust relationship that exists between the foreign and home networks, misbehaving nodes in the foreign network could always prepetrate such attacks without the needs of breaking the authentication

protocol at all. For example, they could simply overstate the amount of traffic that legitimate users generate and receive. Or they could allow access to unauthorized users charging their traffic to the bill of authorized customers. In summary, we feel that this issue must be regulated through the application of business contract agreements, rather than through authentication protocols per se.

The prevention of attacks pertaining to a full adversarial model other than the one mentioned above is outside the scope of this document.

## 5 Alternate network scenarios

As mentioned in section 3.1, depending on how operators decide to implement EAP-SKE in their networks, the two logical entities F-AAA and H-AAA MAY be combined. In this case, EAP-SKE would be terminated at the MN-SUP and H-AAA, without the need of Vendor Specific Extensions to RADIUS. However this would come at the expense of network efficiency and added latency, since three roundtrips with the H-AAA would be required to complete the SKE exchange.

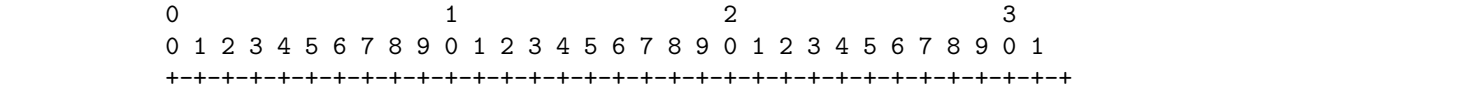
## 6 Recommendation for MAC and PRF algorithms

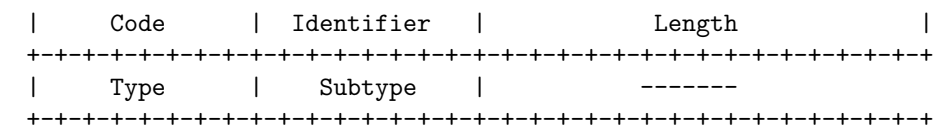
EAP-SKE implementations compliant with this document MUST implement HMAC-SHA1 [14] as MAC function and as PRF, as a minimum. HMAC-MD5 MAY be also implemented, in particular where compatibility with existing RADIUS servers that are already compliant with the requirements of dynamic key distribution for Mobile IP [20] is a concern. In addition, EAP-SKE implementations can optionally implement other MAC and/or PRF algorithms.

## 7 Message formats

### 7.1 EAP-SKE messages

EAP SKE messages are of the following format:



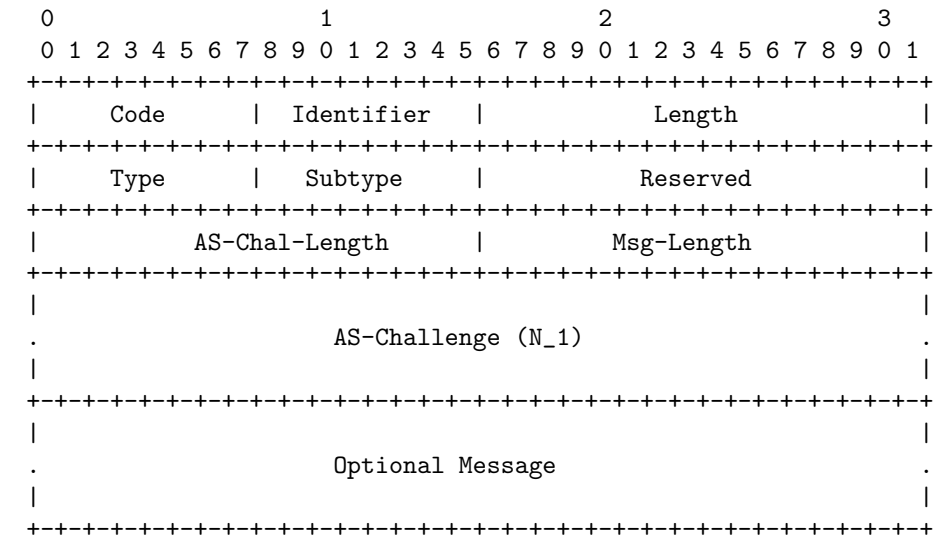



All EAP SKE packets must carry the same value X (TBD) in the Type field. The Subtype field identifies specific SKE packets. At present only Subtype= 1, 2, 3, 4 are defined. The SKE packets with Subtypes other than these MUST BE silently discarded.

The detailed message formats are described in the following:

### 7.1.1 EAP-Request/SKE-AS-Challenge

The format of the packet EAP-Request/SKE-AS-Challenge is shown below.




The semantic of the fields is described below:

Code
1 for Request

Identifier
See [2]

Length
The length of the EAP Request packet.

Type
TBD for EAP-SKE

Subtype
1 for SKE-AS-Challenge

Reserved
This field must be set to zero.

AS-Chal-Length
>=1 and <=28. Length of AS challenge (N_1) in 4-byte words. The challenge size must be at least 4-bytes.

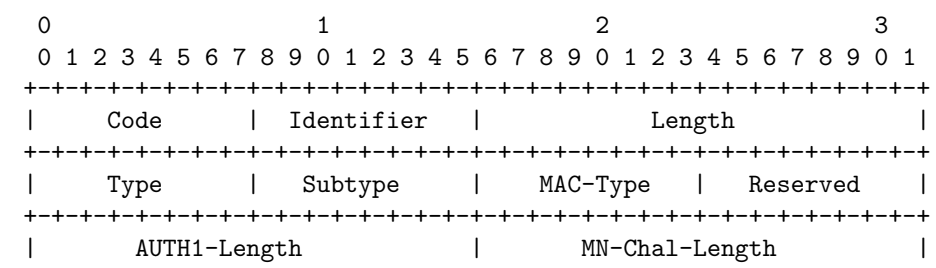
Msg-Length
Length of optional message in 4-bytes words (0 or more words).

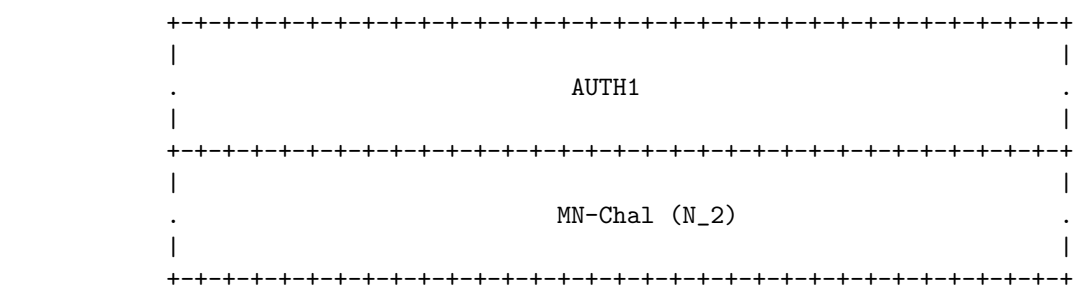
AS-Challenge
Challenge from the F-AAA (N_1).

Optional Message
Optional message must be null terminated and padded to ensure it length is multiple of 4 bytes.

### 7.1.2 EAP-Response/SKE-MN-Challenge

The format of the packet EAP-Response/SKE-MN-Challenge shown below.






The semantic of the fields is described below:

Code
2 for response

Identifier
See [2]

Length
The length of the EAP Response packet.

Type
TBD for EAP-SKE

Subtype
2 for SKE-MN-Challenge

MAC-Type
The MAC algorithm used by the MN-SUP to calculate AUTH1. It MUST be one of the codes listed in section 8.

Reserved
This field MUST be set to 0.

AUTH1-Length
Length of AUTH1, in 4-byte words.

MN-Chal-Length
>=1 and <=28. Length of MN Challenge (N_2), in 4-byte words.

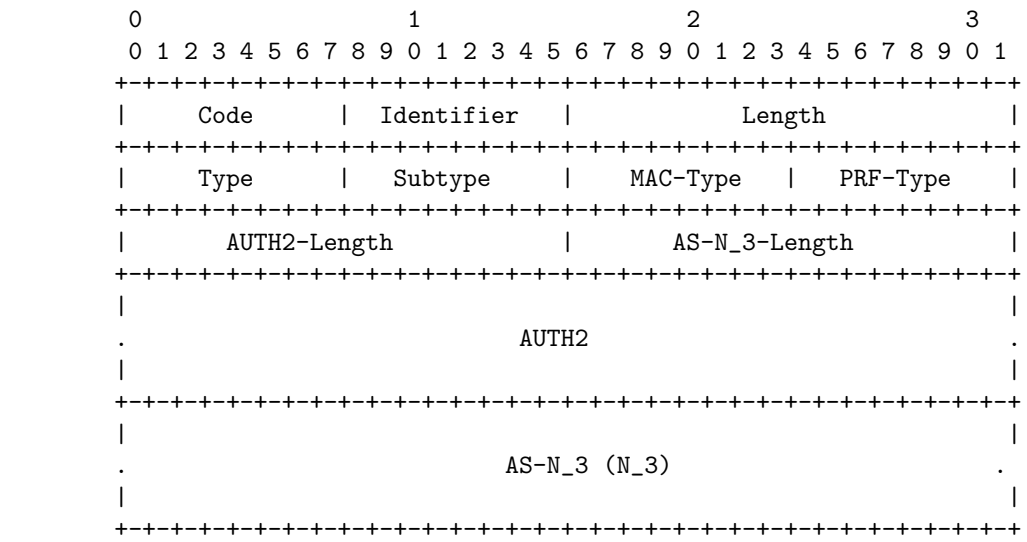
AUTH1

The value AUTH1, computed by MN-SUP as described in section 3.3.

MN-Challenge
The MN Challenge (N_2).

### 7.1.3 EAP-Request/SKE-AS-Verify

The format of the packet EAP-Request/SKE-AS-Verify shown below.




The semantic of the fields is described below:

Code
1 for Request

Identifier
See [2]

Length
The length of the EAP Request packet.

Type

TBD for EAP-SKE

Subtype
3 for SKE-AS-Verify

MAC-Type
The MAC algorithm used by the AS to calculate AUTH2. It MUST be one of the codes listed in section 8.

PRF-Type
The PRF algorithm used by the AS to calculate the EAP Master Secret K_EMS. It MUST be one of the codes listed in section 8.

AUTH2-Length
Length of AUTH2, in 4-byte words.

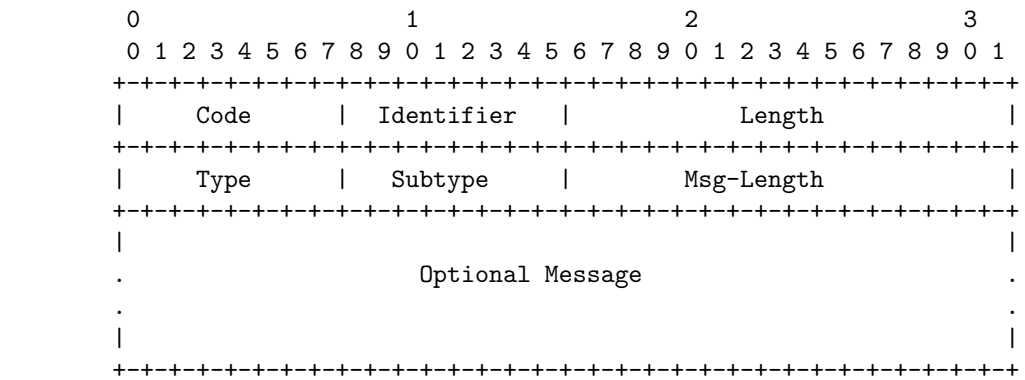
AS-N_3-Length
>=1 and <= 28. Length of AS N_3 nonce, in 4-byte words.

AUTH2
The value AUTH2, computed by AS as described in section 3.3.

AS-N_3
The AS N_3 nonce.

### 7.1.4 EAP-Response/SKE-Success

The format of the packet EAP-Response/SKE-Success shown below.



The semantic of the fields is described below:

Code
2 for Response

Identifier
See [2]

Length
The length of the EAP Response packet.

Type
TBD for EAP-SKE.

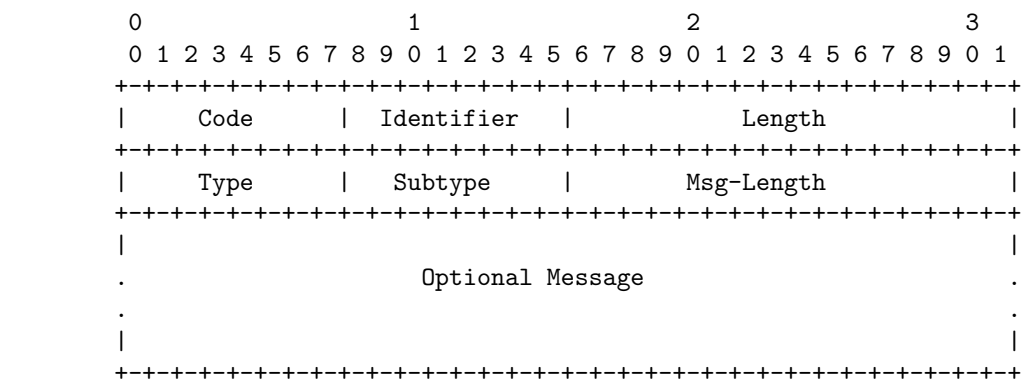
Subtype
4 for SKE-Success.

Msg-Length
Length of optional message in words (0 or more words).

Optional Message
The optional message must be null terminated and padded to ensure its length is multiple of 4 bytes.

### 7.1.5 EAP-Response/SKE-Failure

The format of the packet EAP-Response/SKE-Failure shown below.



The semantic of the fields is described below:

Code
2 for Response

Identifier
See [2]

Length
The length of the EAP Response packet.

Type
TBD for EAP-SKE.

Subtype
5 for SKE-Failure.

Msg-Length
Length of optional message in words (0 or more words).

Optional Message
The optional message must be null terminated and padded to ensure its length is multiple of 4 bytes.

## 7.2 RADIUS messages

SKE parameters are exchanged between the F-AAA and the H-AAA using a RADIUS Access-Request and a RADIUS Access-Accept (in case of successful authentication at the H-AAA), or a RADIUS Access-Reject (in case of unsuccessful authentication).

The RADIUS Access-Request from the F-AAA to the H-AAA MUST contain the following attributes:

- User-Name: containing the user's NAI, copied from the EAP-Response-ID message.
- One Lucent Vendor Specific SKE Attribute: containing N_1, as generated by the F-AAA, AUTH1 and MAC-Type, as copied from the EAP-Response/SKE-MN-Challenge message. The field PRF-Type MUST be '0' in this attribute.

- One Lucent Vendor Specific SKE Attribute: containing N_2, as copied from the EAP-Response/SKE-MN-Challenge message. The fields MAC-Type, PRF-Type and Auth-Length MUST be '0' in this attribute.

The RADIUS Access-Accept from the H-AAA to the F-AAA MUST contain the following attributes:

- One Lucent Vendor Specific SKE Attribute: containing N_3 and AUTH2, as generated by the H-AAA. This attribute also contains the MAC-Type that was used by the H-AAA to generate AUTH2, as well as the PRF-Type that was used by the H-AAA to generate K_EMS.

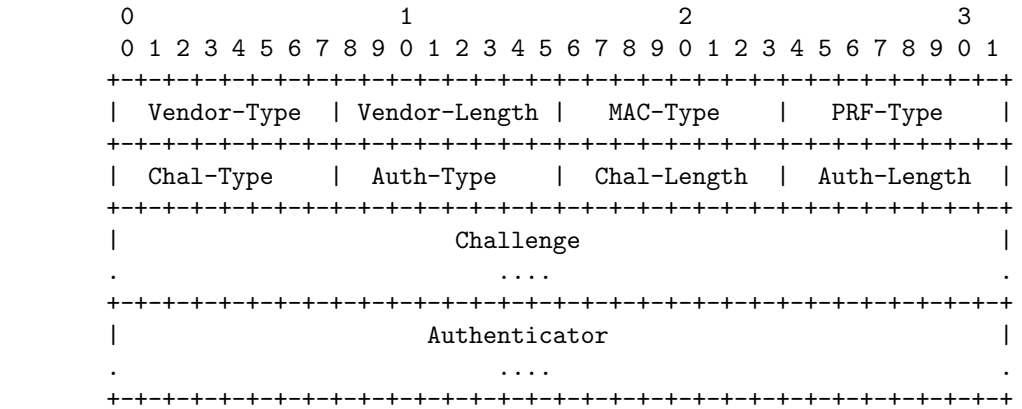
- Master-Session-Key: as defined in [5], containing the Master Session Keys generated by the H-AAA.

The format of the standard attributes used above may be found in [3] or in the relevant documents referenced above. We now define the format of the Lucent Vendor Specific SKE Attribute.

### 7.2.1 RADIUS SKE attribute

The Lucent Vendor Specific SKE attribute is used to carry SKE information elements between the F-AAA and the H-AAA.

A summary of the SKE Attribute format is given below. The fields are transmitted left to right.



Vendor-Type
TBD for SKE-Request.

Vendor-Length
> 8. The length of the message in octets.

MAC-Type
0, or one of the MAC-Type codes listed in section 8.

PRF-Type
0, or one of the PRF-Type codes listed in section 8.

Chal-Type
1 for N_1
2 for N_2
3 for N_3

Auth-Type
0 for NONE
1 for AUTH1
2 for AUTH2

Chal-Length
>= 8. Length of the 'Challenge' field, in octets.

Auth-Length
>= 0. Length of the 'Authenticator' field, in octects.

Challenge
The challenge of the type identified by the field 'Chal-Type'.

Authenticator
The authenticator of the type identified by the field 'Auth-Type'.

## 8 IANA and Protocol Numbering Considerations

IANA has assigned the number TBD for EAP SKE authentication.

EAP SKE messages include a Subtype field. The following Subtypes are specified:

**SKE-AS-Challenge** 1

**SKE-MN-Challenge** 2

**SKE-AS-Verify** 3

**SKE-Success** 4

**SKE-Failure** 5

EAP SKE Subtypes 6-255 are reserved and MUST NOT BE used.

The following codes are defined for the values of the MAC-Type and PRF-Type fields in EAP-SKE messages and in the Lucent Vendor Specific SKE RADIUS Attribute. Other values for these codes MAY be defined in the future.

**MAC-Type** This represents the MAC algorithm that is used by the MN-SUP to generate AUTH1, and by the H-AAA to generate AUTH2. Possible values are:

- 1 for HMAC-SHA1
- 2 for HMAC-MD5

**PRF-Type** This represents the PRF algorithm that is used by the H-AAA to generate K_EMS. Possible values are:

- 1 for HMAC-SHA1
- 2 for HMAC-MD5

## 9 Migration to DIAMETER

In this document the protocol used to transfer the authentication information and the key material from AS-PAE to F-AAA to H-AAA and back is RADIUS, given its wide installed base. Migration to DIAMETER [4] should not present any difficulties, since DIAMETER already provisions mechanisms to collect the same authentication information as RADIUS, and to distribute key material to interested parties. The details of such mechanisms, and how they would be applied to EAP-SKE are outside the scope of this document.

## 10 Open Issues

The scheme as it is described in this document would require the MN to re-authenticate to its H-AAA every time a handoff occurs. The protocol does minimize the number of messages that the MN and H-AAA have to exchange, therefore minimizing the latency of the authentication procedure. However, even 1 RTT to the H-AAA to perform re-authentication could represent a too large latency for certain environments. In such cases, one possible solution would be for the F-AAA, MN and H-AAA to generate arrays of (N_1, N_2, N_3) nonces, and arrays of corresponding (AUTH, K). Such arrays could be cached at the F-AAA and at the MN, so that authentications subsequent to the first one could be performed without the involvement of the H-AAA. Another possible solution would be to perform subsequent authentications between the MN and the F-AAA by using a MAC function applied to random numbers and the Master Session Keys, which would get transferred among AS-PAEs by means of a context-transfer protocol such as the one being defined in the SeaMoby IETF working-group [21].

## Appendix B - Patent Issues

This is to inform you that Lucent Technologies has applied for and/or has patent(s) that relates to the attached submission.

This submission is being made pursuant to the provisions of IETF IPR Policy, RFC 2026, Sections 10.3.1 and 10.3.2.

Lucent Technologies Inc. will offer patent licenses for submissions made by it which are adopted as a standard by your organization as follows:

> If part(s) of a submission by Lucent is included in a standard and Lucent has patents and/or pending applications that are essential to implementation of the included part(s) in said standard, Lucent is prepared to grant - on the basis of reciprocity (grantback) - a license on such included part(s) on reasonable, non-discriminatory terms and conditions.

## 11 Acknowledgments

We would like to thank Peretz Feder, Pete McCann, Simon Mizikovsky and Reuven Shapira from Lucent Technologies for useful discussions on this topic and comments on this draft.

References

[1] Local and Metropolitan Area Networks: Standard for Port Based Network Access Control. Technical report, IEEE P802.1x, 2001.

[2] L. Blunk and J. Volbrecht. PPP Extensible Authentication Protocol (EAP). RFC 2284, IETF, March 1998.

[3] C. Rigney, S. Willens, A. Rubens, and W. Simpson. Remote Authentication Dial In User Service (RADIUS). RFC 2865, IETF, June 2000.

[4] Pat R. Calhoun, Haseeb Akhtar, Jari Arkko, Erik Guttman, Allan C. Rubens, and Glen Zorn. Diameter Base Protocol. Work in progress - Internet Draft, IETF, July 2001. draft-ietf-aaa-diameter-07.txt.

[5] D. Simon. RADIUS Master Session Key Attribute. Work in progress - Internet Draft, IETF, January 2002. draft-simon-radius-key-attr-00.txt.

[6] B. Aboba and D. Simon. The EAP Keying Problem. Work in progress - Internet Draft, IETF, February 2002. draft-aboba-pppext-key-problem-01.txt.

[7] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. RFC 2119, IETF, March 1997.

[8] R. Shirey. Internet Security Glossary. RFC 2828, IETF, May 2000.

[9] B. Aboba and D. Simon. PPP EAP TLS Authentication Protocol. RFC 2716, IETF, October 1999.

[10] G. Tsudik. Message authentication with one-way hash functions. In Proc. INFOCOM'92, 1992.

[11] National Institute of Standards and Technology (NIST). Announcing the Secure Hash Standard. FIPS 180-1, U.S. Department of Commerce, April 1995.

[12] U. Blumenthal. Secure Session Key Generation. Work in progress - Internet Draft, IETF, January 2001. draft-blumenthal-keygen-01.

[13] M. Bellare, R.Canetti, and H. Krawczyk. Keying hash functions for message authentication. In Advances in Cryptology---CRYPTO '96, volume 1109 of Lecture Notes in Computer Science, pages 1--15. Springer-Verlag, 18--22 August 1996.

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

[15] National Institute of Standards and Technology (NIST). Announcing the Advanced Encryption Standard. FIPS ZZZ, U.S. Department of Commerce, February 2001.

[16] O. Goldreich, S. Goldwasser, and Silvio Micali. How to construct random functions. Journal of the ACM, 33(169):210--217, 1986.

[17] W. Diffie, P. van Oorshot, and M. Wiener. Authentication and authenticated key exchange. Designs, Codes and Cryptography, 2(169):107--125, 1992.

[18] H. Krawczyk. Skeme: A versatile secure key exchange mechanism for the internet. In Proc. 1996 Internet Society Symposium on Network and Distributed System Security, pages 114--127, Feb. 1996.

[19] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, IETF, November 1998.

[20] Charles Perkins and Pat Calhoun. AAA Registration Keys for Mobile IP. Work in progress - Internet Draft, IETF, July 2001. draft-ietf-mobileip-aaa-key-08.txt.

[21] H. Syed, G. Kenward, P. Calhoun, M. Nakhjiri, R. Koodli, K. Atwal, M. Smith, and G. Krishnamurthi. General Requirements for a Context Transfer Framework. Work in progress - Internet Draft, IETF, May 2001. draft-ietf-seamoby-ct-reqs-00.txt.

Authors' Addresses

Luca Salgarelli, Milind M. Buddhikot, Juan Garay, Scott Miller,
Uri Blumenthal, Sarvar Patel
Bell Laboratories - Lucent Technologies
101 Crawfords Corner Rd.
Holmdel, NJ 07733, USA

Voice: +1-732-332-6870
Fax: +1-732-949-7397
E-mail: {salga,mbuddhikot,garay,scm,uri,sarvar}@lucent.com

Dorothy Stanley
Agere Systems
2000 North Naperville Rd, Room 5B-441
Naperville, IL 60566, USA
Voice: +1 630 979 1572
Fax: +1 630 979 1572
E-mail: dstanley@agere.com

Peter Dahl, Christopher Carroll
Verizon Wireless
2785 Mitchell Drive, MS 9-2,
Walnut Creek, CA 94598, USA
Voice: +1-925-279-6790
E-mail: {peter.dahl, christopher.carroll}@verizonwireless.com

## 12 Full Copyright Statement