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

Status of This Memo

This document specifies an Internet standards track protocol for the
Internet community, and requests discussion and suggestions for
improvements. Please refer to the current edition of the "Internet
Official Protocol Standards" (STD 1) for the standardization state
and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

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".

<table>
<thead>
<tr>
<th>Group Number</th>
<th>Bit Length</th>
<th>Status</th>
<th>Defined</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1024</td>
<td>MUST-</td>
<td>[RFC2409]</td>
</tr>
<tr>
<td>14</td>
<td>2048</td>
<td>SHOULD+</td>
<td>[RFC3526]</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Defined In</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ENCR_3DES</td>
<td>3</td>
<td>[RFC2451]</td>
<td>MUST-</td>
</tr>
<tr>
<td>ENCR_NULL</td>
<td>11</td>
<td>[RFC2410]</td>
<td>MAY</td>
</tr>
<tr>
<td>ENCR_AES_CBC</td>
<td>12</td>
<td>[AES-CBC]</td>
<td>SHOULD+</td>
</tr>
<tr>
<td>ENCR_AES_CTR</td>
<td>13</td>
<td>[AES-CTR]</td>
<td>SHOULD</td>
</tr>
</tbody>
</table>

3.1.4. IKEv2 Transform Type 2 Algorithms

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Defined In</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESERVED</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRF_HMAC_MD5</td>
<td>1</td>
<td>[RFC2104]</td>
<td>MAY</td>
</tr>
<tr>
<td>PRF_HMAC_SHA1</td>
<td>2</td>
<td>[RFC2104]</td>
<td>MUST</td>
</tr>
<tr>
<td>PRF_AES128_CBC</td>
<td>4</td>
<td>[AESPRF]</td>
<td>SHOULD+</td>
</tr>
</tbody>
</table>
3.1.5. IKEv2 Transform Type 3 Algorithms

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
<th>Defined In</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>NONE</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUTH_HMAC_MD5_96</td>
<td>1</td>
<td>[RFC2403]</td>
<td>MAY</td>
</tr>
<tr>
<td>AUTH_HMAC_SHA1_96</td>
<td>2</td>
<td>[RFC2404]</td>
<td>MUST</td>
</tr>
<tr>
<td>AUTH_AES_XCBC_96</td>
<td>5</td>
<td>[AES-MAC]</td>
<td>SHOULD+</td>
</tr>
</tbody>
</table>

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


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