2. Abstract

The Point-to-Point Protocol (PPP) [RFC1661] provides a standard method for transporting multi-protocol datagrams over point-to-point links. PPP also defines an extensible Link Control Protocol (LCP), which can be used to negotiate authentication methods, as well as an Encryption Control Protocol (ECP) [RFC1968], used to negotiate data encryption over PPP links, and a Compression Control Protocol (CCP) [RFC1962], used to negotiate compression methods. The Extensible Authentication Protocol (EAP) [EAP] is a PPP extension that provides support for additional authentication methods within PPP.

The Internet Security Association and Key Management Protocol [ISAKMP] combined with the Oakley [Oakley] key exchange protocol enables secure, mutually authenticated security association exchanges between two endpoints. This document describes how ISAKMP provides for these mechanisms within EAP.
3. Introduction

The Extensible Authentication Protocol (EAP), described in [EAP], provides a standard mechanism for support of additional authentication methods within PPP. Through the use of EAP, support for a number of authentication schemes may be added, including smart cards, Kerberos, Public Key, One Time Passwords, and others.

When PPP servers are access via public networks it is desirable for a client to be able to cryptographically prove the identity of the server with which it is establishing a connection. Recently EAP-TLS [EAPTLS] has been proposed as a method of EAP which provides mutual authentication. This document describes an alternative to EAP-TLS which uses the Internet Security Association and Key Management Protocol combined with Oakley (ISAKMP/Oakley [IO]) to provide the mutual authentication.

In addition to mutual authentication, as with EAP-TLS, EAP-ISAKMP provides a means to derive session keys for use with PPP encryption protocols. EAP-ISAKMP has a number of unique characteristics not found in other EAP methods which make it a desirable authentication method:

- **Identity Protection** - When used in Main Mode.
- **Authentication Protocol Independence** - public key or shared secret.
- **Supports Revocation information exchange** - when appropriate ISAKMP authentication method is selected.
- **Easy transition from PAP or CHAP** - User password can become ISAKMP Pre-Shared Secret.
- **Designed specifically for session key establishment and related rekeying.**
- **Prefect Forward Secrecy of Session keys.**
- **Ability to shared ISAKMP SA negotiated during PPP establishment with IPSec if so desired.**
- **Code sharing between IPSEC and EAP.**

Readers of this document should familiarize themselves with ISAKMP [ISAKMP], ISAKMP/Oakley [IO], IPSEC-DOI [DOI], PPP-EAP [EAP], PPP_ECP [ECP], and PPP-CCP [CCP].
3.1 Requirements language Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [RFC 2119].

3.2 Terminology

Initiator - The entity which initiates the ISAKMP exchange. EAP has the notion of an authenticator which is the end requesting authentication. While EAP-ISAKMP provides mutual authentication the Initiator is usually the entity commonly associated as being the authenticator, the network access device (NAS, Router, RAS, etc).

Responder - The entity which is replying to an ISAKMP request. Usually the EAP peer.

SA - Security Association. A grouping of parameters which defines the cryptographic protection to apply to data transmitted or received.

Phase 1 - The exchange during which mutual authentication occurs and keys are derived which are used to protect the subsequent session key establishment phase. Results in ISAKMP SA.

Phase 2 - The exchange during which ECP and CCP algorithms are negotiated. As well keys are derived for ECP.

ISAKMP SA - The SA established as a result of an ISAKMP phase 1 exchange. This SA is used to protect ISAKMP phase 2 and ISAKMP Notify exchanges.

EAP SA - The SA established as a result of an ISAKMP phase 2 exchange. This SA contains session keys for use with ECP (if so desired).

ISAKMP/Oakley - The resolution of ISAKMP combined with Oakley as defined in [IO].

4. Protocol Overview

After EAP is negotiated during Link Establishment phase the initiator sends the first EAP packet with the type field set to EAP-ISAKMP. The data portion contains the first ISAKMP packet consisting of the ISAKMP header followed by ISAKMP Exchange dependent data. The first ISAKMP Exchange SHOULD be an ISAKMP/Oakley phase 1 exchange which authenticates the peers and establishes a protection suite. The protection suite is used during session key establishment and notification exchanges, this is the ISAKMP SA. ISAKMP/Oakley defines two types of phase 1 exchanges MAIN MODE and AGGRESSIVE MODE. MAIN MODE allows for identity protection, AGGRESSIVE MODE requires fewer exchanges but at the cost of identity protection. Pictured below is ISAKMP in Main Mode. Appendix A
contains examples of ISAKMP/Oakley in AGGRESSIVE MODE.

MAIN MODE:

Initiator                               Responder
EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, SA     -->

The Responder replies by choosing an acceptable ISAKMP SA (see [ISAKMP], [IO]) and sends back an EAP Response packet.

Initiator                               Responder

The process continues as outlined in [IO]. As an example detailed below are the remaining exchanges when the ISAKMP authentication mode is RSA Signatures:

Initiator                               Responder

EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, KE, Ni -->

EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, KE, Nr

[CERT, ] SIG_I

Upon completion of this exchange the peers have mutually authenticated each other and have arrived at an acceptable ISAKMP SA. If the Initiator does not wish to negotiate PPP encryption or PPP compression using ISAKMP the EAP Success message should be sent to the Responder to signal that the authentication has completed successfully.

Initiator                               Responder

EAP-Success, uID, Length -->

Carter
The EAP-Success message does not require authentication because this has already been provided within ISAKMP.

Note ECP and CCP negotiation may still take place, however ISAKMP will not be able to provide session keys unless a phase 1 exchange is followed by at least one phase 2 exchange.

If PPP Encryption or Compression is to be used and it is desirable to have session keys for PPP Encryption derived the ISAKMP Initiator MUST begin an ISAKMP/Oakley QUICK_MODE Negotiation after the first ISAKMP SA has been negotiated but BEFORE the first EAP-Success message is sent. Subsequent QUICK_MODE negotiations may occur to freshen the keying material.

4.1 QUICK_MODE

Initiator                               Responder

EAP-Request, uID, Length,              EAP-Response, uID, Length,
EAP-ISAKMP, ISAKMP Hdr*, HASH(1), -->   EAP-ISAKMP Hdr*, HASH(2),

EAP-Request, uID, Length,              EAP-Response, uID, Length,
EAP-ISAKMP, ISAKMP Hdr*, HASH(3) -->   EAP-ISAKMP

EAP-Success, uID, Length  -->

Note that in order to maintain the EAP Request - Response exchange format it is necessary for the responder to reply to the Initiator's final HASH(3) ISAKMP message. This is done by sending an empty EAP - Response message with type set to EAP-ISAKMP. This message as well as the final EAP - Success message do not require authentication because this has taken place within ISKAMP (HASH payloads).

After the QUICK MODE exchange has occurred each entity will possess two security associations, one for inbound traffic and one for outbound traffic. The session keys derived for each direction will be unique.

4.2 ISAKMP with Pre-Shared Keys
When ISAKMP/Oakley is used in MAIN MODE (and only MAIN MODE) and the agreed authentication method is Pre-Shared Keys the Initiator MUST take special action in order to assure that both entities have received valid identity information that can be used to identify the correct pre-shared key to use. Therefore after the SA exchange of the ISAKMP SA negotiation in MAIN MODE if the Initiator determines that the agreed ISAKMP authentication method is pre-shared keys the Initiator MUST send an EAP-Identity/Request message to the peer. The EAP-Identity/Request data portion MUST contain an ISAKMP Identity payload with the Initiators identity. The Responder MUST reply with an EAP-Identity/Response with the data portion encapsulating one ISAKMP Identity payload containing the responders identification information. After the Initiator has received the EAP-Identity/Response it continues on with the ISAKMP SA Negotiation.

* note * -- Because the entities require the identity information prior to ISAKMP key establishment the Identities are sent in the clear, therefore one of the primary benefits of MAIN MODE, Identity Protection, is lost. If the Initiator believes that the ISAKMP authentication method will likely be Pre-Shared Keys the Initiator SHOULD use ISAKMP/Oakley in AGGRESSIVE MODE. Aggressive mode allows the entities to examine the ISAKMP identity before lookup of the pre-shared key, however at the expense of Identity protection.

**Initiator**                               **Responder**

EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, SA  --> <-- EAP-Response, uID, Length,
                                      EAP-ISAKMP, ISAKMP Hdr, SA

Initiator examines agreed SA and determines Pre Shared Keys are to be used, therefore must send own ID and receive responders ID.

EAP-Identity/Request, uID,
EAP-ISAKMP, Length, IDii  -->

                                        <-- EAP-Identity/Response, uID,
                                        Length, EAP-ISAKMP, IDir

At this point both sides have received the otherÂs identity information. The ISAKMP exchange continues.

**Initiator**                               **Responder**

EAP-Request, ID, Length,
EAP-ISAKMP, ISAKMP Hdr, -->
KE, Ni

<-- EAP-Response, ID, Length,
EAP-ISAKMP, ISAKMP Hdr, KE, Nr

EAP-Request, ID, Length,
EAP-ISAKMP, ISAKMP Hdr*, -->
IDii, HASH_I

<-- EAP-Response, ID, Length,
EAP-ISAKMP, ISAKMP Hdr*, IDir, HASH_R

EAP-Success, ID, Length -->

4.3 Informational Exchanges

There are two situations where an Informational exchange may take place within ISAKMP/Oakley. These are:

Failed SA Establishment

All other Notify exchanges, including SA Delete.

4.3.1 Failed SA Establishment

When SA establishment fails the entity which has detected the failure may optionally send an ISAKMP notify message to the peer informing the peer of the failure. In the examples below the failure is assumed to occur during phase 1, hence the notify is sent without protection. If the failure were to occur during phase 2 the notify would be sent encrypted with an additional HASH payload to authenticate the message.

4.3.1.1 Responder Side SA Failure

If the failure is detected on the Responder the ISAKMP Notify message MUST be sent. The ISAKMP Notify message is sent in the EAP - Response packet. Upon receiving the error Notify the Initiator MUST send an EAP-Failure message. The responder must send a notify in order to comply with the EAP Request - Response exchange.

Initiator                               Responder
EAP-Request, uID, Length,             EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, SA            EAP-ISAKMP, ISAKMP Hdr*,
-->
                                          IDir, Notify

Responder detects an error and sends back an ISAKMP Notify.

<-- EAP-Response, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, Notify

EAP - Failure - ID - Length -->

4.3.1.2 Initiator Side SA Failure

If the failure occurs on the Initiator the Initiator may send an ISAKMP Notify message in an EAP - Request message to notify the peer of the error and provide additional information. The Responder MUST reply to the notify with an EAP - Response with an empty data field. The Initiator then replies with a final EAP - Failure message. If the Initiator does not wish to send an ISAKMP Notify message it MUST immediately send an EAP - Failure message.

4.3.1.2.1 Initiator Side SA Failure with Notify

Initiator                        Responder
EAP-Request, uID, Length,       
EAP-ISAKMP, ISAKMP Hdr, SA    -->

<-- EAP-Response, uID, Length, 
   EAP-ISAKMP, ISAKMP Hdr, SA

Initiator detects a failure, sends back notify.

EAP-Request, uID, Length,       
EAP-ISAKMP, ISAKMP Hdr, Notify  -->

<-- EAP-Response, uID, Length

EAP-Failure, uID, Length  -->

4.3.1.2.2 Initiator Side SA Failure without Notify

Initiator                        Responder
EAP-Request, uID, Length,       
EAP-ISAKMP, ISAKMP Hdr, SA    -->

<-- EAP-Response, uID, Length, 
   EAP-ISAKMP, ISAKMP Hdr, SA

Initiator detects a failure, immediately sends back EAP-Failure.

EAP-Failure, uID, Length  -->
4.3.2 Other ISAKMP Notify Exchanges

Other types of ISAKMP Notify messages may be sent at any time with ISAKMP. Each notify is sent in side an EAP - Request packet and MUST be acknowledged with an empty EAP - Response packet. Either side may initiate the notify.

4.4 Security Parameter Index and EAP-ISAKMP

Neither ECP nor CCP require the use of SPIs, however in order to generate unique keying material derived using QUICK MODE both peers MUST supply an SPI. The SPI MUST be 4 octets in length. The Responder MUST choose an SPI value which is not equal to the Initiators. There are no other restrictions on the SPI. [SHOULD THIS GO IN DOI SECTION?]

4.5 Re-keying and EAP-ISAKMP

Since ECP does not use SPIs there needs to be some other way for an entity to signal the other that the most recently negotiated key is now in use. To do this the ECP sequence number is reset to 0 to indicate to the peer that the new key/IV was used to protect the packet. [IS THIS A GOOD IDEA?]

4.6 Fragmentation

Refer to [EAP]. [SUGGESTION - MOVE TLS FRAGMENTATION DESCRIPTION TO BASE EAP DOC]

5. EAP Interaction with ECP and CCP

When EAP is used with ISAKMP the negotiation phase of ECP and CCP can be bypassed, instead providing the ECP and CCP components of PPP the values negotiated with EAP-ISAKMP.

6. EAP ISAKMP DOI

[Note this will become a separate document. For simplicity it is currently combined with this document]

For information on EAP-DOI treatments of situation, security policies, and naming schemes please see the IPSEC DOI. The EAP-DOI adds no new information in these areas.

6.1 EAP-ISAKMP Assigned Numbers

The EAP type for this protocol is ?.

The DOI value for this protocol is ?.
The following table lists the values for the Security Protocol Identifiers referenced in the ISKAMP Proposal Payload for the EAP-ISAKMP DOI:

<table>
<thead>
<tr>
<th>Protocol ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
<tr>
<td>PROTO_ISAKMP</td>
<td>1</td>
</tr>
<tr>
<td>PROTO_PPP_ECP</td>
<td>2</td>
</tr>
<tr>
<td>PROTO_PPP_CCP</td>
<td>3</td>
</tr>
</tbody>
</table>

The size of the field is one octet. The values 2-248 are reserved for IANA. Values 249-255 are reserved for private use.

6.1.1 PROTO_ISAKMP

The PROTO_ISAKMP type specifies message protection required during phase 1 of the ISAKMP protocol. The specific protection mechanism used for the EAP DOI is described in [IO]. All implementations within the EAP DOI MUST support PROTO_ISAKMP.

NB: ISAKMP reserves the value (1) across all DOI definitions.

6.1.2 PROTO_PPP_ECP

The PROTO_PPP_ECP type specifies PPP packet confidentiality usually negotiated during the Encryption Control Protocol phase of PPP establishment.

6.1.3 PROTO_PPP_CCP

The PROTO_PPP_CCP type specifies PPP packet compression usually negotiated during the Compression Control Protocol phase of PPP establishment.

6.2 PPP ISAKMP Transform Values

These values are the same as those defined in the IPSEC DOI section 4.4.2.

6.3 PPP ECP Transform Values

The ECP protocol currently defines two transform values, neither is mandatory, used to provide confidentiality of PPP datagrams. The following table lists the defined PPP ECP transform identifiers for the ISAKMP Proposal Payload for the PPP DOI:
Transform ID | Value
-------------|------
ECP_OUI      | 0
ECP_DESE     | 1
ECP_DESE_BIS | 2

The size of the field is one octet. The values 4-254 are reserved to IANA.

6.3.1 ECP_OUI

The ECP_OUI type specifies a proprietary encryption transform. The ECP_OUI type MUST be accompanied by an Encryption OUI attribute which further identifies the specific vendor algorithm and parameters.

6.3.2 ECP_DESE

The ECP_DESE type specifies the DESE transform detailed in RFC1969. This value may be negotiated for compatibility with older systems. The ECP_DESE type MUST be accompanied by an Encryption Nonce attribute.

6.3.3 ECP_DESE_BIS

The ECP_DESE_BIS type specifies the DESE transform detailed in draft-ietf-pppext-des-encrypt-v2-00.txt. Implementations are encouraged to support this transform. The ECP_DESE_BIS type MUST be accompanied by an Encryption Nonce attribute. Note however that this value is chosen by the Initiator (the responder can not specify a value to use for its IV generation), however since ISAKMP/Oakley in Quick Mode derives unique keys for each entity the resulting IV will be different for traffic to each peer. Therefore uniqueness of IVs is preserved.

6.4 PPP CCP Transform Values

The CCP protocol currently defines a number of transform values, none of which are mandatory, used to provide compression of PPP datagrams. The following table lists the defined PPP CCP transform identifiers for the ISAKMP Proposal Payload for the PPP DOI:
<table>
<thead>
<tr>
<th>Transform ID</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OUI</td>
</tr>
<tr>
<td>1</td>
<td>Predictor type 1</td>
</tr>
<tr>
<td>2</td>
<td>Predictor type 2</td>
</tr>
<tr>
<td>3</td>
<td>Puddle Jumper</td>
</tr>
<tr>
<td>4-15</td>
<td>unassigned</td>
</tr>
<tr>
<td>16</td>
<td>Hewlett-Packard PPC</td>
</tr>
<tr>
<td>17</td>
<td>Stac Electronics LZS</td>
</tr>
<tr>
<td>18</td>
<td>Microsoft PPC</td>
</tr>
<tr>
<td>19</td>
<td>Gandalf FZA</td>
</tr>
<tr>
<td>20</td>
<td>V.42bis compression</td>
</tr>
<tr>
<td>21</td>
<td>BSD LZW Compress</td>
</tr>
<tr>
<td>255</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

The size of the field is one octet. Values 22-254 are reserved by IANA.

6.4.1 CCP_OUI

The CCP_OUI type specifies a proprietary encryption transform. The CCP_OUI type MUST be accompanied by a Compress OUI attribute which further identifies the specific vendor algorithm and parameters.

6.4.2 CCP_PREDICTOR_1

Identifies the compression transform detailed in RFC1978.

6.4.3 CCP_PREDICTOR_2

Identifies the compression transform detailed in RFC1978.

6.4.4 CCP_PUDDLE_JUMPER

Identifies the compression transform detailed in RFC1978.

6.4.5 CCP_HP_PCC

Identifies the compression transform detailed in [HPPPC].

6.4.6 CCP_STAC_LZS

Identifies the compression transform detailed in RFC1974.

6.4.7 CCP_MS_PPC

Identifies the compression transform detailed in RFC2118.

6.4.8 CCP_GANDALF_FZA

Identifies the compression transform detailed in RFC1993.

6.4.9 CCP_V42_BIS
Identifies the compression transform detailed in?

6.4.10 CCP_BSD_LZW
Identifies the compression transform detailed in RFC1977.

6.5 EAP Security Association Attributes

The following SA attribute definitions are used in phase 2 of an ISAKMP/Oakley negotiation. Attribute types can be either Basic (B) or Variable-Length (V). Encoding of these attributes is defined in the base ISAKMP specification.

Attribute Classes:

<table>
<thead>
<tr>
<th>Class</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA Life Type</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>SA Life Duration</td>
<td>2</td>
<td>B/V</td>
</tr>
<tr>
<td>Group Description</td>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>Key Length</td>
<td>4</td>
<td>B</td>
</tr>
<tr>
<td>Key Rounds</td>
<td>5</td>
<td>B</td>
</tr>
<tr>
<td>Compress Dictionary Size</td>
<td>6</td>
<td>B</td>
</tr>
<tr>
<td>Compress OUI</td>
<td>7</td>
<td>V</td>
</tr>
<tr>
<td>Encryption Nonce</td>
<td>8</td>
<td>V</td>
</tr>
<tr>
<td>Encryption OUI</td>
<td>9</td>
<td>V</td>
</tr>
<tr>
<td>Session ID</td>
<td>10</td>
<td>B/V</td>
</tr>
</tbody>
</table>

The size of the attribute class field is two octets, with the high bit reserved for ISAKMP B/V encoding indicator. The values 10 - 32000 are reserved for IANA, values 32001 - 32767 are for experimental use.

Class Values

SA Life Type
SA Duration

Specifies the time to live for the overall security association. When the SA expires, all keys negotiated under the association (PPP_ECP) must be renegotiated. The life type values are:

RESERVED 0
Seconds 1
kilobytes 2

Values 3-61439 are reserved for IANA. Values 61440 - 65535 are for experimental use. For a given life type the value of the life duration attribute defines the actual length of the component lifetime either a number of seconds, or a number of kilobytes that can be
protected/compressed.

If unspecified the default value shall be assumed to be 28800 seconds (8 hours).

Group Description

<table>
<thead>
<tr>
<th>Group Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
<tr>
<td>Group 1 - 768 Prime</td>
<td>1</td>
</tr>
<tr>
<td>Group 2 - 1024 Prime</td>
<td>2</td>
</tr>
</tbody>
</table>

Values 2-61439 are reserved for IANA. Values 61440 - 65535 are for experimental use.

This attribute is used for PFS in phase 2 negotiations. See ISAKMP/Oakley for a description of the Diffie-Hellman parameters which make up these groups. If PFS is desired this attribute MUST be sent.

Key Length

<table>
<thead>
<tr>
<th>Key Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
</tbody>
</table>

There is no default value for Key Length. It is used for ECP Transforms which have variable length keys. However transform documents may define a default value for the particular transform.

Key Rounds

<table>
<thead>
<tr>
<th>Key Rounds</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
</tbody>
</table>

There is no default value for Key Rounds. It is used for ECP Transforms which have variable key rounds. However transform documents may define a default value for the particular transform.

Compression Dictionary Size

<table>
<thead>
<tr>
<th>Compression Dictionary Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
</tr>
</tbody>
</table>

Specifies the log2 maximum size of the dictionary. There is no default value.

Compression OUI

Specifies a private vendor compression algorithm using the vendor's Organization Unique Identifier. The first three (3) octets MUST be the most significant three (3) octets of an Ethernet Physical Address, assigned to the vendor by IEEE 802. The next octet may be vendor specific compression algorithm subtype, followed by zero or more octets.
of vendor data.

This attribute MUST accompany requests for transforms of type CCP_OUI.

Encryption Nonce

Supplied by the Initiator to the Responder. This value is used in the
calculation of the initial IV for PPP_ECP encryption.

This attribute MUST accompany requests for transforms of type ECP_DESE
and ECP_DESE_BIS.

Encryption OUI

Specifies a private vendor encryption algorithm using the vendors
Organization Unique Identifier. The first three (3) octets MUST be the
most significant three (3) octets of an Ethernet Physical Address,
assigned to the vendor by IEEE 802. The next octet may be vendor
specific encryption algorithm subtype, followed by zero or more octets
of vendor data.

This attribute MUST accompany requests for transforms of type ECP_OUI.

Session ID

Multilink sessions may be setup between the EAP peer and the
authenticator. To avoid repeated authentication during multilink setup
the EAP server MAY provide a Session ID during the first EAP SA
establishment (EAP authenticator always initiates first EAP SA). The
EAP peer MAY include the Session ID in subsequent EAP SA negotiations
where it acts as the Initiator.

6.5.1 Required Attribute Support

To ensure basic interoperability all implementations MUST be prepared to
negotiate all of the following attributes:

SA Life Type
SA Duration
Encryption Nonce

6.6 Payload Specifications

This DOI does not define any new payload formats other than those
defined in the IPSEC DOI. See IPSEC DOI.

6.7 EAP Key Exchange Types
This DOI defines no new key exchange types.

7.0 Security Considerations

7.1 Revocation Data

If ISAKMP is used with a certificate scheme supporting revocation an EAP authenticator MUST be prepared to provide revocation lists to the EAP peer if the EAP peer requests those lists within ISAKMP. This allows an otherwise Âoff-lineÂ client to obtain the necessary revocation information it needs to validate the EAP authenticatorÂs certificate.

7.2 Pass Through EAP and EAP Servers

EAP discusses the use of a back-end security server allowing EAP data to pass through the NAS on to the server performing the actual EAP authentication. At this time no IETF standard exists which allows the exchange between the NAS and EAP Server to posses the same degree of security as that offered by ISAKMP/Oakley. Therefore it is REQUIRED that if EAP-ISAKMP is to be used with a back-end security server that the link between the NAS and the security server be protected with IPSec or another comparable protocol which provides full packet authentication and confidentiality. If this condition is meet then the session keys derived on the security server may be passed to the NAS.

8. Acknowledgments

This document is the result of combining EAP with ISAKMP/Oakley, it borrows heavily from EAP-TLS and from IPSEC DOI.

9. Copyright notice

Copyright (C) The Internet Society, 1997. All Rights Reserved.

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

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


11. Authors’ Addresses

Greg Carter
Entrust Technologies
750 Heron Road, Suite E08
Ottawa, Ontario
Canada, K1V 1A7

email: greg.carter@entrust.com
A Aggressive Mode ISAKMP

Detailed below is ISAKMP in AGGRESSIVE MODE using an authentication method of Signatures.

Initiator                               Responder

EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, SA, -->
KE, Ni, IDii

The Responder replies by choosing an acceptable ISAKMP SA (see [ISAKMP], [IO]) and sends back an EAP Response packet.

Initiator                               Responder

<-- EAP-Response, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, SA,
KE, Nr, IDir, [CERT,] SIG_R

EAP-Request, uID, Length,
EAP-ISAKMP, ISAKMP Hdr, -->
[CERT,] SIG_I

<-- EAP-Response, uID, Length

In the final response from the Responder an empty EAP-Response packet is sent.