EAP
Internet Draft

H. Tschofenig
D. Kroeselberg
Siemens
Y. Ohba
Toshiba

Document: draft-tschofenig-eap-ikev2-02.txt
Expires: April 2002 October 2003

EAP IKEv2 Method (EAP-IKEv2)

Status of this Memo

Abstract

EAP-IKEv2 is an EAP method which reuses the cryptography and the payloads of IKEv2, creating a flexible EAP method that supports both symmetric and asymmetric authentication. Furthermore protection of legacy authentication mechanisms is supported. This EAP method offers the security benefits of IKEv2 without the goal of establishing IPsec security associations.

Table of Contents

1. Introduction

This document specifies the EAP-IKEv2 authentication method. EAP-IKEv2 is a flexible EAP method which makes the IKEv2 protocolÆs features available for scenarios using EAP-based authentication. The main advantage of EAP-IKEv2 is that it does not define a new cryptographic protocol, but re-uses the IKEv2 authentication protocol, and thereby provides strong, well-analyzed, cryptographic properties as well as broad flexibility. This includes the support of authentication methods and configuration payloads for remote access scenarios.

EAP-IKEv2 can be used directly to mutually authenticate EAP peers. This may be based on either symmetric methods using pre-shared keys, or on asymmetric methods based on public/private key pairs, Certificates and CRLs. In addition, EAP-IKEv2 supports two-phased authentication schemes by establishing a server-authenticated secure tunnel, and by subsequently protecting an EAP authentication allowing for legacy client authentication methods. Hence, EAP-IKEv2 provides a secure EAP tunneling method.

A non-goal of EAP-IKEv2 (and basically the major difference to plain

IKEv2) is the establishment of IPsec security associations, as this would not make much sense in the standard AAA three-party scenario, consisting of an EAP peer, an authenticator (NAS) and a back-end authentication server terminating EAP. IPsec SA establishment may be required locally (i.e., between the EAP peer and some access server). However, SA establishment within an EAP method would only

provide SAs between the EAP peer and the back-end authentication server. Other approaches as e.g., those of the IETF PANA group are considered more appropriate in this case.

2. IKEv2 and EAP-IKEv2 Overview

IKEv2 [Kau03] is a protocol which consists of two exchanges:

(1) an authentication and key exchange protocol which establishes an IKE-SA.

(2) messages and payloads which focus on the negotiation of parameters in order to establish IPsec security associations (i.e., Child-SAs). These payloads contain algorithm parameters and traffic selector fields.

In addition to the above-mentioned parts IKEv2 also includes some payloads and messages which allow configuration parameters to be exchanged primarily for remote access scenarios.

The EAP-IKEv2 method defined by this document uses the IKEv2 payloads and messages used for the initial IKEv2 exchange which establishes an IKE-SA.

IKEv2 provides an improvement over IKEv1 [RFC2409] as described in Appendix A of [Kau03]. Important for this document are the reduced number of initial exchanges, support of legacy authentication, decreased latency of the initial exchange, optional Denial-of-Service (DoS) protection capability and some other fixes (e.g., hash problem). IKEv2 is a cryptographically sound protocol that has received a considerable amount of expert review and that benefits from a long practical experience with IKE.
The goal of EAP-IKEv2 is to inherit these properties within an efficient, secure EAP method.

In addition, IKEv2 provides authentication and key exchange capabilities which allow an entity to use symmetric as well as asymmetric authentication within a single protocol. Such flexibility is considered important for an EAP method and is provided by EAP-IKEv2.

[Per03] provides a good tutorial for IKEv2 design decisions.

EAP-IKEv2 therefore provides

a) well-known IKEv2 symmetric/asymmetric authentication and
b) a new EAP tunneling method.

EAP-IKEv2 provides a secure fragmentation mechanism in which integrity protection is performed for each fragment of an IKEv2 message.

3. Terminology

This document does not introduce new terms other than those defined in [RFC2284] or in [Kau03].

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

4. Protocol overview

This section provides some overview over EAP-IKEv2 message exchanges. Note that some payloads are omitted (such as SAi2 and SAr2) which are mandatory for IKEv2 but are not required in EAP-IKEv2 since they are used to establish an IPsec SA.

IKEv2 uses the same protocol message exchanges for both symmetric and asymmetric authentication. The difference lies only in the computation of the AUTH payload. See Section 2.15 of [Kau03] for more information about the details of the AUTH payload computation. It is even possible to combine symmetric (e.g., from the client to the server) with asymmetric authentication (e.g., from the server to

the client) in a single protocol exchange. Figure 1 depicts such a protocol exchange.

Message exchanges are reused from [Kau03], and are adapted. Since this document does not describe frameworks or particular architectures the message exchange takes place between two parties - between the Initiator (I) and the Responder (R). In context of EAP the Initiator is often called Authenticating Peer whereas the Responder is referred as Authenticator.

The first message flow shows EAP-IKEv2 without the optional DoS protection exchanges. The core EAP-IKEv2 exchange (message (4) - (7)) consists of four messages (two round trips)_only. The first two messages constitute the standard EAP identity exchange and are not mandatory if the EAP server is known.

1) I <-- R: EAP-Request/Identity

2) I --> R: EAP-Response/Identity(Id)

3) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

4) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,0), SAi1, KEi, Ni)

5) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
HDR(A,B), SAr1, KEr, Nr, [CERTREQ])

6) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
HDR(A,B), SK {IDi, [CERT,] [CERTREQ,] [IDr,], AUTH})

7) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
HDR(A,B), SK {IDr, [CERT,] AUTH})

8) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(Finish)

9) I <-- R: EAP-Success

Figure 1: EAP-IKEv2 message flow

The subsequent message flow shows EAP-IKEv2 with DoS protection enabled. The IKEv2 DoS protection mechanism uses cookies and keeps

the responder stateless when it receives the first IKEv2 message, preventing it from performing heavy cryptographic operations based on this first incoming message. As a consequence of DoS protection an additional round trip (message (5) and (6)) is required.

```
1) I <-- R: EAP-Request/Identity

2) I --> R: EAP-Response/Identity(Id)

3) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

4) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR(A,0), SAi1, KEi, Ni)

5) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
         HDR(A,0), N(COOKIE-REQUIRED), N(COOKIE))

6) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
         HDR(A,0), N(COOKIE), SAi1, KEi, Ni)

7) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
         HDR(A,B), SAr1, KEr, Nr, [CERTREQ])

8) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
         HDR(A,B), SK {IDi, [CERT,] [CERTREQ,] [IDr,], AUTH})

9) I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
         HDR(A,B), SK {IDr, [CERT,] AUTH})
```



```
10) I --> R: EAP-Response/EAP-Type=EAP-IKEv2(Finish)

11) I <-- R: EAP-Success
```

Figure 2: EAP-IKEv2 with Cookies

The Secure Legacy Authentication (SLA) EAP message exchange shown in Figure 3 is taken from Section 2.16 of [Kau03] and adapted. It provides an example of a successful inner EAP exchange using the EAP-SIM Authentication method [HS03], which is secured by the IKE-SA.

Implementations MUST ensure that infinite recursions of EAP and EAP-IKEv2 exchanges are not allowed. (TBD: some limit necessary)

```
I <-- R: EAP-Request/Identity

I --> R: EAP-Response/Identity(Id)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(Start)

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
         HDR, SAi1, KEi, Ni)

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(
         HDR, SAr1, KEr, Nr, [CERTREQ])

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
         HDR, SK {IDi, [CERTREQ,] [IDr,]})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR,
         SK {IDr, [CERT,] AUTH, EAP(EAP-Request /SIM
         /Start(AT_VERSION_LIST))})

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-
         Response/SIM/Start(AT_NONCE_MT, AT_SELECTED_VERSION)),
         [AUTH]})

I <-- R: EAP-Request/EAP-Type=EAP-IKEv2(HDR, SK {EAP(EAP-
         Request/SIM/Challenge(AT_RAND, AT_MAC)), [AUTH]})

I --> R: EAP-Response/EAP-Type=EAP-IKEv2(
         HDR, SK {EAP(EAP-Response/SIM/Challenge(AT_MAC) ), [AUTH]})

I <-- R: EAP-Success
```

Figure 3: EAP-IKEv2 SLA with EAP-SIM Authentication

Please note that the message flow in Figure 3 does not include an EAP-Request/Identity and the corresponding EAP-Response/Identity message inside the EAP-IKEv2 tunnel. Although it would be possible to perform such an exchange IKEv2 suggests using the IDi payload for this purpose. As a consequence the initiators identity is not protected against active attacks.

Since the goal of this EAP method is not to establish an IPsec SA some payloads used in IKEv2 are omitted. In particularly the following messages and payloads are not required:

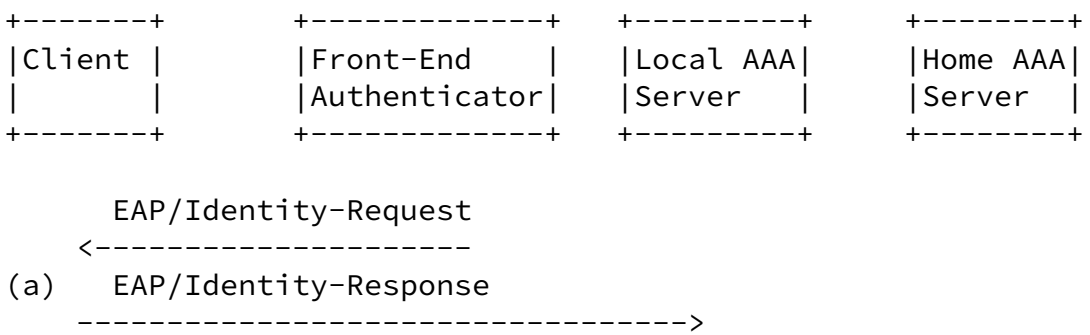
- Traffic Selectors
- IPsec SA negotiation payloads
  (e.g., CREATE_CHILD_SA exchange or SAx2 payloads)
- ECN Notification
- Port handling
- NAT traversal

Some of these messages and payloads are optional in IKEv2. In general it does not make sense to directly negotiate IPsec SAs with EAP-IKEv2, as such SAs were unlikely to be used between the EAP endpoints.

IKEv2 also provides functionality for the initiator to request address information from the responder as described in Section 2.19 of [Kau03]. Using this functionality it is possible for an end host to securely request address configuration information from the local network.

5. Identities used in EAP-IKEv2

A number of different places allow to convey identity information in IKEv2, when combined with EAP. This section describes their function within the different exchanges of EAP-IKEv2. Note that EAP-IKEv2 does not introduce more identities than any other tunneling approach. Figure 4 shows which identities are used during the individual phases of the protocol.



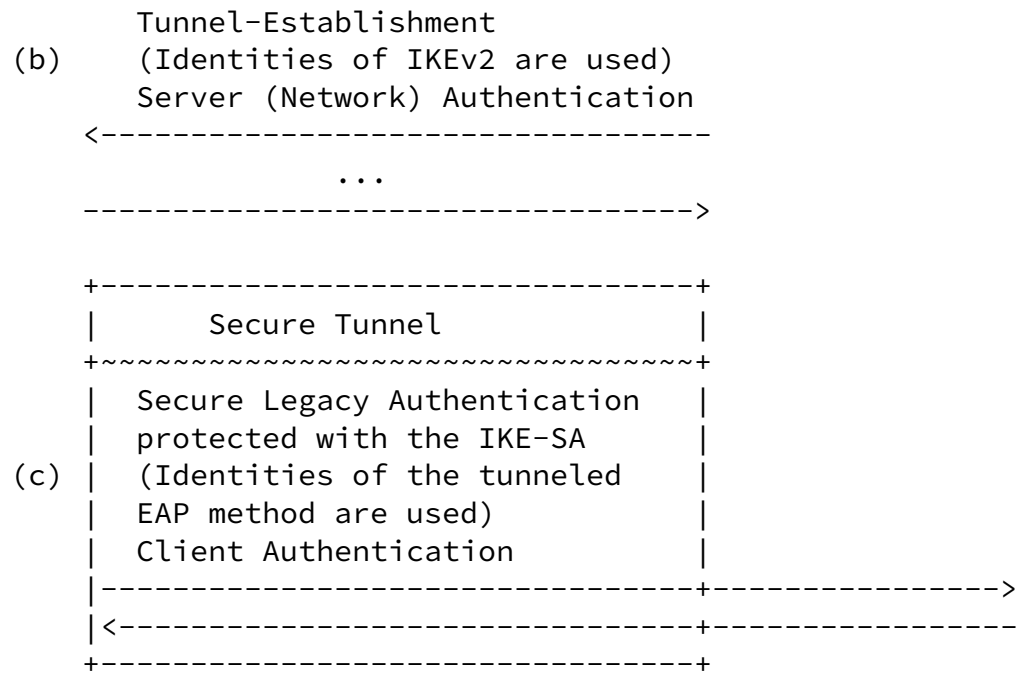


Figure 4: Identities used in EAP-IKEv2

a) The first part of the (outer) EAP message exchange provides information about the identities of the EAP endpoints. This message exchange mainly is an identity request/response. This exchange is optional if the EAP server is known already or can be learned by other means.

b) The identities used within EAP-IKEv2 for both the initiator and the responder. The initiator identity is often associated with a user identity such as a fully-qualified RFC 822 email address. The identity of the responder might be a FQDN. The identity is of importance for authorization.

c) For secure legacy authentication an EAP message exchange is protected with the established IKE-SA as shown in Figure 3. This exchange again adds EAP identities.

This inner EAP message exchange serves the purpose of client authentication. The two identities used thereby are the EAP identity (i.e., a NAI) and possibly a separate identity for the selected EAP method.

The large number of identities is required due to nesting of authentication methods and due to overloaded function of the identity for routing (i.e., authentication end point indication). The number of recursions of EAP and IKEv2 is limited, see Section 4.

Hence with this additional (nested) EAP exchange the end point of the EAP-IKEv2 exchange might not be the same as the end point of the

inner EAP exchange which is protected by the IKE-SA (which in this case is not protected by the IKE-SA any more between the EAP-IKEv2 endpoint and the endpoint of the inner EAP exchange, but might be protected by other means that are not considered in this document).

6. Packet Format

The IKEv2 payloads, which are defined in [Kau03], are embedded into the Data field of the standard EAP Request/Response packets. The Code, Identifier, Length and Type field is described in [RFC2284]. The Type-Data field carries a one byte Flags field following the IKEv2 payloads. Each IKEv2 payload starts with a header field HDR (see [Kau03]).

The packet format is shown in Figure 5.

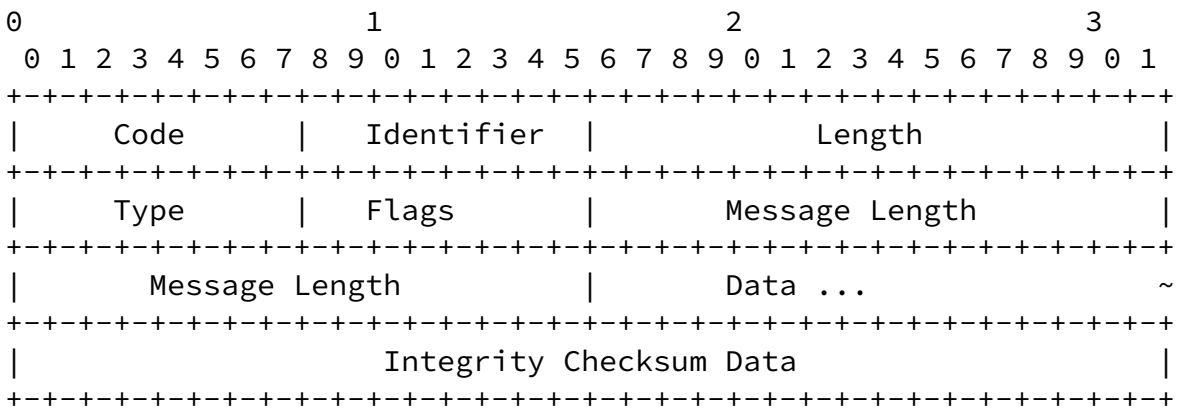
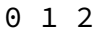


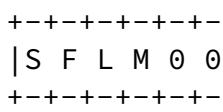
Figure 5: Packet Format

No additional packet formats other than those defined in [Kau03] are required for this EAP method.

The Flags field is required to indicate Start and Finish messages which are required due to the asymmetric nature of IKEv2 and the Request/Response message handling of EAP.

Currently five bits of the eight bit flags field are defined. The remaining bits are set to zero.





S = EAP-IKEv2 start message
F = EAP-IKEv2 finish message

L = Length included
M = More fragments
I = Integrity Checksum Data included

EAP-IKEv2 messages which have neither the S nor the F flag set contain regular IKEv2 message payloads inside the Data field.

With regard to fragmentation we follow the suggestions and descriptions given in Section 2.8 of [PS+03]: The L indicates that a length field is present and the M flag indicates fragments. The L flag MUST be set for the first fragment and the M flag MUST be set on all fragments expect for the last one. Each fragment sent must subsequently be acknowledged.

The Message Length field is four octets long and present only if the L bit is set. This field provides the total message length that is being fragmented (i.e., the length of the Data field.).

The Integrity Checksum Data is the cryptographic checksum of the entire EAP message starting with the Code field through the Data field. This field presents only if the I bit is set. The field immediately follows the Data field without adding any padding octet before or after itself. The checksum MUST be computed for each fragment (including the case where the entire IKEv2 message is carried in a single fragment) by using the same key (i.e., SK_ai or SK_ar) that is used for computing the checksum for the IKEv2 Encrypted payload in the encapsulated IKEv2 message. The Integrity Checksum Data field is omitted for other packets. To minimize DoS attacks on fragmented packets, messages that are not protected SHOULD NOT be fragmented. Note that IKE_SA_INIT messages are the only ones that are not encrypted or integrity protected, however, such messages are not likely to be fragmented since they do not carry certificates.

The EAP Type for this EAP method is <TBD>.

7. Retransmission

Since EAP authenticators support a timer-based retransmission mechanism for EAP Requests and EAP peers retransmit the last valid EAP Response when receiving a duplicate EAP Request message, IKEv2 messages MUST NOT be retransmitted based on timers, when used as EAP authentication method.

8. Key derivation

The EAP-IKEv2 method described in this document generates session keys. These session keys are used to establish an IKE-SA which provides protection of subsequent EAP-IKEv2 payloads. To export a session key as part of the EAP keying framework [AS+03] it is required to derive additional session keys for usage with EAP (i.e., MSK, EMSK and IV). It is good cryptographic security practice to use different keys for different "applications". Hence we suggest reusing the key derivation function suggested in Section 2.17 of [Kau03] to create keying material KEYMAT.

The key derivation function defined is KEYMAT = prf+(SK_d, Ni | Nr), where Ni and Nr are the Nonces from the IKE_SA_INIT exchange.

According to [AS+03] the keying material of MSK, EMSK and IV have to be at minimum 64, 64 and 64 octets long.

The produced keying material for MSK, EMSK and IV MUST be twice the minimum size (i.e., 128 octets).

9. Error Handling

As described in the IKEv2 specification, there are many kinds of errors that can occur during IKE processing (i.e., processing the Data field of EAP-IKEv2 Request and Response messages) and detailed processing rules. EAP-IKEv2 follows the error handling rules specified in the IKEv2 specification for errors on the Data field of EAP-IKEv2 messages, with the following additional rules:

- For an IKEv2 error that triggers an initiation of an IKEv2 exchange (i.e., an INFORMATIONAL exchange), an EAP-IKEv2 message that contains the IKEv2 request that is generated for the IKEv2 exchange MUST be sent to the peering entity. In this case, the EAP message that caused the IKEv2 error MUST be treated as a valid EAP message.

- For an IKEv2 error for which the IKEv2 message that caused the error is discarded without triggering an initiation of an IKEv2 exchange, the EAP message that carries the the erroneous IKEv2 message MUST be treated as an invalid EAP message and discarded as if it were not received at EAP layer.

For an error occurred outside the Data field of EAP-IKEv2 messages, including defragmentation failures, integrity check failures, errors in Flag and Message Length fields, the EAP message that caused the error MUST be treated as an invalid EAP message and discarded as if it were not received at EAP layer.

When the EAP-IKEv2 method runs on a backend EAP server, an outstanding EAP Request is not retransmitted based on timer and thus there is a possibility of EAP conversation stall when the EAP server receives an invalid EAP Response. To avoid this, the EAP server MAY retransmit the outstanding EAP Request in response to an invalid EAP Response. Alternatively, the EAP server MAY send a new EAP Request in response to an invalid EAP Response with assigning a new Identifier and putting the last transmitted IKEv2 message in the Data field.

10. Fast Resume

TLS provides the capability of resuming a session. This offers primarily performance improvement for a new authentication and key exchange protocol run. In order to resume a session two approaches can be taken:

a) Generic approach
b) Method-specific approach

The idea of approach (a) is to

- force each EAP method to create an EAP SA. This SA is kept at the EAP peer and the EAP server and is used for subsequent exchanges.
- built this functionality into EAP itself.

Approach (b) is already used by existing methods using TLS. Choosing (b) does not require any changes to EAP itself since each EAP method has to implement its own mechanism.

So far it has not been decided which approach should be suggested for EAP. In any case it seems that a generic approach contains some difficulties since EAP methods need to negotiate the necessary parameters with are required to build the EAP SA (lifetime, algorithms, identifiers, etc.). Furthermore, it is necessary to cover error cases which happen if the wrong AAA server is selected (due to failover or load balancing) and the EAP SA is not found.

For both cases it is necessary to establish to keep some state information. An additional motivation for establishing state is the ability to provide passive user identity confidentiality as exercised in [AH03]. Subsequent protocol exchanges use a pseudonym instead of the long-term user identity.

Additionally it is necessary to list some requirements for establishing an EAP SA and for running a fast resume. For example, does the fast resume exchange need to provide key agreement or key transport functionality?

Once the above-raised issues have been addressed in the EAP working group a solution will be added to EAP-IKEv2.

11. Security Considerations

The security of the proposed EAP method is intentionally based on IKEv2 [Kau03]. Man-in-the-middle attacks discovered in the context of tunneled authentication protocols (see [AN03] and [PL+03]) are applicable to IKEv2 if legacy authentication with EAP [RFC2284] is used. To counter this threat IKEv2 provides a compound authentication by including the EAP provided session key inside the AUTH payload.

12. Open Issues

The following issues are still under consideration:

- Reducing the number of messages

The message flows given in this document finish with an EAP-Success message. In some cases it might be possible to skip these messages. Furthermore it is possible to omit the first exchange if the identity can be learned by other means.

- Notifications

IKEv2 provides the concept of notifications to exchange messages at any time (e.g., dead peer detection). It remains for further study which of these messages are required for this EAP method.

- Roles of initiator and responder

Figure 4 shows the initiator starting the EAP-IKEv2 exchange. However, there is also the possibility to have the EAP server to start the exchange which saves roundtrips. It remains for further study to analyze the resulting security properties.

13. Normative References

[RFC2284] L. Blunk and J. Vollbrecht: "PPP Extensible Authentication Protocol (EAP)", RFC 2284, March 1998.

[Kau03] C. Kaufman: "Internet Key Exchange (IKEv2) Protocol", internet draft, Internet Engineering Task Force, October 2003. Work in progress.

[RFC2119] S. Bradner: "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Internet Engineering Task Force, March 1997.

14. Informative References

[AN03] N. Asokan, V. Niemi, and K. Nyberg: "Man-in-the-middle in

tunnelled authentication", In the Proceedings of the 11th International Workshop on Security Protocols, Cambridge, UK, April 2003. To be published in the Springer-Verlag LNCS series.

[PL+03] J. Puthenkulam, V. Lortz, A. Palekar, D. Simon, and B. Aboba, "The compound authentication binding problem," internet draft, Internet Engineering Task Force, 2003. Work in progress.

[RFC2409] Harkins, D., Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[Per03] R. Perlman: "Understanding IKEv2: Tutorial, and rationale for decisions", internet draft, Internet Engineering Task Force, 2003. Work in progress.

[AS+03] B. Aboba, D. Simon and J. Arkko: " EAP Key Management Framework", internet draft, Internet Engineering Task Force, October, 2003. Work in progress.

[HS03] H. Haverinen, J. Salowey: "EAP SIM Authentication", internet draft, Internet Engineering Task Force, 2003. Work in progress.

[PS+03] A. Palekar, D. Simon, G. Zorn and S. Josefsson: "Protected EAP Protocol (PEAP)", internet draft, Internet Engineering Task Force, March 2003. Work in progress.

[AH03] J. Arkko and H. Haverinen: "EAP AKA Authentication", internet draft, Internet Engineering Task Force, June 2003. Work in progress.

Acknowledgments

We would like to thank Bernard Aboba, Jari Arkko, Paoulo Pagliusi and John Vollbrecht for their comments to this draft.

Additionally we would like to thank members of the PANA design team (namely D. Forsberg and A. Yegin) for their comments and input to the initial version of the draft.

Finally we would like to thank the members of the EAP keying design

team for their discussion in the area of the EAP Key Management Framework.

Author's Addresses

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Hannes.Tschofenig@siemens.com

Dirk Kroeselberg
Siemens AG
Otto-Hahn-Ring 6
81739 Munich
Germany
EMail: Dirk.Kroeselberg@siemens.com

Yoshihiro Ohba
Toshiba America Research, Inc.
P.O. Box 136
Convent Station, NJ, 07961-0136
USA
Phone: +1 973 829 5174
Email: yohba@tari.toshiba.com

Full Copyright Statement

Acknowledgement

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