Network Working Group
Request for Comments: 4307
Category: Standards Track

J. Schiller
Massachusetts Institute of Technology

December 2005

Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)

Status of This Memo

Copyright Notice

Abstract

The IPsec series of protocols makes use of various cryptographic algorithms in order to provide security services. The Internet Key Exchange (IKE (RFC 2409) and IKEv2) provide a mechanism to negotiate which algorithms should be used in any given association. However, to ensure interoperability between disparate implementations, it is necessary to specify a set of mandatory-to-implement algorithms to ensure that there is at least one algorithm that all implementations will have available. This document defines the current set of algorithms that are mandatory to implement as part of IKEv2, as well as algorithms that should be implemented because they may be promoted to mandatory at some future time.

1. Introduction

The Internet Key Exchange protocol provides for the negotiation of cryptographic algorithms between both endpoints of a cryptographic

association. Different implementations of IPsec and IKE may provide different algorithms. However, the IETF desires that all implementations should have some way to interoperate. In particular, this requires that IKE define a set of mandatory-to-implement algorithms because IKE itself uses such algorithms as part of its own negotiations. This requires that some set of algorithms be specified as "mandatory-to-implement" for IKE.

The nature of cryptography is that new algorithms surface continuously and existing algorithms are continuously attacked. An algorithm believed to be strong today may be demonstrated to be weak tomorrow. Given this, the choice of mandatory-to-implement algorithm should be conservative so as to minimize the likelihood of it being compromised quickly. Thought should also be given to performance considerations as many uses of IPsec will be in environments where performance is a concern.

Finally, we need to recognize that the mandatory-to-implement algorithm(s) may need to change over time to adapt to the changing world. For this reason, the selection of mandatory-to-implement algorithms was removed from the main IKEv2 specification and placed in this document. As the choice of algorithm changes, only this document should need to be updated.

Ideally, the mandatory-to-implement algorithm of tomorrow should already be available in most implementations of IPsec by the time it is made mandatory. To facilitate this, we will attempt to identify those algorithms (that are known today) in this document. There is no guarantee that the algorithms we believe today may be mandatory in the future will in fact become so. All algorithms known today are subject to cryptographic attack and may be broken in the future.

2. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

We define some additional terms here:

SHOULD+ This term means the same as SHOULD. However, it is likely that an algorithm marked as SHOULD+ will be promoted at some future time to be a MUST.

SHOULD- This term means the same as SHOULD. However, an algorithm marked as SHOULD- may be deprecated to a MAY in a future version of this document.

MUST- This term means the same as MUST. However, we expect at some point that this algorithm will no longer be a MUST in a future document. Although its status will be determined at a later time, it is reasonable to expect that if a future revision of a document alters the status of a MUST- algorithm, it will remain at least a SHOULD or a SHOULD-.

3. Algorithm Selection

3.1. IKEv2 Algorithm Selection

3.1.1. Encrypted Payload Algorithms

The IKEv2 Encrypted Payload requires both a confidentiality algorithm and an integrity algorithm. For confidentiality, implementations MUST- implement 3DES-CBC and SHOULD+ implement AES-128-CBC. For integrity, HMAC-SHA1 MUST be implemented.

3.1.2. Diffie-Hellman Groups

There are several Modular Exponential (MODP) groups that are defined for use in IKEv2. They are defined in both the [IKEv2] base document and in the MODP extensions document. They are identified by group number. Any groups not listed here are considered as "MAY be implemented".

Group Number	Bit Length	Status	Defined
2	1024 MODP Group	MUST-	[RFC2409]
14	2048 MODP Group	SHOULD+	[RFC3526]

3.1.3. IKEv2 Transform Type 1 Algorithms

IKEv2 defines several possible algorithms for Transfer Type 1 (encryption). These are defined below with their implementation status.

Name	Number	Defined In	Status
RESERVED	0		
ENCR_3DES	3	[RFC2451]	MUST-
ENCR_NULL	11	[RFC2410]	MAY
ENCR_AES_CBC	12	[AES-CBC]	SHOULD+
ENCR_AES_CTR	13	[AES-CTR]	SHOULD

3.1.4. IKEv2 Transform Type 2 Algorithms

Transfer Type 2 Algorithms are pseudo-random functions used to generate random values when needed.

Name	Number	Defined In	Status
RESERVED	0		
PRF_HMAC_MD5	1	[RFC2104]	MAY
PRF_HMAC_SHA1	2	[RFC2104]	MUST
PRF_AES128_CBC	4	[AESPRF]	SHOULD+

3.1.5. IKEv2 Transform Type 3 Algorithms

Transfer Type 3 Algorithms are Integrity algorithms used to protect data against tampering.

Name	Number	Defined In	Status
NONE	0		
AUTH_HMAC_MD5_96	1	[RFC2403]	MAY
AUTH_HMAC_SHA1_96	2	[RFC2404]	MUST
AUTH_AES_XCBC_96	5	[AES-MAC]	SHOULD+

4. Security Considerations

The security of cryptographic-based systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

This document concerns itself with the selection of cryptographic algorithms for the use of IKEv2, specifically with the selection of "mandatory-to-implement" algorithms. The algorithms identified in this document as "MUST implement" or "SHOULD implement" are not known to be broken at the current time, and cryptographic research so far leads us to believe that they will likely remain secure into the foreseeable future. However, this isn't necessarily forever. We would therefore expect that new revisions of this document will be issued from time to time that reflect the current best practice in this area.

5. Normative References

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

[IKEv2] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[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.

[RFC2451] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.

[RFC2410] Glenn, R. and S. Kent, "The NULL Encryption Algorithm and Its Use With IPsec", RFC 2410, November 1998.

[AES-CBC] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.

[AES-CTR] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[AESPRF] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 3664, January 2004.

[RFC2403] Madson, C. and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, November 1998.

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[AES-MAC] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.

Author's Address

Jeffrey I. Schiller
Massachusetts Institute of Technology
Room W92-190
77 Massachusetts Avenue
Cambridge, MA 02139-4307
USA

Phone: +1 (617) 253-0161
EMail: jis@mit.edu

Full Copyright Statement

Intellectual Property

Acknowledgement

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