

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 10, 2011

Y. Sheffer
Independent
S. Fluhrer
Cisco
March 9, 2011

HUSH: Using HUmanly memorable SHared secrets with IKEv2
draft-sheffer-ipsecme-hush-02

Abstract

This document defines a new mode for IKEv2, where both peers can authenticate using a short, humanly memorable shared secret. This mode is based on the EKE protocol.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2011.

Copyright Notice

Copyright (c) 2011 IETFTrust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Internet-Draft

HUSH

March 2011

Table of Contents

1.	Introduction	3
2.	Terminology	3
3.	Overview	3
4.	Protocol Sequence	4
4.1.	The IKE_SA_INIT Exchange	5
4.2.	The IKE_HUSH Exchange	5
4.2.1.	Message #1	5
4.2.2.	Message #2	6
4.2.3.	Message #3	7
4.2.4.	Message #4	7
5.	Protocol Formats	7
5.1.	Encrypt Payload	7
5.2.	Protect Payload	8
6.	Cryptographic Details	9
6.1.	Diffie-Hellman Groups	9
7.	IANA Considerations	10
8.	Security Considerations	10
9.	Acknowledgements	10
10.	References	11
10.1.	Normative References	11
10.2.	Informative References	11
Appendix A.	Change Log	12
A.1.	-01	12
A.2.	-00	12
	Authors' Addresses	12

Internet-Draft

HUSH

March 2011

1. Introduction

There is strong interest in a simple method for bootstrapping an IKE [[RFC4306](#)] security association between two peers, requiring neither PKI nor AAA infrastructure. Although IKEv2 supports EAP-based authentication in part to provide for this capability, it has been claimed that the use of an extra authentication layer/protocol adds little benefit and increases complexity.

This protocol integrates the well known EKE protocol [[BM92](#)] into IKEv2, to provide password-based authentication. Some of the benefits of this protocol are:

- o EKE is a well known protocol, which has had multiple deep cryptographic analyses applied to it.
- o EKE provides the benefit of a well known, clear IPR status.

This protocol is not intended for use in enterprise-scale remote access. As a result, only the basic authentication capability is provided. Some capabilities typically associated with the use of passwords for remote access include: password change and expiry, password recovery, and enforcement of password strength policy.

In this preliminary version of the protocol many issues are not yet covered, such as:

- o Integration with other IKE elements, e.g. optional notifications, Session Resumption...
- o Error handling.
- o Generation of a high-quality PSK, so that the password doesn't need to be used for each authentication. Secure signalling of PSK possession.
- o Security analysis.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

3. Overview

This protocol attempts to preserve the general structure of IKEv2, minimizing the number of new constructs and round trips, while retaining IKE's security guarantees, including identity protection. The resulting protocol only adds one round trip to the shared secret

Sheffer & Fluhrer

Expires September 10, 2011

[Page 3]

Internet-Draft

HUSH

March 2011

authentication mode of IKEv2.

At a high level, the message exchange is as follows:

Initiator	Responder
-----	-----
HDR, SAI1, KEi, Ni, N(HUSH)	
	<- HDR, SAR1, KEr, Nr, N(HUSH)
HDR, SK{IDi, [IDr,], SAI2, TSi, TSr, Encrypt{Yi}} ->	
	<- HDR, SK{IDr, Encrypt{Yr}, Protect{Nr2}}
HDR, SK{AUTH, Protect{Ni2 Nr2}}	
	<- HDR, SK{AUTH, SAR2, TSi, TSr, Protect{Ni2}}

The changes to IKEv2 are summarized in the following list:

- o The regular IKE_SA_INIT exchange is followed by a new 2-round trip exchange, IKE_HUSH.
- o Negotiation of the new mode using notifications in IKE_SA_INIT.
- o Negotiation of cryptographic algorithms: the encryption algorithm, the integrity protection algorithm and the pseudo-random function are the ones negotiated for the IKE SA itself, while a new Diffie-

Hellman group is selected for the HUSH exchange, by extending the SAI1/SAr1 negotiation.

- o A new encrypted payload type, denoted Encrypt{}, for exchanging an ephemeral public key encrypted by the password.
- o A new encrypted and integrity-protected payload type, denoted Protect{}, for exchanging nonces encrypted by the EKE shared secret.
- o The IKE AUTH payloads provide cryptographic binding of the IKE shared secret with the password-based authentication.

[4.](#) Protocol Sequence

The protocol consists of a slightly extended IKE_SA_INIT exchange, followed by the 4-message IKE_HUSH exchange.

[4.1.](#) The IKE_SA_INIT Exchange

During this exchange, the initiator sends an empty HUSH_SUPPORTED notification. If the responder understands this protocol and wishes to use it, it sends back another empty HUSH_SUPPORTED notification.

In addition, this protocol defines a new transform type for the IKE protocol, called "EKE D-H Group". Possible transforms are the EKE groups defined in [Section 6.1](#). This transform type is negotiated between the initiator and responder with the usual SAI1, SAr1 payloads. If the initiator suspects that the responder does not support this protocol, it SHOULD also include a proposal that omits this transform, to allow the negotiation to revert to regular IKE. During successful negotiation, an EKE D-H Group MUST be negotiated if (and only if) the responder indicates support for this protocol.

[4.2.](#) The IKE_HUSH Exchange

This exchange consists of two message pairs, and includes all payloads normally contained in the IKE_AUTH exchange. These latter payloads are not described in this subsection, in order to focus on the new HUSH payloads.

[4.2.1.](#) Message #1

The initiator computes

$$Y_i = g^x \bmod N,$$

where x is a randomly chosen number in the range $2 \dots N-1$, as defined by the negotiated Diffie-Hellman group. The randomly chosen number is the private key, and the calculated value is the corresponding public key. Each of the peers **MUST** use a fresh, random value for x on each run of the protocol.

Note: If Elliptic Curve Diffie-Hellman is used in a future version of this protocol, the corresponding additive group operations are to be understood.

The initiator generates the Encrypt payload ([Section 5.1](#)),

$$\text{Encrypt}(\text{prf}+(\text{password}, \text{"HUSH Password"}), Y_i),$$

where the literal string is encoded using ASCII with no zero terminator. The `prf+` notation is as defined in [\[RFC4306\]](#). When using block ciphers, it may be necessary to pad Y_i on the right, to fit the encryption algorithm's block size. In such cases, random padding **MUST** be used, and this randomness is critical to the security

of the protocol. Randomness recommendations can be found in [\[RFC4086\]](#).

If the password needs to be stored on the server, it is **RECOMMENDED** to store the randomized password value, i.e. `prf+(password, ...)`, as a password-equivalent, rather than the cleartext password.

If the password is non-ASCII, it **SHOULD** be normalized by the sender before the message is constructed. The normalization method is SASLprep, [\[RFC4013\]](#). Note that the password is not null-terminated.

[4.2.2.](#) Message #2

Similarly to Message #1, the responder picks a random private key, generates an ephemeral public key Y_r , encrypts it by the expanded

password and includes the resulting Encrypt payload in the message:

```
Encrypt(prf+(password, "HUSH Password"), Yr),
```

The responder now calculates

```
EkeSharedSecret = prf(0+,  $g^{(x*y)} \bmod N$ )
```

where the first argument to "prf" is a string of zero octets whose length is the output size of the base hash algorithm, e.g. 20 octets for HMAC-SHA1; the result is of the same length. This extra application of the pseudo-random function is the "extraction step" of [\[RFC5869\]](#).

The responder computes the encryption and authentication (integrity protection) keys:

```
Ke2, Ka2 = prf+(EkeSharedSecret, "HUSH encryption and  
authentication" | IDi | IDr)
```

Now the responder can generate the Protect payload included in the message:

```
Protect(Ke2, Ka2, Nr2),
```

where Nr2 is a randomly generated binary string (nonce). Nr2 has length equal to the block size of the negotiated encryption algorithm for block ciphers, or 32 octets if this algorithm is a stream cipher. The responder sends this value as an Encrypt payload.

[4.2.3.](#) Message #3

The initiator computes the EkeSharedSecret, Ke2 and Ka2 values as above.

It then picks a random nonce Ni2, of the same format as Nr2, concatenates the two nonces, and generates

Protect(Ke2, Ka2, Ni2 | Nr2),

In addition, it computes

AUTH = prf(prf(Shared Secret, Ni2 | Nr2 | IDi | IDr),
<InitiatorSignedOctets>)

where the Shared Secret is the regular IKE shared secret, created by the IKE_SA_INIT exchange.

[4.2.4.](#) Message #4

The responder verifies Nr2 and the received AUTH payload, and MUST terminate the protocol if either of them fails to verify. The responder generates

Protect(Ke2, Ka2, Ni2)

and

AUTH = prf(prf(Shared Secret, Ni2 | Nr2 | IDi | IDr),
<ResponderSignedOctets>)

The initiator MUST verify the Ni2 and AUTH values when receiving Message #4.

[5.](#) Protocol Formats

[5.1.](#) Encrypt Payload

This payload contains encrypted, but non-integrity protected, data. Unfortunately the simpler term "Encrypted Payload" is used by IKEv2 for a payload that contains encrypted and integrity-protected data.

Compared to the IKE Encrypted Payload, this payload does not contain other embedded payloads. The payload is denoted Encrypt(key, data), and defined thus:



1. The generator is a primitive element of the group.
2. The most significant 64 bits of the prime number are 1.
3. The group's order p is a "safe prime", i.e. $(p-1)/2$ is also prime.

The last requirement is related to the strength of the Diffie Hellman

Internet-Draft

HUSH

March 2011

algorithm, rather than the password encryption. It also makes it easy to verify that the generator is primitive.

We have defined the following groups. The Value column is used when negotiating the group. Additional groups may be defined through IANA allocation. Future non-MODP groups require a document to define their interaction with this protocol.

Name	Length	Value	Description
Reserved		0	
DHGROUP_EKE_2	1024	1	The prime number of Group 2 [RFC4306], with the generator 5 (decimal)
DHGROUP_EKE_5	1536	2	The prime number of Group 5 [RFC3526], g=31
DHGROUP_EKE_14	2048	3	The prime number of Group 14 [RFC3526], g=11
DHGROUP_EKE_15	3072	4	The prime number of Group 15 [RFC3526], g=5
DHGROUP_EKE_16	4096	5	The prime number of Group 16 [RFC3526], g=5
Available for allocation via IANA		6-127	
Reserved for private use		128-255	

7. IANA Considerations

TBD: one notification, one transform type, two payloads, a new exchange. Also a new DH group registry.

8. Security Considerations

Will be added.

9. Acknowledgements

Much of this protocol is derived from [[I-D.sheffer-emu-eap-eke](#)], and authors (and reviewers) of that draft are acknowledged.

Sheffer & Fluhner Expires September 10, 2011 [Page 10]

Internet-Draft HUSH March 2011

10. References

10.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", [RFC 3526](#), May 2003.
- [RFC4306] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", [RFC 4306](#), December 2005.

10.2. Informative References

- [BM92] Bellovin, S. and M. Merritt, "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks", Proc. IEEE Symp. on Research in Security and Privacy , May 1992.
- [BM93] Bellovin, S. and M. Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise", Proc. 1st ACM Conference on Computer and Communication Security , 1993.
- [BMP00] Boyko, V., MacKenzie, P., and S. Patel, "Provably Secure Password Authenticated Key Exchange Using Diffie-Hellman", Advances in Cryptology, EUROCRYPT 2000 , 2000.

[[I-D.sheffer-emu-eap-eke](#)]

Sheffer, Y., Zorn, G., Tschofenig, H., and S. Fluhrer, "An EAP Authentication Method Based on the EKE Protocol", [draft-sheffer-emu-eap-eke-09](#) (work in progress), October 2010.

- [PA97] Patel, S., "Number Theoretic Attacks On Secure Password Schemes", Proceedings of the 1997 IEEE Symposium on Security and Privacy , 1997.
- [RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", [RFC 4013](#), February 2005.
- [RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", [BCP 106](#), [RFC 4086](#), June 2005.
- [RFC5869] Krawczyk, H. and P. Eronen, "HMAC-based Extract-and-Expand

Sheffer & Fluhrer Expires September 10, 2011 [Page 11]

Internet-Draft HUSH March 2011

Key Derivation Function (HKDF)", [RFC 5869](#), May 2010.

[Appendix A](#). Change Log

Note to RFC Editor: please remove this section before publication.

[A.1](#). -01

Reissued, changed the derivation of the payload encryption and authentication keys.

[A.2](#). -00

Initial version, a very rough draft.

Authors' Addresses

Yaron Sheffer
Independent

Email: yarolf.ietf@gmail.com

Scott Fluhner
Cisco Systems.
1414 Massachusetts Ave.
Boxborough, MA 01719
USA

Email: sfluhner@cisco.com