S. Beaulieu
R. Pereira
Cisco Systems

Internet Draft
Document: <draft-beaulieu-ike-xauth-02.txt>
Expires April 2002

October 2001

Extended Authentication within IKE (XAUTH)

Status of this Memo

1. Abstract

[IKE] allows a device to set up a secure session by using a bidirectional authentication method using either pre-shared keys or digital certificates. However [IKE] does not provide a method to leverage legacy authentication methods which are widely deployed today.

This document describes a method for using existing unidirectional authentication mechanisms such as RADIUS, SecurID, and OTP within IPsec's ISAKMP protocol. The purpose of this draft is not to replace or enhance the existing authentication mechanisms described in [IKE], but rather to allow them to be extended using legacy authentication mechanisms.

This protocol is designed in such a way that extended authentication may be accomplished using any mode of operation for phase 1 (i.e. Main Mode or Aggressive Mode) as well as any authentication method supported by [IKE]. This protocol may also be easily extended to support new modes or authentication methods. This protocol does however require that the phase 1 authentication method be fully secure.

The authors currently intend this document to be published as an Informational RFC, not a standards-track document, so that the many IPsec implementations that have implemented to earlier drafts of this protocol can have a single stable reference.

Comments regarding this draft should be sent to ietf-xauth@vpnc.org or to the authors.

2. 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 [2].

3. Introduction

The following technique allows IPsec's ISAKMP/Oakley [IKE] protocol to support extended authentication mechanisms like two-factor authentication, challenge/response and other remote access unidirectional authentication methods.

These authentication mechanisms have a large deployment in remote access applications and many IT departments have requirements for these unidirectional authentication mechanisms.

This draft defines packet formats for a protocol which allows you to carry legacy authentication information from one peer to another. It does so by extending the [IKECFG] protocol. This protocol requires a sufficient level of security from the phase 1 SA authentication.

This protocol may be used in conjunction with a multitude of combinations of modes (i.e. Main Mode, Aggressive Mode, etc) and authentication methods (i.e. Pre-Shared keys, RSA Signatures, DSS Signatures, etc). This protocol has also been designed to work with any new modes and authentication methods.

This draft also specifies how to accomplish legacy authentication when used with the existing modes and authentication methods defined in IKE (the assumption here being that they offer "sufficient" level of security to protect the XAUTH exchange). This is accomplished by extending the [IKE] protocol.

The document has been published as informational as the IPSRA working group will not accept any protocol which extends ISAKMP or IKE. Furthermore, the IPsec working group refuses to accept any protocols that deal with remote access.

At the time of the writing of this draft, the IPSRA working group has still not defined a protocol to solve the issue of legacy

authentication. XAUTH has been in existance for several years, and has successfully proven interoperability. Several vendors have implemented and deployed the protocol. Several vendors wish to implement the protocol but have had problems finding the protocol specification. For this reason, the draft is being republished as informational to give new vendors an opportunity to interoperate with the many existing vendors who implement this protocol today.

3.1. Changes since last revision.

The last revision of this document was published as "draft-beaulieu-ike-xauth-01.txt"

o clarified text regarding CHALLENGE attribute
o clarified text regarding NEXT-PIN attribute

3.2. Extended Authentication

Two-factor authentication and challenge/response schemes like SDI's SecurID and RADIUS are forms of authentication that allow a gateway, firewall, or network access server to offload the user administration and authentication to a central management server. IPsec's ISAKMP/Oakley protocol supports certificates (RSA & DSS), shared-secret, and Kerberos as authentication methods, but since the authentication methods described within this document are only unidirectional authentication methods (client to a gateway/firewall), they cannot be used by themselves, but must be used in conjunction with the other standard ISAKMP authentication methods.

The technique described within this document utilizes ISAKMP to transfer the user's authentication information (name, password) to the gateway/firewall (edge device) in a secured ISAKMP message. The edge device would then use the appropriate protocol (RADIUS,

SecurID, OTP) to authenticate the user. This allows the authentication server to be within the private network that the edge device is protecting.

3.3. Reader Prerequisites

It is assumed that the reader is familiar with the terms and concepts described in the "Security Architecture for the Internet Protocol" [ArchSec] and "IP Security Document Roadmap" [Thayer97] documents.

Readers are advised to be familiar with both [IKE] and [ISAKMP] as well as [IKECFG] since this document is an extension to that document.

4. Vendor ID

XAUTH currently uses attribute numbers from the private ranges of both [IKE] and [IKECFG]. In order to ensure interoperability with future and past implementations of XAUTH a Vendor ID has been added. The Vendor ID payload is sent during the phase 1 exchange as per [ISAKMP]. The vendor id payload SHOULD be authenticated whenever possible. Two IKE implementations which support the [KIV] document will be capable of doing this. The Vendor ID for this revision of XAUTH is the following 8 bytes.

Vendor ID = 0x09002689DFD6B712

If an implementation receives the aforementioned Vendor ID, it can assume that the peer also has implemented this protocol and therefore is a "mutually consenting party".

If this document ever advances to the standard-track, then new numbers will be assigned by IANA from the appropriate number spaces of [IKE] and [IKECFG], thus eliminating the need for a Vendor ID payload.

5. Extended Authentication Method

This specification allows for extended authentication by allowing an edge device to request extended authentication from an IPsec host

(end-node), thus forcing the host to respond with its extended authentication credentials. The edge device will then respond with a failed or passed message.

When the edge device requests extended authentication, it will specify the type of extra authentication and any parameters required for it. These parameters MAY be the attributes that it requires for authentication and they MAY be information required for the IPsec host's reply (e.g. challenge string).

The Extended Authentication transaction is terminated either when the edge device starts a SET/ACK exchange which includes an XAUTH_STATUS attribute or when the remote device sends a XAUTH_STATUS attribute in a REPLY message. Please note that a remote device can not set XAUTH_STATUS to anything but FAIL.

The edge device MAY request multiple different authentication transactions within one Extended Authentication transaction. This is done by having multiple REQUEST/REPLY pairs, initiated by the edge device, before the transaction is terminated as described above. Each REQUEST/REPLY pair MAY have a different value for XAUTH_TYPE.

As with CHAP [CHAP], this protocol can also be used to periodically authenticate the user during the lifetime of a security association.

If the IPsec host does not have support for the authentication method requested by the edge device, then it would send back a REPLY with the XAUTH_STATUS attribute set to FAIL, thus failing the authentication but completing the transaction.

The Extended Authentication mechanism does not effect the nature of the phase 1 authentication mechanism in any way. Both peers MUST authenticate each other via the authentication methods described in [IKE] or some other authentication method in the ISAKMP framework. There are Security Considerations involved in at least one of the authentication methods in [IKE] and this is described in "Security Considerations" below.

This method provides unidirectional authentication only, meaning that only one device is authenticated using both IKE authentication methods and Extended Authentication.

Here are some types of extended authentication that this

specification supports:

5.1 Simple Authentication

Where a user name and password are required for authentication.

```
IPsec Host                                              Edge Device
--------------                                        -----------------
                                     <-- REQUEST(NAME="" PASSWORD="")
REPLY(NAME="joe" PASSWORD="foobar") -->
                                                    <-- SET(STATUS=OK)
ACK(STATUS) -->
```

Some authentication mechanisms hide the user password by some type of encryption mechanism.

```
IPsec Host                                              Edge Device
--------------                                        -----------------
                             <-- REQUEST(TYPE=RADIUS-CHAP
                             CHALLENGE="123456" NAME="" PASSWORD="")
REPLY(TYPE=RADIUS-CHAP NAME="joe" PASSWORD="E4901AB7") -->
                                                    <-- SET(STATUS=OK)
ACK(STATUS) -->
```

NOTE: This is a conceptual example of RADIUS-CHAP, for a more detailed example, see Appendix A.

5.2 Challenge/Response

Where a challenge from the edge device must be incorporated with the reply. This makes each reply different.

```
IPsec Host                                              Edge Device
--------------                                        -----------------
                                   <-- REQUEST(NAME="" PASSWORD="")
REPLY(NAME="joe" PASSWORD="foobar") -->
               <-- REQUEST(MESSAGE="Enter your password followed by
                                your pin number" NAME="" PASSWORD="")
REPLY(NAME="joe" PASSWORD="foobar0985124") -->
                                                    <-- SET(STATUS=OK)
ACK(STATUS) -->
```

If, however, the edge device knows that a challenge will be required it may skip the first exchange as follows:

```
IPsec Host                                                Edge Device
--------------                                          -----------------
               <-- REQUEST(MESSAGE="Enter your password followed by
                                  your pin number" NAME="" PASSWORD="")
REPLY(NAME="joe" PASSWORD="foobar0985124") -->
                                                      <-- SET(STATUS=OK)
ACK(STATUS) -->
```

5.3 Two-Factor Authentication

This authentication method combines something the user knows (their password) and something that the user has (a token card).

```
IPsec Host                                                  Edge Device
--------------                                            -----------------
                       <-- REQUEST(NAME="" PASSWORD="" PASSCODE="")
REPLY(NAME="joe" PASSWORD="foobar" PASSCODE="3412") -->
                                                      <-- SET(STATUS=OK)
ACK(STATUS) -->
```

Some mechanisms allow for another optional request of the passcode.

```
IPsec Host                                                  Edge Device
--------------                                            -----------------
                       <-- REQUEST(NAME="" PASSWORD="" PASSCODE="")
REPLY(NAME="joe" PASSWORD="foobar" PASSCODE="323415") -->
                       <-- REQUEST(NAME="" PASSWORD="" PASSCODE="")
REPLY(NAME="joe" PASSWORD="foobar" PASSCODE="513212") -->
                                                      <-- SET(STATUS=OK)
ACK(STATUS) -->
```

5.4 One-Time-Password

Similar to the Challenge/Response method, this method allows authentication that is secure against passive attacks based on replaying captured passwords.


```
IPsec Host                                                Edge Device
--------------                                          -----------------
                           <-- REQUEST(TYPE=OTP CHALLENGE, NAME="")
REPLY(TYPE=OTP_CHALLENGE, NAME="joe")-->
                <-- REQUEST(TYPE=OTP CHALLENGE="otp-md5 499 ke1234"
```

```
                                                    NAME="" PASSWORD="")
   REPLY(TYPE=OTP NAME="joe" PASSWORD="5bf0 75d9 959d 036f") -->
                                                      <-- SET(STATUS=OK)
   ACK(STATUS) -->
```

5.5 User Previously Authenticated

Some situations may occur where the edge device has already authenticated the host and no new authentication is required. This may happen when either the host or the edge device must rekey an existing phase 1 SA. It is important that this method not be used, unless the implementation can be sure that the current phase 1 SA was created with the same peer as the initial phase 1 SA, which was previously authenticated using XAUTH. There is currently no way defined to ensure that two separate phase 1 SAs actually belong to the same peer. One method suggested is to use the ID from the phase 1 negotiation (available in Main Mode and Aggressive Mode) but only if the ID is unique to the user and cannot not be forged. This concept is herein referred to as "ID-Checking".

Implementation hint:

o In order to accomplish ID-Checking for Phase 1 Authenticated With a Pre-Shared Key (as defined in [IKE]), the pre-shared key lookup must be based on the phase 1 ID. Please note that this method only currently works for Aggressive Mode, and may work with modes defined in the future. A static IP address could also be used for shared secret lookup, however, the binding of the user to XAUTH session would have to use the IP address instead of the ID.

o In order to accomplish ID-Checking for IKE Phase 1 Authenticated With Signatures (as defined in [IKE]), the implementation must ensure that the ID provided in the phase 1 exchange matches the ID in the peer's certificate which must be signed by a trusted third party.

In the situation where the peer does not require additional authentication, the following method is used.

```
   IPsec Host                                            Edge Device
   -------------                                       ----------------
                                                      <-- SET(STATUS=OK)
   ACK(STATUS) -->
```

5.6 Other Useful Examples

More useful examples are found in Appendix A.

6 Extensions to ISAKMP-Config

This protocol uses the mechanisms described in ISAKMP-Config [IKECFG] to accomplish its authentication transaction. This protocol uses Configuration Attributes from the private range of Isakmp-Config [IKECFG]. To ensure interoperability with past and future versions of Extended Authentication, a Vendor ID is provided in section 2.

All ISAKMP-Config messages in an extended authentication transaction MUST contain the same ISAKMP-Config transaction identifier. The Message ID in the ISAKMP header follows the rules defined by the ISAKMP-Config protocol.

This protocol can therefore be used in conjunction with any existing basic ISAKMP authentication method as defined in [IKE].

This authentication MUST be used after a phase 1 exchange has completed and before any other exchange with the exception of Info mode exchanges. If the extended authentication fails, then the phase 1 SA MUST be immediately deleted. The edge device MAY choose to retry an extended authentication request if the user failed to be authenticated, but must do so in the same ISAKMP-Config transaction, and MUST NOT send the SET message until the user is authenticated, or until the edge device wishes to stop retrying and fail the user.

Extended Authentication MAY be initiated by the edge device at any time after the initial authentication exchange. For example, RADIUS servers may specify that a user only be authenticated for a certain time period. Once that time period has elapsed (minus a possible jitter), the edge device may request a new Extended Authentication exchange. If the Extended Authentication exchange fails, the edge device MUST tear down all phase 1 and phase 2 SAs associated with the user.

The following are extensions to the ISAKMP-Config [IKECFG] specification to support Extended Authentication.

6.1 Message Types

Type	Value
ISAKMP-CFG-REQUEST	(as defined in [IKECFG])
ISAKMP-CFG-REPLY	(as defined in [IKECFG])
ISAKMP-CFG-SET	(as defined in [IKECFG])

ISAKMP-CFG-ACK (as defined in [IKECFG])

ISAKMP-CFG-REQUEST - This message is sent from an edge device to an IPsec host trying to request extended authentication. Attributes that it requires sent back in the reply MUST be included with a length of zero (0). Attributes required for the authentication reply, such as a challenge string MUST be included with the proper values filled in.

ISAKMP-CFG-REPLY - This message MUST contain the filled in authentication attributes that were requested by the edge device or if the proper authentication attributes can not be retrieved, then this message MUST contain the XAUTH-STATUS attribute with a value of FAIL.

ISAKMP-CFG-SET - This message is sent from an edge device and is only used, within the scope of this document, to state the success of the authentication. This message MUST only include the success of failure of the authentication and MAY contain some clarification text.

ISAKMP-CFG-ACK - This message is sent from the IPsec host acknowledging receipt of the authentication result. Its attributes are not relevant and MAY be skipped entirely, thus no attributes SHOULD be included. This last message in the authentication transaction is used solely as an acknowledgement of the previous message and to eliminate problems with unacknowledged messages over UDP.

6.2 Attributes

Attribute	Value	Type
XAUTH-TYPE	16520	Basic
XAUTH-USER-NAME	16521	Variable ASCII string
XAUTH-USER-PASSWORD	16522	Variable ASCII string
XAUTH-PASSCODE	16523	Variable ASCII string
XAUTH-MESSAGE	16524	Variable ASCII string
XAUTH-CHALLENGE	16525	Variable ASCII string
XAUTH-DOMAIN	16526	Variable ASCII string
XAUTH-STATUS	16527	Basic
XAUTH-NEXT-PIN	16528	Variable ASCII string
XAUTH-ANSWER	16529	Variable ASCII string

NOTE: Variable ASCII strings need not be NULL-terminated, as the

length field in the attribute header is sufficient to properly format the strings.

XAUTH-TYPE - The type of extended authentication requested whose values are described in the next section. This is an optional attribute for the ISAKMP_CFG_REQUEST and ISAKMP_CFG_REPLY messages. If the XAUTH-TYPE is not present, then it is assumed to be Generic. The XAUTH-TYPE in a REPLY MUST be identical to the XAUTH-TYPE in the REQUEST. If the XAUTH-TYPE was not present in the REQUEST, then it MUST NOT be present in the REPLY. However, an XAUTH transaction MAY

have multiple REQUEST/REPLY pairs with different XAUTH-TYPE values in each pair.

XAUTH-USER-NAME - The user name MAY be any unique identifier of the user such as a login name, an email address, or a X.500 Distinguished Name.

XAUTH-USER-PASSWORD - The user's password.

XAUTH-PASSCODE - A token card's passcode.

XAUTH-MESSAGE - A textual message from an edge device to an IPsec host. The message may contain a textual challenge or instruction. An example of this would be "Enter your password followed by your pin number". The message may also contain a reason why authentication failed or succeeded. This message SHOULD be displayed to the user.

XAUTH-CHALLENGE - A challenge string sent from the edge device to the IPsec host for it to include in its calculation of a password. This attribute SHOULD only be sent in an ISAKMP-CFG-REQUEST message. Typically, the XAUTH-TYPE attribute dictates how the receiving device should handle the challenge. For example, RADIUS-CHAP uses the challenge to hide the password. The XAUTH-CHALLENGE attribute MUST NOT be used when XAUTH-TYPE is set to generic.

XAUTH-DOMAIN - The domain to be authenticated in. This value will have different meaning depending on the authentication type.

XAUTH-STATUS - A variable that is used to denote authentication success (OK=1) or failure (FAIL=0). This attribute MUST be sent in the ISAKMP-CFG-SET message, in which case it may be set to either OK or FAIL, and MAY be sent in a REPLY message by a remote peer, in which case it MUST be set to FAIL.

XAUTH-NEXT-PIN - A variable which is used when the edge device is

requesting that the user choose a new pin number. This attribute MUST NOT be used in conjunction with any attributes other than XAUTH-MESSAGE and / or XAUTH-TYPE.

XAUTH-ANSWER - A variable length ASCII string used to send input to the edge device. An edge device MAY include this attribute in a REQUEST message in order to prompt an answer from the user, though it MUST be accompanied by an XAUTH-MESSAGE attribute. This attribute MUST NOT be used in conjunction with any attributes other than XAUTH-TYPE or XAUTH-MESSAGE.

6.3 Authentication Types

Value	Authentication Required
0	Generic
1	RADIUS-CHAP
2	OTP
3	S/KEY
4-32767	Reserved for future use
32768-65535	Reserved for private use

Generic - A catch-all type that allows for future extensibility and a generic mechanism to request authentication information. This method allows for any type of extended authentication which does not require specific processing, and should be used whenever possible. This is the default setting if no XAUTH_TYPE is present.

RADIUS-CHAP - RADIUS-CHAP is one method of authentication defined in [RADIUS] which uses a challenge to hide the password. In order to use the CHAP functionality defined in [RADIUS], the XAUTH_TYPE MUST be set to RADIUS-CHAP. For all other methods defined in [RADIUS] (i.e. PAP), the XAUTH_TYPE MUST be set to Generic.

OTP - One-Time-Passwords as defined in [OTP] uses a challenge string to request a certain generated password. The request SHOULD contain a user name, password and a challenge string while the reply MUST contain the user name and the generated password. The challenge string is formatted as defined in [OTPEXT].
S/KEY - This one-time-password scheme defined in [SKEY] was the precursor to OTP, thus the same rules applies.

7. XAUTH Notification

It is important the edge device be able to notify the remote device of its intent to prompt for extended authentication. If such a mechanism were not present, the remote device would send a Quick Mode message, or a Mode-Cfg message before authentication was complete, and the state machines would get pretty complicated.

We present here two methods of accomplishing this. The first is the simplest, and most intuitive. However it is not possible to achieve this within the [IKE] protocol as it stands today, and is therefore not recommended. It has been added to this document for completeness, and may be used in future versions of this document.

7.1 Notification payloads within a phase 1 exchange

The following method is used to notify the remote device that an XAUTH exchange will follow the phase 1 exchange. Once the edge device does a policy lookup for the peer, the edge device appends a notify payload to any phase 1 exchange packet, indicating that an XAUTH exchange will follow. Note, that this notify payload is unauthenticated unless both devices support the mechanisms described in [KIV]. Therefore, implementations MUST NOT use this method unless they are also using the mechanisms described in [KIV].

Once again, this method is not part of the XAUTH protocol in its present form. It has only been added here for completeness, and may be used in future versions of this document.

No payload definitions, assigned numbers, or vendor ID payloads will be provided for this method, as it is currently not part of the XAUTH protocol. These may be defined in the future if enough interest is shown, and if [KIV] becomes a standardized within the IPsec working group.

7.2 Notification via Authentication Method Types

The following method is used to negotiate the use of XAUTH via the SA payload. New authentication methods are defined which allow the edge device to choose an authentication method which mandates XAUTH. This allows the edge device to notify the remote device that an XAUTH exchange will follow the phase 1 exchange. Edge devices which conform to this document MUST support this method.

The following values relate to the ISAKMP authentication method

attribute used in proposals. They optionally allow an XAUTH implementation to propose use of extended authentication after the initial phase 1 authentication. Values are taken from the private use range defined in [IKE] and should be used among mutually consenting parties. To ensure interoperability and avoid collisions, a Vendor ID is provided in the "Vendor ID" section of this document.

Method	Value
XAUTHInitPreShared	65001
XAUTHRespPreShared	65002
XAUTHInitDSS	65003
XAUTHRespDSS	65004
XAUTHInitRSA	65005
XAUTHRespRSA	65006
XAUTHInitRSAEncryption	65007
XAUTHRespRSAEncryption	65008
XAUTHInitRSARevisedEncryption	65009
XAUTHRespRSARevisedEncryption	65010

An Extended Authentication proposal has two characteristics.

The first is the direction of the authentication. Each type identifies whether the Initiator or the Responder is the device which should be authenticated using XAUTH. For example XAUTHInitPreShared is a type which demands that the Initiator be authenticated.

Note that an edge device would typically initiate with one of the following:

- XAUTHRespPreShared
- XAUTHRespDSS
- XAUTHRespRSA
- XAUTHRespRSAEncryption
- XAUTHRespRSARevisedEncryption

and would typically only accept proposals with the following authentication methods:

- XAUTHInitPreShared
- XAUTHInitDSS
- XAUTHInitRSA
- XAUTHInitRSAEncryption
- XAUTHInitRSARevisedEncryption

The second characteristic is the IKE Authentication method to be used. The following table illustrates which keywords in the methods described above relate to which Authentication Methods described in [IKE] Appendix A.

```
"PreShared"            -> pre-shared key
"DSS"                  -> DSS signatures
"RSA"                  -> RSA signatures
"RSAEncryption"        -> Encryption with RSA
"RSARevisedEncryption" -> Revised encryption with RSA
```

8. Other Scenarios for Extended Authentication

Although this document described a scenario where an IPsec host (eg. mobile user) was being authenticated by an edge device (eg. firewall/gateway), the methods described can also be used for edge device to edge device authentication as well as IPsec host to IPsec host authentication.

9. Extensibility

Although this protocol was initially developed for the corporate "Road Warrior" with a dynamic IP address to connect to a corporate Net, there may be certain applications where static IP addresses are used by the "Road Warrior" or where this protocol is used in a non remote-user environment where the IP address is static. There are Security Considerations for certain applications of this protocol in certain deployment scenarios. Please consult the "Security Considerations" section below for more detail.

[IKE] defines many different ways to authenticate a user and generate keying material. There are two basic phase 1 modes defined: Main Mode and Aggressive Mode. There are also at least 5 different authentication schemes which can be used with each mode.

New authentication schemes are being developed and surely more will be standardized in the future. Similarly new phase 1 modes are being proposed to address weaknesses or missing functionality in Main Mode and/or Aggressive mode.

It is for this reason that XAUTH was designed to be fully

extensible. Since XAUTH extends the phase 1 authentication provided by [IKE], it is an important design goal that a legacy user authentication scheme in IPsec be able to use the strengths of current and future authentication and key generation schemes.

XAUTH accomplishes this by working with all modes which allow the negotiation of a phase 1 authentication method in ISAKMP. Any new authentication methods defined in the future which are not addressed by this document need simply to take values from the "consenting parties" ranges of [IKE]. Such an example would be the introduction of Encryption with El-Gamal and Revised Encryption with El-Gamal, and [HYBRID]. Furthermore, any new modes defined such as Base Mode, will automatically be able to use the functionality of XAUTH as no new numbers are needed.

Finally, any new or forgotten Legacy User Authentication Schemes which are not part of XAUTH can be easily incorporated by taking numbers from the "consenting parties" ranges of XAUTH, or by requesting reserved numbers from IANA.

10. Security Considerations

Care should be taken when sending sensitive information over public networks such as the Internet. A user's password should never be sent in the clear and when sent encrypted, the destination MUST have been previously authenticated. The use of ISAKMP-Config [IKECFG] addresses these issues.

The protocol described in this memo strictly extends the authentication methods described in [IKE]. It does not in any way affect the authenticated nature of the phase 1 security association. In fact, this protocol heavily relies on the authenticated nature of the phase 1 SA. Without complete phase 1 authentication, this protocol does not provide protection against man-in-the-middle attacks. Therefore it MUST NOT be used without normal phase 1 authentication. This protocol was designed to be extensible, and can be used in many possible combinations of phase 1 Modes and authentication methods. However, certain combinations of scenarios could lead to weaker than desired security, and are therefore discouraged.

When using XAUTH with Pre-Shared keys, where the peer's IP address is dynamic, Main Mode SHOULD NOT be used, and is STRONGLY DISCOURAGED. In this particular scenario, the phase 1 authentication becomes suspect as the administrator has little choice but to use

one single Shared-Key for all users, and group-shared keys are susceptible to "social engineering attacks".

However, the choice of implementation of this functionality is left up to the implementers of this protocol. There may be some applications where this functionality is desired. Some examples are: proof of concept deployments and small deployments where the proper management of a group shared-key is less difficult.

If at some point restrictions are introduced in one of the IPsec Standard RFC documents which prohibit the use of group pre-shared keys, then this protocol will, by default, conform, and these Security Considerations will no longer be of concern.

11. References

[1] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

[CHAP] W. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC1994

[DIAMETER] P. Calhoun, A. Rubens, "DIAMETER - Base Protocol", draft-calhoun-diameter-02.txt

[HYBRID] M. Litvin, R. Shamir, T. Zegman, "A Hybrid Authentication Mode for IKE", draft-ietf-ipsec-isakmp-hybrid-auth-05

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

[IKECFG] D. Dukes, R. Pereira, "The ISAKMP Configuration Method", draft-dukes-ike-mode-cfg-01.txt

[KIV] Kivinen, T., "Fixing IKE Phase 1 & 2 Authentication HASHs", "draft-ietf-ipsec-ike-hash-revised-01.txt", work in progress.

[OTP] N. Haller, C. Metz, P. Nesser, M. Straw, "A One-Time Password System", RFC2289

[OTPEXT] C. Metz, "OTP Extended Responses", RFC 2243

[RADIUS] C. Rigney, A. Rubens, W. Simpson, S. Willens, "Remote

[SKEY] N. Haller, "The S/KEY One-Time Password System", RFC1760

[TACACS] C. Finseth, "An Access Control Protocol, Sometimes Called TACACS", RFC1492

[TACACS+] D. Carrel, L. Grant, "The TACACS+ Protocol Version 1.77", draft-grant-tacacs-01.txt

12. Acknowledgments

The authors would like to thank Tamir Zegmen, Moshe Litvin, Dan Harkins and all those from the IPsec community who have helped improve the XAUTH protocol. We would also like to thank Tim Jenkins, Ajai Puri, Laurie Shields, Andrew Krywaniuk, Gabriela Dinescu, Paul Kierstead and Scott Fanning for their continued support, and many sanity checks along the way.

13. Author's Addresses

Stephane Beaulieu
<stephane@cisco.com>
Cisco Systems Co.
+1 (613) 721-3678

Roy Pereira
<royp@cisco.com>
Cisco Systems
+1 (408) 526-6793

14. Expiration

This draft expires April, 2002

Full Copyright Statement

Appendix A

This appendix gives more useful examples of Extended Authentication.

SDI through RADIUS
==========

The following 3 examples show examples of SDI running through RADIUS. Since the edge device does not necessarily know that we are indeed doing SDI, the edge device will typically send everything in terms of Username and Password. This of course results in the user being prompted with a password dialog when it isn't really a password which is required. This tends to be a little confusing, but it is really a limitation of RADIUS.

NOTE: The edge device may choose to try and detect these situations and send better suited XAUTH attributes (such as XAUTH ANSWER or XAUTH NEXT PIN). The Client is typically protocol agnostic and will prompt the user for whatever attributes the edge device requests.

Example A-1:
==========

Secure ID Next PIN mode via RADIUS (Scenario 1 - SDI generated next pin)

```
IPsec Client                                          IPsec Gateway
------------                                          -------------
                          <-- REQUEST(Username = '', Password = '')
REPLY(Username = 'joe', Password = '1637364856') -->
                     <-- REQUEST(Username = '', Password = '',
```

```
                        XAUTH_MESSAGE = 'The system has assigned you a
                        new PIN of '1234', please re-enter your
                        username and password')
   REPLY(Username = 'joe', Password = '1234764456') -->
                                              <-- SET(XAUTH_STATUS = OK)
   ACK(XAUTH_STATUS) -->

   Example A-2:
   ============
   Secure ID Next PIN mode via RADIUS (Scenario 2 - User generated next
   pin)

   IPsec Client                                          IPsec Gateway
   ------------                                          -------------
                             <-- REQUEST(Username = '', Password = '')
   REPLY(Username = 'joe', Password = '1637364856') -->
                        <-- REQUEST(Username = '', Password = '',
                        XAUTH_MESSAGE = 'Enter your new PIN containing
                        4-6 digits')
   REPLY(Username = 'joe', Password = '1234') -->
```



```
                              <-- REQUEST(Username = '', Password = '')
   REPLY(Username = 'joe', Password = '1234764456') -->
                                              <-- SET(XAUTH_STATUS = OK)
   ACK(XAUTH_STATUS) -->


   Example A-3:
   ============

   Secure ID Next PIN mode via RADIUS (Scenario 3 - RADIUS server
   offers choice of generating new PIN)

   IPsec Client                                          IPsec Gateway
   ------------                                          -------------
                             <-- REQUEST(Username = '', Password = '')
   REPLY(Username = 'joe', Password = '1637364856') -->
                        <-- REQUEST(Username = '', Password = '',
                        XAUTH_MESSAGE = 'You must start using a new
                        PIN.  Would you like to generate your own PIN
                        (y/n)?)
   REPLY(Username = 'joe', Password = 'y') -->
                        <-- REQUEST(Username = '', Password = '', XAUTH
                        MESSAGE = 'Enter your new PIN containing 4-6
                        digits')
   REPLY(Username = 'joe', Password = '1234') -->
```

```
                              <-- REQUEST(Username = '', Password = '')
   REPLY(Username = 'joe', Password = '1234764456'
                                          <-- SET(XAUTH_STATUS = OK)
   ACK(XAUTH_STATUS) -->
```

Native SDI
==========

When doing native SDI between the edge device and the SDI server, the edge device has more information about what type of information is required from the user. The edge device can therefore use more intuitive attributes in certain situations as compared with the RADIUS examples above.

Example A-4:
============
Secure ID Next PIN mode(Scenario 1 - SDI generated next pin)

```
   IPsec Client                                          IPsec Gateway
   ------------                                          -------------
                            <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1637364856') -->
                        <-- REQUEST(Username = '', Passcode = '',
                        XAUTH_MESSAGE = 'The system has assigned you a
```



```
                        new PIN of '1234', please re-enter your
                        username and passcode')
   REPLY(Username = 'joe', Passcode = '1234764456') -->
                                              <-- SET(STATUS = OK)
   ACK(STATUS) -->
```

Example A-5:
============

Secure ID Next PIN mode(Scenario 2 - User generated next pin)

```
   IPsec Client                                          IPsec Gateway
   ------------                                          -------------
                            <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1637364856') -->
                        <-- REQUEST(NEXT PIN = '', XAUTH_MESSAGE =
                        'Enter your new PIN containing 4-6 digits')
   REPLY(NEXT_PIN = '1234') -->
```

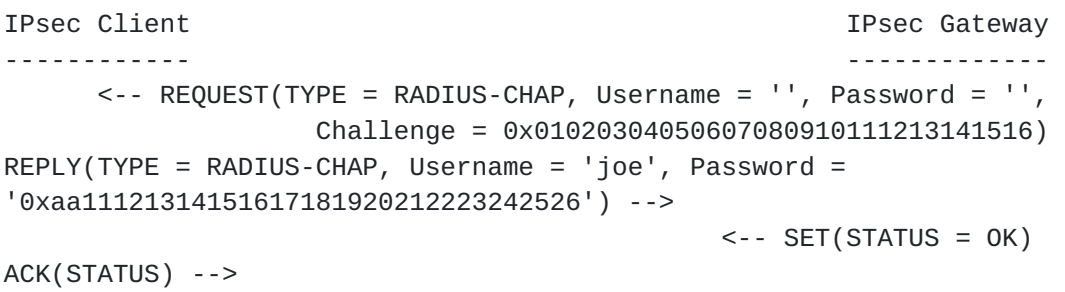
```
                            <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1234764456') -->
                                                  <-- SET(STATUS = OK)
   ACK(STATUS) -->


   Example A-6:
   ============
   Secure ID Next PIN mode(Scenario 3 - SDI server offers choice of
   generating new PIN)

   IPsec Client                                          IPsec Gateway
   ------------                                          -------------
                            <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1637364856') -->
                        <-- REQUEST(ANSWER = '', XAUTH_MESSAGE = 'You
                        must start using a new PIN.  Would you like to
                        generate your own PIN (y/n)?)
   REPLY(ANSWER = 'y') -->
                        <-- REQUEST(NEXT_PIN = '', XAUTH MESSAGE =
                        'Enter your new PIN containing 4-6 digits')
   REPLY(NEXT PIN = '1234') -->
                             <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1234764456'
                                               <-- SET(STATUS = OK)
   ACK(STATUS) -->



   Example A-7:
   ============
   SDI next cardcode

   IPsec Client                                          IPsec Gateway
```



```
   ------------                                          -------------
                            <-- REQUEST(Username = '', Passcode = '')
   REPLY(Username = 'joe', Passcode = '1637364856') -->
                        <-- REQUEST(Username = '', Passcode = '',
                        XAUTH_MESSAGE = 'Your token is out of sync with
                        the server, please enter a new passcode.')
   REPLY(Username = 'joe', Passcode = '1637904324') -->
                                                  <-- SET(STATUS = OK)
   ACK(STATUS) -->
```

RADIUS Chap Challenge
=====================

Example A-8:
============



where the Challenge in the REQUEST is the random number generated by the edge device, and the Password in the reply contains the ID used to calculate the hash 'aa' concatenated with the hash of the (ID+secret+challenge).