Guidelines for Implementation and Priorities in Testing IKEv2

Technical Document

Version 2.0.0b

IPv6 Promotion Council
Certification WG
IPsec SWG

Modification Record

Version 2.0.0b May 19, 2011

Version 1.1.0 May 20, 2010

- RFC4868 and RFC5114 was added to "Related standards".
- The condition of pseudo-random function of IKE SA changed.
  BASIC : PRF_HMAC_SHA1
  ADVANCED : PRF_AES128_XCBC, PRF_HMAC_SHA2_256
- The condition of Integrity Algorithm of IKE SA changed.
  BASIC : AUTH_HMAC_SHA1_96
  ADVANCED : AUTH_AES_XCBC_96,AUTH_HMAC_SHA2_256_128
  Not Supported : AUTH_HMAC_MD5_96
- The condition of Diffie-Hellman Group of IKE SA changed.
  BASIC : Group 2
  ADVANCED : Group14, Group24
- The condition of Integrity Algorithm of CHILD SA changed.
  BASIC : AUTH_HMAC_SHA1_96
  ADVANCED : AUTH_AES_XCBC_96,AUTH_HMAC_SHA2_256_128
  Not Supported : AUTH_HMAC_MD5_96
- RFC4306 page 51 line 2825 : test requirement changed to ADVANCED because PRF_HMAC_SHA2_256 is ADVANCED algorithm.
- RFC4306 page 51 line 2839 : test requirement changed to ADVANCED because AUTH_HMAC_SHA2_256_128 is ADVANCED algorithm.
- RFC4306 page 51 line 2852 : test requirement changed to ADVANCED because Group 24 is ADVANCED group.

Version 1.0.2 Mar 25, 2010

- The condition of ID types Receiving and Sending changed.
  BASIC : IPV6_ADDR
  ADVANCED : FQDN , RFC822_ADDR ,
  Not Supported : IPV4_ADDR, KEY_ID
- The condition of encryption algorithms of IKE SA changed.
  BASIC : ENCR_3DES
  ADVANCED : ENCR_AES_CBC
- RFC4306 page 4 line 189 and 199 : test number changed because EN.I.1.3.2.1 and SGW.I.1.3.2.1 were removed.
- RFC4306 page 11 line 577, 581 and 590 :
  target and test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 12 line 638 : target and test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 13 line 720 : test number changed because EN.I.1.3.2.1 and SGW.I.1.3.2.1 were removed.
- RFC4306 page 15 line 830 :
  target and test number changed because EN.I.1.1.3.4, EN.R.1.1.3.3, SGW.I.1.1.3.3 and SGW.I.1.1.3.4 were removed.
- RFC4306 page 15 line 834 and 836 : test number changed because EN.I.1.1.3.5 and SGW.I.1.1.3.5 were removed.
- RFC4306 page 17 line 906 : test number changed because EN.I.1.1.3.7 and SGW.I.1.1.3.7 were removed.
- RFC4306 page 18 line 968, 969 and 987 : test number changed because EN.I.1.3.1.1 and SGW.I.1.1.3.1 were removed.
- RFC4306 page 19 line 1035 test requirement changed to Not Support because of untestable.
- RFC4306 page 19 line 1046 target and test number changed because EN.R.1.1.5.1 and SGW.R.1.1.5.1 were removed.
- RFC4306 page 20 line 1070 target and test number changed because EN.R.1.1.5.1 and SGW.R.1.1.5.1 were removed.
- RFC4306 page 21 line 1164 : test requirement changed to ADVANCED because DH#14 is ADVENCED group.
- RFC4306 page 28 line 1520, 1522, 1525, 1542, 1544 and 1584 :
  test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 29 line 1586 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 36 line 1986 :
  test number changed because EN.I.1.1.6.8, EN.I.1.1.8.1, EN.I.1.1.8.2, EN.I.1.3.4.1, EN.R.1.1.8.1, EN.R.1.1.8.2, EN.R.1.1.8.3, SGW.I.1.1.6.8, SGW.1.1.8.1, SGW.I.1.1.8.2, SGW.I.1.3.4.1, SGW.R.1.1.8.1, SGW.R.1.1.8.2 and SGW.R.1.1.8.3 were removed.
- RFC4306 page 36 line 1988, 2002 and 2005 : test requirement changed to Not Support because of untestable.
- RFC4306 page 41 line 2263 and 2277 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 42 line 2302, 2320, 2324, 2334, 2342, 2347 and 2361 :
  test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 43 line 2376, 2379, 2393 and 2400 :
  test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 44 line 2414,2420,2430,2440 and 2459 :
  test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 45 line 2492,2508 and 2511 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 45 line 2508 and 2511 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
- RFC4306 page 46 line 2560 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.

・RFC4306 page 48 line 2638 and 2669 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4306 page 50 line 2754 : test number changed because EN.I.1.3.3.1 and SGW.I.1.3.3.1 were removed.
・RFC4306 page 51 line 2811 :
test number changed because EN.I.1.6.1.1, EN.R.1.6.1.1, SGW.I.1.6.1.1 and SGW.R.1.6.1.1 were removed.
・RFC4306 page 55 line 3049 : test requirement changed to ADVANCED because DH#14 is ADVANCED group.
・RFC4306 page 56 line 3114 : test requirement changed to ADVANCED because DH#14 is ADVANCED group.
・RFC4306 page 57 line 3179 : test number changed because this test is not support expect ID_IPV6_ADDR, FQDN and RFC822_ADDR,
・RFC4306 page 57 line 3183 : test requirement changed to ADVANCED because only with RSA-DSS auth.
・RFC4306 page 58 line 3198 : test requirement changed to ADVANCED because only with RSA-DSS auth.
・RFC4306 page 58 line 3204, 3212, 3217 and 3222 :
test number changed because this test is not support expect ID_IPV6_ADDR, FQDN and RFC822_ADDR,.
・RFC4306 page 58 line 3233, 3235 and 3236 :
test requirement changed Not Support because ID_FQDN and RFC822_ADDR are available only with RSA-DSS auth.
・RFC4306 page 66 line 3678 and 3690 : test requirement change to Not Support because of untestable.
・RFC4306 page 67 line 3705 and 3722 : test requirement change to Not Support because of untestable.
・RFC4306 page 67 line 3733 :
target and test number changed because EN.I.1.1.6.8 and SGW.1.1.6.8 were removed and EN.R.1.1.6.9 and SGW.R.1.1.6.9 were added..
・RFC4306 page 67 line 3737 : test requirement changed ADVANCED to because DH#14 is ADVANCED group.
・RFC4306 page 68 line 3793 : test requirement change to Not Support because of untestable.
・RFC4306 page 69 line 3817 : test requirement change to Not Support because of untestable.
・RFC4306 page 71 line 3926, 3930, 3933 :
target and test number changed because EN.R.1.1.5.1 and SGW.I.R.1.5.1 were removed.
・RFC4306 page 72 line 4011 : test number changed because EN.I.1.1.3.7, SGW.I.1.1.3.7 and SGW.I.1.1.3.8 were removed.
・RFC4306 page 72 line 4025 : test number changed because EN.I.1.1.3.7, SGW.I.1.1.3.7 and SGW.I.1.1.3.8 were removed.
・RFC4306 page 73 line 4038 : test number changed because EN.I.1.1.3.7, SGW.I.1.1.3.7 and SGW.I.1.1.3.8 were removed.
・RFC4306 page 73 line 4042 : test number changed because EN.I.1.1.3.7, SGW.I.1.1.3.7 and SGW.I.1.1.3.8 were removed.
・RFC4306 page 73 line 4045 : test number changed because EN.I.1.1.3.7, SGW.I.1.1.3.7 and SGW.I.1.1.3.8 were removed.
・RFC4306 page 77 line 4276, 4280, 4284 and 4292 :
test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4306 page 78 line 4318, 4337, 4345, 4348, 4357 and 4360 :
test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4306 page 79 line 4374 and 4381 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4306 page 86 line 4794 : test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4306 page 86 line 4797 : target and test number changed because EN.I.1.3.1.1 and SGW.I.1.3.1.1 were removed.
・RFC4718 page 4 line 218 : test requirement changed to Not Support because of untestable.
・RFC4718 page 4 line 240 : test requirement changed to ADVANCED because DH#14 is ADVANCED group.
・RFC4718 page 5 line 254 : test number changed because EN.R.1.1.5.1 and SGW.R.1.1.5.1 were removed.
・RFC4718 page 7 line 375 : target andtest number changed because EN.R.1.1.5.3 and SGW.R.1.1.5.3 were removed.
・RFC4718 page 26 line 1417 : test requirement changed to ADVANCE because PRF_AES128_XCBC is ADVANCED algorithm.
・RFC4718 page 28 line 1548, 1580 and 1592 : test requirement changed to Not Support because of untestable.
・RFC4718 page 32 line 1766, 1776, 1783 and 1804 : test requirement changed to Not Support because of untestable.
・RFC4718 page 33 line 1811 and 1818 : test requirement changed to Not Support because of untestable.
・RFC4718 page 34 line 1859, 1865 and 1873 : test requirement changed to Not Support because of untestable.
・RFC4718 page 35 line 1913, 1920, 1935, 1946 and 1956 : test requirement changed to Not Support because of untestable.
・RFC4718 page 36 line 1970, 1977, 1980, 1985, 1991, 1994, 1997, 2000 and 2003 :
test requirement changed to Not Support because of untestable.
・RFC4718 page 37 line 2034 : test requirement changed to Not Support because of untestable.
・RFC4718 page 48 line 2646 and 2650 : test number changed because EN.R.1.1.8.3 and SGW.R.1.1.8.3 were removed.
・RFC4718 page 48 line 2661 : test requirement changed to Not Support because of untestable.
・RFC4718 page 56 line 3110 and 3114 :
target and test number changed because EN.R.1.3.1.1 and SGW.R.1.3.1.1 were removed.

Version 1.0.1 June 5, 2009

・The condition of ID types Receiving and Sending changed.

BASIC : IPV6_ADDR
Not Supported : IPV4_ADDR,FQDN , RFC822_ADDR , KEY_ID

・Function of "restarting the entire IKE_SA"is Not Suppoted.

Version 1.0.0 December 11, 2008

・Initial release.

Table of Contents

1. Overview

This document gives guidelines for implementing the functions specified in IKEv2 prescribed in IETF to ensure interoperability.

The IKEv2 Test Profile consists of two volumes. One is “Guidelines for Implementation and Priorities in Testing”, which means this document, and the other is “Test Specifications”. The contents of this document include following items.

- Guidelines for the implementation of the nodes supporting IKEv2
- Specifications of the IKEv2 sequences and payload type in each message between the nodes supporting IKEv2 (i.e. SGW and End-Node)
- Priorities for the testing of each node function according to the function’s importance to interoperability.

This document is in complete accord with the IETF RFC specifications for IKEv2 but includes some extra information for clarification and thus more strongly ensures interoperability.

Term Description

-**End-Node**

IPv6 host which can terminate IKEv2 protocol.

End-Node is denoted by “EN” in this document.

-**Security Gateway**

IPv6 node including a router or a firewall that intermediate system which support IKEv2 protocols.

Security Gateway is denoted by “SGW” in this document.

2. Scope of the IKEv2 Guidelines for Implementation

2.1. Reference Network Architecture

Figure 2-1 shows the network architecture covered by IKEv2 Guidelines for Implementation.

- I/F1 is an interface that showed the protocol confirmation between EN and EN.
- I/F2 is an interface that showed the protocol confirmation between EN and SGW.
- I/F3 is an interface that showed the protocol confirmation between SGW and SGW.

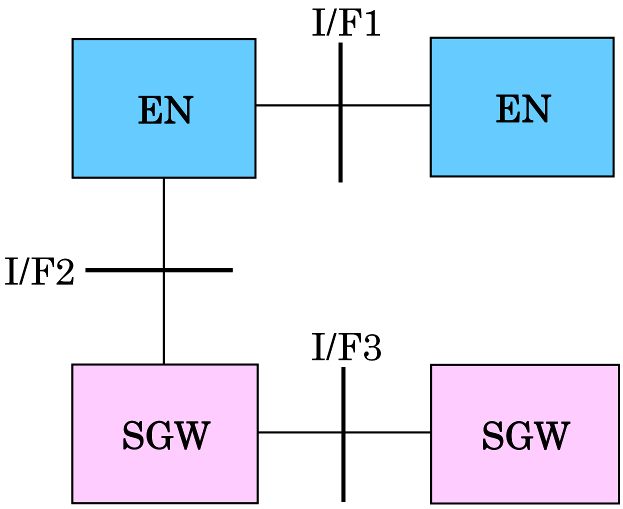


Figure 2-1 Reference Network Architecture

This document only covers IKEv2 specifications. Testing of generic IPv6 functions is beyond the scope of this test; however; some of the generic IPv6 functions are necessary to IKEv2 functions and are thus supported in this test.

2.2. Related standards

This document covers the functions specified in the following IETF RFCs.

(1) RFC4301 "Security Architecture for the Internet Protocol" (http://www.ietf.org/rfc/rfc4301.txt)

(2) RFC4302 "IP Authentication Header" (http://www.ietf.org/rfc/rfc4302.txt)

(3) RFC4303 "IP Encapsulating Security Payload (ESP)" (http://www.ietf.org/rfc/rfc4303.txt)

(4) RFC4305 "Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)"

(http://www.ietf.org/rfc/rfc4305.txt)

(5) RFC4307 "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)"
(http://www.ietf.org/rfc/rfc4307.txt)

(6) RFC5996 "Internet Key Exchange Protocol Version 2 (IKEv2)"
(http://www.rfc-editor.org/rfc/rfc5996.txt)

(7) RFC2404 "The Use of HMAC-SHA-1-96 within ESP and AH"
(http://www.ietf.org/rfc/rfc2404.txt)

(8) RFC2410 "The NULL Encryption Algorithm and Its Use With IPsec"
(http://www.ietf.org/rfc/rfc2410.txt)

(9) RFC2451 "The ESP CBC-Mode Cipher Algorithms"
(http://www.ietf.org/rfc/rfc2451.txt)

(10) RFC3526 "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)"
(http://www.ietf.org/rfc/rfc3526.txt)

(11) RFC3566 "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec"
(http://www.ietf.org/rfc/rfc3566.txt)

(12) RFC3602 "The AES-CBC Cipher Algorithm and Its Use with IPsec"
(http://www.ietf.org/rfc/rfc3602.txt)

(13) RFC3686 "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)"
(http://www.ietf.org/rfc/rfc3686.txt)

(14) RFC4434 "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)"
(http://www.ietf.org/rfc/rfc4301.txt)

(15)RFC4868 "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec"
(http://www.ietf.org/rfc/rfc4868.txt)

(16) RFC5114 "Additional Diffie-Hellman Groups for Use with IETF Standards"
(http://www.ietf.org/rfc/rfc5114.txt)

3. Classification of IKEv2 functions

This section describes ways to classify the IKEv2 functions needed for interoperability and provided as test functions in the IKEv2 Conformance Test.

3.1. Viewpoints of the classification

The classification of IKEv2 functions is from the following viewpoints.

(A) IETF specification

(B) Test Requirements

(A) IETF specification

IETF specification refers to the classification of each of the IKEv2 functions from the viewpoint of importance for implementation as indicated by usage of the keywords below in the RFCs.

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are defined in RFC 2119.

(B) Test Requirements

Test Requirement is the classification from the viewpoint of the importance for testing and interoperability. Test Requirement has three classification Rank; that is, ”BASIC”, “ADVANCED”, “Not support”.

Testing of the functions classified as “BASIC” are included in the minimum test package, for the testing functions which are essential to conformance and interoperability.

Testing of the functions classified as “ADVANCED” are optional; this depends on the application to be used. The testing of “ADVANCED” (Optional Test) items is selectively incorporated in the test package according to the functions to be supported by the EN / SGW.

Testing of the functions classified as “Not supported” are functions that need not to support.

Table 3-1 shows the definition of Test Requirements

Table 3-1 Definitions of Test Requirements

	Definitions of Test Requirement
BASIC (Required Test)	These functions are essential to conformance and interoperability and should basically be implemented. Testing of the functions classified as "BASIC" is included in the minimum test package, for the testing of functions that are essential to conformance and interoperability.
ADVANCED (Optional Test)	Implementation of these functions is optional. Testing of the functions classified as "ADVANCED" may not be needed; this depends on the application to be used. The testing of "ADVANCED" (Optional Test) items is selectively incorporated in the test package according to the functions to be supported by the EN / SGW.
Not support	Testing of the functions classified as "Not supported" are functions that need not to support.

3.2. The method to classify the IKEv2 functions

Table 3-2 shows the relationships between IETF Specification and Test Requirement in this document.

IKEv2 functions with descriptions “MUST”, “SHOULD”, “MUST NOT” and “SHOULD NOT” in the IETF RFC are basically classified as BASIC, however some of these functions are described as ADVANCED or Not support, if necessary.

In the same way, IKEv2 functions with descriptions “MAY” and “No description” are basically classified as Not support, however some of these functions are described as BASIC or ADVANCED, if necessary.

Table 3-2 Relationship of classifications between IETF Specification and Test Requirement

(A) IETF	(B) Test Requirement
MUST MUST NOT	BASIC (Required Test)
SHOULD SHOULD NOT	ADVANCED (Optional Test)
MAY	Not Supported
No descriptions	Not Supported

supported

not supported

As reference, the classification of functions as BASIC, ADVANCED and Not Supported is described for each node about a typical IKEv2 function at Table 3-3 to Table 3-4. The classification of Notify Status Types is described at Table 3-5.

Table 3-3 IKEv2 functions and its classifications for EN

Function		EN		
		BASIC	ADVANCED	Not Supported
Initial Exchanges	Initiator or Responder	Initiator, Responder	-	-
	Sending proposal	simple transform in single proposal(patternA)	complex transform in single proposal(patternB) multiple proposals(patternC)	-
	Receiving proposal	simple transform in single proposal(patternA) complex transform in single proposal(patternB) multiple proposals(patternC)	-	-
	Retransmission	Supported	-	-
	ID Type receiving	IPV6_ADDR	FQDN, RFC822_ADDR	IPV4_ADDR, KEY_ID,
	ID Type sending	IPV6_ADDR	FQDN, RFC822_ADDR	IPV4_ADDR, KEY_ID,
	Auth method	Pre-shared Key	RSA Digital Signature	DSS Digital Signature
	Certificate Encoding	-	X.509 Certificate - Signature	-
	Traffic Selector Type	TS_IPV6_ADDR_RANGE	-	TS_IPV4_ADDR_RANGE
	Configuration Type	-	CFG_REQUEST, CFG_REPLY	CFG_SET, CFG_ACK
	Configuration Attribute Type	-	INTERNAL_IP6_ADDRESS	-
	EAP Authentication	-	-	Not support
	NAT Traversal	-	-	Not support
	Cookies	-	sending, receiving	-
	Vendor ID	-	-	Not support

<table>
<tr><th colspan="2" rowspan="2">Function</th><th colspan="3">EN</th></tr>
<tr><th>BASIC</th><th>ADVANCED</th><th>Not Supported</th></tr>
<tr><td rowspan="5">IKE_SA</td><td>Transform Type(IKE)</td><td>ENCR,
PRF,
INTEG,
D-H</td><td>-</td><td>-</td></tr>
<tr><td>Encryption Algorithm (ENCR)</td><td>ENCR_3DES</td><td>ENCR_AES_CBC</td><td>ENCR_DES,</td></tr>
<tr><td>Pseudo-random Function(PRF)</td><td>PRF_HMAC_SHA1</td><td>PRF_AES128_XCBC,
PRF_HMAC_SHA2_256</td><td>PRF_HMAC_MD5,</td></tr>
<tr><td>Integrity Algorithm (INTEG)</td><td>AUTH_HMAC_SHA1_96</td><td>AUTH_AES_XCBC_96,
AUTH_HMAC_SHA2_256_128</td><td>AUTH_HMAC_MD5_96,</td></tr>
<tr><td>Diffie-Hellman Group (D-H)</td><td>Group2
(1024 MODP)</td><td>Group14
(2048 MODP)
Group24
(2048-bit MODP Group with 256-bit Prime Order Subgroup)</td><td>-</td></tr>
<tr><td rowspan="7">CHILD_SA</td><td>IPsec mode</td><td>Transport</td><td>Tunnel</td><td>-</td></tr>
<tr><td>Security Protocol</td><td>ESP</td><td>-</td><td>AH</td></tr>
<tr><td>Transform Type(ESP)</td><td>ENCR,
INTEG,
ESN</td><td>-</td><td>-</td></tr>
<tr><td>Encryption Algorithm (ENCR)</td><td>ENCR_3DES</td><td>ENCR_NULL,
ENCR_AES_CBC,
ENCR_AES_CTR</td><td>ENCR_DES,</td></tr>
<tr><td>Integrity Algorithm (INTEG)</td><td>AUTH_HMAC_SHA1_96</td><td>AUTH_AES_XCBC_96,
NONE,
AUTH_HMAC_SHA2_256_128</td><td>AUTH_HMAC_MD5_96,</td></tr>
<tr><td>Extended Sequence Numbers(ESN)</td><td>No Extended Sequence Numbers</td><td>Extended Sequence Number</td><td>-</td></tr>
<tr><td>IPcomp</td><td>-</td><td>-</td><td>Not support</td></tr>
</table>

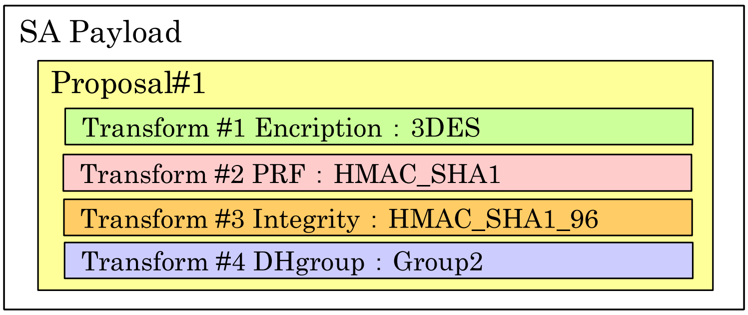
Function		EN		
		BASIC	ADVANCED	Not Supported
CREATE_ CHILD_SA Exchange	Initiator or Responder	Initiator, Responder	-	-
	Sending proposal	simple transform in single proposal(patternA)	complex transform in single proposal(patternB) multiple proposals(patternC)	-
	Receiving proposal	simple transform in single proposal(patternA) complex transform in single proposal(patternB) multiple proposals(patternC)	-	-
	Retransmission	Support	-	-
	Rekeying	IKE_SA rekeying, CHILD_SA rekeying	-	restarting the entire IKE_SA
	additional CHILD_SA	-	Support	-
	perfect forward secrecy(PFS)	-	Support	-
	Traffic Selector Type	IPV6_ADDR_RANGE	-	IPV4_ADDR_RANGE
	Vendor ID	-	-	Not support
INFOMATI ONAL Exchange	Initiator or Responder	Initiator, Responder	-	-
	Retransmission	Support	-	-
	Liveness Check	Sending, Responding	-	-
	Delete SA	IKE_SA delete, CHILD_SA delete	-	-
	Multiple SPIs deletion	receiving	-	Sending
	Request peer's version	-	-	requesting, answering
	Vendor ID	-	-	Not support

Table 3-4 IKEv2 functions and its classifications for SGW

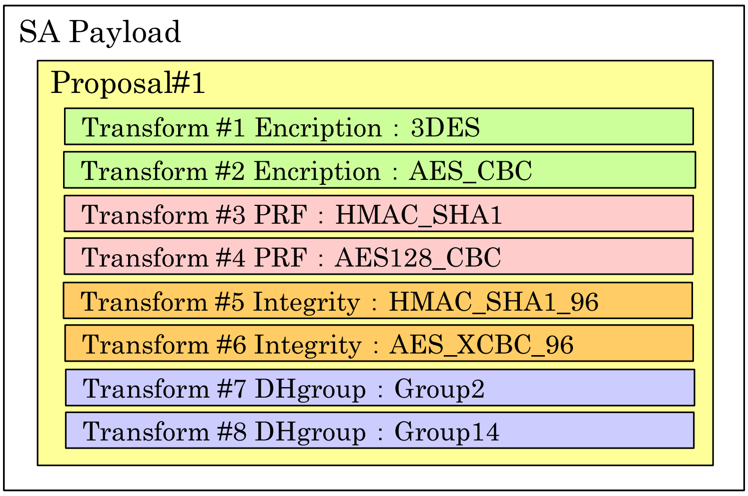
Function		SGW		
		BASIC	ADVANCED	Not Supported
Initial Exchanges	Initiator or Responder	Initiator, Responder	-	-
	Sending proposal	simple transform in single proposal(patternA)	complex transform in single proposal(patternB) multiple proposals(patternC)	-
	Receiving proposal	simple transform in single proposal(patternA) complex transform in single proposal(patternB) multiple proposals(patternC)	-	-
	Retransmission	Supported	-	-
	ID Type receiving	IPV6_ADDR	FQDN, RFC822_ADDR	IPV4_ADDR, KEY_ID,
	ID Type sending	IPV6_ADDR	FQDN, RFC822_ADDR,	IPV4_ADDR, KEY_ID,
	Auth method	Pre-shared Key	RSA Digital Signature	DSS Digital Signature
	Certificate Encoding	-	X.509 Certificate - Signature	-
	Traffic Selector Type	TS_IPV6_ADDR_RANGE	-	TS_IPV4_ADDR_RANGE
	Configuration Type	-	CFG_REQUEST, CFG_REPLY	CFG_SET, CFG_ACK
	Configuration Attribute Type	-	INTERNAL_IP6_ADDRESS	-
	EAP Authentication	-	-	Not support
	NAT Traversal	-	-	Not support
	Cookies	-	sending, receiving	-
	Vendor ID	-	-	Not support

Function		SGW		
		BASIC	ADVANCED	Not Supported
IKE_SA	Transform Type(IKE)	ENCR, PRF, INTEG, D-H	-	-
	Encryption Algorithm (ENCR)	ENCR_3DES	ENCR_AES_CBC	ENCR_DES,
	Pseudo-random Function(PRF)	PRF_HMAC_SHA1	PRF_AES128_XCBC, PRF_HMAC_SHA2_256	PRF_HMAC_MD5,
	Integrity Algorithm (INTEG)	AUTH_HMAC_SHA1_96	AUTH_AES_XCBC_96, AUTH_HMAC_SHA2_256_128	AUTH_HMAC_MD5_96,
	Diffie-Hellman Group (D-H)	Group2 (1024 MODP)	Group14 (2048 MODP) Group24 (2048-bit MODP Group with 256-bit Prime Order Subgroup)	-
CHILD_SA	IPsec mode	Tunnel	-	-
	Security Protocol	ESP	-	AH
	Transform Type(ESP)	ENCR, INTEG, ESN	-	-
	Encryption Algorithm (ENCR)	ENCR_3DES	ENCR_NULL, ENCR_AES_CBC, ENCR_AES_CTR	ENCR_DES,
	Integrity Algorithm (INTEG)	AUTH_HMAC_SHA1_96	AUTH_AES_XCBC_96, NONE AUTH_HMAC_SHA2_256_128	AUTH_HMAC_MD5_96,
	Extended Sequence Numbers(ESN)	No Extended Sequence Numbers	Extended Sequence Number	-
	IPcomp	-	-	Not support

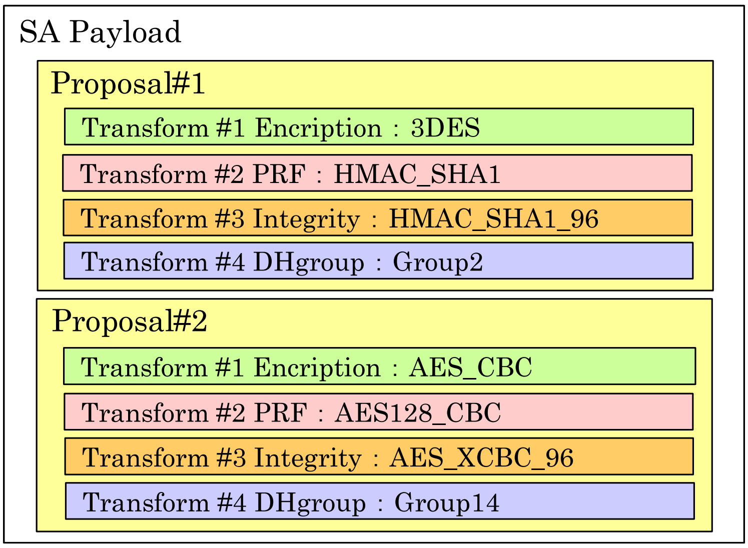
Function		SGW		
		BASIC	ADVANCED	Not Supported
CREATE_ CHILD_SA Exchange	Initiator or Responder	Initiator, Responder	-	-
	Sending proposal	simple transform in single proposal(patternA)	complex transform in single proposal(patternB) multiple proposals(patternC)	-
	Receiving proposal	simple transform in single proposal(patternA) complex transform in single proposal(patternB) multiple proposals(patternC)	-	-
	Retransmission	Support	-	-
	Rekeying	IKE_SA rekeying, CHILD_SA rekeying	-	restarting the entire IKE_SA
	additional CHILD_SA	-	Support	-
	perfect forward secrecy(PFS)	-	Support	-
	Traffic Selector Type	IPV6_ADDR_RANGE	-	IPV4_ADDR_RANGE
	Vendor ID	-	-	Not support
INFOMATI ONAL Exchange	Initiator or Responder	Initiator, Responder	-	-
	Retransmission	Support	-	-
	Liveness Check	Sending, Responding	-	-
	Delete SA	IKE_SA delete, CHILD_SA delete	-	-
	Multiple SPIs deletion	receiving	-	Sending
	Request peer's version	-	-	requesting, answering
	Vendor ID	-	-	Not support



patternA: simple transform in single proposal



patternB: complex transform in single proposal



patternC: multiple proposals

Table 3-5 IKEv2 Notify message for Status types and its classifications

Status Types	Classification	recital
INITIAL_CONTACT	ADVANCED	sending and receiving
SET_WINDOW_SIZE	Not support	
ADDITIONAL_TS_POSSIBLE	Not support	
IPCOMP_SUPPORTED	Not support	
NAT_DETECTION_SOURCE_IP	Not support	IPv6 network
NAT_DETECTION_DESTINATION_IP	Not support	IPv6 network
COOKIE	ADVANCED	
USE_TRANSPORT_MODE(only EN)	BASIC	use only End-Node
HTTP_CERT_LOOKUP_SUPPORTED	Not support	
REKEY_SA	BASIC	Rekey function BASIC
ESP_TFC_PADDING_NOT_SUPPORTED	BASIC	TFC Padding function Not support in IPsec Guidelines(IPsec v2)
NON_FIRST_FRAGMENTS_ALSO	Not support	
CHILD_SA_NOT_FOUND	Not support	
TEMPORARY_FAILURE	BASIC	

4. IKEv2 Sequences and Payloads

This section describes the IKEv2 sequences and payloads used in the IKEv2 Guidelines for Implementation. Sequences of test packet are sent to the target and expects to receive corresponding acknowledgement packets from the target. Details of the test sequences and payloads utilized in each test are given in the Test Specification documents. A gray color payload means the encrypted in the figure of payload and a double allow means the IPsec communication in the figure of sequence.

4.1. IKEv2 BASIC sequences and payloads

This section consist of two items, initial exchange and rekey.

The Initial exchange sequences and payloads are shown from Figure 4-1 to Figure 4-8.

The rekey exchange sequences and payloads are shown from Figure 4-9 to Figure 4-20.

4.1.1. Initial exchange

4.1.1.1. EN to EN

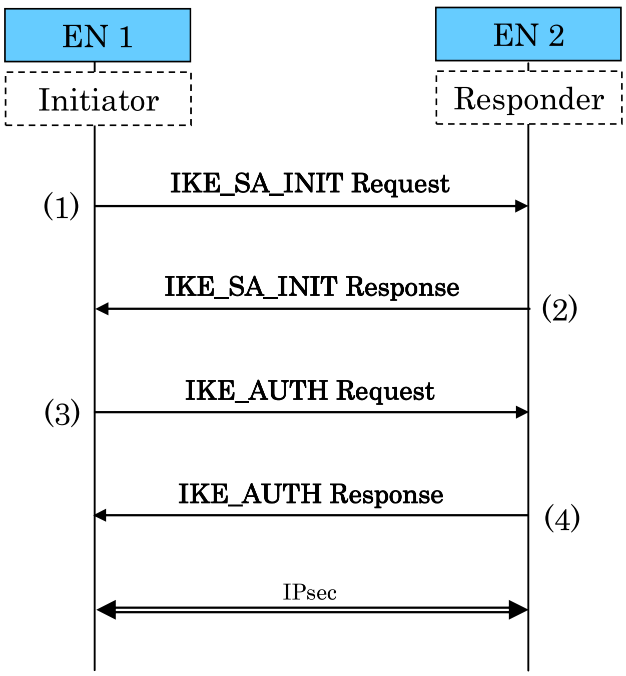
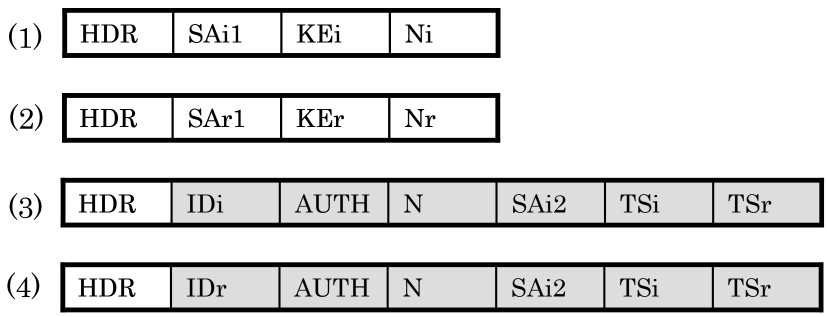


Figure 4-1 EN to EN



N : USE_TRANSPORT_MODE

Figure 4-2 EN to EN Payloads

4.1.1.2. EN to SGW

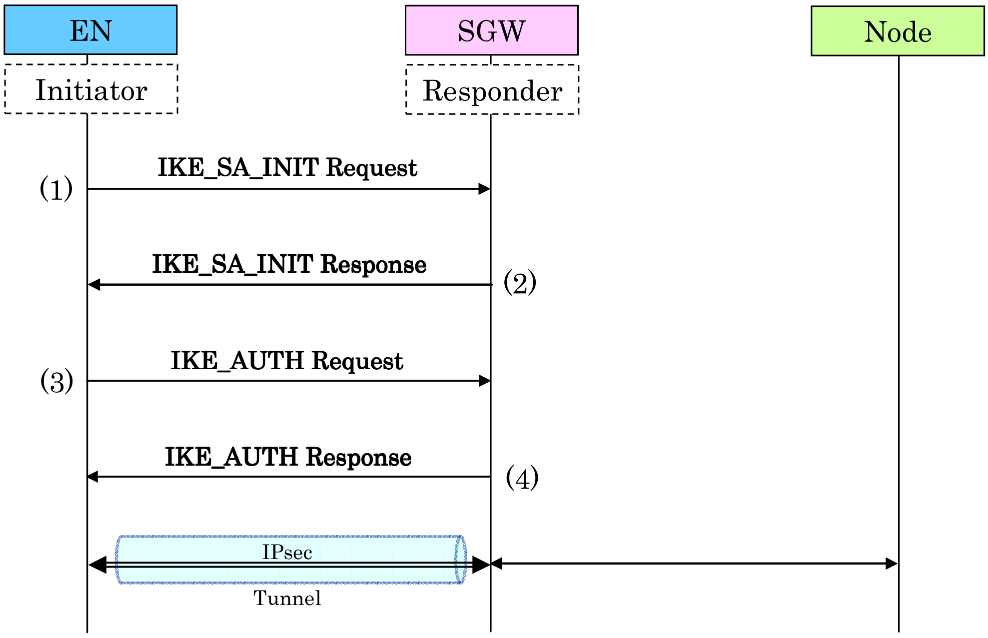


Figure 4-3 EN to SGW

(1)	HDR	SAi1	KEi	Ni		
(2)	HDR	SAr1	KEr	Nr		
(3)	HDR	IDi	AUTH	SAi2	TSi	TSr
(4)	HDR	IDr	AUTH	SAi2	TSi	TSr

Figure 4-4 EN to SGW Payloads

4.1.1.3. SGW to EN

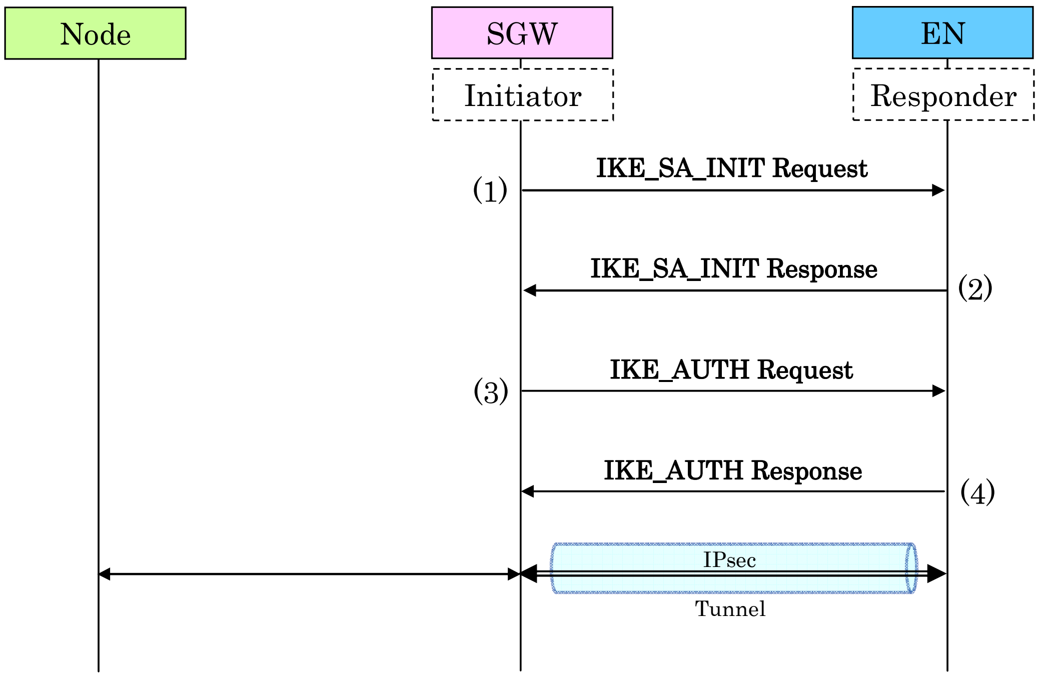


Figure 4-5 SGW to EN

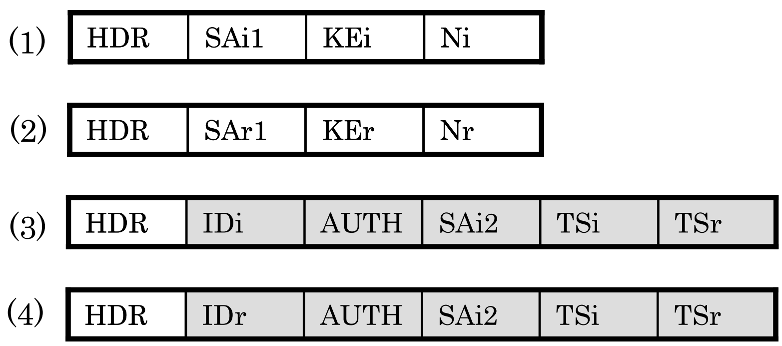


Figure 4-6 SGW to EN Payloads

4.1.1.4. SGW to SGW

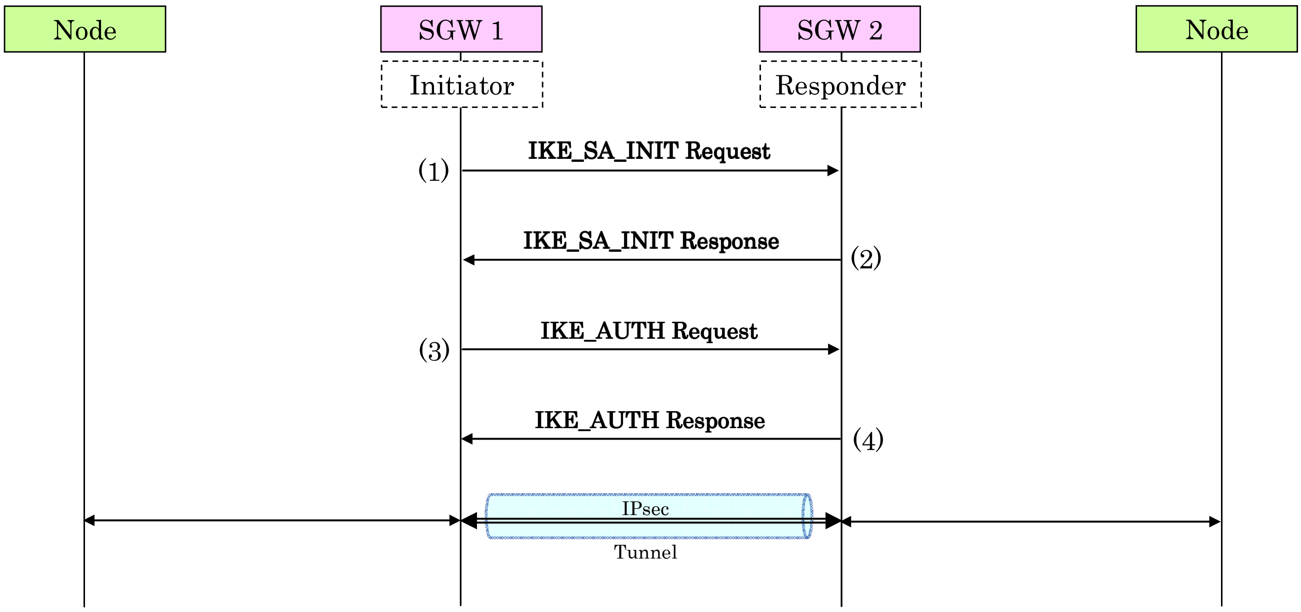


Figure 4-7 SGW to SGW

(1)	HDR	SAi1	KEi	Ni		
(2)	HDR	SAr1	KEr	Nr		
(3)	HDR	IDi	AUTH	SAi2	TSi	TSr
(4)	HDR	IDr	AUTH	SAi2	TSi	TSr

Figure 4-8 SGW to SGW Payloads

4.1.2. Rekey

4.1.2.1. Rekey by EN to EN

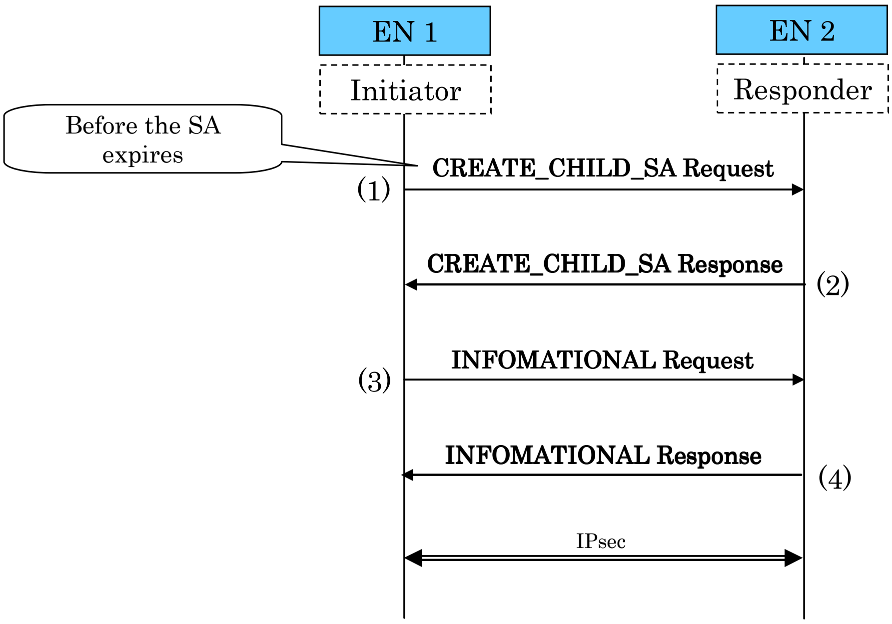


Figure 4-9 Rekey by EN to EN

(1)	HDR	N	N	SA	Ni	[KEi]	TSi	TSr
(2)	HDR	N	SA	Nr	[KEr]	TSi	TSr	
(3)	HDR	D						
(4)	HDR	D						

(1) first N : REKEY_SA
second N : USE_TRANSPORT_MODE
(2) first N : USE_TRANSPORT_MODE
[KEi],[KEr] : PFS

Figure 4-10 CHILD_SA Rekey Payloads by EN to EN

(1)	HDR	SA	Ni	[KEi]
(2)	HDR	SA	Nr	[KEr]
(3)	HDR	D		
(4)	HDR	[D]		

[D] : Optionally Delete Payload in the response is omitted.
[KEi],[KEr] : PFS

Figure 4-11 IKE_SA Rekey Payloads by EN to EN

4.1.2.2. Rekey by EN to SGW

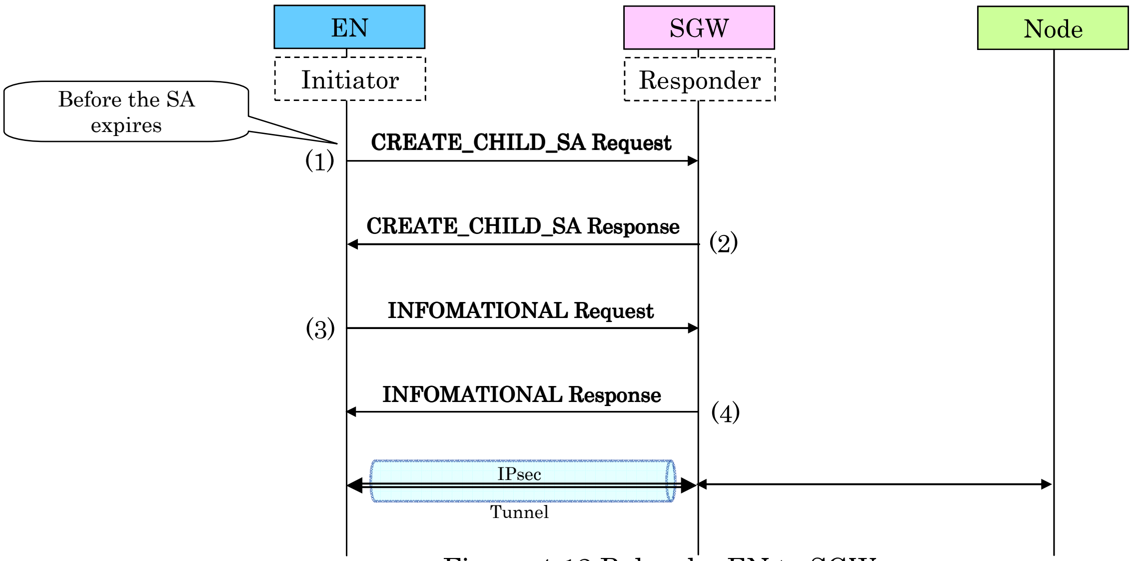


Figure 4-12 Rekey by EN to SGW

(1)	HDR	N	SA	Ni	[KEi]	TSi	TSr
(2)	HDR	SA	Nr	[KEr]	TSi	TSr	
(3)	HDR	D					
(4)	HDR	D					

N : REKEY_SA
[KEi],[KEr] : PFS

Figure 4-13 CHILD_SA Rekey Payloads by EN to SGW

(1)	HDR	SA	Ni	[KEi]
(2)	HDR	SA	Nr	[KEr]
(3)	HDR	D		
(4)	HDR	[D]		

[D] : Optionally Delete Payload in the response is omitted.
[KEi],[KEr] : PFS

Figure 4-14 IKE_SA Rekey Payloads by EN to SGW

4.1.2.3. Rekey by SGW to EN

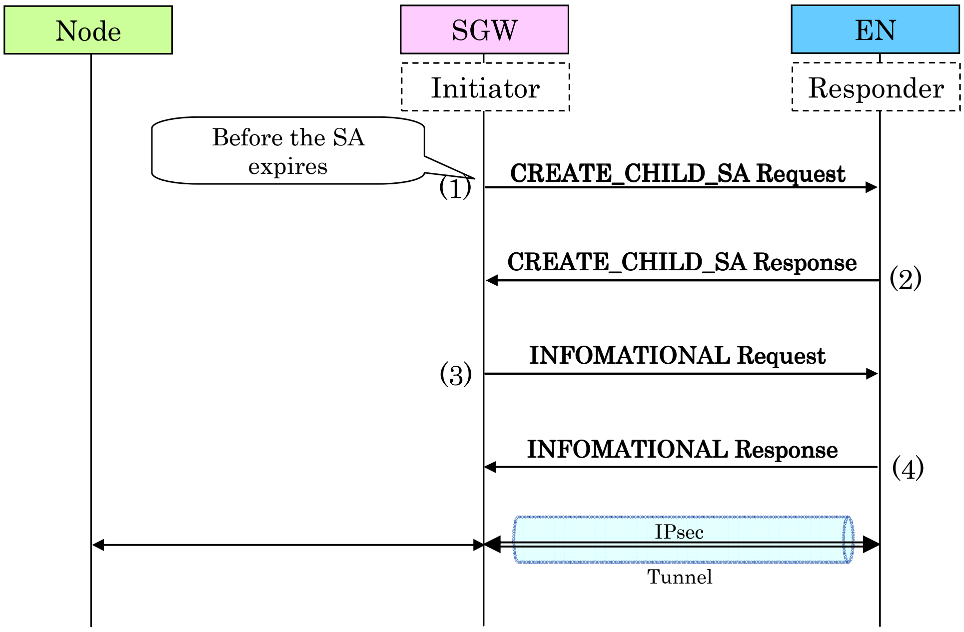


Figure 4-15 Rekey by SGW to EN

(1)	HDR	N	SA	Ni	[KEi]	TSi	TSr
(2)	HDR	SA	Nr	[KEr]	TSi	TSr	
(3)	HDR	D					
(4)	HDR	D					

N : REKEY_SA
[KEi],[KEr] : PFS

Figure 4-16 CHILD_SA Rekey Payloads by SGW to EN

(1)	HDR	SA	Ni	[KEi]
(2)	HDR	SA	Nr	[KEr]
(3)	HDR	D		
(4)	HDR	[D]		

[D] : Optionally Delete Payload in the response is omitted.
[KEi],[KEr] : PFS

Figure 4-17 IKE_SA Rekey Payloads by SGW to EN

4.1.2.4. Rekey by SGW to SGW

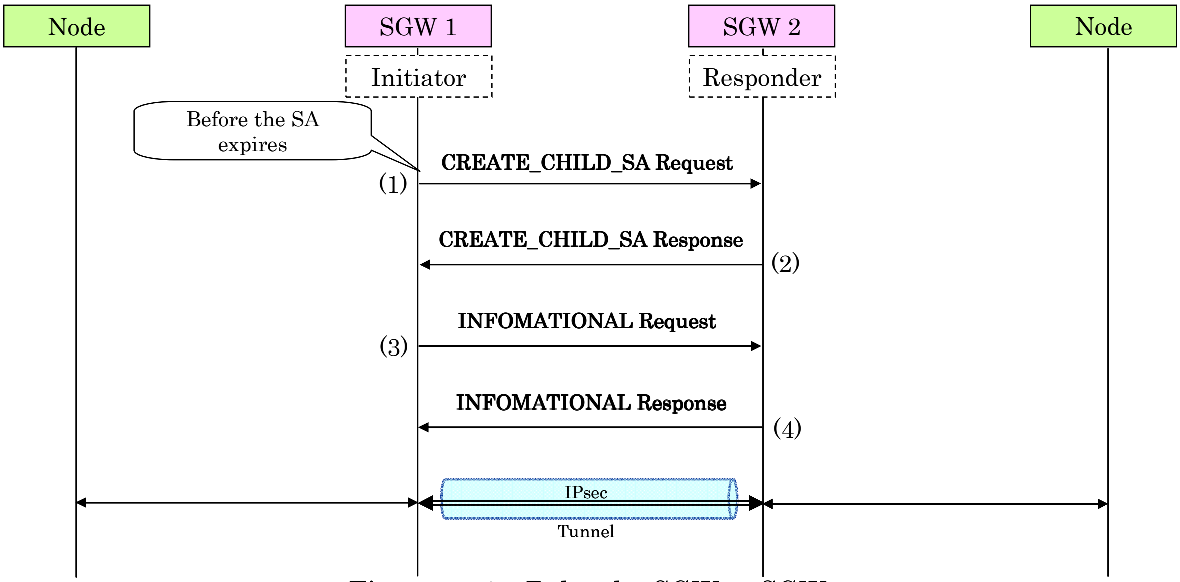


Figure 4-18 Rekey by SGW to SGW

(1)	HDR	N	SA	Ni	[KEi]	TSi	TSr
(2)	HDR	SA	Nr	[KEr]	TSi	TSr	
(3)	HDR	D					
(4)	HDR	D					

N : REKEY_SA
[KEi],[KEr] : PFS

Figure 4-19 CHILD_SA Rekey Payloads by SGW to SGW

(1)	HDR	SA	Ni	[KEi]
(2)	HDR	SA	Nr	[KEr]
(3)	HDR	D		
(4)	HDR	[D]		

[D] : Optionally Delete Payload in the response is omitted.
[KEi],[KEr] : PFS

Figure 4-20 IKE_SA Rekey Payloads by SGW to SGW

4.2. IKEv2 ADVANCED sequences and payloads

This section consist of two cases, mutual authentication using public key signature and Extensible Authentication Protocol (EAP) method. EAP method utilize in this document is EAP with MD5 (EAP-MD5).

The authentication using public key signature sequences and payloads are shown from Figure 4-21 to Figure 4-28.

EAP-MD5 sequences and payloads are shown in Appendix_A.

EAP-TLS sequences and payloads are shown in Appendix_B.

4.2.1. Mutual authentication using public signature

4.2.1.1. Mutual authentication using public key signature in the case of EN-EN

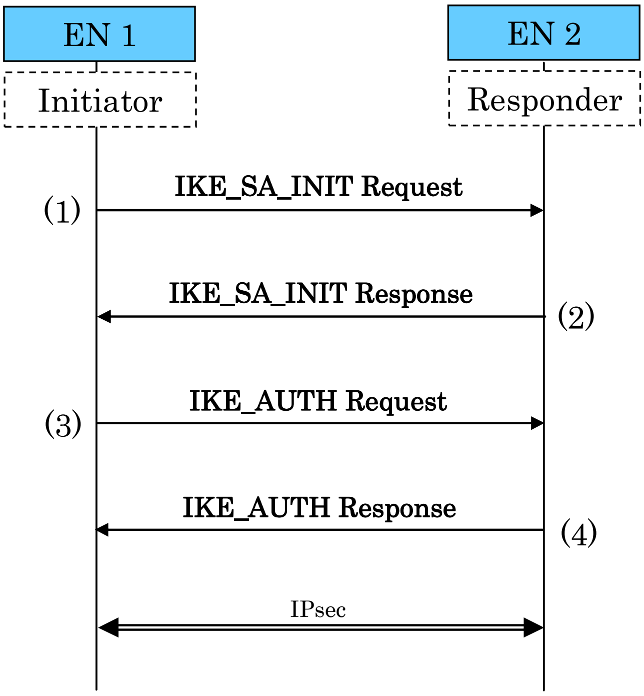


Figure 4-21 Mutual authentication using public key signature by EN to EN

(1)	HDR	SAi1	KEi	Ni					
(2)	HDR	SAr1	KEr	Nr	CERTREQ				
(3)	HDR	IDi	CERT	CERTREQ	AUTH	N	SAi2	TSi	TSr
(4)	HDR	IDr	CERT	AUTH	N	SAi2	TSi	TSr	

N : USE_TRANSPORT_MODE

Figure 4-22 Payloads mutual authentication using public key signature by EN to EN

4.2.1.2. Mutual authentication using public key signature in the case of EN-SGW

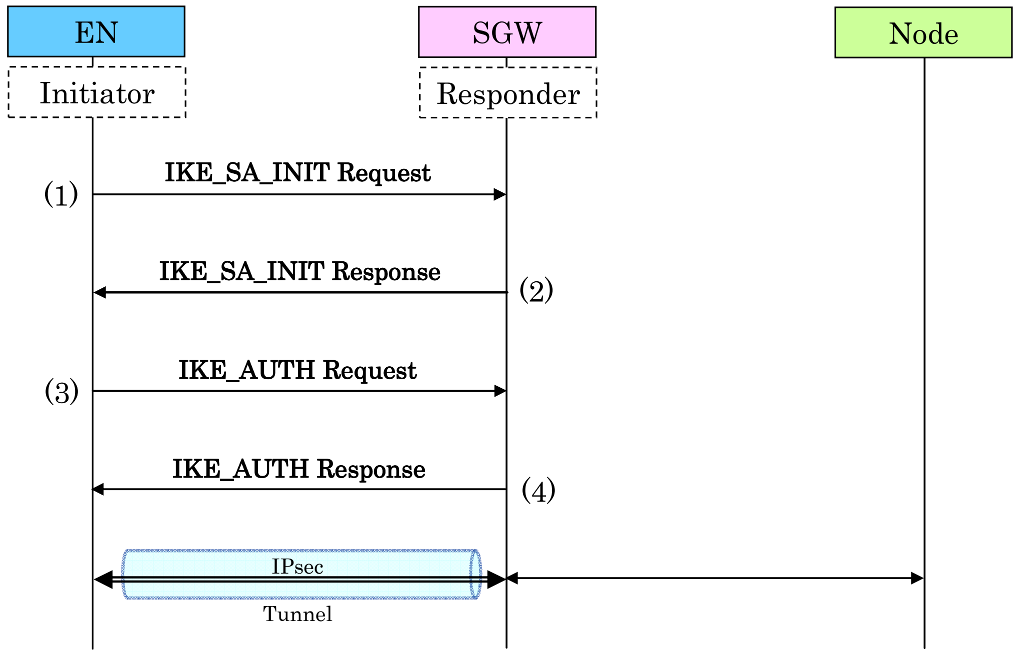


Figure 4-23 Mutual authentication using public key signature by EN to SGW

(1)	HDR	SAi1	KEi	Ni				
(2)	HDR	SAr1	KEr	Nr	CERTREQ			
(3)	HDR	IDi	CERT	CERTREQ	AUTH	SAi2	TSi	TSr
(4)	HDR	IDr	CERT	AUTH	SAi2	TSi	TSr	

Figure 4-24 Payloads mutual authentication using public key signature by EN to SGW

4.2.1.3. Mutual authentication using public key signature in the case of SGW-EN

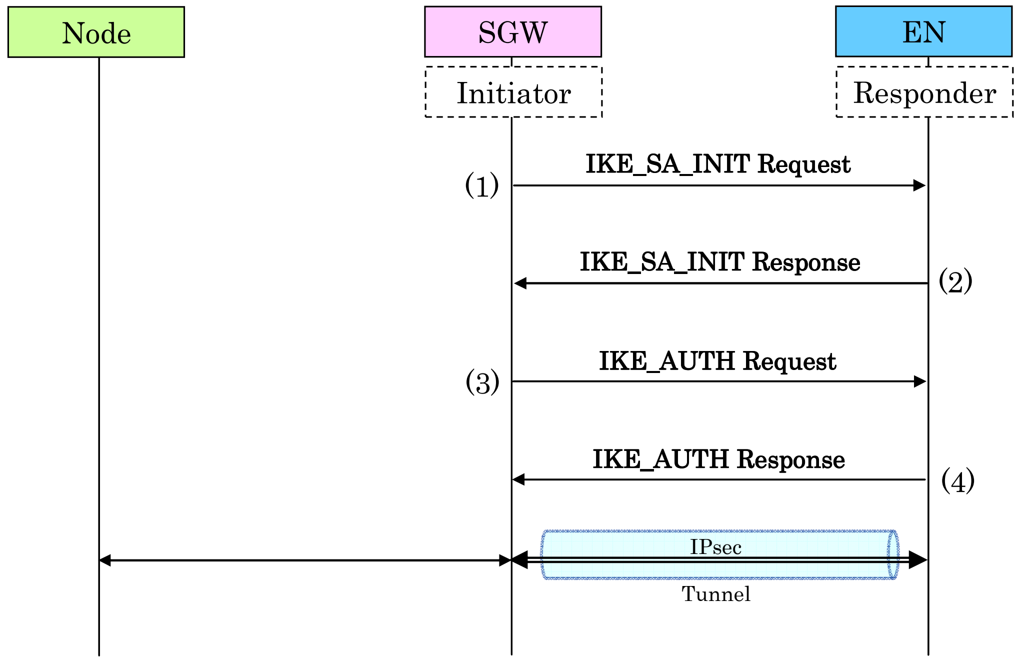


Figure 4-25 Mutual authentication using public key signature by SGW to EN

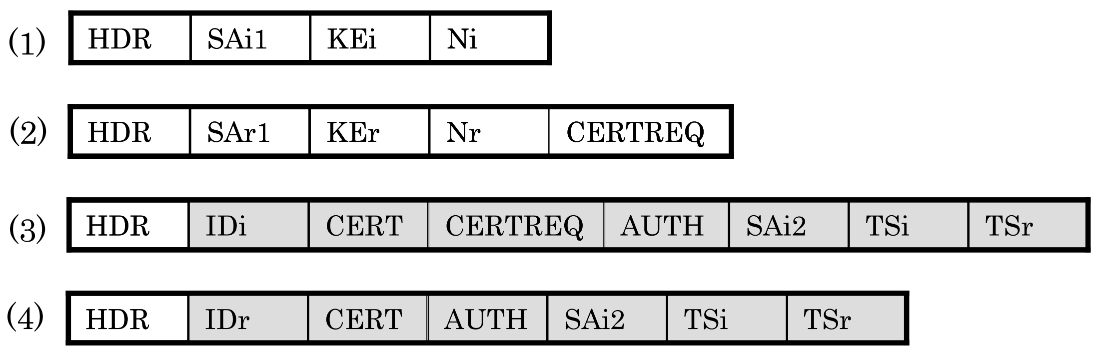


Figure 4-26 Payloads mutual authentication using public key signature by SGW to EN

4.2.1.4. Mutual authentication using public key signature in the case of SGW-SGW

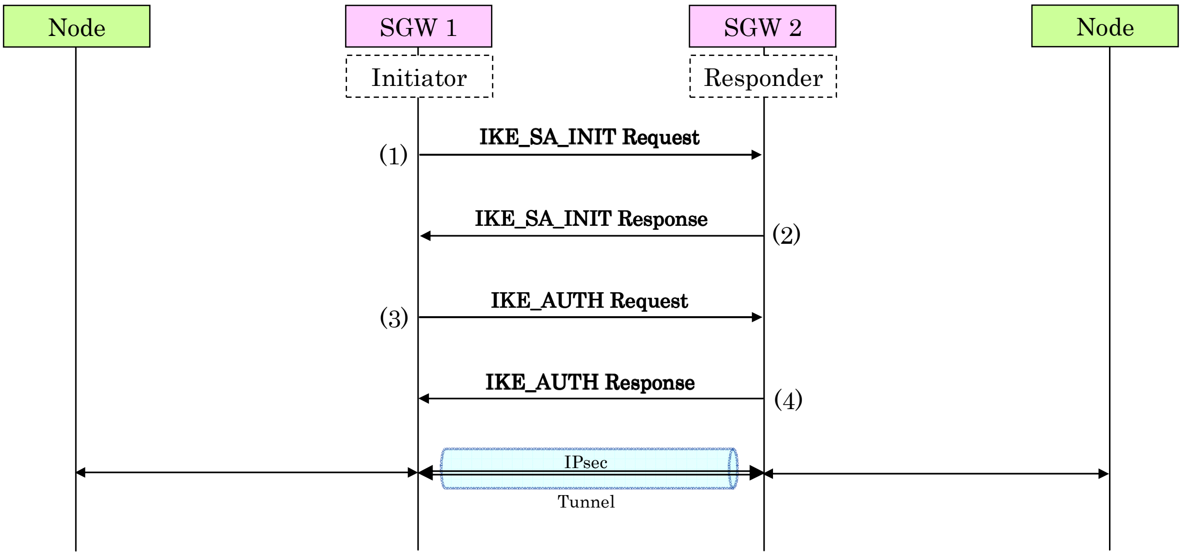


Figure 4-27 Mutual authentication using public key signature by SGW to SGW

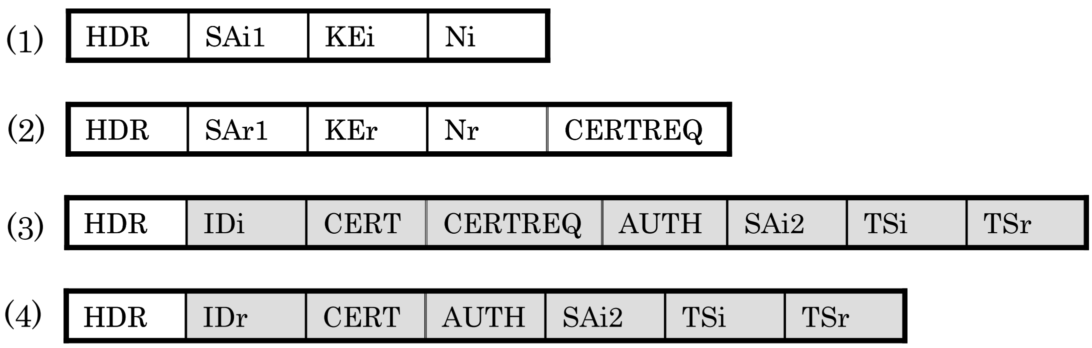


Figure 4-28 Payloads mutual authentication using public key signature by SGW to SGW

5. Priorities of IKEv2 function for testing

This chapter describes the detail IKEv2 functional classifications on the basis of the classifications given in chapter 3.

Priorities of IKEv2 function for testing in RFC5996 are shown at 2.

Notes

- "**page**" gives the corresponding page number in RFC5996.
- "**line**" gives the corresponding line number in RFC5996.
- "**sentence**" gives the statement in RFC5996.
- "**RFC requirement**" gives the corresponding requirement like "MUST" etc. in RFC5996.
- "**test requirement**" gives the corresponding requirement of the conformance test in RFC5996.
- "**target**" gives the corresponding target of the test as follows.

  If the test requirement is "BASIC" or "ADVANCED", the value of this column indicates one or more supporting functions including EN(initiator), EN(responder) , SGW(initiator) and SGW(responder).

  If the test requirement is "Not support",the value of this column is blank.
- "**test number / reason**" gives the corresponding test numbers or reasons as follows.

  If the test requirement is "BASIC" or "ADVANCED", the value of this column indicates the test number of the conformance test specification.

  If the test requirement is "Not support", the value of this column indicates the reason why the part of this description is "Not support".

Table 5-2 IKEv2 functions and its classifications for RFC5996

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
1	15	Abstract				
1	17	This document describes version 2 of the Internet Key Exchange (IKE) protocol. IKE is a component of IPsec used for performing mutual authentication and establishing and maintaining security associations (SAs). This document replaces and updates RFC 4306, and includes all of the clarifications from RFC 4718.		Not support		Explanation
1	23	Status of this Memo				
1	25	This is an Internet Standards Track document.		Not support		Explanation
1	28	This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.		Not support		Explanation
1	33	Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5996.		Not support		Explanation
2	40	Copyright Notice				
2	42	Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.		Not support		Explanation
2	45	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.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
2	55	This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.		Not support		Explanation
5	212	1. Introduction				
5	214	IP Security (IPsec) provides confidentiality, data integrity, access control, and data source authentication to IP datagrams. These services are provided by maintaining shared state between the source and the sink of an IP datagram. This state defines, among other things, the specific services provided to the datagram, which cryptographic algorithms will be used to provide the services, and the keys used as input to the cryptographic algorithms.		Not support		Explanation
5	222	Establishing this shared state in a manual fashion does not scale well. Therefore, a protocol to establish this state dynamically is needed. This document describes such a protocol -- the Internet Key Exchange (IKE). Version 1 of IKE was defined in RFCs 2407 [DOI], 2408 [ISAKMP], and 2409 [IKEV1]. IKEv2 replaced all of those RFCs. IKEv2 was defined in [IKEV2] (RFC 4306) and was clarified in [Clarif] (RFC 4718). This document replaces and updates RFC 4306 and RFC 4718. IKEv2 was a change to the IKE protocol that was not backward compatible. In contrast, the current document not only provides a clarification of IKEv2, but makes minimum changes to the IKE protocol. A list of the significant differences between RFC 4306 and this document is given in Section 1.7.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
5	235	IKE performs mutual authentication between two parties and establishes an IKE security association (SA) that includes shared secret information that can be used to efficiently establish SAs for Encapsulating Security Payload (ESP) [ESP] or Authentication Header (AH) [AH] and a set of cryptographic algorithms to be used by the SAs to protect the traffic that they carry. In this document, the term "suite" or "cryptographic suite" refers to a complete set of algorithms used to protect an SA. An initiator proposes one or more suites by listing supported algorithms that can be combined into suites in a mix-and-match fashion. IKE can also negotiate use of IP Compression (IPComp) [IP-COMP] in connection with an ESP or AH SA. The SAs for ESP or AH that get set up through that IKE SA we call "Child SAs".		Not support		Explanation
5	249	All IKE communications consist of pairs of messages: a request and a response. The pair is called an "exchange", and is sometimes called "request/response pair". The first exchange of messages establishing an IKE SA are called a the IKE_SA_INIT and IKE_AUTH exchanges; subsequent IKE exchanges are called the CREATE_CHILD_SA or INFORMATIONAL exchanges. In the common case, there is a single IKE_SA_INIT exchange and a single IKE_AUTH exchange (a total of four messages) to establish the IKE SA and the first Child SA. In exceptional cases, there may be more than one of each of these exchanges.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
5	258	In all cases, all IKE_SA_INIT exchanges **MUST** complete before any other exchange type, then all IKE_AUTH exchanges **MUST** complete, and following that any number of CREATE_CHILD_SA and INFORMATIONAL exchanges may occur in any order.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.2.1 EN.I.1.1.2.2 EN.I.1.1.2.3 EN.I.1.1.2.4 EN.I.1.2.2.1 EN.I.1.2.2.2 EN.R.1.1.2.1 EN.R.1.1.2.2 EN.R.1.2.2.1 EN.R.1.3.2.1 SGW.I.1.1.2.1 SGW.I.1.1.2.2 SGW.I.1.1.2.3 SGW.I.1.1.2.4 SGW.I.1.2.2.1 SGW.I.1.2.2.2 SGW.R.1.1.2.1 SGW.R.1.1.2.2 SGW.R.1.2.2.1 SGW.R.1.3.2.1
6	261	In some scenarios, only a single Child SA is needed between the IPsec endpoints, and therefore there would be no additional exchanges. Subsequent exchanges **MAY** be used to establish additional Child SAs between the same authenticated pair of endpoints and to perform housekeeping functions.	MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
6	268	An IKE message flow always consists of a request followed by a response. It is the responsibility of the requester to ensure reliability. If the response is not received within a timeout interval, the requester needs to retransmit the request (or abandon the connection).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.2.1 EN.I.1.1.2.2 EN.I.1.1.2.3 EN.I.1.1.2.4 EN.I.1.2.2.1 EN.I.1.2.2.2 EN.R.1.1.2.1 EN.R.1.1.2.2 EN.R.1.2.2.1 EN.R.1.3.2.1 SGW.I.1.1.2.1 SGW.I.1.1.2.2 SGW.I.1.1.2.3 SGW.I.1.1.2.4 SGW.I.1.2.2.1 SGW.I.1.2.2.2 SGW.R.1.1.2.1 SGW.R.1.1.2.2 SGW.R.1.2.2.1 SGW.R.1.3.2.1
6	274	The first exchange of an IKE session, IKE_SA_INIT, negotiates security parameters for the IKE SA, sends nonces, and sends Diffie-Hellman values.		Not support		Explanation
6	278	The second exchange, IKE_AUTH, transmits identities, proves knowledge of the secrets corresponding to the two identities, and sets up an SA for the first (and often only) AH or ESP Child SA (unless there is failure setting up the AH or ESP Child SA, in which case the IKE SA is still established without the Child SA).		Not support		Explanation
6	284	The types of subsequent exchanges are CREATE_CHILD_SA (which creates a Child SA) and INFORMATIONAL (which deletes an SA, reports error conditions, or does other housekeeping). Every request requires a response. An INFORMATIONAL request with no payloads (other than the empty Encrypted payload required by the syntax) is commonly used as a check for liveness. These subsequent exchanges cannot be used until the initial		Not support		Explanation

<table>
<tr><th colspan="2">Section</th><th rowspan="2">Sentence</th><th>RFC</th><th>Test</th><th rowspan="2">Target</th><th rowspan="2">Comments</th></tr>
<tr><th>page</th><th>line</th><th>requirement</th><th>Requirements</th></tr>
<tr><td></td><td></td><td>exchanges have completed.</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>292</td><td>In the description that follows, we assume that no errors occur.
Modifications to the flow when errors occur are described in
Section 2.21.</td><td></td><td>Not support</td><td></td><td>Explanation</td></tr>
<tr><td>6</td><td>296</td><td>1.1. Usage Scenarios</td><td></td><td></td><td></td><td></td></tr>
<tr><td>6</td><td>298</td><td>IKE is used to negotiate ESP or AH SAs in a number of different
scenarios, each with its own special requirements.</td><td></td><td>Not support</td><td></td><td>Explanation</td></tr>
<tr><td>7</td><td>301</td><td>1.1.1. Security Gateway to Security Gateway Tunnel Mode</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">303</td><td rowspan="2">+-+-+-+-+-+ +-+-+-+-+-+
| | IPsec | |
Protected |Tunnel | tunnel |Tunnel |
Protected
Subnet <-->|Endpoint |<---------->|Endpoint |<--> Subnet
| | | |
+-+-+-+-+-+ +-+-+-+-+-+

Figure 1: Security Gateway to Security Gateway Tunnel</td><td rowspan="2"></td><td>BASIC</td><td>SGW(initiator)
SGW(responder)</td><td>SGW.I.1.1.1.1
SGW.I.1.1.1.2
SGW.I.1.1.1.3
SGW.R.1.1.1.1
SGW.R.1.1.1.2
SGW.R.1.1.1.3</td></tr>
<tr><td>Not support</td><td>EN(initiator)
EN(responder)</td><td>EN is out of
scope here.</td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">312</td><td rowspan="2">In this scenario, neither endpoint of the IP connection implements
IPsec, but network nodes between them protect traffic for part of
the way. Protection is transparent to the endpoints, and depends
on ordinary routing to send packets through the tunnel endpoints
for processing. Each endpoint would announce the set of
addresses "behind" it, and packets would be sent in tunnel mode
where the inner IP header would contain the IP addresses of the
actual endpoints.</td><td rowspan="2"></td><td>BASIC</td><td>SGW(initiator)
SGW(responder)</td><td>SGW.I.1.1.1.3
SGW.R.1.1.1.3</td></tr>
<tr><td>Not support</td><td>EN(initiator)
EN(responder)</td><td>EN is out of
scope here.</td></tr>
<tr><td>7</td><td>320</td><td>1.1.2. Endpoint-to-Endpoint Transport Mode</td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">7</td><td rowspan="2">322</td><td rowspan="2">+-+-+-+-+-+
+-+-+-+-+-+
| | IPsec transport |
|
|Protected| or tunnel mode SA
|Protected|
|Endpoint |<--------------------------------------->|Endpoint |</td><td rowspan="2"></td><td>Not support</td><td>SGW(initiator)
SGW(responder)</td><td>SGW is out of
scope here.</td></tr>
<tr><td>BASIC</td><td>EN(initiator)
EN(responder)</td><td>EN.I.1.1.1.1
EN.I.1.1.1.2
EN.I.1.1.1.3
EN.R.1.1.1.1
EN.R.1.1.1.2</td></tr>
</table>

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		| | | | +-+-+-+-+-+ +-+-+-+-+-+ Figure 2: Endpoint to Endpoint				EN.R.1.1.1.3
7	331	In this scenario, both endpoints of the IP connection implement IPsec, as required of hosts in [IPSECARCH]. Transport mode will commonly be used with no inner IP header. A single pair of addresses will be negotiated for packets to be protected by this SA. These endpoints **MAY** implement application-layer access controls based on the IPsec authenticated identities of the participants. This scenario enables the end-to-end security that has been a guiding principle for the Internet since [ARCHPRINC], [TRANSPARENCY], and a method of limiting the inherent problems with complexity in networks noted by [ARCHGUIDEPHIL]. Although this scenario may not be fully applicable to the IPv4 Internet, it has been deployed successfully in specific scenarios within intranets using IKEv1. It should be more broadly enabled during the transition to IPv6 and with the adoption of IKEv2.	MAY	Not support	SGW(initiator) SGW(responder)	SGW is out of scope here.
				BASIC	EN(initiator) EN(responder)	EN.I.1.1.1.3 EN.R.1.1.1.3
8	346	It is possible in this scenario that one or both of the protected endpoints will be behind a network address translation (NAT) node, in which case the tunneled packets will have to be UDP encapsulated so that port numbers in the UDP headers can be used to identify individual endpoints "behind" the NAT (see Section 2.23).		Not support		Explanation
8	352	1.1.3. Endpoint to Security Gateway in Tunnel Mode				
8	354	+-+-+-+-+-+ +-+-+-+-+-+ | | IPsec | | Protected |Protected| tunnel |Tunnel | Subnet		BASIC	SGW(initiator) SGW(responder)	SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.2.1.1.1 SGW.R.2.1.1.2

<table>
<tr><th colspan="2">Section</th><th rowspan="2">Sentence</th><th>RFC</th><th>Test</th><th rowspan="2">Target</th><th rowspan="2">Comments</th></tr>
<tr><th>page</th><th>line</th><th>requirement</th><th>Requirements</th></tr>
<tr><td></td><td></td><td>|Endpoint |<------------------------>|Endpoint |<--- and/or
| | | | Internet
+-+-+-+-+-+ +-+-+-+-+-+

Figure 3: Endpoint to Security Gateway Tunnel</td><td></td><td>ADVANCED</td><td>EN(initiator)
EN(responder)</td><td>EN.I.2.1.1.1
EN.I.2.1.1.2
EN.R.2.1.1.1
EN.R.2.1.1.2</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">363</td><td rowspan="2">In this scenario, a protected endpoint (typically a portable roaming computer) connects back to its corporate network through an IPsec- protected tunnel. It might use this tunnel only to access information on the corporate network, or it might tunnel all of its traffic back through the corporate network in order to take advantage of protection provided by a corporate firewall against Internet-based attacks. In either case, the protected endpoint will want an IP address associated with the security gateway so that packets returned to it will go to the security gateway and be tunneled back. This IP address may be static or may be dynamically allocated by the security gateway. In support of the latter case, IKEv2 includes a mechanism (namely, configuration payloads) for the initiator to request an IP address owned by the security gateway for use for the duration of its SA.</td><td rowspan="2"></td><td>BASIC</td><td>SGW(initiator)
SGW(responder)</td><td>SGW.I.2.1.1.2
SGW.R.2.1.1.2</td></tr>
<tr><td>ADVANCED</td><td>EN(initiator)
EN(responder)</td><td>EN.I.2.1.1.2
EN.R.2.1.1.2</td></tr>
<tr><td rowspan="2">8</td><td rowspan="2">378</td><td rowspan="2">In this scenario, packets will use tunnel mode. On each packet from the protected endpoint, the outer IP header will contain the source IP address associated with its current location (i.e., the address that will get traffic routed to the endpoint directly), while the inner IP header will contain the source IP address assigned by the security gateway (i.e., the address that will get traffic routed to the security gateway for forwarding to the endpoint). The outer destination address will always be that of the security gateway, while the inner destination address will be the ultimate destination for the packet.</td><td rowspan="2"></td><td>BASIC</td><td>SGW(initiator)
SGW(responder)</td><td>SGW.I.2.1.1.1
SGW.I.2.1.1.2
SGW.R.2.1.1.1
SGW.R.2.1.1.2
SGW.R.2.1.2.1
SGW.R.2.1.2.2
SGW.R.2.1.2.3</td></tr>
<tr><td>ADVANCED</td><td>EN(initiator)
EN(responder)</td><td>EN.I.2.1.1.2
EN.I.2.1.2.1
EN.I.2.1.2.2
EN.R.2.1.1.1
EN.R.2.1.1.2</td></tr>
<tr><td>8</td><td>389</td><td>In this scenario, it is possible that the protected endpoint will be behind a NAT. In that case, the IP address as seen by the security gateway will not be the same as the IP address sent by the protected endpoint, and packets will have to be UDP encapsulated</td><td></td><td>Not support</td><td></td><td>Explanation</td></tr>
</table>

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		in order to be routed properly. Interaction with NATs is covered in detail in Section 2.23.				
9	396	1.1.4. Other Scenarios				
9	398	Other scenarios are possible, as are nested combinations of the above. One notable example combines aspects of Sections 1.1.1 and 1.1.3. A subnet may make all external accesses through a remote security gateway using an IPsec tunnel, where the addresses on the subnet are routed to the security gateway by the rest of the Internet. An example would be someone's home network being virtually on the Internet with static IP addresses even though connectivity is provided by an ISP that assigns a single dynamically assigned IP address to the user's security gateway (where the static IP addresses and an IPsec relay are provided by a third party located elsewhere).		Not support		Explanation
9	409	1.2. The Initial Exchanges				
9	411	Communication using IKE always begins with IKE_SA_INIT and IKE_AUTH exchanges (known in IKEv1 as Phase 1). These initial exchanges normally consist of four messages, though in some scenarios that number can grow.		Not support		Explanation
9	414	All communications using IKE consist of request/response pairs.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3
9	415	We'll describe the base exchange first, followed by variations.		Not support		Explanation
9	416	The first pair of messages (IKE_SA_INIT) negotiate cryptographic algorithms, exchange nonces, and do a Diffie-Hellman exchange		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		[DH].				
9	420	The second pair of messages (IKE_AUTH) authenticate the previous messages, exchange identities and certificates, and establish the first Child SA.		Not support		Explanation
9	422	Parts of these messages are encrypted and integrity protected with keys established through the IKE_SA_INIT exchange, so the identities are hidden from eavesdroppers and all fields in all the messages are authenticated.		Not support		Explanation
9	425	See Section 2.14 for information on how the encryption keys are generated. (A man-in-the-middle attacker who cannot complete the IKE_AUTH exchange can nonetheless see the identity of the initiator.)		Not support		Explanation
9	430	All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the IKE_SA_INIT exchange.		Not support		Explanation
9	432	These subsequent messages use the syntax of the Encrypted Payload described in Section 3.14, encrypted with keys that are derived as described in Section 2.14. All subsequent messages include an Encrypted Payload, even if they are referred to in the text as "empty".		Not support		Explanation
9	436	For the CREATE_CHILD_SA, IKE_AUTH, or INFORMATIONAL exchanges, the message following the header is encrypted and the message including the header is integrity protected using the cryptographic algorithms negotiated for the IKE SA.		Not support		Explanation
10	441	Every IKE message contains a Message ID as part of its fixed header. This Message ID is used to match up requests and responses, and to identify retransmissions of messages.		Not support		Explanation
10	445	In the following descriptions, the payloads contained in the message are indicated by names as listed below.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
10	448	Notation Payload --- AUTH Authentication CERT Certificate CERTREQ Certificate Request CP Configuration D Delete EAP Extensible Authentication HDR IKE Header (not a payload) IDi Identification - Initiator IDr Identification - Responder KE Key Exchange Ni, Nr Nonce N Notify SA Security Association SK Encrypted and Authenticated TSi Traffic Selector - Initiator TSr Traffic Selector - Responder V Vendor ID		Not support		Explanation
10	468	The details of the contents of each payload are described in section 3. Payloads that may optionally appear will be shown in brackets, such as [CERTREQ]; this indicates that a certificate request payload can optionally be included.		Not support		Explanation
10	473	The initial exchanges are as follows:		Not support		Explanation
10	475	Initiator Responder --- HDR, SAi1, KEi, Ni -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 SGW.I.1.1.1.1
10	479	HDR contains the Security Parameter Indexes (SPIs), version numbers, and flags of various sorts. The SAi1 payload states the cryptographic algorithms the initiator supports for the IKE SA. The KE payload sends the initiator's Diffie-Hellman value. Ni is the initiator's nonce.		Not support		Explanation
10	485	<-- HDR, SAr1, KEr, Nr, [CERTREQ]		BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.1 SGW.R.1.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
11	487	The responder chooses a cryptographic suite from the initiator's offered choices and expresses that choice in the SAr1 payload, completes the Diffie-Hellman exchange with the KEr payload, and sends its nonce in the Nr payload.		Not support		Explanation
11	492	At this point in the negotiation, each party can generate SKEYSEED, from which all keys are derived for that IKE SA. The messages that follow are encrypted and integrity protected in their entirety, with the exception of the message headers. The keys used for the encryption and integrity protection are derived from SKEYSEED and are known as SK_e (encryption) and SK_a (authentication, a.k.a. integrity protection); see Sections 2.13 and 2.14 for details on the key derivation. A separate SK_e and SK_a is computed for each direction. In addition to the keys SK_e and SK_a derived from the Diffie-Hellman value for protection of the IKE SA, another quantity SK_d is derived and used for derivation of further keying material for Child SAs. The notation SK { ... } indicates that these payloads are encrypted and integrity protected using that direction's SK_e and SK_a.		Not support		Explanation
11	506	HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.2 SGW.I.1.1.1.2
11	510	The initiator asserts its identity with the IDi payload, proves knowledge of the secret corresponding to IDi and integrity protects the contents of the first message using the AUTH payload (see Section 2.15). It might also send its certificate(s) in CERT payload(s) and a list of its trust anchors in CERTREQ payload(s).		Not support		Explanation
11	514	If any CERT payloads are included, the first certificate provided **MUST** contain the public key used to verify the AUTH field.	MUST	Not support		Difficult to test
11	518	The optional payload IDr enables the initiator to specify to which of the responder's identities it wants to talk.		Not support		Explanation
11	519	This is useful when the machine on which the responder is running is hosting multiple identities at the same IP address. If the IDr proposed by the initiator is not acceptable to the responder, the responder might use some other IDr to finish the exchange. If the initiator then does not accept the fact that responder used an IDr		Not Support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		different than the one that was requested, the initiator can close the SA after noticing the fact.				
11	528	The traffic selectors (TSi and TSr) are discussed in Section 2.9.		Not support		Explanation
11	530	The initiator begins negotiation of a Child SA using the SAi2 payload. The final fields (starting with SAi2) are described in the description of the CREATE_CHILD_SA exchange.		Not support		Explanation
12	534	<-- HDR, SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr}		BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.2 SGW.R.1.1.1.2
12	537	The responder asserts its identity with the IDr payload, optionally sends one or more certificates (again with the certificate containing the public key used to verify AUTH listed first), authenticates its identity and protects the integrity of the second message with the AUTH payload, and completes negotiation of a Child SA with the additional fields described below in the CREATE_CHILD_SA exchange.		Not support		Explanation
12	544	Both parties in the IKE_AUTH exchange **MUST** verify that all signatures and Message Authentication Codes (MACs) are computed correctly. If either side uses a shared secret for authentication, the names in the ID payload **MUST** correspond to the key used to generate the AUTH payload.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.3 EN.R.1.1.1.2 SGW.I.1.1.1.3 SGW.R.1.1.1.2
12	549	Because the initiator sends its Diffie-Hellman value in the IKE_SA_INIT, it must guess the Diffie-Hellman group that the responder will select from its list of supported groups.		Not support		Explanation
12	551	If the initiator guesses wrong, the responder will respond with a Notify payload of type INVALID_KE_PAYLOAD indicating the selected group.		ADVANCED *Because DH#14 is ADVANCED group.	EN(responder) SGW(responder)	EN.R.1.1.6.7 SGW.R.1.1.6.7
12	553	In this case, the initiator **MUST** retry the IKE_SA_INIT with the corrected Diffie-Hellman group.	MUST	ADVANCED *Because	EN(initiator) SGW(initiator)	EN.I.1.1.6.7 SGW.I.1.1.6.7

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
				DH#14 is ADVANCED group.		
12	555	The initiator **MUST** again propose its full set of acceptable cryptographic suites because the rejection message was unauthenticated and otherwise an active attacker could trick the endpoints into negotiating a weaker suite than a stronger one that they both prefer.	MUST	Not support		
12	561	If creating the Child SA during the IKE_AUTH exchange fails for some reason, the IKE SA is still created as usual. The list of Notify message types in the IKE_AUTH exchange that do not prevent an IKE SA from being set up include at least the following: NO_PROPOSAL_CHOSEN, TS_UNACCEPTABLE, SINGLE_PAIR_REQUIRED, INTERNAL_ADDRESS_FAILURE, and FAILED_CP_REQUIRED.		BASIC		EN.I.1.1.1.6
12	568	If the failure is related to creating the IKE SA (for example, an AUTHENTICATION_FAILED Notify error message is returned), the IKE SA is not created.		Not Support		Explanation
12	570	Note that although the IKE_AUTH messages are encrypted and integrity protected, if the peer receiving this Notify error message has not yet authenticated the other end (or if the peer fails to authenticate the other end for some reason), the information needs to be treated with caution.		Not support		Explanation
12	574	More precisely, assuming that the MAC verifies correctly, the sender of the error Notify message is known to be the responder of the IKE_SA_INIT exchange, but the sender's identity cannot be assured.		Not support		Explanation
13	579	Note that IKE_AUTH messages do not contain KEi/KEr or Ni/Nr payloads. Thus, the SA payloads in the IKE_AUTH exchange cannot contain Transform Type 4 (Diffie-Hellman Group) with any value other than NONE. Implementations SHOULD omit the whole transform substructure instead of sending value NONE.	SHOULD	BASIC	Both	EN.I.1.1.1.3 EN.R.1.1.1.3 SGW.I.1.1.1.3 SGW.R.1.1.1.3
13	585	1.3. The CREATE_CHILD_SA Exchange				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
13	587	The CREATE_CHILD_SA exchange is used to create new Child SAs and to rekey both IKE SAs and Child SAs. This exchange consists of a single request/response pair, and some of its function was referred to as a phase 2 exchange in IKEv1. It **MAY** be initiated by either end of the IKE SA after the initial exchanges are completed.	MAY	Not support		(ref.) RFC4718 4.1,5.1 -start-
13	593	An SA is rekeyed by creating a new SA and then deleting the old one. This section describes the first part of rekeying, the creation of new SAs; Section 2.8 covers the mechanics of rekeying, including moving traffic from old to new SAs and the deletion of the old SAs. The two sections must be read together to understand the entire process of rekeying.		Not support		Explanation
13	600	Either endpoint may initiate a CREATE_CHILD_SA exchange, so in this section the term initiator refers to the endpoint initiating this exchange.		Not support		Explanation
13	602	An implementation **MAY** refuse all CREATE_CHILD_SA requests within an IKE SA.	MAY	Not support		Not need to test
13	605	The CREATE_CHILD_SA request **MAY** optionally contain a KE payload for an additional Diffie-Hellman exchange to enable stronger guarantees of forward secrecy for the Child SA.	MAY	Not support		Explanation
13	607	The keying material for the Child SA is a function of SK_d established during the establishment of the IKE SA, the nonces exchanged during the CREATE_CHILD_SA exchange, and the Diffie-Hellman value (if KE payloads are included in the CREATE_CHILD_SA exchange).		Not support		Explanation
13	613	If a CREATE_CHILD_SA exchange includes a KEi payload, at least one of the SA offers **MUST** include the Diffie-Hellman group of the KEi.	MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.2.3.7 SGW.I.1.2.3.7
13	614	The Diffie-Hellman group of the KEi **MUST** be an element of the group the initiator expects the responder to accept (additional Diffie-Hellman groups can be proposed).	MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.2.3.7 SGW.I.1.2.3.7
13	617	If the responder selects a proposal using a different Diffie-Hellman group (other than NONE), the responder **MUST** reject the request and indicate its preferred Diffie-Hellman group in the	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.2.5.7 SGW.R.1.2.5.7

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		INVALID_KE_PAYLOAD Notify payload.				
13	620	There are two octets of data associated with this notification: the accepted Diffie-Hellman Group number in big endian order.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.7 EN.R.1.1.6.7 SGW.I.1.1.6.7 SGW.R.1.1.6.7
13	621	In the case of such a rejection, the CREATE_CHILD_SA exchange fails, and the initiator will probably retry the exchange with a Diffie-Hellman proposal and KEi in the group that the responder gave in the INVALID_KE_PAYLOAD Notify payload.		Not Support		INVALID_KE_PAYLOAD test is done at Initial Exchanges
14	627	The responder sends a NO_ADDITIONAL_SAS notification to indicate that a CREATE_CHILD_SA request is unacceptable because the responder is unwilling to accept any more Child SAs on this IKE SA.		Not support		NO_ADDITIONAL_SAS is out of the scope
14	629	This notification can also be used to reject IKE SA rekey. Some minimal implementations may only accept a single Child SA setup in the context of an initial IKE exchange and reject any subsequent attempts to add more.		Not support		Explanation
14	635	1.3.1. Creating New Child SAs with the CREATE_CHILD_SA Exchange				
14	637	A Child SA may be created by sending a CREATE_CHILD_SA request. The CREATE_CHILD_SA request for creating a new Child SA is:		Not support		Explanation
14	640	Initiator Responder --- HDR, SK {SA, Ni, [KEi], TSi, TSr} -->		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.2.1.1 SGW.I.1.2.1.1
14	645	The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, optionally a Diffie-Hellman value in the KEi payload, and the proposed traffic selectors for the proposed Child SA in the TSi and TSr payloads.		Not support		Explanation
14	650	The CREATE_CHILD_SA response for creating a new Child SA is:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
14	652	<-- HDR, SK {SA, Nr, [KEr], TSi, TSr}		ADVANCED	EN(responder) SGW(responder)	EN.R.1.2.1.1 SGW.R.1.2.1.1
14	655	The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if KEi was included in the request and the selected cryptographic suite includes that group.		Not support		Explanation
14	660	The traffic selectors for traffic to be sent on that SA are specified in the TS payloads in the response, which may be a subset of what the initiator of the Child SA proposed.		Not support		Explanation
14	664	The USE_TRANSPORT_MODE notification **MAY** be included in a request message that also includes an SA payload requesting a Child SA.	MAY	BASIC	EN(initiator) EN(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2
14	665	It requests that the Child SA use transport mode rather than tunnel mode for the SA created.		Not support		Explanation
14	667	If the request is accepted, the response **MUST** also include a notification of type USE_TRANSPORT_MODE. If the responder declines the request, the Child SA will be established in tunnel mode.	MUST	BASIC	EN(responder)	EN.R.1.1.1.2
14	670	If this is unacceptable to the initiator, the initiator **MUST** delete the SA.	MUST	Not support		Explanation (Difficult to test)
14	671	Note: Except when using this option to negotiate transport mode, all Child SAs will use tunnel mode.		Not support		Explanation
15	674	The ESP_TFC_PADDING_NOT_SUPPORTED notification asserts that the sending endpoint will not accept packets that contain Traffic Flow Confidentiality (TFC) padding over the Child SA being negotiated. If neither endpoint accepts TFC padding, this notification is included in both the request and the response. If this notification is included in only one of the messages, TFC padding can still be sent in the other direction.		Not support		(ref.)RFC4718 4.5
15	682	The NON_FIRST_FRAGMENTS_ALSO notification is used for fragmentation control. See [IPSECARCH] for a fuller explanation. Both parties need to agree to sending non-first fragments before either party does so. It is enabled only if NON_FIRST_FRAGMENTS_ALSO notification is included in both		Not support		NON_FIRST_FRAGMENTS_ALSO is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		the request proposing an SA and the response accepting it. If the responder does not want to send or receive non-first fragments, it only omits NON_FIRST_FRAGMENTS_ALSO notification from its response, but does not reject the whole Child SA creation.				
15	691	An IPCOMP_SUPPORTED notification, covered in Section 2.22, can also be included in the exchange.		Not support		Explanation
15	694	A failed attempt to create a Child SA **SHOULD NOT** tear down the IKE SA: there is no reason to lose the work done to set up the IKE SA.	SHOULD NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.12 EN.R.1.1.6.9 SGW.I.1.1.6.12 SGW.R.1.1.6.9
15	696	See Section 2.21 for a list of error messages that might occur if creating a Child SA fails.		Not support		Explanation
15	699	1.3.2. Rekeying IKE SAs with the CREATE_CHILD_SA Exchange				
15	701	The CREATE_CHILD_SA request for rekeying an IKE SA is:		Not support		Explanation
15	703	Initiator Responder --- HDR, SK {SA, Ni, KEi} -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.4.1 SGW.I.1.2.4.1
15	707	The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, and a Diffie-Hellman value in the KEi payload. The KEi payload **MUST** be included.	MUST	BASIC	Both	EN.I.1.2.4.1 EN.R.1.2.6.1 SGW.I.1.2.4.1 SGW.R.1.2.6.1
15	709	A new initiator SPI is supplied in the SPI field of the SA payload.		Not Support		Explanation
15	710	Once a peer receives a request to rekey an IKE SA or sends a request to rekey an IKE SA, it **SHOULD NOT** start any new CREATE_CHILD_SA exchanges on the IKE SA that is being rekeyed.	SHOULD NOT	Not Support		untestable
15	714	The CREATE_CHILD_SA response for rekeying an IKE SA is:		Not support		Explanation
15	716	<-- HDR, SK {SA, Nr, KEr}		BASIC	EN(responder) SGW(responder)	EN.R.1.2.6.1 SGW.R.1.2.6.1
15	718	The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if the selected cryptographic suite		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		includes that group. A new responder SPI is supplied in the SPI field of the SA payload.				
16	723	The new IKE SA has its message counters set to 0, regardless of what they were in the earlier IKE SA. The first IKE requests from both sides on the new IKE SA will have message ID 0. The old IKE SA retains its numbering, so any further requests (for example, to delete the IKE SA) will have consecutive numbering. The new IKE SA also has its window size reset to 1, and the initiator in this rekey exchange is the new "original initiator" of the new IKE SA.		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.4.2 SGW.I.1.2.4.2
16	731	Section 2.18 also covers IKE SA rekeying in detail.		Not support		Explanation
16	733	1.3.3. Rekeying Child SAs with the CREATE_CHILD_SA Exchange				
16	735	The CREATE_CHILD_SA request for rekeying a Child SA is:		Not support		Explanation
16	737	Initiator Responder --- HDR, SK {N(REKEY_SA), SA, Ni, [KEi], TSi, TSr} -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.1.1 SGW.I.1.2.1.1
16	742	The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, optionally a Diffie-Hellman value in the KEi payload, and the proposed traffic selectors for the proposed Child SA in the TSi and TSr payloads.		Not support		Explanation
16	747	The notifications described in Section 1.3.1 may also be sent in a rekeying exchange. Usually, these will be the same notifications that were used in the original exchange; for example, when rekeying a transport mode SA, the USE_TRANSPORT_MODE notification will be used.		BASIC	Both	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
16	752	The REKEY_SA notification **MUST** be included in a CREATE_CHILD_SA exchange if the purpose of the exchange is to replace an existing ESP or AH SA.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
16	754	The SA being rekeyed is identified by the SPI field in the Notify payload; this is the SPI the exchange initiator would expect in inbound ESP or AH packets. There is no data associated with		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		this Notify message type. The Protocol ID field of the REKEY_SA notification is set to match the protocol of the SA we are rekeying, for example, 3 for ESP and 2 for AH.				
16	761	The CREATE_CHILD_SA response for rekeying a Child SA is:		Not support		Explanation
16	763	<-- HDR, SK {SA, Nr, [KEr], TSi, TSr}		BASIC	EN(responder) SGW(responder)	EN.R.1.2.1.1 SGW.R.1.2.1.1
16	766	The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if KEi was included in the request and the selected cryptographic suite includes that group.		Not support		Explanation
17	771	The traffic selectors for traffic to be sent on that SA are specified in the TS payloads in the response, which may be a subset of what the initiator of the Child SA proposed.		Not support		Explanation
17	775	1.4. The INFORMATIONAL Exchange				
17	777	At various points during the operation of an IKE SA, peers may desire to convey control messages to each other regarding errors or notifications of certain events. To accomplish this, IKE defines an INFORMATIONAL exchange.		Not support		Explanation
17	780	INFORMATIONAL exchanges **MUST** ONLY occur after the initial exchanges and are cryptographically protected with the negotiated keys.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
17	782	Note that some informational messages, not exchanges, can be sent outside the context of an IKE SA. Section 2.21 also covers error messages in great detail.		Not support		Explanation
17	786	Control messages that pertain to an IKE SA **MUST** be sent under that IKE SA.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
17	787	Control messages that pertain to Child SAs **MUST** be sent under the protection of the IKE SA that generated them (or its successor if the IKE SA was rekeyed).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.8

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.3.9
17	791	Messages in an INFORMATIONAL exchange contain zero or more Notification, Delete, and Configuration payloads.		Not support		Explanation
17	792	The recipient of an INFORMATIONAL exchange request **MUST** send some response; otherwise, the sender will assume the message was lost in the network and will retransmit it.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.10 EN.R.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.9
17	795	That response **MAY** be an empty message. The request message in an INFORMATIONAL exchange **MAY** also contain no payloads. This is the expected way an endpoint can ask the other endpoint to verify that it is alive.	MAY MAY	BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
17	800	The INFORMATIONAL exchange is defined as:		Not support		Explanation
17	802	Initiator Responder --- HDR, SK {[N,] [D,] [CP,] ...} --> <-- HDR, SK {[N,] [D,] [CP], ...}		BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
17	809	The processing of an INFORMATIONAL exchange is determined by its component payloads.		Not support		Explanation
17	812	1.4.1. Deleting an SA with INFORMATIONAL Exchanges				
17	814	ESP and AH SAs always exist in pairs, with one SA in each direction. When an SA is closed, both members of the pair **MUST** be closed (that is, deleted).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.8 SGW.R.1.1.3.9

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
17	816	Each endpoint **MUST** close its incoming SAs and allow the other endpoint to close the other SA in each pair. To delete an SA, an INFORMATIONAL exchange with one or more Delete payloads is sent listing the SPIs (as they would be expected in the headers of inbound packets) of the SAs to be deleted.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.8 SGW.R.1.1.3.9
18	820	The recipient **MUST** close the designated SAs.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.8 SGW.R.1.1.3.9
18	821	Note that one never sends delete payloads for the two sides of an SA in a single message. If there are many SAs to delete at the same time, one includes Delete payloads for the inbound half of each SA pair in the INFORMATIONAL exchange.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.6 EN.R.1.1.3.5 SGW.I.1.1.3.6 SGW.R.1.1.3.5
18	826	Normally, the response in the INFORMATIONAL exchange will contain delete payloads for the paired SAs going in the other direction. There is one exception. If, by chance, both ends of a set of SAs independently decide to close them, each may send a delete payload and the two requests may cross in the network.		Not support		Explanation
18	830	If a node receives a delete request for SAs for which it has already issued a delete request, it **MUST** delete the outgoing SAs while processing the request and the incoming SAs while processing the response.	MUST	Not support		untestable
18	833	In that case, the responses **MUST NOT** include delete payloads for the deleted SAs, since that would result in duplicate deletion and could in theory delete the wrong SA.	MUST NOT	Not support		untestable
18	838	Similar to ESP and AH SAs, IKE SAs are also deleted by sending an Informational exchange. Deleting an IKE SA implicitly closes any remaining Child SAs negotiated under it. The response to a		BASIC	Both	EN.I.1.1.3.6 EN.R.1.1.3.5 SGW.I.1.1.3.6

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		request that deletes the IKE SA is an empty INFORMATIONAL response.				SGW.R.1.1.3.5
18	843	Half-closed ESP or AH connections are anomalous, and a node with auditing capability should probably audit their existence if they persist. Note that this specification does not specify time periods, so it is up to individual endpoints to decide how long to wait.		Not support		Explanation
18	846	A node **MAY** refuse to accept incoming data on half-closed connections but **MUST NOT** unilaterally close them and reuse the SPIs.	MAY MUST NOT	Not support		untestable
18	848	If connection state becomes sufficiently messed up, a node **MAY** close the IKE SA, as described above. It can then rebuild the SAs it needs on a clean base under a new IKE SA.	MAY	Not support		untestable
18	853	1.5. Informational Messages outside of an IKE SA				
18	855	There are some cases in which a node receives a packet that it cannot process, but it may want to notify the sender about this situation.		Not support		Explanation
18	858	o If an ESP or AH packet arrives with an unrecognized SPI. This might be due to the receiving node having recently crashed and lost state, or because of some other system malfunction or attack.		Not support		Explanation
18	862	o If an encrypted IKE request packet arrives on port 500 or 4500 with an unrecognized IKE SPI. This might be due to the receiving node having recently crashed and lost state, or because of some other system malfunction or attack.		Not support		Explanation
19	867	o If an IKE request packet arrives with a higher major version number than the implementation supports.		Not support		Explanation
19	870	In the first case, if the receiving node has an active IKE SA to the IP address from whence the packet came, it **MAY** send an INVALID_SPI notification of the wayward packet over that IKE SA in an INFORMATIONAL exchange.	MAY	Not support		Not need to test
19	873	The Notification Data contains the SPI of the invalid packet. The recipient of this notification cannot tell whether the SPI is for AH or ESP, but this is not important because the SPIs are supposed to be different for the two.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
19	876	If no suitable IKE SA exists, the node **MAY** send an informational message without cryptographic protection to the source IP address, using the source UDP port as the destination port if the packet was UDP (UDP-encapsulated ESP or AH).	MAY	Not support		Explanation
19	880	In this case, it should only be used by the recipient as a hint that something might be wrong (because it could easily be forged).		Not support		Explanation
19	882	This message is not part of an INFORMATIONAL exchange, and the receiving node **MUST NOT** respond to it because doing so could cause a message loop.	MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.2 EN.R.1.1.3.2 SGW.I.1.1.3.2 SGW.R.1.1.3.2
19	884	The message is constructed as follows:		Not support		Explanation
19	886	there are no IKE SPI values that would be meaningful to the recipient of such a notification;		Not support		Explanation
19	888	using zero values or random values are both acceptable, this being the exception to the rule in Section 3.1 that prohibits zero IKE Initiator SPIs.		Not support		Explanation
19	890	The Initiator flag is set to 1, the Response flag is set to 0, and the version flags are set in the normal fashion;		Not support		Explanation
19	891	these flags are described in Section 3.1.		Not support		Explanation
19	893	In the second and third cases, the message is always sent without cryptographic protection (outside of an IKE SA), and includes either an INVALID_IKE_SPI or an INVALID_MAJOR_VERSION notification (with no notification data). The message is a response message, and thus it is sent to the IP address and port from whence it came with the same IKE SPIs and the Message ID and Exchange Type are copied from the request. The Response flag is set to 1, and the version flags are set in the normal fashion.		BASIC	Both	INVALID_MAJOR_VERSION EN.R.1.1.4.2 SGW.R.1.1.4.2 INVALID_IKE_SPI -> untestable
19	902	1.6. Requirements Terminology				
19	904	Definitions of the primitive terms in this document (such as Security Association or SA) can be found in [IPSECARCH]. It should be noted that parts of IKEv2 rely on some of the processing rules in [IPSECARCH], as described in various sections of this document.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
19	909	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 [MUSTSHOULD].		Not support		Explanation
20	913	1.7. Significant Differences Between RFC 4306 and This Document				
20	915	This document contains clarifications and amplifications to IKEv2 [IKEV2]. Many of the clarifications are based on [Clarif]. The changes listed in that document were discussed in the IPsec Working Group and, after the Working Group was disbanded, on the IPsec mailing list. That document contains detailed explanations of areas that were unclear in IKEv2, and is thus useful to implementers of IKEv2.		Not support		Explanation
20	923	The protocol described in this document retains the same major version number (2) and minor version number (0) as was used in RFC 4306. That is, the version number is *not* changed from RFC 4306. The small number of technical changes listed here are not expected to affect RFC 4306 implementations that have already been deployed at the time of publication of this document.		Not support		Explanation
20	930	This document makes the figures and references a bit more consistent than they were in [IKEV2].		Not support		Explanation
20	933	IKEv2 developers have noted that the SHOULD-level requirements in RFC 4306 are often unclear in that they don't say when it is OK to not obey the requirements. They also have noted that there are MUST-level requirements that are not related to interoperability. This document has more explanation of some of these requirements. All non-capitalized uses of the words SHOULD and MUST now mean their normal English sense, not the interoperability sense of [MUSTSHOULD].		Not support		Explanation
20	941	IKEv2 (and IKEv1) developers have noted that there is a great deal of material in the tables of codes in Section 3.10.1 in RFC 4306. This leads to implementers not having all the needed information in the main body of the document. Much of the material from those tables has been moved into the associated parts of the main body of the document.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
20	948	This document removes discussion of nesting AH and ESP. This was a mistake in RFC 4306 caused by the lag between finishing RFC 4306 and RFC 4301. Basically, IKEv2 is based on RFC 4301, which does not include "SA bundles" that were part of RFC 2401. While a single packet can go through IPsec processing multiple times, each of these passes uses a separate SA, and the passes are coordinated by the forwarding tables. In IKEv2, each of these SAs has to be created using a separate CREATE_CHILD_SA exchange.		Not support		Explanation
20	957	This document removes discussion of the INTERNAL_ADDRESS_EXPIRY configuration attribute because its implementation was very problematic.		Not support		Explanation
20	959	Implementations that conform to this document **MUST** ignore proposals that have configuration attribute type 5, the old value for INTERNAL_ADDRESS_EXPIRY.	MUST	Not support		Explanation
21	961	This document also removed INTERNAL_IP6_NBNS as a configuration attribute.		Not support		Explanation
21	964	This document removes the allowance for rejecting messages in which the payloads were not in the "right" order; now implementations **MUST NOT** reject them. This is due to the lack of clarity where the orders for the payloads are described.	MUST NOT	Not support		Explanation
21	969	The lists of items from RFC 4306 that ended up in the IANA registry were trimmed to only include items that were actually defined in RFC 4306. Also, many of those lists are now preceded with the very important instruction to developers that they really should look at the IANA registry at the time of development because new items have been added since RFC 4306.		Not support		Explanation
21	976	This document adds clarification on when notifications are and are not sent encrypted, depending on the state of the negotiation at the time.		Not support		Explanation
21	980	This document discusses more about how to negotiate combined-mode ciphers.		Not support		Explanation
21	983	In section 1.3.2, changed "The KEi payload SHOULD be included" was changed to be "The KEi payload **MUST** be included". This		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		also led to changes in section 2.18.				
21	987	In Section 2.1, there is new material covering how the initiator's SPI and/or IP is used to differentiate if this is a "half-open" IKE SA or a new request.		Not support		Explanation
21	991	This document clarifies the use of the critical flag in Section 2.5.		Not support		Explanation
21	993	In Section 2.8, changed "Note that, when rekeying, the new Child SA MAY have different traffic selectors and algorithms than the old one" was changed to "Note that, when rekeying, the new Child SA SHOULD NOT have different traffic selectors and algorithms than the old one".		Not support		Explanation
21	998	The new Section 2.8.2 covers simultaneous IKE SA rekeying.		Not support		Explanation
21	1000	The new Section 2.9.2 covers traffic selectors in rekeying.		Not support		Explanation
21	1002	This document adds the restriction in Section 2.13 that all pseudorandom functions (PRFs) used with IKEv2 **MUST** take variable-sized keys. This should not affect any implementations because there were no standardized PRFs that have fixed-size keys.		Not support		Explanation
22	1007	Section 2.18 requires doing a Diffie-Hellman exchange when rekeying the IKE_SA. In theory, RFC 4306 allowed a policy where the Diffie-Hellman exchange was optional, but this was not useful (or appropriate) when rekeying the IKE_SA.		Not support		Explanation
22	1012	Section 2.21 has been greatly expanded to cover the different cases where error responses are needed and the appropriate responses to them.		Not support		Explanation
22	1016	Section 2.23 clarified that, in NAT traversal, now both UDP encapsulated IPsec packets and non-UDP-encapsulated IPsec packets need to be understood when receiving.		Not support		Explanation
22	1020	Added Section 2.23.1 to describe NAT traversal when transport mode is requested.		Not support		Explanation
22	1023	Added Section 2.25 to explain how to act when there are timing collisions when deleting and/or rekeying SAs, and two new error notifications (TEMPORARY_FAILURE and		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		CHILD_SA_NOT_FOUND) were defined.				
22	1028	In Section 3.6, "Implementations **MUST** support the HTTP method for hash-and-URL lookup. The behavior of other URL methods is not currently specified, and such methods **SHOULD NOT** be used in the absence of a document specifying them."		Not support		Explanation
22	1033	In Section 3.15.3, a pointer to a new document that is related to configuration of IPv6 addresses.		Not support		Explanation
22	1036	Appendix C was expanded and clarified.		Not support		Explanation
22	1039	2. IKE Protocol Details and Variations				
22	1041	IKE normally listens and sends on UDP port 500, though IKE messages may also be received on UDP port 4500 with a slightly different format (see Section 2.23).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
22	1043	Since UDP is a datagram (unreliable) protocol, IKE includes in its definition recovery from transmission errors, including packet loss, packet replay, and packet forgery. IKE is designed to function so long as (1) at least one of a series of retransmitted packets reaches its destination before timing out; and (2) the channel is not so full of forged and replayed packets so as to exhaust the network or CPU capacities of either endpoint. Even in the absence of those minimum performance requirements, IKE is designed to fail cleanly (as though the network were broken).		Not support		Explanation
22	1053	Although IKEv2 messages are intended to be short, they contain structures with no hard upper bound on size (in particular, digital certificates), and IKEv2 itself does not have a mechanism for fragmenting large messages. IP defines a mechanism for fragmentation of oversized UDP messages, but implementations vary in the maximum message size supported. Furthermore, use of IP fragmentation opens an implementation to denial-of-service (DoS) attacks [DOSUDPPROT]. Finally, some NAT and/or firewall implementations may block IP fragments.		Not support		Explanation
23	1063	All IKEv2 implementations **MUST** be able to send, receive, and process IKE messages that are up to 1280 octets long, and they	MUST SHOULD	Not support		Difficult to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		SHOULD be able to send, receive, and process messages that are up to 3000 octets long.				
23	1066	IKEv2 implementations need to be aware of the maximum UDP message size supported and **MAY** shorten messages by leaving out some certificates or cryptographic suite proposals if that will keep messages below the maximum.	MAY	Not support		Explanation
23	1069	Use of the "Hash and URL" formats rather than including certificates in exchanges where possible can avoid most problems. Implementations and configuration need to keep in mind, however, that if the URL lookups are possible only after the Child SA is established, recursion issues could prevent this technique from working.		Not support		Explanation
23	1076	The UDP payload of all packets containing IKE messages sent on port 4500 **MUST** begin with the prefix of four zeros; otherwise, the receiver won't know how to handle them.	MUST	Not Support		NAT traversal is out of the scope
23	1080	2.1. Use of Retransmission Timers				
23	1082	All messages in IKE exist in pairs: a request and a response. The setup of an IKE SA normally consists of two exchanges. Once the IKE SA is set up, either end of the security association may initiate requests at any time, and there can be many requests and responses "in flight" at any given moment. But each message is labeled as either a request or a response, and for each exchange, one end of the security association is the initiator and the other is the responder.		Not support		Explanation
23	1090	For every pair of IKE messages, the initiator is responsible for retransmission in the event of a timeout.		Not support		Explanation
23	1091	The responder **MUST** never retransmit a response unless it receives a retransmission of the request.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.2.1 EN.R.1.1.2.2 EN.R.1.2.2.1 EN.R.1.3.2.1 SGW.R.1.1.2.1 SGW.R.1.1.2.2 SGW.R.1.2.2.1 SGW.R.1.3.2.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
23	1093	In that event, the responder **MUST** ignore the retransmitted request except insofar as it causes a retransmission of the response.	MUST	Not support		Difficult to test
23	1095	The initiator **MUST** remember each request until it receives the corresponding response.	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.2.1 EN.I.1.1.2.3 EN.I.1.2.2.1 SGW.I.1.1.2.1 SGW.I.1.1.2.3 SGW.I.1.2.2.1
23	1096	The responder **MUST** remember each response until it receives a request whose sequence number is larger than or equal to the sequence number in the response plus its window size (see Section 2.3).	MUST	Not support		Window size is "Not support"
23	1099	In order to allow saving memory, responders are allowed to forget the response after a timeout of several minutes. If the responder receives a retransmitted request for which it has already forgotten the response, it **MUST** ignore the request (and not, for example, attempt constructing a new response).	MUST	Not Support		test condition is ambiguous (several minutes)
24	1105	IKE is a reliable protocol: the initiator **MUST** retransmit a request until it either receives a corresponding response or until it deems the IKE SA to have failed. In the latter case, the initiator discards all state associated with the IKE SA and any Child SAs that were negotiated using that IKE SA.	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.2.2 EN.I.1.1.2.4 EN.I.1.2.2.2 SGW.I.1.1.2.2 SGW.I.1.1.2.4 SGW.I.1.2.2.2
24	1109	A retransmission from the initiator **MUST** be bitwise identical to the original request. That is, everything starting from the IKE Header (the IKE SA initiator's SPI onwards) must be bitwise identical; items before it (such as the IP and UDP headers) do not have to be identical.	MUST	BASIC	Both	EN.I.1.1.2.1 EN.I.1.1.2.3 EN.I.1.2.2.1 SGW.I.1.1.2.1 SGW.I.1.1.2.3 SGW.I.1.2.2.1
24	1115	Retransmissions of the IKE_SA_INIT request require some special handling. When a responder receives an IKE_SA_INIT request, it has to determine whether the packet is a retransmission belonging to an existing "half-open" IKE SA (in which case the		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		responder retransmits the same response), or a new request (in which case the responder creates a new IKE SA and sends a fresh response), or it belongs to an existing IKE SA where the IKE_AUTH request has been already received (in which case the responder ignores it).				
24	1124	It is not sufficient to use the initiator's SPI and/or IP address to differentiate between these three cases because two different peers behind a single NAT could choose the same initiator SPI.		Not support		Explanation
24	1126	Instead, a robust responder will do the IKE SA lookup using the whole packet, its hash, or the Ni payload.		Not support		Explanation
24	1130	The retransmission policy for one-way messages is somewhat different from that for regular messages. Because no acknowledgement is ever sent, there is no reason to gratuitously retransmit one-way messages. Given that all these messages are errors, it makes sense to send them only once per "offending" packet, and only retransmit if further offending packets are received. Still, it also makes sense to limit retransmissions of such error messages.		Not support		Explanation
24	1138	2.2. Use of Sequence Numbers for Message ID				
24	1140	Every IKE message contains a Message ID as part of its fixed header. This Message ID is used to match up requests and responses and to identify retransmissions of messages.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.2.2 EN.I.1.1.2.4 EN.I.1.2.2.2 EN.R.1.1.2.1 EN.R.1.1.2.2 EN.R.1.2.2.1 EN.R.1.3.2.1 SGW.I.1.1.2.2 SGW.I.1.1.2.4 SGW.I.1.2.2.2 SGW.R.1.1.2.1 SGW.R.1.1.2.2 SGW.R.1.2.2.1 SGW.R.1.3.2.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
24	1142	Retransmission of a message **MUST** use the same Message ID as the original message.	MUST	BASIC	Both	EN.I.1.1.2.1 EN.I.1.1.2.3 EN.I.1.2.2.1 SGW.I.1.1.2.1 SGW.I.1.1.2.3 SGW.I.1.2.2.1
24	1145	The Message ID is a 32-bit quantity, which is zero for the IKE_SA_INIT messages (including retries of the message due to responses such as COOKIE and INVALID_KE_PAYLOAD), and incremented for each subsequent exchange.		BASIC	Both	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
24	1148	Thus, the first pair of IKE_AUTH messages will have an ID of 1,		BASIC	Both	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2
24	1149	the second (when EAP is used) will be 2, and so on.		Not Support		EAP authentication is out of the scope
24	1150	The Message ID is reset to zero in the new IKE SA after the IKE SA is rekeyed.		BASIC	Both	EN.I.1.2.4.1 EN.R.1.2.6.1 SGW.I.1.2.4.1 SGW.R.1.2.6.1
25	1153	Each endpoint in the IKE Security Association maintains two "current" Message IDs: the next one to be used for a request it initiates and the next one it expects to see in a request from the other end.		Not support		Explanation
25	1156	These counters increment as requests are generated and received. Responses always contain the same message ID as the corresponding request.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2
25	1158	That means that after the initial exchange, each integer n may appear as the message ID in four distinct messages: the nth request from the original IKE initiator, the corresponding response, the nth request from the original IKE responder, and the corresponding response. If the two ends make a very different number of requests, the Message IDs in the two directions can be		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		very different. There is no ambiguity in the messages, however, because the Initiator and Response flags in the message header specify which of the four messages a particular one is.				
25	1168	Throughout this document, "initiator" refers to the party who initiated the exchange being described. The "original initiator" always refers to the party who initiated the exchange that resulted in the current IKE SA. In other words, if the "original responder" starts rekeying the IKE SA, that party becomes the "original initiator" of the new IKE SA.		Not support		Explanation
25	1175	Note that Message IDs are cryptographically protected and provide protection against message replays.		Not support		Explanation
25	1176	In the unlikely event that Message IDs grow too large to fit in 32 bits, the IKE SA **MUST** be closed or rekeyed.	MUST	Not support		2^32 waiting difficult
25	1180	2.3. Window Size for Overlapping Requests				
25	1182	The SET_WINDOW_SIZE notification asserts that the sending endpoint is capable of keeping state for multiple outstanding exchanges, permitting the recipient to send multiple requests before getting a response to the first.		Not support		Explanation
25	1185	The data associated with a SET_WINDOW_SIZE notification **MUST** be 4 octets long and contain the big endian representation of the number of messages the sender promises to keep.	MUST	Not support		SET_WINDOW_ SIZE is out of the scope
25	1188	The window size is always one until the initial exchanges complete.		Not support		Explanation
25	1190	An IKE endpoint **MUST** wait for a response to each of its messages before sending a subsequent message unless it has received a SET_WINDOW_SIZE Notify message from its peer informing it that the peer is prepared to maintain state for multiple outstanding messages in order to allow greater throughput.	MUST	Not support		Not need to test
25	1196	After an IKE SA is set up, in order to maximize IKE throughput, an IKE endpoint **MAY** issue multiple requests before getting a response to any of them, up to the limit set by its peer's SET_WINDOW_SIZE. These requests may pass one another over the network.	MAY	Not support		Not need to test
25	1199	An IKE endpoint **MUST** be prepared to accept and process a	MUST	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		request while it has a request outstanding in order to avoid a deadlock in this situation.				
26	1202	An IKE endpoint may also accept and process multiple requests while it has a request outstanding.		Not support		Difficult to test
26	1205	An IKE endpoint **MUST NOT** exceed the peer's stated window size for transmitted IKE requests. In other words, if the responder stated its window size is N, then when the initiator needs to make a request X, it **MUST** wait until it has received responses to all requests up through request X-N.	MUST NOT MUST	Not support		window size is "Not support"
26	1209	An IKE endpoint **MUST** keep a copy of (or be able to regenerate exactly) each request it has sent until it receives the corresponding response.	MUST	Not support		Internal process
26	1211	An IKE endpoint **MUST** keep a copy of (or be able to regenerate exactly) the number of previous responses equal to its declared window size in case its response was lost and the initiator requests its retransmission by retransmitting the request.	MUST	Not support		Internal process
26	1216	An IKE endpoint supporting a window size greater than one ought to be capable of processing incoming requests out of order to maximize performance in the event of network failures or packet reordering.		Not support		window size is "Not support"
26	1220	The window size is normally a (possibly configurable) property of a particular implementation, and is not related to congestion control (unlike the window size in TCP, for example).		Not support		Explanation
26	1222	In particular, that the responder should do when it receives a SET_WINDOW_SIZE notification containing a smaller value than is currently in effect is not defined. Thus, there is currently no way to reduce the window size of an existing IKE SA; you can only increase it. When rekeying an IKE SA, the new IKE SA starts with window size 1 until it is explicitly increased by sending a new SET_WINDOW_SIZE notification.		Not support		Explanation
26	1231	The INVALID_MESSAGE_ID notification is sent when an IKE message ID outside the supported window is received.		Not support		Explanation
26	1232	This Notify message **MUST NOT** be sent in a response;	MUST NOT	Not support		Explanation
26	1233	the invalid request **MUST NOT** be acknowledged.	MUST NOT	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
26	1234	Instead, inform the other side by initiating an INFORMATIONAL exchange with Notification data containing the four-octet invalid message ID.		Not support		Explanation
26	1236	Sending this notification is OPTIONAL, and notifications of this type **MUST** be rate limited.	MUST	Not support		Explanation
26	1239	2.4. State Synchronization and Connection Timeouts				
26	1241	An IKE endpoint is allowed to forget all of its state associated with an IKE SA and the collection of corresponding Child SAs at any time. This is the anticipated behavior in the event of an endpoint crash and restart. It is important when an endpoint either fails or reinitializes its state that the other endpoint detect those conditions and not continue to waste network bandwidth by sending packets over discarded SAs and having them fall into a black hole.		Not support		Explanation
27	1249	The INITIAL_CONTACT notification asserts that this IKE SA is the only IKE SA currently active between the authenticated identities.		Not support		untestable
27	1250	It **MAY** be sent when an IKE SA is established after a crash, and the recipient **MAY** use this information to delete any other IKE SAs it has to the same authenticated identity without waiting for a timeout.	MAY MAY	Not support		Explanation
27	1254	This notification **MUST NOT** be sent by an entity that may be replicated (e.g., a roaming user's credentials where the user is allowed to connect to the corporate firewall from two remote systems at the same time).	MUST NOT	Not support		Difficult to test
27	1257	The INITIAL_CONTACT notification, if sent, **MUST** be in the first IKE_AUTH request or response, not as a separate exchange afterwards; receiving parties **MAY** ignore it in other messages.	MUST MAY	Not support		untestable
27	1262	Since IKE is designed to operate in spite of DoS attacks from the network, an endpoint **MUST NOT** conclude that the other endpoint has failed based on any routing information (e.g., ICMP messages) or IKE messages that arrive without cryptographic protection (e.g., Notify messages complaining about unknown SPIs).	MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.1 EN.I.1.1.3.2 EN.R.1.1.3.1 EN.R.1.1.3.2 SGW.I.1.1.3.1 SGW.I.1.1.3.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.3.1 SGW.R.1.1.3.2
27	1266	An endpoint **MUST** conclude that the other endpoint has failed only when repeated attempts to contact it have gone unanswered for a timeout period or when a cryptographically protected INITIAL_CONTACT notification is received on a different IKE SA to the same authenticated identity.	MUST	Not Support		INITIAL_CONTACT is out of the scope
27	1270	An endpoint should suspect that the other endpoint has failed based on routing information and initiate a request to see whether the other endpoint is alive.		Not support		Explanation
27	1273	To check whether the other side is alive, IKE specifies an empty INFORMATIONAL message that (like all IKE requests) requires an acknowledgement (note that within the context of an IKE SA, an "empty" message consists of an IKE header followed by an Encrypted payload that contains no payloads).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.6 EN.R.1.1.3.4 EN.R.1.1.3.5 SGW.I.1.1.3.6 SGW.R.1.1.3.4 SGW.R.1.1.3.5
27	1277	If a cryptographically protected (fresh, i.e., not retransmitted) message has been received from the other side recently, unprotected Notify messages MAY be ignored. Implementations MUST limit the rate at which they take actions based on unprotected messages.	MAY MUST	Not support		Not need to test
27	1283	The number of retries and length of timeouts are not covered in this specification because they do not affect interoperability. It is suggested that messages be retransmitted at least a dozen times over a period of at least several minutes before giving up on an SA, but different environments may require different rules.		Not support		Explanation
27	1287	To be a good network citizen, retransmission times **MUST** increase exponentially to avoid flooding the network and making an existing congestion situation worse.	MUST	Not support		Difficult to test
27	1290	If there has only been outgoing traffic on all of the SAs associated with an IKE SA, it is essential to confirm liveness of the other		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		endpoint to avoid black holes.				
27	1292	If no cryptographically protected messages have been received on an IKE SA or any of its Child SAs recently, the system needs to perform a liveness check in order to prevent sending messages to a dead peer.(This is sometimes called "dead peer detection" or "DPD", although it is really detecting live peers, not dead ones.)		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.3.11 SGW.I.1.1.3.11
28	1297	Receipt of a fresh cryptographically protected message on an IKE SA or any of its Child SAs ensures liveness of the IKE SA and all of its Child SAs.		Not support		Explanation
28	1299	Note that this places requirements on the failure modes of an IKE endpoint.		Not support		Difficult to test
28	1301	An implementation needs to stop sending over any SA if some failure prevents it from receiving on all of the associated SAs.		Not support		Difficult to test
28	1302	If a system creates Child SAs that can fail independently from one another without the associated IKE SA being able to send a delete message, then the system **MUST** negotiate such Child SAs using separate IKE SAs.	MUST	Not support		Difficult to test
28	1308	There is a DoS attack on the initiator of an IKE SA that can be avoided if the initiator takes the proper care. Since the first two messages of an SA setup are not cryptographically protected, an attacker could respond to the initiator's message before the genuine responder and poison the connection setup attempt. To prevent this, the initiator **MAY** be willing to accept multiple responses to its first message, treat each as potentially legitimate, respond to it, and then discard all the invalid half-open connections when it receives a valid cryptographically protected response to any one of its requests. Once a cryptographically valid response is received, all subsequent responses should be ignored whether or not they are cryptographically valid.	MAY	Not support		Not need to test
28	1321	Note that with these rules, there is no reason to negotiate and agree upon an SA lifetime. If IKE presumes the partner is dead, based on repeated lack of acknowledgement to an IKE message, then the IKE SA and all Child SAs set up through that IKE SA are deleted.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
28	1326	An IKE endpoint may at any time delete inactive Child SAs to recover resources used to hold their state.		Not support		Explanation
28	1327	If an IKE endpoint chooses to delete Child SAs, it **MUST** send Delete payloads to the other end notifying it of the deletion.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.8 EN.R.1.1.3.9 EN.R.1.2.3.1 EN.R.1.2.3.2 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.8 SGW.R.1.1.3.9 SGW.R.1.2.3.1 SGW.R.1.2.3.2
28	1329	It **MAY** similarly time out the IKE SA.	MAY	Not support		Not need to test
28	1330	Closing the IKE SA implicitly closes all associated Child SAs. In this case, an IKE endpoint **SHOULD** send a Delete payload indicating that it has closed the IKE SA unless the other endpoint is no longer responding.	SHOULD	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.R.1.1.3.6 EN.R.1.1.3.7 SGW.R.1.1.3.6 SGW.R.1.1.3.7
28	1335	2.5. Version Numbers and Forward Compatibility				
28	1337	This document describes version 2.0 of IKE, meaning the major version number is 2 and the minor version number is 0. This document is a replacement for [IKEV2]. It is likely that some implementations will want to support version 1.0 and version 2.0, and in the future, other versions.		Not support		Explanation
29	1343	The major version number should be incremented only if the packet formats or required actions have changed so dramatically that an older version node would not be able to interoperate with a newer version node if it simply ignored the fields it did not understand and took the actions specified in the older specification.		Not support		Explanation
29	1347	The minor version number indicates new capabilities, and MUST be ignored by a node with a smaller minor version number, but used for informational purposes by the node with the larger minor	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.1 SGW.R.1.1.4.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		version number. For example, it might indicate the ability to process a newly defined Notify message type. The node with the larger minor version number would simply note that its correspondent would not be able to understand that message and therefore would not send it.				
29	1356	If an endpoint receives a message with a higher major version number, it **MUST** drop the message and **SHOULD** send an unauthenticated Notify message of type INVALID_MAJOR_VERSION containing the highest (closest) version number it supports.	MUST SHOULD	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.2 SGW.R.1.1.4.2
29	1359	If an endpoint supports major version n, and major version m, it **MUST** support all versions between n and m.	MUST	Not support		Explanation
29	1361	If it receives a message with a major version that it supports, it **MUST** respond with that version number.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.1 SGW.R.1.1.1.1
29	1362	In order to prevent two nodes from being tricked into corresponding with a lower major version number than the maximum that they both support, IKE has a flag that indicates that the node is capable of speaking a higher major version number.		Not support		Explanation
29	1368	Thus, the major version number in the IKE header indicates the version number of the message, not the highest version number that the transmitter supports. If the initiator is capable of speaking versions n, n+1, and n+2, and the responder is capable of speaking versions n and n+1, then they will negotiate speaking n+1, where the initiator will set a flag indicating its ability to speak a higher version.		Not support		Explanation
29	1374	If they mistakenly (perhaps through an active attacker sending error messages) negotiate to version n, then both will notice that the other side can support a higher version number, and they **MUST** break the connection and reconnect using version n+1.	MUST	Not support		V-bit in IKE header is always "0" at IKEv2 tests.
29	1379	Note that IKEv1 does not follow these rules, because there is no way in v1 of noting that you are capable of speaking a higher version number. So an active attacker can trick two v2-capable nodes into speaking v1. When a v2-capable node negotiates down to v1, it should note that fact in its logs.		Not support		Internal process

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
29	1385	Also, for forward compatibility, all fields marked RESERVED **MUST** be set to zero by an implementation running version 2.0,	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
29	1386	and their content **MUST** be ignored by an implementation running version 2.0 ("Be conservative in what you send and liberal in what you receive" [IP]). In this way, future versions of the protocol can use those fields in a way that is guaranteed to be ignored by implementations that do not understand them.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.1 EN.I.1.1.11.2 EN.I.1.2.7.1 EN.R.1.1.11.1 EN.R.1.1.11.2 EN.R.1.2.9.1 EN.R.1.3.3.1 SGW.I.1.1.11.1 SGW.I.1.1.11.2 SGW.I.1.2.7.1 SGW.R.1.1.11.1 SGW.R.1.1.11.2 SGW.R.1.2.9.1 SGW.R.1.3.3.1
	1391	Similarly, payload types that are not defined are reserved for future use; implementations of a version where they are undefined **MUST** skip over those payloads and ignore their contents.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.4.1 EN.I.1.1.4.2 EN.R.1.1.4.3 EN.R.1.1.4.4 SGW.I.1.1.4.1 SGW.I.1.1.4.2 SGW.R.1.1.4.3 SGW.R.1.1.4.4
	1395	IKEv2 adds a "critical" flag to each payload header for further flexibility for forward compatibility.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.4.1 EN.I.1.1.4.2 EN.R.1.1.4.3 EN.R.1.1.4.4 SGW.I.1.1.4.1 SGW.I.1.1.4.2 SGW.R.1.1.4.3 SGW.R.1.1.4.4
	1396	If the critical flag is set and the payload type is unrecognized, the message **MUST** be rejected and the response to the IKE request containing that payload **MUST** include a Notify payload UNSUPPORTED_CRITICAL_PAYLOAD, indicating an	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.4.2 EN.R.1.1.4.4 SGW.I.1.1.4.2 SGW.R.1.1.4.4

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		unsupported critical payload was included.				
	1400	In that Notify payload, the notification data contains the one-octet payload type.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.4 SGW.R.1.1.4.4
	1401	If the critical flag is not set and the payload type is unsupported, that payload **MUST** be ignored.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.4.1 EN.R.1.1.4.3 SGW.I.1.1.4.1 SGW.R.1.1.4.3
	1403	Payloads sent in IKE response messages **MUST NOT** have the critical flag set. Note that the critical flag applies only to the payload type, not the contents.	MUST NOT	BASIC	Both	EN.R.1.1.1.1 EN.R.1.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2
	1405	If the payload type is recognized, but the payload contains something that is not (such as an unknown transform inside an SA payload, or an unknown Notify Message Type inside a Notify payload), the critical flag is ignored.		Not Support		behavior after ignored is not prescribed
	1410	Although new payload types may be added in the future and may appear interleaved with the fields defined in this specification, implementations **SHOULD** send the payloads defined in this specification in the order shown in the figures in Sections 1 and 2;	SHOULD	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	1414	implementations **MUST NOT** reject as invalid a message with those payloads in any other order.	MUST NOT	BASIC	EN(responder) SGW(responder)	[Changed] In RFC 4306, "implementations **SHOULD** reject as invalid a message with those payloads in any other order."
	1417	2.6. IKE SA SPIs and Cookies				
	1419	The initial two eight-octet fields in the header, called the "IKE SPIs", are used as a connection identifier at the beginning of IKE packets.		Not support		Explanation
	1421	Each endpoint chooses one of the two SPIs and **MUST** choose them so as to be unique identifiers of an IKE SA. An SPI value of zero is special: it indicates that the remote SPI value is not yet known by the sender.	MUST	Not support		Internal process
	1426	Incoming IKE packets are mapped to an IKE SA only using the		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		packet's SPI, not using (for example) the source IP address of the packet.				
	1429	Unlike ESP and AH where only the recipient's SPI appears in the header of a message, in IKE the sender's SPI is also sent in every message. Since the SPI chosen by the original initiator of the IKE SA is always sent first, an endpoint with multiple IKE SAs open that wants to find the appropriate IKE SA using the SPI it assigned must look at the Initiator flag in the header to determine whether it assigned the first or the second eight octets.		Not support		Explanation
	1437	In the first message of an initial IKE exchange, the initiator will not know the responder's SPI value and will therefore set that field to zero.		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 SGW.I.1.1.1.1
	1439	When the IKE_SA_INIT exchange does not result in the creation of an IKE SA due to INVALID_KE_PAYLOAD, NO_PROPOSAL_CHOSEN, or COOKIE (see Section 2.6), the responder's SPI will be zero also in the response message.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.8 SGW.R.1.1.6.8
	1442	However, if the responder sends a non-zero responder SPI, the initiator should not reject the response for only that reason.		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.6.11 SGW.I.1.1.6.11
	1446	Two expected attacks against IKE are state and CPU exhaustion, where the target is flooded with session initiation requests from forged IP addresses. These attacks can be made less effective if a responder uses minimal CPU and commits no state to an SA until it knows the initiator can receive packets at the address from which it claims to be sending them.		Not support		Explanation
	1453	When a responder detects a large number of half-open IKE SAs, it SHOULD reply to IKE_SA_INIT requests with a response containing the COOKIE notification.	SHOULD	Not support		test condition is ambiguous (a large number of half-open IKE SAs)
	1455	The data associated with this notification **MUST** be between 1 and 64 octets in length (inclusive), and its generation is described later in this section.	MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1
	1457	If the IKE_SA_INIT response includes the COOKIE notification, the initiator **MUST** then retry the IKE_SA_INIT request, and	MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		include the COOKIE notification containing the received data as the first payload, and all other payloads unchanged. The initial exchange will then be as follows:				
	1463	Initiator Responder --- HDR(A,0), SAi1, KEi, Ni --> <-- HDR(A,0), N(COOKIE) HDR(A,0), N(COOKIE), SAi1, KEi, Ni --> <-- HDR(A,B), SAr1, KEr, Nr, [CERTREQ] HDR(A,B), SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} --> <-- HDR(A,B), SK {IDr, [CERT,] AUTH, SAr2, TSi, TSr}		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1
	1477	The first two messages do not affect any initiator or responder state except for communicating the cookie.		Not support		Explanation
	1478	In particular, the message sequence numbers in the first four messages will all be zero		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1
	1479	and the message sequence numbers in the last two messages will be one.		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1
	1480	A' is the SPI assigned by the initiator, while 'B' is the SPI assigned by the responder.		Not support		Explanation
	1484	An IKE implementation can implement its responder cookie generation in such a way as to not require any saved state to recognize its valid cookie when the second IKE_SA_INIT message arrives. The exact algorithms and syntax used to generate cookies do not affect interoperability and hence are not specified here. The following is an example of how an endpoint could use cookies to implement limited DoS protection.		Not support		Internal process
	1492	A good way to do this is to set the responder cookie to be:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1494	Cookie = <VersionIDofSecret> | Hash(Ni | IPi | SPIi | <secret>)		Not support		Explanation
	1496	where <secret> is a randomly generated secret known only to the responder and periodically changed and | indicates concatenation. <VersionIDofSecret> should be changed whenever <secret> is regenerated.		Not support		Explanation
	1499	The cookie can be recomputed when the IKE_SA_INIT arrives the second time and compared to the cookie in the received message. If it matches, the responder knows that the cookie was generated since the last change to <secret> and that IPi must be the same as the source address it saw the first time. Incorporating SPIi into the calculation ensures that if multiple IKE SAs are being set up in parallel they will all get different cookies (assuming the initiator chooses unique SPIi's). Incorporating Ni in the hash ensures that an attacker who sees only message 2 can't successfully forge a message 3.		Not support		Explanation
	1508	Also, incorporating SPIi in the hash prevents an attacker from fetching one cookie from the other end, and then initiating many IKE_SA_INIT exchanges all with different initiator SPIs (and perhaps port numbers) so that the responder thinks that there are a lot of machines behind one NAT box that are all trying to connect.		Not support		Explanation
	1515	If a new value for <secret> is chosen while there are connections in the process of being initialized, an IKE_SA_INIT might be returned with other than the current <VersionIDofSecret>. The responder in that case **MAY** reject the message by sending another response with a new cookie or it **MAY** keep the old value of <secret> around for a short time and accept cookies computed from either one. The responder should not accept cookies indefinitely after <secret> is changed, since that would defeat part of the DoS protection. The responder should change the value of <secret> frequently, especially if under attack.	MAY MAY	Not support		(ref.) RFC4718 2.5
	1526	When one party receives an IKE_SA_INIT request containing a cookie whose contents do not match the value expected, that party **MUST** ignore the cookie and process the message as if no cookie	MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.2 SGW.1.1.5.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		had been included; usually this means sending a response containing a new cookie. The initiator should limit the number of cookie exchanges it tries before giving up, possibly using exponential back-off.				
	1531	An attacker can forge multiple cookie responses to the initiator's IKE_SA_INIT message, and each of those forged cookie replies will cause two packets to be sent: one packet from the initiator to the responder (which will reject those cookies), and one response from responder to initiator that includes the correct cookie.		Not support		Explanation
	1538	A note on terminology: the term "cookies" originates with Karn and Simpson [PHOTURIS] in Photuris, an early proposal for key management with IPsec, and it has persisted. The Internet Security Association and Key Management Protocol (ISAKMP) [ISAKMP] fixed message header includes two eight-octet fields called "cookies", and that syntax is used by both IKEv1 and IKEv2, although in IKEv2 they are referred to as the "IKE SPI" and there is a new separate field in a Notify payload holding the cookie.		Not support		Explanation
	1547	2.6.1. Interaction of COOKIE and INVALID_KE_PAYLOAD				
	1549	There are two common reasons why the initiator may have to retry the IKE_SA_INIT exchange: the responder requests a cookie or wants a different Diffie-Hellman group than was included in the KEi payload.		Not support		Explanation
	1552	If the initiator receives a cookie from the responder, the initiator needs to decide whether or not to include the cookie in only the next retry of the IKE_SA_INIT request, or in all subsequent retries as well.		Not support		Explanation
	1557	If the initiator includes the cookie only in the next retry, one additional round trip may be needed in some cases.		Not support		Explanation
	1558	An additional round trip is needed also if the initiator includes the cookie in all retries, but the responder does not support this. For instance, if the responder includes the KEi payloads in cookie calculation, it will reject the request by sending a new cookie.		Not support		Explanation
	1564	If both peers support including the cookie in all retries, a slightly shorter exchange can happen.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1567	Initiator Responder --- HDR(A,0), SAi1, KEi, Ni --> <-- HDR(A,0), N(COOKIE) HDR(A,0), N(COOKIE), SAi1, KEi, Ni --> <-- HDR(A,0), N(INVALID_KE_PAYLOAD) HDR(A,0), N(COOKIE), SAi1, KEi', Ni --> <-- HDR(A,B), SAr1, KEr, Nr		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.2 SGW.I.1.1.5.2
	1576	Implementations **SHOULD** support this shorter exchange, but **MUST NOT** fail if other implementations do not support this shorter exchange.	SHOULD MUST NOT	Not support		Explanation
	1579	2.7. Cryptographic Algorithm Negotiation				
	1581	The payload type known as "SA" indicates a proposal for a set of choices of IPsec protocols (IKE, ESP, or AH) for the SA as well as cryptographic algorithms associated with each protocol.		Not support		Explanation
	1585	An SA payload consists of one or more proposals. Each proposal includes one protocol. Each protocol contains one or more transforms -- each specifying a cryptographic algorithm. Each transform contains zero or more attributes (attributes are needed only if the transform ID does not completely specify the cryptographic algorithm).		Not support		Explanation
	1592	This hierarchical structure was designed to efficiently encode proposals for cryptographic suites when the number of supported suites is large because multiple values are acceptable for multiple transforms. The responder **MUST** choose a single suite, which may be any subset of the SA proposal following the rules below.	MUST	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1598	Each proposal contains one protocol. If a proposal is accepted, the SA response **MUST** contain the same protocol. The responder **MUST** accept a single proposal or reject them all and return an error. The error is given in a notification of type NO_PROPOSAL_CHOSEN.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.I.1.1.6.4 EN.I.1.1.6.6 EN.R.1.1.6.1 EN.R.1.1.6.2 EN.R.1.1.6.4 EN.R.1.1.6.6 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.I.1.1.6.4 SGW.I.1.1.6.6 SGW.R.1.1.6.1 SGW.R.1.1.6.2 SGW.R.1.1.6.4 SGW.R.1.1.6.6
	1603	Each IPsec protocol proposal contains one or more transforms. Each transform contains a transform type. The accepted cryptographic suite **MUST** contain exactly one transform of each type included in the proposal. For example: if an ESP proposal includes transforms ENCR_3DES, ENCR_AES w/keysize 128, ENCR_AES w/keysize 256, AUTH_HMAC_MD5, and AUTH_HMAC_SHA, the accepted suite **MUST** contain one of the ENCR_ transforms and one of the AUTH_ transforms. Thus, six combinations are acceptable.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.I.1.1.6.3 EN.I.1.1.6.5 EN.R.1.1.6.1 EN.R.1.1.6.2 EN.R.1.1.6.3 EN.R.1.1.6.5 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.I.1.1.6.3 SGW.I.1.1.6.5 SGW.R.1.1.6.1 SGW.R.1.1.6.2 SGW.R.1.1.6.3 SGW.R.1.1.6.5
	1612	If an initiator proposes both normal ciphers with integrity protection as well as combined-mode ciphers, then two proposals are needed. One of the proposals includes the normal ciphers		Not support		combined-mode is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		with the integrity algorithms for them, and the other proposal includes all the combined-mode ciphers without the integrity algorithms (because combined-mode ciphers are not allowed to have any integrity algorithm other than "none").				
	1620	2.8. Rekeying				
	1622	IKE, ESP, and AH security associations use secret keys that should be used only for a limited amount of time and to protect a limited amount of data. This limits the lifetime of the entire security association.		Not support		Internal process
	1625	When the lifetime of a security association expires, the security association **MUST NOT** be used.	MUST NOT	BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.3.3 EN.I.1.2.4.3 SGW.I.1.2.3.3 SGW.I.1.2.4.3
	1626	If there is demand, new security associations **MAY** be established. Reestablishment of security associations to take the place of ones that expire is referred to as "rekeying".	MAY	Not support		Explanation
	1631	To allow for minimal IPsec implementations, the ability to rekey SAs without restarting the entire IKE SA is optional. An implementation **MAY** refuse all CREATE_CHILD_SA requests within an IKE SA.	MAY	Not support		Explanation
	1633	If an SA has expired or is about to expire and rekeying attempts using the mechanisms described here fail, an implementation **MUST** close the IKE SA and any associated Child SAs and then **MAY** start new ones.	MUST MAY	BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.3.6 SGW.I.1.2.3.6
	1637	Implementations may wish to support in-place rekeying of SAs, since doing so offers better performance and is likely to reduce the number of packets lost during the transition.		Not support		Explanation
	1641	To rekey a Child SA within an existing IKE SA, create a new, equivalent SA (see Section 2.17 below), and when the new one is established, delete the old one.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.1 EN.I.1.2.4.1 EN.R.1.2.5.1 EN.R.1.2.6.4 SGW.I.1.2.3.1 SGW.I.1.2.4.1 SGW.R.1.2.5.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.2.6.4
	1643	Note that, when rekeying, the new Child SA **SHOULD NOT** have different traffic selectors and algorithms than the old one.	SHOULD NOT	BASIC	Both	EN.I.1.2.3.4 EN.I.1.2.3.5 EN.I.1.2.4.4 EN.I.1.2.4.5 EN.I.1.2.4.7 EN.R.1.2.5.3 EN.R.1.2.5.4 EN.R.1.2.6.5 EN.R.1.2.6.6 EN.R.1.2.6.7 SGW.I.1.2.3.4 SGW.I.1.2.3.5 SGW.I.1.2.4.4 SGW.I.1.2.4.5 SGW.I.1.2.4.7 SGW.R.1.2.5.3 SGW.R.1.2.5.4 SGW.R.1.2.6.5 SGW.R.1.2.6.6 SGW.R.1.2.6.7
	1647	To rekey an IKE SA, establish a new equivalent IKE SA (see Section 2.18 below) with the peer to whom the old IKE SA is shared using a CREATE_CHILD_SA within the existing IKE SA. An IKE SA so created inherits all of the original IKE SA's Child SAs,		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.4.1 EN.R.1.2.6.4 SGW.I.1.2.4.1 SGW.R.1.2.6.4
	1650	and the new IKE SA is used for all control messages needed to maintain those Child SAs. After the new equivalent IKE SA is created, the initiator deletes the old IKE SA, and the Delete payload to delete itself **MUST** be the last request sent over the old	MUST	Not Support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		IKE SA.				
	1656	SAs should be rekeyed proactively, i.e., the new SA should be established before the old one expires and becomes unusable. Enough time should elapse between the time the new SA is established and the old one becomes unusable so that traffic can be switched over to the new SA.		Not support		Difficult to test
	1662	A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary.		Not support		Explanation
	1665	If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying.		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.1.1 SGW.I.1.2.1.1
	1667	If an SA has been inactive for a long time and if an endpoint would not initiate the SA in the absence of traffic, the endpoint **MAY** choose to close the SA instead of rekeying it when its lifetime expires. It can also do so if there has been no traffic since the last time the SA was rekeyed.	MAY	Not support		Not need to test
	1673	Note that IKEv2 deliberately allows parallel SAs with the same traffic selectors between common endpoints. One of the purposes of this is to support traffic quality of service (QoS) differences among the SAs (see [DIFFSERVFIELD], [DIFFSERVARCH], and section 4.1 of [DIFFTUNNEL]). Hence unlike IKEv1, the combination of the endpoints and the traffic selectors may not uniquely identify an SA between those endpoints, so the IKEv1 rekeying heuristic of deleting SAs on the basis of duplicate traffic selectors **SHOULD NOT** be used.	SHOULD NOT	Not support		Explanation
	1682	There are timing windows -- particularly in the presence of lost packets -- where endpoints may not agree on the state of an SA. The responder to a CREATE_CHILD_SA **MUST** be prepared to accept messages on an SA before sending its response to the creation request, so there is no ambiguity for the initiator. The initiator **MAY** begin sending on an SA as soon as it processes the response. The initiator, however, cannot receive on a newly	MUST MAY	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		created SA until it receives and processes the response to its CREATE_CHILD_SA request. How, then, is the responder to know when it is OK to send on the newly created SA?				
	1692	From a technical correctness and interoperability perspective, the responder **MAY** begin sending on an SA as soon as it sends its response to the CREATE_CHILD_SA request. In some situations, however, this could result in packets unnecessarily being dropped, so an implementation **MAY** defer such sending.	MAY MAY	Not support		Explanation
	1698	The responder can be assured that the initiator is prepared to receive messages on an SA if either (1) it has received a cryptographically valid message on the other half of the SA pair, or (2) the new SA rekeys an existing SA and it receives an IKE request to close the replaced SA. When rekeying an SA, the responder continues to send traffic on the old SA until one of those events occurs. When establishing a new SA, the responder **MAY** defer sending messages on a new SA until either it receives one or a timeout has occurred.	MAY	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.8 EN.I.1.2.4.6 EN.R.1.2.5.6 EN.R.1.2.6.3 SGW.I.1.2.3.8 SGW.I.1.2.4.6 SGW.R.1.2.5.6 SGW.R.1.2.6.3
	1706	If an initiator receives a message on an SA for which it has not received a response to its CREATE_CHILD_SA request, it interprets that as a likely packet loss and retransmits the CREATE_CHILD_SA request. An initiator **MAY** send a dummy ESP message on a newly created ESP SA if it has no messages queued in order to assure the responder that the initiator is ready to receive messages.	MAY	Not support		Difficult to test
	1713	2.8.1. Simultaneous Child SA rekeying				
	1715	If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests **SHOULD** be jittered (delayed by a random amount of time after the need for rekeying is noticed).	SHOULD	Not support		Explanation
	1721	This form of rekeying may temporarily result in multiple similar SAs between the same pairs of nodes. When there are two SAs eligible to receive packets, a node MUST accept incoming packets through either SA.	MUST SHOULD	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		If redundant SAs are created though such a collision, the SA created with the lowest of the four nonces used in the two exchanges SHOULD be closed by the endpoint that created it.				
	1726	"Lowest" means an octet-by-octet comparison (instead of, for instance, comparing the nonces as large integers). In other words, start by comparing the first octet; if they're equal, move to the next octet, and so on. If you reach the end of one nonce, that nonce is the lower one.		Not support		Explanation
	1730	The node that initiated the surviving rekeyed SA should delete the replaced SA after the new one is established.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.1 EN.I.1.2.4.1 EN.R.1.2.5.1 EN.R.1.2.6.4 SGW.I.1.2.3.1 SGW.I.1.2.4.1 SGW.R.1.2.5.1 SGW.R.1.2.6.4
	1734	The following is an explanation on the impact this has on implementations. Assume that hosts A and B have an existing Child SA pair with SPIs (SPIa1,SPIb1), and both start rekeying it at the same time:		Not support		Explanation
	1739	Host A Host B -- send req1: N(REKEY_SA,SPIa1), SA(..,SPIa2,..),Ni1,.. --> <-- send req2: N(REKEY_SA,SPIb1), SA(..,SPIb2,..),Ni2 recv req2 <--		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.3 SGW.I.1.2.6.3
	1747	At this point, A knows there is a simultaneous rekeying happening. However, it cannot yet know which of the exchanges will have the lowest nonce, so it will just note the situation and respond as usual.		Not support		Explanation
	1752	send resp2: SA(..,SPIa3,..), Nr1,.. -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.3 SGW.I.1.2.6.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		--> recv req1				
	1756	Now B also knows that simultaneous rekeying is going on. It responds as usual.		Not support		Explanation
	1759	<-- send resp1: SA(..,SPIb3,..), Nr2,.. recv resp1 <-- --> recv resp2		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.3 SGW.I.1.2.6.3
	1764	At this point, there are three Child SA pairs between A and B (the old one and two new ones). A and B can now compare the nonces. Suppose that the lowest nonce was Nr1 in message resp2; in this case, B (the sender of req2) deletes the redundant new SA, and A (the node that initiated the surviving rekeyed SA), deletes the old one.		Not support		Explanation
	1770	send req3: D(SPIa1) --> <-- send req4: D(SPIb2) --> recv req3 <-- send resp3: D(SPIb1) recv req4 <-- send resp4: D(SPIa3) -->		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.3 SGW.I.1.2.6.3
	1777	The rekeying is now finished.		Not support		Explanation
	1779	However, there is a second possible sequence of events that can happen if some packets are lost in the network, resulting in retransmissions. The rekeying begins as usual, but A's first packet (req1) is lost.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1784	Host A Host B --- send req1: N(REKEY_SA,SPIa1), SA(..,SPIa2,..), Ni1,.. --> (lost) <-- send req2: N(REKEY_SA,SPIb1), SA(..,SPIb2,..),Ni2 recv req2 <-- send resp2: SA(..,SPIa3,..), Nr1,.. --> --> recv resp2 <-- send req3: D(SPIb1) recv req3 <-- send resp3: D(SPIa1) --> --> recv resp3		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.4 SGW.I.1.2.6.4
	1800	From B's point of view, the rekeying is now completed, and since it has not yet received A's req1, it does not even know that there was simultaneous rekeying. However, A will continue retransmitting the message, and eventually it will reach B.		Not support		Explanation
	1805	resend req1 --> --> recv req1		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.4 SGW.I.1.2.6.4
	1808	To B, it looks like A is trying to rekey an SA that no longer exists; thus, B responds to the request with something non-fatal such as CHILD_SA_NOT_FOUND.		Not support		Explanation
	1812	<-- send resp1: N(CHILD_SA_NOT_FOUND) recv resp1 <--		Not support		Difficult to test
	1815	When A receives this error, it already knows there was simultaneous rekeying, so it can ignore the error message.		Not support		Explanation
	1818	2.8.2. Simultaneous IKE SA Rekeying				
	1820	Probably the most complex case occurs when both peers try to rekey the IKE_SA at the same time. Basically, the text in Section 2.8 applies to this case as well; however, it is important to ensure that the Child SAs are inherited by the correct IKE_SA.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1825	The case where both endpoints notice the simultaneous rekeying works the same way as with Child SAs. After the CREATE_CHILD_SA exchanges, three IKE SAs exist between A and B: the old IKE SA and two new IKE SAs. The new IKE SA containing the lowest nonce **SHOULD** be deleted by the node that created it, and the other surviving new IKE SA **MUST** inherit all the Child SAs.	SHOULD MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.5 SGW.I.1.2.6.5
	1832	In addition to normal simultaneous rekeying cases, there is a special case where one peer finishes its rekey before it even notices that other peer is doing a rekey. If only one peer detects a simultaneous rekey, redundant SAs are not created. In this case, when the peer that did not notice the simultaneous rekey gets the request to rekey the IKE SA that it has already successfully rekeyed, it **SHOULD** return TEMPORARY_FAILURE because it is an IKE SA that it is currently trying to close (whether or not it has already sent the delete notification for the SA). If the peer that did notice the simultaneous rekey gets the delete request from the other peer for the old IKE SA, it knows that the other peer did not detect the simultaneous rekey, and the first peer can forget its own rekey attempt.	SHOULD	Not support		untestable (it is difficult to distinguish simultaneous rekeying from new rekeying)
	1845	Host A Host B --- send req1: SA(..,SPIa1,..),Ni1,.. --> <-- send req2: SA(..,SPIb1,..),Ni2,.. --> recv req1 <-- send resp1: SA(..,SPIb2,..),Nr2,.. recv resp1 <-- send req3: D() --> --> recv req3		Not support		Explanation
	1856	At this point, host B sees a request to close the IKE_SA. There's not much more to do than to reply as usual. However, at this point host B should stop retransmitting req2, since once host A receives resp3, it will delete all the state associated with the old IKE_SA and will not be able to reply to it.		BASIC	EN(initiator) SGW(initiator)	EN.I.1.2.6.6 SGW.I.1.2.6.6

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	1862	<-- send resp3: ()		Not support		Explanation
	1864	The TEMPORARY_FAILURE notification was not included in RFC 4306, and support of the TEMPORARY_FAILURE notification is not negotiated. Thus, older peers that implement RFC 4306 but not this document may receive these notifications. In that case, they will treat it the same as any other unknown error notification, and will stop the exchange. Because the other peer has already rekeyed the exchange, doing so does not have any ill effects.		Not Support		implementation with RFC4306 is out of the scope
	1872	2.8.3. Rekeying the IKE SA Versus Reauthentication				
	1874	Rekeying the IKE SA and reauthentication are different concepts in IKEv2. Rekeying the IKE SA establishes new keys for the IKE SA and resets the Message ID counters, but it does not authenticate the parties again (no AUTH or EAP payloads are involved).		Not support		Explanation
	1879	Although rekeying the IKE SA may be important in some environments, reauthentication (the verification that the parties still have access to the long-term credentials) is often more important.		Not support		Explanation
	1883	IKEv2 does not have any special support for reauthentication. Reauthentication is done by creating a new IKE SA from scratch (using IKE_SA_INIT/IKE_AUTH exchanges, without any REKEY_SA notify payloads), creating new Child SAs within the new IKE SA (without REKEY_SA Notify payloads), and finally deleting the old IKE SA (which deletes the old Child SAs as well).		Not support		Explanation
	1890	This means that reauthentication also establishes new keys for the IKE SA and Child SAs. Therefore, while rekeying can be performed more often than reauthentication, the situation where "authentication lifetime" is shorter than "key lifetime" does not make sense.		Not support		Explanation
	1895	While creation of a new IKE SA can be initiated by either party (initiator or responder in the original IKE SA), the use of EAP and/or configuration payloads means in practice that reauthentication has to be initiated by the same party as the		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		original IKE SA. IKEv2 does not currently allow the responder to request reauthentication in this case; however, there are extensions that add this functionality such as [REAUTH].				
	1903	2.9. Traffic Selector Negotiation				
	1905	When an RFC4301-compliant IPsec subsystem receives an IP packet that matches a "protect" selector in its Security Policy Database (SPD), the subsystem protects that packet with IPsec. When no SA exists yet, it is the task of IKE to create it. Maintenance of a system's SPD is outside the scope of IKE, although some implementations might update their SPD in connection with the running of IKE (for an example scenario, see Section 1.1.3).		Not support		Internal process
	1913	Traffic Selector (TS) payloads allow endpoints to communicate some of the information from their SPD to their peers. These must be communicated to IKE from the SPD (for example, the PF_KEY API [PFKEY] uses the SADB_ACQUIRE message). TS payloads specify the selection criteria for packets that will be forwarded over the newly set up SA. This can serve as a consistency check in some scenarios to assure that the SPDs are consistent. In others, it guides the dynamic update of the SPD.		Not support		Explanation
	1922	Two TS payloads appear in each of the messages in the exchange that creates a Child SA pair. Each TS payload contains one or more Traffic Selectors. Each traffic selector consists of an address range (IPv4 or IPv6), a port range, and an IP protocol ID.		Not support		Explanation
	1927	The first of the two TS payloads is known as TSi (Traffic Selector-initiator). The second is known as TSr (Traffic Selector-responder). TSi specifies the source address of traffic forwarded from (or the destination address of traffic forwarded to) the initiator of the Child SA pair. TSr specifies the destination address of the traffic forwarded to (or the source address of the traffic forwarded from) the responder of the Child SA pair.		Not support		Explanation
	1931	For example, if the original initiator requests the creation of a Child SA pair, and wishes to tunnel all traffic from subnet 198.51.100.* on the initiator's side to subnet 192.0.2.* on the responder's side, the initiator would include a single traffic selector		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		in each TS payload. TSi would specify the address range (198.51.100.0 - 198.51.100.255) and TSr would specify the address range (192.0.2.0 - 192.0.2.255). Assuming that proposal was acceptable to the responder, it would send identical TS payloads back.				
	1943	IKEv2 allows the responder to choose a subset of the traffic proposed by the initiator. This could happen when the configurations of the two endpoints are being updated but only one end has received the new information. Since the two endpoints may be configured by different people, the incompatibility may persist for an extended period even in the absence of errors. It also allows for intentionally different configurations, as when one end is configured to tunnel all addresses and depends on the other end to have the up-to-date list.		Not support		Explanation
	1952	When the responder chooses a subset of the traffic proposed by the initiator, it narrows the traffic selectors to some subset of the initiator's proposal (provided the set does not become the null set).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.7.1 EN.R.1.1.7.1 SGW.I.1.1.7.1 SGW.R.1.1.7.1
	1955	If the type of traffic selector proposed is unknown, the responder ignores that traffic selector, so that the unknown type is not returned in the narrowed set.		Not support		multiple TS is out of the scope
	1959	To enable the responder to choose the appropriate range in this case, if the initiator has requested the SA due to a data packet, the initiator **SHOULD** include as the first traffic selector in each of TSi and TSr a very specific traffic selector including the addresses in the packet triggering the request.	SHOULD	Not support		Internal process
	1963	In the example, the initiator would include in TSi two traffic selectors: the first containing the address range (198.51.100.43 - 198.51.100.43) and the source port and IP protocol from the packet and the second containing (198.51.100.0 - 198.51.100.255) with all ports and IP protocols. The initiator would similarly include two traffic selectors in TSr.		Not support		Explanation
	1968	If the initiator creates the Child SA pair not in response to an arriving packet, but rather, say, upon startup, then there may be no specific addresses the initiator prefers for the initial tunnel over		Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		any other. In that case, the first values in TSi and TSr can be ranges rather than specific values.				
	1975	The responder performs the narrowing as follows:		Not support		Explanation
	1977	o If the responder's policy does not allow it to accept any part of the proposed traffic selectors, it responds with a TS_UNACCEPTABLE Notify message.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.7.2 SGW.R.1.1.7.2
	1981	o If the responder's policy allows the entire set of traffic covered by TSi and TSr, no narrowing is necessary, and the responder can return the same TSi and TSr values.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.2 SGW.R.1.1.1.2
	1985	o If the responder's policy allows it to accept the first selector of TSi and TSr, then the responder **MUST** narrow the traffic selectors to a subset that includes the initiator's first choices. In this example above, the responder might respond with TSi being (198.51.100.43 - 198.51.100.43) with all ports and IP protocols.	MUST	Not support		Explanation
	1991	o If the responder's policy does not allow it to accept the first selector of TSi and TSr, the responder narrows to an acceptable subset of TSi and TSr.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.7.1 SGW.R.1.1.7.1
	1995	When narrowing is done, there may be several subsets that are acceptable but their union is not. In this case, the responder arbitrarily chooses one of them, and **MAY** include an ADDITIONAL_TS_POSSIBLE notification in the response. The ADDITIONAL_TS_POSSIBLE notification asserts that the responder narrowed the proposed traffic selectors but that other traffic selectors would also have been acceptable, though only in a separate SA. There is no data associated with this Notify type.	MAY	Not support		Explanation
	2002	This case will occur only when the initiator and responder are configured differently from one another.		Not support		Explanation
	2004	If the initiator and responder agree on the granularity of tunnels, the initiator will never request a tunnel wider than the responder will accept.		Not Support		ADDITIONAL_TS_POSSIBLE is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2008	It is possible for the responder's policy to contain multiple smaller ranges, all encompassed by the initiator's traffic selector, and with the responder's policy being that each of those ranges should be sent over a different SA. Continuing the example above, the responder might have a policy of being willing to tunnel those addresses to and from the initiator, but might require that each address pair be on a separately negotiated Child SA. If the initiator didn't generate its request based on the packet, but (for example) upon startup, there would not be the very specific first traffic selectors helping the responder to select the correct range. There would be no way for the responder to determine which pair of addresses should be included in this tunnel, and it would have to make a guess or reject the request with a SINGLE_PAIR_REQUIRED Notify message.		Not support		Explanation
	2022	The SINGLE_PAIR_REQUIRED error indicates that a CREATE_CHILD_SA request is unacceptable because its sender is only willing to accept traffic selectors specifying a single pair of addresses.		Not support		Explanation
	2024	The requestor is expected to respond by requesting an SA for only the specific traffic it is trying to forward.		Not support		Explanation
	2028	Few implementations will have policies that require separate SAs for each address pair. Because of this, if only some parts of the TSi and TSr proposed by the initiator are acceptable to the responder, responders **SHOULD** narrow the selectors to an acceptable subset rather than use SINGLE_PAIR_REQUIRED.	SHOULD	BASIC	Both	EN.I.1.1.7.1 EN.R.1.1.7.1 SGW.I.1.1.7.1 SGW.R.1.1.7.1
	2034	2.9.1. Traffic Selectors Violating Own Policy				
	2036	When creating a new SA, the initiator needs to avoid proposing traffic selectors that violate its own policy. If this rule is not followed, valid traffic may be dropped. If you use decorrelated policies from [IPSECARCH], this kind of policy violations cannot happen.		Not support		Explanation
	2042	This is best illustrated by an example. Suppose that host A has a policy whose effect is that traffic to 198.51.100.66 is sent via host B encrypted using AES, and traffic to all other hosts in 198.51.100.0/24 is also sent via B, but must use 3DES. Suppose		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		also that host B accepts any combination of AES and 3DES.				
	2048	If host A now proposes an SA that uses 3DES, and includes TSr containing (198.51.100.0-198.51.100.255), this will be accepted by host B. Now, host B can also use this SA to send traffic from 198.51.100.66, but those packets will be dropped by A since it requires the use of AES for this traffic. Even if host A creates a new SA only for 198.51.100.66 that uses AES, host B may freely continue to use the first SA for the traffic. In this situation, when proposing the SA, host A should have followed its own policy, and included a TSr containing ((198.51.100.0-198.51.100.65),(198.51.100.67-198.51.100.255)) instead.		Not support		Explanation
	2059	In general, if (1) the initiator makes a proposal "for traffic X (TSi/TSr), do SA", and (2) for some subset X' of X, the initiator does not actually accept traffic X' with SA, and (3) the initiator would be willing to accept traffic X' with some SA' (!=SA), valid traffic can be unnecessarily dropped since the responder can apply either SA or SA' to traffic X'.		Not support		Explanation
	2066	2.10. Nonces				
	2068	The IKE_SA_INIT messages each contain a nonce. These nonces are used as inputs to cryptographic functions. The CREATE_CHILD_SA request and the CREATE_CHILD_SA response also contain nonces. These nonces are used to add freshness to the key derivation technique used to obtain keys for Child SA, and to ensure creation of strong pseudorandom bits from the Diffie-Hellman key.		Not support		Explanation
	2073	Nonces used in IKEv2 **MUST** be randomly chosen, **MUST** be at least 128 bits in size, and **MUST** be at least half the key size of the negotiated pseudo-random function (PRF).	MUST MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	2076	However, the initiator chooses the nonce before the outcome of the negotiation is known. Because of that, the nonce has to be long enough for all the PRFs being proposed.		Not support		Explanation
	2078	If the same random number source is used for both keys and nonces, care must be taken to ensure that the latter use does not		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		compromise the former.				
	2082	2.11. Address and Port Agility				
	2084	IKE runs over UDP ports 500 and 4500, and implicitly sets up ESP and AH associations for the same IP addresses over whitch it runs. The IP addresses and ports in the outer header are, however, not themselves cryptographically protected, and IKE is designed to work even through Network Address Translation (NAT) boxes.		Not support		Explanation
	2088	An implementation **MUST** accept incoming requests even if the source port is not 500 or 4500, and **MUST** respond to the address and port from which the request was received. It **MUST** specify the address and port at which the request was received as the source address and port in the response.	MUST MUST MUST	Not support		NAT traversal is out of the scope
	2092	IKE functions identically over IPv4 or IPv6.		Not support		Explanation
	2095	2.12. Reuse of Diffie-Hellman Exponentials				
	2097	IKE generates keying material using an ephemeral Diffie-Hellman exchange in order to gain the property of "perfect forward secrecy". This means that once a connection is closed and its corresponding keys are forgotten, even someone who has recorded all of the data from the connection and gets access to all of the long-term keys of the two endpoints cannot reconstruct the keys used to protect the conversation without doing a brute force search of the session key space.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.7 EN.R.1.2.5.5 SGW.I.1.2.3.7 SGW.R.1.2.5.5
	2106	Achieving perfect forward secrecy requires that when a connection is closed, each endpoint **MUST** forget not only the keys used by the connection but also any information that could be used to recompute those keys.	MUST	Not support		Explanation
	2111	Because computing Diffie-Hellman exponentials is computationally expensive, an endpoint may find it advantageous to reuse those exponentials for multiple connection setups. There are several reasonable strategies for doing this. An endpoint could choose a new exponential only periodically though this could result in less-than-perfect forward secrecy if some connection lasts for less than the lifetime of the exponential. Or it could keep track of which exponential was used for each connection and delete		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		the information associated with the exponential only when some corresponding connection was closed. This would allow the exponential to be reused without losing perfect forward secrecy at the cost of maintaining more state.				
	2124	Whether and when to reuse Diffie-Hellman exponentials are private decisions in the sense that they will not affect interoperability. An implementation that reuses exponentials **MAY** choose to remember the exponential used by the other endpoint on past exchanges and if one is reused to avoid the second half of the calculation. See [REUSE] for a security analysis of this practice and for additional security considerations when reusing ephemeral Diffie-Hellman keys.	MAY	Not support		Explanation
	2132	2.13. Generating Keying Material				
	2134	In the context of the IKE SA, four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudorandom function (PRF). The PRF is used for the construction of keying material for all of the cryptographic algorithms used in both the IKE SA and the Child SAs.		Not support		Explanation
	2141	We assume that each encryption algorithm and integrity protection algorithm uses a fixed-size key and that any randomly chosen value of that fixed size can serve as an appropriate key.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2143	For algorithms that accept a variable-length key, a fixed key size **MUST** be specified as part of the cryptographic transform negotiated (see Section 3.3.5 for the definition of the Key Length transform attribute).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2146	For algorithms for which not all values are valid keys (such as DES or 3DES with key parity), the algorithm by which keys are derived from arbitrary values **MUST** be specified by the cryptographic transform.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2150	For integrity protection functions based on Hashed Message Authentication Code (HMAC), the fixed key size is the size of the output of the underlying hash function.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	2154	It is assumed that PRFs accept keys of any length, but have a preferred key size. The preferred key size **MUST** be used as the length of SK_d, SK_pi, and SK_pr (see Section 2.14). For PRFs based on the HMAC construction, the preferred key size is equal to the length of the output of the underlying hash function. Other types of PRFs **MUST** specify their preferred key size.	MUST MUST	Not support		Internal process
	2161	Keying material will always be derived as the output of the negotiated PRF algorithm. Since the amount of keying material needed may be greater than the size of the output of the PRF, the		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		PRF is used iteratively. The term "prf+" describes a function that outputs a pseudo-random stream based on the inputs to a pseudorandom function called "prf".				
	2168	In the following, | indicates concatenation. prf+ is defined as:		Not support		Explanation
	2170	prf+ (K,S) = T1 | T2 | T3 | T4 | ... where: T1 = prf (K, S | 0x01) T2 = prf (K, T1 | S | 0x02) T3 = prf (K, T2 | S | 0x03) T4 = prf (K, T3 | S | 0x04) ...		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	2179	This continues until all the material needed to compute all required keys has been output from prf+. The keys are taken from the output string without regard to boundaries (e.g., if the required keys are a 256-bit Advanced Encryption Standard (AES)		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		key and a 160-bit HMAC key, and the prf function generates 160 bits, the AES key will come from T1 and the beginning of T2, while the HMAC key will come from the rest of T2 and the beginning of T3).				
	2187	The constant concatenated to the end of each prf function is a single octet. The prf+ function is not defined beyond 255 times the size of the prf function output.		Not support		Explanation
	2191	2.14. Generating Keying Material for the IKE SA				
	2193	The shared keys are computed as follows. A quantity called SKEYSEED is calculated from the nonces exchanged during the IKE_SA_INIT exchange and the Diffie-Hellman shared secret established during that exchange.		Not support		Explanation
	2196	SKEYSEED is used to calculate seven other secrets: SK_d used for deriving new keys for the Child SAs established with this IKE SA; SK_ai and SK_ar used as a key to the integrity protection algorithm for authenticating the component messages of subsequent exchanges; SK_ei and SK_er used for encrypting (and of course decrypting) all subsequent exchanges; and SK_pi and SK_pr, which are used when generating an AUTH payload.		Not support		Explanation
	2202	The lengths of SK_d, SK_pi, and SK_pr **MUST** be the preferred key length of the PRF agreed upon.	MUST	Not support		Internal process
	2205	SKEYSEED and its derivatives are computed as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2207	SKEYSEED = prf(Ni | Nr, g^ir) {SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr } = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	2212	(indicating that the quantities SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, and SK_pr are taken in order from the generated bits of the prf+). g^ir is the shared secret from the ephemeral Diffie-Hellman exchange. g^ir is represented as a string of octets in big endian order padded with zeros if necessary to make it the length of the modulus. Ni and Nr are the nonces, stripped of any headers. For historical backward-compatibility reasons, there are two PRFs that are treated specially in this calculation. If the negotiated PRF is AES-XCBC-PRF-128 [AESXCBCPRF128] or AES-CMAC-PRF-128 [AESCMACPRF128], only the first 64 bits of Ni and the first 64 bits of Nr are used in calculating SKEYSEED, but all the bits are used for input to the prf+ function.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2225	The two directions of traffic flow use different keys. The keys used to protect messages from the original initiator are SK_ai and SK_ei. The keys used to protect messages in the other direction are SK_ar and SK_er.		Not support		Explanation
	2230	2.15. Authentication of the IKE SA				
	2232	When not using extensible authentication (see Section 2.16), the peers are authenticated by having each sign (or MAC using a padded shared secret as the key, as described later in this section) a block of data.		Not support		Explanation
	2235	In these calculations, IDi' and IDr' are the entire ID payloads excluding the fixed header.		Not support	Not support	Explanation
	2236	For the responder, the octets to be signed start with the first octet of the first SPI in the header of the second message (IKE_SA_INIT response) and end with the last octet of the last payload in the second message. Appended to this (for the purposes of computing the signature) are the initiator's nonce Ni (just the value, not the payload containing it), and the value prf(SK_pr, IDr'). Note that neither the nonce Ni nor the value prf(SK_pr, IDr') are transmitted.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1
○	2243	Similarly, the initiator signs the first message (IKE_SA_INIT request), starting with the first octet of the first SPI in the header and ending with the last octet of the last payload. Appended to this (for purposes of computing the signature) are the responder's nonce Nr, and the value prf(SK_pi, IDi').		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.2 EN.I.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1
	2248	It is critical to the security of the exchange that each side sign the other side's nonce.		Not support		Explanation
	2251	The initiator's signed octets can be described as:		Not support		Explanation
	2253	InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI GenIKEHDR = [four octets 0 if using port 4500] | RealIKEHDR RealIKEHDR = SPIi | SPIr | . . . | Length RealMessage1 = RealIKEHDR | RestOfMessage1 NonceRPayload = PayloadHeader | NonceRData InitiatorIDPayload = PayloadHeader | RestOfInitIDPayload		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		RestOfInitIDPayload = IDType ︱ RESERVED ︱ InitIDData MACedIDForI = prf(SK_pi, RestOfInitIDPayload)				
	2262	The responder's signed octets can be described as:		Not support		Explanation
	2264	ResponderSignedOctets = RealMessage2 ︱ NonceIData ︱ MACedIDForR GenIKEHDR = [four octets 0 if using port 4500] ︱ RealIKEHDR RealIKEHDR = SPIi ︱ SPIr ︱ . . . ︱ Length RealMessage2 = RealIKEHDR ︱ RestOfMessage2 NonceIPayload = PayloadHeader ︱ NonceIData ResponderIDPayload = PayloadHeader ︱ RestOfRespIDPayload RestOfRespIDPayload = IDType ︱ RESERVED ︱ RespIDData MACedIDForR = prf(SK_pr, RestOfRespIDPayload)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2
	2273	Note that all of the payloads are included under the signature, including any payload types not defined in this document. If the first message of the exchange is sent multiple times (such as with a responder cookie and/or a different Diffie-Hellman group), it is the latest version of the message that is signed.		Not support		Explanation
	2279	Optionally, messages 3 and 4 **MAY** include a certificate, or certificate chain providing evidence that the key used to compute a digital signature belongs to the name in the ID payload. The signature or MAC will be computed using algorithms dictated by the type of key used by the signer, and specified by the Auth Method field in the Authentication payload. There is no requirement that the initiator and responder sign with the same cryptographic algorithms. The choice of cryptographic algorithms depends on the type of key each has. In particular, the initiator may be using a shared key while the responder may have a public signature key and certificate. It will commonly be the case (but it is not required) that if a shared secret is used for authentication that the same key is used in both directions.	MAY	ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.1 EN.I.1.1.10.2 EN.I.1.1.10.3 EN.R.1.1.10.1 EN.R.1.1.10.2 EN.R.1.1.10.3 SGW.I.1.1.10.1 SGW.I.1.1.10.2 SGW.I.1.1.10.3 SGW.R.1.1.10.1 SGW.R.1.1.10.2 SGW.R.1.1.10.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Note that it is a common but typically insecure practice to have a shared key derived solely from a user-chosen password without incorporating another source of randomness.				
	2295	This is typically insecure because user-chosen passwords are unlikely to have sufficient unpredictability to resist dictionary attacks and these attacks are not prevented in this authentication method. (Applications using password-based authentication for bootstrapping and IKE SA should use the authentication method in Section 2.16, which is designed to prevent off-line dictionary attacks.) The pre-shared key needs to contain as much unpredictability as the strongest key being negotiated.		Not support		Explanation
	2303	In the case of a pre-shared key, the AUTH value is computed as: For the initiator: AUTH = prf(prf(Shared Secret, "Key Pad for IKEv2"), <InitiatorSignedOctets>) For the responder: AUTH = prf(prf(Shared Secret, "Key Pad for IKEv2"), <ResponderSignedOctets>) where the string "Key Pad for IKEv2" is 17 ASCII characters without null termination. The shared secret can be variable length. The pad string is added so that if the shared secret is derived from a password, the IKE implementation need not store the password in cleartext, but rather can store the value prf(Shared Secret,"Key Pad for IKEv2"), which could not be used as a password equivalent for protocols other than IKEv2. As noted above, deriving the shared secret from a password is not secure. This construction is used because it is anticipated that people will do it anyway.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2321	The management interface by which the Shared Secret is provided **MUST** accept ASCII strings of at least 64 octets	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1
	2323	and **MUST NOT** add a null terminator before using them as shared secrets.	MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1
	2324	It **MUST** also accept a hex encoding of the Shared Secret.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.4 EN.R.1.1.10.4 SGW.I.1.1.10.4 SGW.R.1.1.10.4
	2325	The management interface **MAY** accept other encodings if the algorithm for translating the encoding to a binary string is specified.	MAY	Not support		Not need to test
	2329	There are two types of EAP authentication (described in Section 2.16), and each type uses different values in the AUTH computations shown above. If the EAP method is key-generating, substitute MSK for the Shared Secret in the computation. For non-key-generating methods, substitute SK_pi and SK_pr, respectively, for the Shared Secret in the two AUTH computations.		Not support		Explanation
	2336	2.16. Extensible Authentication Protocol Methods				
	2338	In addition to authentication using public key signatures and shared secrets, IKE supports authentication using methods defined in RFC 3748 [EAP]. Typically, these methods are asymmetric (designed for a user authenticating to a server), and they may not be mutual.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2341	For this reason, these protocols are typically used to authenticate the initiator to the responder and **MUST** be used in conjunction with a public key signature based authentication of the responder to the initiator.	MUST	Not support		EAP is out of the scope
	2345	These methods are often associated with mechanisms referred to as "Legacy Authentication" mechanisms.		Not support		Explanation
	2348	While this document references [EAP] with the intent that new methods can be added in the future without updating this specification, some simpler variations are documented here. [EAP] defines an authentication protocol requiring a variable number of messages.		Not support		Explanation
	2352	Extensible Authentication is implemented in IKE as additional IKE_AUTH exchanges that **MUST** be completed in order to initialize the IKE SA.	MUST	Not support		EAP is out of the scope
	2356	An initiator indicates a desire to use extensible authentication by leaving out the AUTH payload from the first message in the IKE_AUTH exchange. (Note that the AUTH payload is required for non-EAP authentication, and is thus not marked as optional in the rest of this document.) By including an IDi payload but not an AUTH payload, the initiator has declared an identity but has not proven it. If the responder is willing to use an extensible authentication method, it will place an Extensible Authentication Protocol (EAP) payload in the response of the IKE_AUTH exchange and defer sending SAr2, TSi, and TSr until initiator authentication is complete in a subsequent IKE_AUTH exchange. In the case of a minimal extensible authentication, the initial SA establishment will appear as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2369	Initiator Responder -- HDR, SAi1, KEi, Ni --> <-- HDR, SAr1, KEr, Nr, [CERTREQ] HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} --> <-- HDR, SK {IDr, [CERT,] AUTH, EAP } HDR, SK {EAP} --> <-- HDR, SK {EAP (success)} HDR, SK {AUTH} --> <-- HDR, SK {AUTH, SAr2, TSi, TSr }		Not support		Explanation
	2383	As described in Section 2.2, when EAP is used, each pair of IKE SA initial setup messages will have their message numbers incremented; the first pair of AUTH messages will have an ID of 1, the second will be 2, and so on.		Not support		EAP authentication is out of the scope
	2388	For EAP methods that create a shared key as a side effect of authentication, that shared key **MUST** be used by both the initiator and responder to generate AUTH payloads in messages 7 and 8 using the syntax for shared secrets specified in Section 2.15. The shared key from EAP is the field from the EAP specification named MSK. This shared key generated during an IKE exchange **MUST NOT** be used for any other purpose.	MUST MUST NOT	Not support		EAP is out of the scope
	2396	EAP methods that do not establish a shared key **SHOULD NOT** be used, as they are subject to a number of man-in-the-middle attacks [EAPMITM] if these EAP methods are used in other protocols that do not use a server-authenticated tunnel. Please see the Security Considerations section for more details.	SHOULD NOT	Not support		EAP is out of the scope
	2400	If EAP methods that do not generate a shared key are used, the AUTH payloads in messages 7 and 8 **MUST** be generated using SK_pi and SK_pr, respectively.	MUST	Not support		EAP is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2404	The initiator of an IKE SA using EAP needs to be capable of extending the initial protocol exchange to at least ten IKE_AUTH exchanges in the event the responder sends notification messages and/or retries the authentication prompt.		Not support		EAP is out of the scope
	2407	Once the protocol exchange defined by the chosen EAP authentication method has successfully terminated, the responder **MUST** send an EAP payload containing the Success message.	MUST	Not support		EAP is out of the scope
	2410	Similarly, if the authentication method has failed, the responder **MUST** send an EAP payload containing the Failure message.	MUST	Not support		EAP is out of the scope
	2411	The responder **MAY** at any time terminate the IKE exchange by sending an EAP payload containing the Failure message.	MAY	Not support		EAP is out of the scope
	2415	Following such an extended exchange, the EAP AUTH payloads **MUST** be included in the two messages following the one containing the EAP Success message.	MUST	Not support		EAP is out of the scope
	2419	When the initiator authentication uses EAP, it is possible that the contents of the IDi payload is used only for AAA routing purposes and selecting which EAP method to use. This value may be different from the identity authenticated by the EAP method.		Not support		Explanation
	2422	It is important that policy lookups and access control decisions use the actual authenticated identity.		Not support		Explanation
	2424	Often the EAP server is implemented in a separate AAA server that communicates with the IKEv2 responder. In this case, the authenticated identity, if different from that in the IDi payload, has to be sent from the AAA server to the IKEv2 responder.		Not support		Explanation
	2430	2.17. Generating Keying Material for Child SAs				
	2432	A single Child SA is created by the IKE_AUTH exchange, and additional Child SAs can optionally be created in CREATE_CHILD_SA exchanges. Keying material for them is generated as follows:		Not support		Explanation
	2436	KEYMAT = prf+(SK_d, Ni | Nr)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.3 EN.R.1.1.1.3 SGW.I.1.1.1.3 SGW.R.1.1.1.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2438	Where Ni and Nr are the nonces from the IKE_SA_INIT exchange if this request is the first Child SA created or the fresh Ni and Nr from the CREATE_CHILD_SA exchange if this is a subsequent creation.		Not support		Explanation
	2442	For CREATE_CHILD_SA exchanges including an optional Diffie-Hellman exchange, the keying material is defined as:		Not support		Explanation
	2445	KEYMAT = prf+(SK_d, g^ir (new) | Ni | Nr)		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.7 EN.R.1.2.5.5 SGW.I.1.2.3.7 SGW.R.1.2.5.5
	2447	where g^ir (new) is the shared secret from the ephemeral Diffie-Hellman exchange of this CREATE_CHILD_SA exchange (represented as an octet string in big endian order padded with zeros in the high-order bits if necessary to make it the length of the modulus).		Not support		Explanation
	2452	A single CHILD_SA negotiation may result in multiple security associations. ESP and AH SAs exist in pairs (one in each direction), so two SAs are created in a single Child SA negotiation for them. Furthermore, Child SA negotiation may include some future IPsec protocol(s) in addition to, or instead of, ESP or AH (for example, ROHC_INTEG as described in [ROHCV2]).		Not support		Explanation
	2457	In any case, keying material for each child SA **MUST** be taken from the expanded KEYMAT using the following rules:	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.3 EN.R.1.1.1.3 SGW.I.1.1.1.3 SGW.R.1.1.1.3
	2461	o All keys for SAs carrying data from the initiator to the responder are taken before SAs going from the responder to the initiator.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.3 EN.R.1.1.1.3 SGW.I.1.1.1.3 SGW.R.1.1.1.3
	2464	o If multiple IPsec protocols are negotiated, keying material for each Child SA is taken in the order in which the protocol headers will appear in the encapsulated packet.	MUST	Not support		Explanation
	2468	o If an IPsec protocol requires multiple keys, the order in which they are taken from the SA's keying material needs to be described	MUST MUST	Not support		Internal process

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		in the protocol's specification. For ESP and AH, [IPSECARCH] defines the order, namely: the encryption key (if any) **MUST** be taken from the first bits and the integrity key (if any) **MUST** be taken from the remaining bits.				
	2475	Each cryptographic algorithm takes a fixed number of bits of keying material specified as part of the algorithm, or negotiated in SA payloads (see Section 2.13 for description of key lengths, and Section 3.3.5 for the definition of the Key Length transform attribute).		Not support		Explanation
	2481	2.18. Rekeying IKE SAs Using a CREATE_CHILD_SA Exchange				
	2483	The CREATE_CHILD_SA exchange can be used to rekey an existing IKE SA (see Section 1.3.2 and Section 2.8). New initiator and responder SPIs are supplied in the SPI fields in the Proposal structures inside the Security Association (SA) payloads (not the SPI fields in the IKE header). The TS payloads are omitted when rekeying an IKE SA. SKEYSEED for the new IKE SA is computed using SK_d from the existing IKE SA as follows:		Not support		Explanation
	2491	SKEYSEED = prf(SK_d (old), g^ir (new) | Ni | Nr)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.4.2 EN.R.1.2.6.3 SGW.I.1.2.4.2 SGW.R.1.2.6.3
	2493	where g^ir (new) is the shared secret from the ephemeral Diffie-Hellman exchange of this CREATE_CHILD_SA exchange (represented as an octet string in big endian order padded with zeros if necessary to make it the length of the modulus) and Ni and Nr are the two nonces stripped of any headers.		Not support		Explanation
	2499	The old and new IKE SA may have selected a different PRF. Because the rekeying exchange belongs to the old IKE SA, it is the old IKE SA's PRF that is used to generate SKEYSEED.		Not support		Explanation
	2503	The main reason for rekeying the IKE SA is to ensure that the compromise of old keying material does not provide information about the current keys, or vice versa. Therefore, implementations **MUST** perform a new Diffie-Hellman exchange when rekeying the IKE SA. In other words, an initiator **MUST NOT** propose the value "NONE" for the Diffie-Hellman transform, and a responder	MUST MUST NOT MUST NOT	BASIC	Both	EN.I.1.2.4.1 SGW.I.1.2.4.1 [EN.R.P29.L250 3.ADD] [SGW.R.P29.L25

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		MUST NOT accept such a proposal. This means that a successful exchange rekeying the IKE SA always includes the KEi/KEr payloads.				03.ADD]
	2512	The new IKE SA **MUST** reset its message counters to 0.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.4.2 EN.R.1.2.6.3 SGW.I.1.2.4.2 SGW.R.1.2.6.3
	2514	SK_d, SK_ai, SK_ar, SK_ei, and SK_er are computed from SKEYSEED as specified in Section 2.14, using SPIi, SPIr, Ni, and Nr from the new exchange, and using the new IKE SA's PRF.		Not support		Explanation
	2518	2.19. Requesting an Internal Address on a Remote Network				
	2520	Most commonly occurring in the endpoint-to-security-gateway scenario, an endpoint may need an IP address in the network protected by the security gateway and may need to have that address dynamically assigned. A request for such a temporary address can be included in any request to create a Child SA (including the implicit request in message 3) by including a CP payload.		Not support		Explanation
	2525	Note, however, it is usual to only assign one IP address during the IKE_AUTH exchange. That address persists at least until the deletion of the IKE SA.		Not support		Explanation
	2529	This function provides address allocation to an IPsec Remote Access Client (IRAC) trying to tunnel into a network protected by an IPsec Remote Access Server (IRAS).		Not support		Explanation
	2531	Since the IKE_AUTH exchange creates an IKE SA and a Child SA, the IRAC **MUST** request the IRAS-controlled address (and optionally other information concerning the protected network) in the IKE_AUTH exchange.	MUST	ADVANCED	EN(initiator)	EN.I.2.1.2.1
	2534	The IRAS may procure an address for the IRAC from any number of sources such as a DHCP/BOOTP server or its own address pool.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2538	Initiator Responder --- HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, CP(CFG_REQUEST), SAi2, TSi, TSr} --> <-- HDR, SK {IDr, [CERT,] AUTH, CP(CFG_REPLY), SAr2, TSi, TSr}		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1
	2548	In all cases, the CP payload **MUST** be inserted before the SA payload. In variations of the protocol where there are multiple IKE_AUTH exchanges, the CP payloads **MUST** be inserted in the messages containing the SA payloads.	MUST MUST	ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 SGW.R.2.1.2.1
	2553	CP(CFG_REQUEST) **MUST** contain at least an INTERNAL_ADDRESS attribute (either IPv4 or IPv6)	MUST	ADVANCED	EN(initiator)	EN.I.2.1.2.1
	2554	but **MAY** contain any number of additional attributes the initiator wants returned in the response.	MAY	Not support		Not need to test
	2557	For example, message from initiator to responder:		Not support		Explanation
	2559	CP(CFG_REQUEST)= INTERNAL_ADDRESS() TSi = (0, 0-65535,0.0.0.0-255.255.255.255) TSr = (0, 0-65535,0.0.0.0-255.255.255.255)		Not support		Explanation
	2564	NOTE: Traffic selectors contain (protocol, port range, address range).		Not support		Explanation
	2567	Message from responder to initiator:		Not support		Explanation
	2569	CP(CFG_REPLY)= INTERNAL_ADDRESS(192.0.2.202) INTERNAL_NETMASK(255.255.255.0) INTERNAL_SUBNET(192.0.2.0/255.255.255.0) TSi = (0, 0-65535,192.0.2.202-192.0.2.202) TSr = (0, 0-65535,192.0.2.0-192.0.2.255)		Not support		Explanation
	2576	All returned values will be implementation dependent. As can be seen in the above example, the IRAS **MAY** also send other	MAY MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		attributes that were not included in CP(CFG_REQUEST) and **MAY** ignore the non-mandatory attributes that it does not support.				
	2581	The responder **MUST NOT** send a CFG_REPLY without having first received a CP(CFG_REQUEST) from the initiator, because we do not want the IRAS to perform an unnecessary configuration lookup if the IRAC cannot process the REPLY.	MUST NOT	ADVANCED	SGW(responder)	SGW.R.2.1.2.3
	2586	In the case where the IRAS's configuration requires that CP be used for a given identity IDi, but IRAC has failed to send a CP(CFG_REQUEST), IRAS **MUST** fail the request, and terminate the Child SA creation with a FAILED_CP_REQUIRED error. The FAILED_CP_REQUIRED is not fatal to the IKE SA; it simply causes the Child SA creation to fail. The initiator can fix this by later starting a new configuration payload request. There is no associated data in the FAILED_CP_REQUIRED error.	MUST	ADVANCED	SGW(responder)	SGW.R.2.1.2.3
	2595	2.20. Requesting the Peer's Version				
	2597	An IKE peer wishing to inquire about the other peer's IKE software version information **MAY** use the method below. This is an example of a configuration request within an INFORMATIONAL exchange, after the IKE SA and first Child SA have been created.	MAY	Not support		Not need to test
	2602	An IKE implementation **MAY** decline to give out version information prior to authentication or even after authentication in case some implementation is known to have some security weakness. In that case, it **MUST** either return an empty string or no CP payload if CP is not supported.	MAY MUST	Not support		Explanation
	2608	Initiator Responder --- HDR, SK{CP(CFG_REQUEST)} --> <-- HDR, SK{CP(CFG_REPLY)} CP(CFG_REQUEST)= APPLICATION_VERSION("") CP(CFG_REPLY) APPLICATION_VERSION("foobar v1.3beta, (c) Foo Bar		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Inc.")				
	2619	2.21. Error Handling				
	2621	There are many kinds of errors that can occur during IKE processing.		Not support		Explanation
	2622	The general rule is that if a request is received that is badly formatted, or unacceptable for reasons of policy (such as no matching cryptographic algorithms), the response contains a Notify payload indicating the error.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.7 EN.R.1.1.4.2 EN.R.1.1.4.4 EN.R.1.1.6.7 EN.R.1.1.7.2 EN.R.1.2.4.1 SGW.I.1.1.6.7 SGW.R.1.1.4.2 SGW.R.1.1.4.4 SGW.R.1.1.6.7 SGW.R.1.1.7.2 SGW.R.1.2.4.1
	2625	The decision whether or not to send such a response depends whether or not there is an authenticated IKE SA.		Not support		Explanation
	2628	If there is an error parsing or processing a response packet, the general rule is to not send back any error message because responses should not generate new requests (and a new request would be the only way to send back an error message). Such errors in parsing or processing response packets should still cause the recipient to clean up the IKE state (for example, by sending a DELETE for a bad SA).		Not support		just general rule
	2635	Only authentication failures (AUTHENTICATION_FAILED and EAP failure) and malformed messages (INVALID_SYNTAX) lead to a deletion of the IKE SA without requiring an explicit	MAY	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		INFORMATIONAL exchange carrying a DELETE payload. Other error conditions **MAY** require such an exchange if policy dictates that this is needed. If the exchange is terminated with EAP Failure, an AUTHENTICATION_FAILED notification is not sent.				
	2643	2.21.1. Error Handling in IKE_SA_INIT				
	2645	Errors that occur before a cryptographically protected IKE SA is established need to be handled very carefully. There is a trade-off between wanting to help the peer to diagnose a problem and thus responding to the error, and wanting to avoid being part of a DoS attack based on forged messages.		Not support		Explanation
	2651	In an IKE_SA_INIT exchange, any error notification causes the exchange to fail. Note that some error notifications such as COOKIE, INVALID_KE_PAYLOAD or INVALID_MAJOR_VERSION may lead to a subsequent successful exchange.		Not support		Explanation
	2654	Because all error notifications are completely unauthenticated, the recipient should continue trying for some time before giving up. The recipient should not immediately act based on the error notification unless corrective actions are defined in this specification, such as for COOKIE, INVALID_KE_PAYLOAD, and INVALID_MAJOR_VERSION.		Not support		test condition is ambiguous (immediately act)
	2661	2.21.2. Error Handling in IKE_AUTH				
	2663	All errors that occur in an IKE_AUTH exchange, causing the authentication to fail for whatever reason (invalid shared secret, invalid ID, untrusted certificate issuer, revoked or expired certificate, etc.) **SHOULD** result in an AUTHENTICATION_FAILED notification.	SHOULD	BASIC		[EN.R.P57.L2663.ADD] [SGW.R.P57.L2663.ADD]
	2667	If the error occurred on the responder, the notification is returned in the protected response, and is usually the only payload in that response. Although the IKE_AUTH messages are encrypted and integrity protected, if the peer receiving this notification has not authenticated the other end yet, that peer needs to treat the information with caution.		BASIC		[EN.R.P57.L2663.ADD] [SGW.R.P57.L2663.ADD]

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2674	If the error occurs on the initiator, the notification **MAY** be returned in a separate INFORMATIONAL exchange, usually with no other payloads. This is an exception for the general rule of not starting new exchanges based on errors in responses.	MAY	Not support		Explanation
	2679	Note, however, that request messages that contain an unsupported critical payload, or where the whole message is malformed (rather than just bad payload contents), **MUST** be rejected in their entirety, and **MUST** only lead to an UNSUPPORTED_CRITICAL_PAYLOAD or INVALID_SYNTAX Notification sent as a response.	MUST MUST	Not support		test condition is ambiguous (malformed message)
	2684	The receiver should not verify the payloads related to authentication in this case.		Not support		Explanation
	2686	If authentication has succeeded in the IKE_AUTH exchange, the IKE SA is established; however, establishing the Child SA or requesting configuration information may still fail. This failure does not automatically cause the IKE SA to be deleted.		BASIC	Both	EN.I.1.1.6.12 EN.R.1.1.6.9 SGW.I.1.1.6.12 SGW.R.1.1.6.9
	2689	Specifically, a responder may include all the payloads associated with authentication (IDr, Cert and AUTH) while sending error notifications for the piggybacked exchanges (FAILED_CP_REQUIRED, NO_PROPOSAL_CHOSEN, and so on), and the initiator **MUST NOT** fail the authentication because of this.	MUST NOT	BASIC	EN(responder) SGW(responder)	EN.R.1.2.4.1 EN.R.1.2.6.9 SGW.R.1.2.4.1 SGW.R.1.2.6.9
	2694	The initiator MAY, of course, for reasons of policy later delete such an IKE SA.	MAY	Not support		Explanation
	2697	In an IKE_AUTH exchange, or in the INFORMATIONAL exchange immediately following it (in case an error happened when processing a response to IKE_AUTH), the UNSUPPORTED_CRITICAL_PAYLOAD, INVALID_SYNTAX, and AUTHENTICATION_FAILED notifications are the only ones to cause the IKE SA to be deleted or not created, without a DELETE payload. Extension documents may define new error notifications with these semantics, but **MUST NOT** use them unless the peer has been shown to understand them, such as by using the Vendor ID payload.	MUST NOT	Not support		Explanation
	2706	2.21.3. Error Handling after IKE SA is Authenticated			SGW(responder)	

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2708	After the IKE SA is authenticated all requests having errors **MUST** result in a response notifying about the error.	MUST	Not support		Explanation
	2711	In normal situations, there should not be cases where a valid response from one peer results in an error situation in the other peer, so there should not be any reason for a peer to send error messages to the other end except as a response. Because sending such error messages as an INFORMATIONAL exchange might lead to further errors that could cause loops, such errors **SHOULD NOT** be sent. If errors are seen that indicate that the peers do not have the same state, it might be good to delete the IKE SA to clean up state and start over.	SHOULD NOT	Not support		Explanation
	2721	If a peer parsing a request notices that it is badly formatted (after it has passed the message authentication code checks and window checks) and it returns an INVALID_SYNTAX notification, then this error notification is considered fatal in both peers, meaning that the IKE SA is deleted without needing an explicit DELETE payload.		Not support		INVALID_SYNTAX is out of the scope
	2727	2.21.4. Error Handling Outside IKE SA				
	2729	A node needs to limit the rate at which it will send messages in response to unprotected messages.		Not support		Explanation
	2732	If a node receives a message on UDP port 500 or 4500 outside the context of an IKE SA known to it (and the message is not a request to start an IKE SA), this may be the result of a recent crash of the node.		Not support		Explanation
	2735	If the message is marked as a response, the node can audit the suspicious event but **MUST NOT** respond.	MUST NOT	Not support		untestable
	2736	If the message is marked as a request, the node can audit the suspicious event and **MAY** send a response.	MAY	Not support		Not need to test
	2738	If a response is sent, the response **MUST** be sent to the IP address and port from where it came with the same IKE SPIs and the Message ID copied. The response **MUST NOT** be cryptographically protected and **MUST** contain an INVALID_IKE_SPI Notify payload. The INVALID_IKE_SPI notification indicates an IKE message was received with an	MUST MUST NOT MUST	Not support		untestable

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		unrecognized destination SPI; this usually indicates that the recipient has rebooted and forgotten the existence of an IKE SA.				
	2746	A peer receiving such an unprotected Notify payload **MUST NOT** respond and **MUST NOT** change the state of any existing SAs. The message might be a forgery or might be a response that a genuine correspondent was tricked into sending.	MUST NOT MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.1 EN.I.1.1.3.2 EN.R.1.1.3.1 EN.R.1.1.3.2 SGW.I.1.1.3.1 SGW.I.1.1.3.2 SGW.R.1.1.3.1 SGW.R.1.1.3.2
	2749	A node should treat such a message (and also a network message like ICMP destination unreachable) as a hint that there might be problems with SAs to that IP address and should initiate a liveness check for any such IKE SA. An implementation **SHOULD** limit the frequency of such tests to avoid being tricked into participating in a DoS attack.	SHOULD	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.1 EN.I.1.1.3.2 EN.R.1.1.3.1 EN.R.1.1.3.2 SGW.I.1.1.3.1 SGW.I.1.1.3.2 SGW.R.1.1.3.1 SGW.R.1.1.3.2
	2756	If an error occurs outside the context of an IKE request (e.g., the node is getting ESP messages on a nonexistent SPI), the node **SHOULD** initiate an INFORMATIONAL exchange with a Notify payload describing the problem.	SHOULD	Not support		untestable
	2761	A node receiving a suspicious message from an IP address (and port, if NAT traversal is used) with which it has an IKE SA **SHOULD** send an IKE Notify payload in an IKE INFORMATIONAL exchange over that SA.	SHOULD	Not support		Not need to test
	2764	The recipient **MUST NOT** change the state of any SAs as a result, but may wish to audit the event to aid in diagnosing malfunctions.	MUST NOT	Not support		Explanation
	2767	2.22. IPComp				
	2769	Use of IP compression [IP-COMP] can be negotiated as part of the setup of a Child SA. While IP compression involves an extra header in each packet and a compression parameter index (CPI), the virtual "compression association" has no life outside the ESP or		Not support		IPComp is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		AH SA that contains it. Compression associations disappear when the corresponding ESP or AH SA goes away. It is not explicitly mentioned in any DELETE payload.				
	2777	Negotiation of IP compression is separate from the negotiation of cryptographic parameters associated with a Child SA.		Not support		IPComp is out of the scope
	2778	A node requesting a Child SA MAY advertise its support for one or more compression algorithms through one or more Notify payloads of type IPCOMP_SUPPORTED.	MAY	Not support		IPComp is out of the scope
	2781	This Notify message may be included only in a message containing an SA payload negotiating a Child SA and indicates a willingness by its sender to use IPComp on this SA.		Not support		IPComp is out of the scope
	2783	The response MAY indicate acceptance of a single compression algorithm with a Notify payload of type IPCOMP_SUPPORTED.	MAY	Not support		IPComp is out of the scope
	2785	These payloads MUST NOT occur in messages that do not contain SA payloads.	MUST NOT	Not support		IPComp is out of the scope
	2788	The data associated with this Notify message includes a two-octet IPComp CPI followed by a one-octet transform ID optionally followed by attributes whose length and format are defined by that transform ID. A message proposing an SA may contain multiple IPCOMP_SUPPORTED notifications to indicate multiple supported algorithms. A message accepting an SA may contain at most one.		Not support		IPComp is out of the scope
	2795	The transform IDs are listed here. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		IPComp is out of the scope
	2801	Name Number Defined In ------------------------------------ IPCOMP_OUI 1 IPCOMP_DEFLATE 2 RFC 2394 IPCOMP_LZS 3 RFC 2395 IPCOMP_LZJH 4 RFC 3051		Not support		IPComp is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2808	Although there has been discussion of allowing multiple compression algorithms to be accepted and to have different compression algorithms available for the two directions of a Child SA,		Not support		IPComp is out of the scope
	2811	implementations of this specification **MUST NOT** accept an IPComp algorithm that was not proposed, **MUST NOT** accept more than one, and **MUST NOT** compress using an algorithm other than one proposed and accepted in the setup of the Child SA.	MUST NOT MUST NOT MUST NOT	Not support		IPComp is out of the scope
	2816	A side effect of separating the negotiation of IPComp from cryptographic parameters is that it is not possible to propose multiple cryptographic suites and propose IP compression with some of them but not others.		Not support		IPComp is out of the scope
	2821	In some cases, Robust Header Compression (ROHC) may be more appropriate than IP Compression. [ROHCV2] defines the use of ROHC with IKEv2 and IPsec.		Not support		IPComp is out of the scope
	2825	2.23. NAT Traversal				
	2827	Network Address Translation (NAT) gateways are a controversial subject. This section briefly describes what they are and how they are likely to act on IKE traffic. Many people believe that NATs are evil and that we should not design our protocols so as to make them work better. IKEv2 does specify some unintuitive processing rules in order that NATs are more likely to work.		Not support		Explanation
	2834	NATs exist primarily because of the shortage of IPv4 addresses, though there are other rationales. IP nodes that are "behind" a NAT have IP addresses that are not globally unique, but rather are assigned from some space that is unique within the network behind the NAT but that are likely to be reused by nodes behind other NATs. Generally, nodes behind NATs can communicate with other nodes behind the same NAT and with nodes with globally unique addresses, but not with nodes behind other NATs. There are exceptions to that rule. When those nodes make connections to nodes on the real Internet, the NAT gateway "translates" the IP source address to an address that will be routed back to the gateway. Messages to the gateway from the Internet have their destination addresses "translated" to the internal address that will		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		route the packet to the correct endnode.				
	2848	NATs are designed to be "transparent" to endnodes. Neither software on the node behind the NAT nor the node on the Internet requires modification to communicate through the NAT. Achieving this transparency is more difficult with some protocols than with others. Protocols that include IP addresses of the endpoints within the payloads of the packet will fail unless the NAT gateway understands the protocol and modifies the internal references as well as those in the headers. Such knowledge is inherently unreliable, is a network layer violation, and often results in subtle problems.		Not support		Explanation
	2858	Opening an IPsec connection through a NAT introduces special problems. If the connection runs in transport mode, changing the IP addresses on packets will cause the checksums to fail and the NAT cannot correct the checksums because they are cryptographically protected. Even in tunnel mode, there are routing problems because transparently translating the addresses of AH and ESP packets requires special logic in the NAT and that logic is heuristic and unreliable in nature. For that reason, IKEv2 will use UDP encapsulation of IKE and ESP packets. This encoding is slightly less efficient but is easier for NATs to process. In addition, firewalls may be configured to pass UDP-encapsulated IPsec traffic but not plain, unencapsulated ESP/AH or vice versa.		Not support		Explanation
	2871	It is a common practice of NATs to translate TCP and UDP port numbers as well as addresses and use the port numbers of inbound packets to decide which internal node should get a given packet. For this reason, even though IKE packets **MUST** be sent from and to UDP port 500 or 4500, they **MUST** be accepted coming from any port and responses **MUST** be sent to the port from whence they came. This is because the ports may be modified as the packets pass through NATs. Similarly, IP addresses of the IKE endpoints	MUST MUST MUST	Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		are generally not included in the IKE payloads because the payloads are cryptographically protected and could not be transparently modified by NATs.				
	2882	Port 4500 is reserved for UDP-encapsulated ESP and IKE. An IPsec endpoint that discovers a NAT between it and its correspondent (as described below) MUST send all subsequent traffic from port 4500, which NATs should not treat specially (as they might with port 500).		Not support		NAT traversal is out of the scope
	2887	An initiator can use port 4500 for both IKE and ESP, regardless of whether or not there is a NAT, even at the beginning of IKE. When either side is using port 4500, sending ESP with UDP encapsulation is not required, but understanding received UDP encapsulated ESP packets is required. UDP encapsulation **MUST NOT** be done on port 500. If NAT-T is supported (that is, if NAT_DETECTION_*_IP payloads were exchanged during IKE_SA_INIT), all devices **MUST** be able to receive and process both UDP encapsulated ESP and non-UDP encapsulated ESP packets at any time. Either side can decide whether or not to use UDP encapsulation for ESP irrespective of the choice made by the other side. However, if a NAT is detected, both devices **MUST** use UDP encapsulation for ESP.	MUST NOT MUST MUST	Not support		NAT traversal is out of the scope
	2900	The specific requirements for supporting NAT traversal [NATREQ] are listed below. Support for NAT traversal is optional. In this section only, requirements listed as **MUST** apply only to implementations supporting NAT traversal.	MUST	Not support		Not need to test
	2905	o Both IKE initiator and responder **MUST** include in their IKE_SA_INIT packets Notify payloads of type NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP. Those payloads can be used to detect if there is NAT between the hosts, and which end is behind the NAT. The location of the payloads in the IKE_SA_INIT packets is just after the Ni and Nr payloads (before the optional CERTREQ payload).	MUST	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2913	o The data associated with the NAT_DETECTION_SOURCE_IP notification is a SHA-1 digest of the SPIs (in the order they appear in the header), IP address, and port from which this packet was sent.		Not support		Explanation
	2916	There **MAY** be multiple NAT_DETECTION_SOURCE_IP payloads in a message if the sender does not know which of several network attachments will be used to send the packet.	MAY	Not support		NAT traversal is out of the scope
	2920	o The data associated with the NAT_DETECTION_DESTINATION_IP notification is a SHA-1 digest of the SPIs (in the order they appear in the header), IP address, and port to which this packet was sent.		Not support		Explanation
	2925	o The recipient of either the NAT_DETECTION_SOURCE_IP or NAT_DETECTION_DESTINATION_IP notification **MAY** compare the supplied value to a SHA-1 hash of the SPIs, source or recipient IP address (respectively), address, and port, and if they don't match it **SHOULD** enable NAT traversal.	MAY SHOULD	Not support		Explanation
	2929	In the case there is a mismatch of the NAT_DETECTION_SOURCE_IP hash with all of the NAT_DETECTION_SOURCE_IP payloads received, the recipient **MAY** reject the connection attempt if NAT traversal is not supported. In the case of a mismatching NAT_DETECTION_DESTINATION_IP hash, it means that the system receiving the NAT_DETECTION_DESTINATION_IP payload is behind a NAT and that system **SHOULD** start sending keepalive packets as defined in [UDPENCAPS]; alternately, it **MAY** reject the connection attempt if NAT traversal is not supported.	MAY SHOULD MAY	Not support		NAT traversal is out of the scope
	2939	o If none of the NAT_DETECTION_SOURCE_IP payload(s) received matches the expected value of the source IP and port found from the IP header of the packet containing the payload, it means that the system sending those payloads is behind NAT (i.e., someone along the route changed the source address of the original packet to match the address of the NAT box). In this case, the system receiving the payloads should allow dynamic update of the other systems' IP address, as described later.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	2948	o The IKE initiator **MUST** check the NAT_DETECTION_SOURCE_IP or NAT_DETECTION_DESTINATION_IP payloads if present and if they do not match the addresses in the outer packet **MUST** tunnel all future IKE and ESP packets associated with this IKE SA over UDP port 4500.	MUST MUST	Not support		NAT traversal is out of the scope
	2954	o To tunnel IKE packets over UDP port 4500, the IKE header has four octets of zero prepended and the result immediately follows the UDP header. To tunnel ESP packets over UDP port 4500, the ESP header immediately follows the UDP header. Since the first four octets of the ESP header contain the SPI, and the SPI cannot validly be zero, it is always possible to distinguish ESP and IKE messages.		Not support		Explanation
	2962	o Implementations **MUST** process received UDP-encapsulated ESP packets even when no NAT was detected.	MUST	Not support		NAT traversal is out of the scope
	2965	o The original source and destination IP address required for the transport mode TCP and UDP packet checksum fixup (see [UDPENCAPS]) are obtained from the traffic selectors associated with the exchange. In the case of transport mode NAT traversal, the traffic selectors **MUST** contain exactly one IP address, which is then used as the original IP address. This is covered in greater detail in Section 2.23.1.	MUST	Not support		Not need to test
	2973	There are cases where a NAT box decides to remove mappings that are still alive (for example, the keepalive interval is too long, or the NAT box is rebooted). This will be apparent to a host if it receives a packet whose integrity protection validates, but has a different port, address, or both from the one that was associated with the SA in the validated packet. When such a validated packet is found, a host that does not support other methods of recovery such as MOBIKE [MOBIKE], and that is not behind a NAT, **SHOULD** send all packets (including retransmission packets) to the IP address and port in the validated packet, and **SHOULD** store this as the new address and port combination for the SA (that is, they **SHOULD** dynamically update the address). A host behind a NAT **SHOULD NOT** do this type of dynamic address update if a	SHOULD SHOULD SHOULD SHOULD NOT	Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		validated packet has different port and/or address values because it opens a possible DoS attack (such as allowing an attacker to break the connection with a single packet). Also, dynamic address update should only be done in response to a new packet; otherwise, an attacker can revert the addresses with old replayed packets. Because of this, dynamic update can only be done safely if replay protection is enabled. When IKEv2 is used with MOBIKE, dynamically updating the addresses described above interferes with MOBIKE's way of recovering from the same situation. See Section 3.8 of [MOBIKE] for more information.				
	2997	2.23.1. Transport Mode NAT Traversal				
	2999	Transport mode used with NAT Traversal requires special handling of the traffic selectors used in the IKEv2. The complete scenario looks like:		Not support		NAT traversal is out of the scope
	3003	+------+ +------+ +------+ +------+ |Client| IP1 | NAT | IPN1 IPN2 | NAT | IP2 |Server| |node |<------>| A |<---------->| B |<------->| | +------+ +------+ +------+ +------+		Not support		NAT traversal is out of the scope
	3008	(Other scenarios are simplifications of this complex case, so this discussion uses the complete scenario.)		Not support		NAT traversal is out of the scope
	3011	In this scenario, there are two address translating NATs: NAT A and NAT B. NAT A is dynamic NAT that maps the clients source address IP1 to IPN1. NAT B is static NAT configured so that connections coming to IPN2 address are mapped to the gateways address IP2, that is, IPN2 destination address is mapped to IP2. This allows the client to connect to a server by connecting to the IPN2. NAT B does not necessarily need to be a static NAT, but the client needs to know how to connect to the server, and it can only do that if it somehow knows the outer address of the NAT B, that is, the IPN2 address. If NAT B is a static NAT, then its address can be configured to the client's configuration. Other options would be find it using some other protocol (like DNS), but those are outside of scope of IKEv2.		Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3024	In this scenario, both client and server are configured to use transport mode for the traffic originating from the client node and destined to the server.		Not support		NAT traversal is out of the scope
	3028	When the client starts creating the IKEv2 SA and Child SA for sending traffic to the server, it may have a triggering packet with source IP address of IP1, and a destination IP address of IPN2. Its PAD and SPD needs to have configuration matching those addresses (or wildcard entries covering them). Because this is transport mode, it uses exactly same addresses as the traffic selectors and outer IP address of the IKE packets. For transport mode, it **MUST** use exactly one IP address in the TSi and TSr payloads. It can have multiple traffic selectors if it has, for example, multiple port ranges that it wants to negotiate, but all TSi entries must use IP1-IP1 range as the IP addresses, and all TSr entries must have the IPN2-IPN2 range as IP addresses. The first traffic selector of TSi and TSr **SHOULD** have very specific traffic selectors including protocol and port numbers, such as from the packet triggering the request.	MUST SHOULD	Not support		NAT traversal is out of the scope
	3043	NAT A will then replace the source address of the IKE packet from IP1 to IPN1, and NAT B will replace the destination address of the IKE packet from IPN2 to IP2, so when the packet arrives to the server it will still have the exactly same traffic selectors which were sent by the client, but the IP address of the IKE packet has been replaced to IPN1 and IP2.		Not support		NAT traversal is out of the scope
	3050	When the server receives this packet, it normally looks in the Peer Authorization Database (PAD) described in RFC 4301 [IPSECARCH] based on the ID and then searches the SPD based on the traffic selectors. Because IP1 does not really mean anything to the server (it is the address client has behind the NAT), it is useless to do a lookup based on that if transport mode is used. On the other hand, the server cannot know whether transport mode is allowed by its policy before it finds the matching SPD entry.		Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3059	In this case, the server should first check that the initiator requested transport mode, and then do address substitution on the traffic selectors. It needs to first store the old traffic selector IP addresses to be used later for the incremental checksum fixup (the IP address in the TSi can be stored as the original source address and the IP address in the TSr can be stored as the original destination address). After that, if the other end was detected as being behind a NAT, the server replaces the IP address in TSi payloads with the IP address obtained from the source address of the IKE packet received (that is, it replaces IP1 in TSi with IPN1). If the server's end was detected to be behind NAT, it replaces the IP address in the TSr payloads with the IP address obtained from the destination address of the IKE packet received (that is, it replaces IPN2 in TSr with IP2).		Not support		NAT traversal is out of the scope
	3074	After this address substitution, both the traffic selectors and the IKE UDP source/destination addresses look the same, and the server does SPD lookup based on those new traffic selectors. If an entry is found and it allows transport mode, then that entry is used. If an entry is found but it does not allow transport mode, then the server **MAY** undo the address substitution and redo the SPD lookup using the original traffic selectors. If the second lookup succeeds, the server will create an SA in tunnel mode using real traffic selectors sent by the other end.	MAY	Not support		NAT traversal is out of the scope
	3084	This address substitution in transport mode is needed because the SPD is looked up using the addresses that will be seen by the local host. This also will make sure the SAD entries for the tunnel exit checks and return packets is added using the addresses as seen by the local operating system stack.		Not support		NAT traversal is out of the scope
	3090	The most common case is that the server's SPD will contain wildcard entries matching any addresses, but this allows also making different SPD entries, for example, for different known NATs' outer addresses.		Not support		NAT traversal is out of the scope
	3094	After the SPD lookup, the server will do traffic selector narrowing based on the SPD entry it found. It will again use the already-substituted traffic selectors, and it will thus send back		Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		traffic selectors having IPN1 and IP2 as their IP addresses; it can still narrow down the protocol number or port ranges used by the traffic selectors. The SAD entry created for the Child SA will have the addresses as seen by the server, namely IPN1 and IP2.				
	3102	When the client receives the server's response to the Child SA, it will do similar processing. If the transport mode SA was created, the client can store the original returned traffic selectors as original source and destination addresses. It will replace the IP addresses in the traffic selectors with the ones from the IP header of the IKE packet: it will replace IPN1 with IP1 and IP2 with IPN2. Then it will use those traffic selectors when verifying the SA against sent traffic selectors, and when installing the SAD entry.		Not support		NAT traversal is out of the scope
	3111	A summary of the rules for NAT-traversal in transport mode is:		Not support		NAT traversal is out of the scope
	3113	For the client proposing transport mode:		Not support		NAT traversal is out of the scope
	3115	- The TSi entries **MUST** have exactly one IP address, and that **MUST** match the source address of the IKE SA.	MUST MUST	Not support		NAT traversal is out of the scope
	3118	- The TSr entries **MUST** have exactly one IP address, and that **MUST** match the destination address of the IKE SA.	MUST MUST	Not support		NAT traversal is out of the scope
	3121	- The first TSi and TSr traffic selectors **SHOULD** have very specific traffic selectors including protocol and port numbers, such as from the packet triggering the request.	SHOULD	Not support		NAT traversal is out of the scope
	3125	- There **MAY** be multiple TSi and TSr entries.	MAY	Not support		NAT traversal is out of the scope
	3127	- If transport mode for the SA was selected (that is, if the server included USE_TRANSPORT_MODE notification in its response):		Not support		NAT traversal is out of the scope
	3130	- Store the original traffic selectors as the received source and destination address.		Not support		NAT traversal is out of the scope
	3133	- If the server is behind a NAT, substitute the IP address in the TSr entries with the remote address of the IKE SA.		Not support		NAT traversal is out of the scope
	3136	- If the client is behind a NAT, substitute the IP address in the TSi entries with the local address of the IKE SA.		Not support		NAT traversal is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3139	- Do address substitution before using those traffic selectors for anything else other than storing original content of them. This includes verification that traffic selectors were narrowed correctly by other end, creation of the SAD entry, and so on.		Not support		NAT traversal is out of the scope
	3145	For the responder, when transport mode is proposed by client:		Not support		NAT traversal is out of the scope
	3147	- Store the original traffic selector IP addresses as received source and destination address, both in case we need to undo address substitution, and to use as the "real source and destination address" specified by [UDPENCAPS], and for TCP/UDP checksum fixup.		Not support		NAT traversal is out of the scope
	3152	- If the client is behind a NAT, substitute the IP address in the TSi entries with the remote address of the IKE SA.		Not support		NAT traversal is out of the scope
	3155	- If the server is behind a NAT substitute the IP address in the TSr entries with the local address of the IKE SA.		Not support		NAT traversal is out of the scope
	3158	- Do PAD and SPD lookup using the ID and substituted traffic selectors.		Not support		NAT traversal is out of the scope
	3161	- If no SPD entry was found, or if found SPD entry does not allow transport mode, undo the traffic selector substitutions. Do PAD and SPD lookup again using the ID and original traffic selectors, but also searching for tunnel mode SPD entry (that is, fall back to tunnel mode).		Not support		NAT traversal is out of the scope
	3167	- However, if a transport mode SPD entry was found, do normal traffic selection narrowing based on the substituted traffic selectors and SPD entry. Use the resulting traffic selectors when creating SAD entries, and when sending traffic selectors back to the client.		Not support		NAT traversal is out of the scope
	3173	2.24. Explicit Congestion Notification (ECN)				
	3175	When IPsec tunnels behave as originally specified in [IPSECARCH-OLD], ECN usage is not appropriate for the outer IP headers because tunnel decapsulation processing discards ECN congestion indications to the detriment of the network. ECN support for IPsec tunnels for IKEv1-based IPsec requires multiple operating modes and negotiation (see [ECN]).		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3180	IKEv2 simplifies this situation by requiring that ECN be usable in the outer IP headers of all tunnel-mode Child SAs created by IKEv2.		Not support		Explanation
	3182	Specifically, tunnel encapsulators and decapsulators for all tunnel-mode SAs created by IKEv2 **MUST** support the ECN full-functionality option for tunnels specified in [ECN] and **MUST** implement the tunnel encapsulation and decapsulation processing specified in [IPSECARCH] to prevent discarding of ECN congestion indications.	MUST MUST	Not support		ECN is out of the scope
	3189	2.25. Exchange Collisions				
	3191	Because IKEv2 exchanges can be initiated by either peer, it is possible that two exchanges affecting the same SA partly overlap. This can lead to a situation where the SA state information is temporarily not synchronized, and a peer can receive a request that it cannot process in a normal fashion.		Not support		Explanation
	3197	Obviously, using a window size greater than 1 leads to more complex situations, especially if requests are processed out of order. This section concentrates on problems that can arise even with a window size of 1, and recommends solutions.		Not support		Explanation
	3202	A TEMPORARY_FAILURE notification **SHOULD** be sent when a peer receives a request that cannot be completed due to a temporary condition such as a rekeying operation.	SHOULD	Not support		Explanation
	3204	When a peer receives a TEMPORARY_FAILURE notification, it **MUST NOT** immediately retry the operation; it **MUST** wait so that the sender may complete whatever operation caused the temporary condition.	MUST NOT MUST	Not support		test condition is ambiguous (immediately)
	3207	The recipient **MAY** retry the request one or more times over a period of several minutes.	MAY	Not support		Explanation
	3208	If a peer continues to receive TEMPORARY_FAILURE on the same IKE SA after several minutes, it **SHOULD** conclude that the state information is out-of-sync and close the IKE SA.	SHOULD	Not support		test condition is ambiguous (several minutes)
	3213	A CHILD_SA_NOT_FOUND notification **SHOULD** be sent when a peer receives a request to rekey a Child SA that does not exist. The SA that the initiator attempted to rekey is indicated by the	SHOULD	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		SPI field in the Notify Payload, which is copied from the SPI field in the REKEY_SA notification.				
	3217	A peer that receives a CHILD_SA_NOT_FOUND notification **SHOULD** silently delete the Child SA (if it still exists) and send a request to create a new Child SA from scratch (if the Child SA does not yet exist).	SHOULD	Not support		untestable (it is difficult to send rekey request toward deleted CHILD SA)
	3222	2.25.1. Collisions While Rekeying or Closing Child SAs				
	3224	If a peer receives a request to rekey a Child SA that it is currently trying to close, it **SHOULD** reply with TEMPORARY_FAILURE.	SHOULD	Not support		untestable (it is difficult to send delete request for CHILD SA)
	3225	If a peer receives a request to rekey a Child SA that it is currently rekeying, it **SHOULD** reply as usual, and **SHOULD** prepare to close redundant SAs later based on the nonces (see Section 2.8.1).	SHOULD SHOULD	BASIC	Both	EN.I.1.2.6.3 SGW.I.1.2.6.3
	3228	If a peer receives a request to rekey a Child SA that does not exist, it **SHOULD** reply with CHILD_SA_NOT_FOUND.	SHOULD	Not support		untestable (it is difficult to make an environment with CHILD SA deleted)
	3232	If a peer receives a request to close a Child SA that it is currently trying to close, it **SHOULD** reply without Delete payloads (see Section 1.4.1).	SHOULD	Not Support		untestable (it is difficult to send delete request for CHILD SA)
	3234	If a peer receives a request to close a Child SA that it is currently rekeying, it **SHOULD** reply as usual, with a Delete payload.	SHOULD	BASIC		[EN.R.P69.L3234.ADD] [SGW.R.P69.L3234.ADD]
	3236	If a peer receives a request to close a Child SA that does not exist, it **SHOULD** reply without Delete payloads.	SHOULD	Not support		untestable (it is difficult to

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						send delete request for CHILD SA)
	3239	If a peer receives a request to rekey the IKE SA, and it is currently creating, rekeying, or closing a Child SA of that IKE SA, it **SHOULD** reply with TEMPORARY_FAILURE.	SHOULD	Not support		untestable (it is difficult to send delete request or rekey request for CHILD SA)
	3243	2.25.2. Collisions While Rekeying or Closing IKE SAs				
	3245	If a peer receives a request to rekey an IKE SA that it is currently rekeying, it **SHOULD** reply as usual, and **SHOULD** prepare to close redundant SAs and move inherited Child SAs later based on the nonces (see Section 2.8.2).	SHOULD SHOULD	BASIC	Both	EN.I.1.2.6.6 SGW.I.1.2.6.6
	3248	If a peer receives a request to rekey an IKE SA that it is currently trying to close, it **SHOULD** reply with TEMPORARY_FAILURE.	SHOULD	Not support		untestable (it is difficult to send delete request for IKE SA)
	3252	If a peer receives a request to close an IKE SA that it is currently rekeying, it **SHOULD** reply as usual, and forget about its own rekeying request.	SHOULD	BASIC		[EN.R.P69.L3252.ADD] [SGW.R.P69.L3252.ADD]
	3254	If a peer receives a request to close an IKE SA that it is currently trying to close, it **SHOULD** reply as usual, and forget about its own close request.	SHOULD	Not Support		untestable (it is difficult to send delete request for IKE SA)
	3258	If a peer receives a request to create or rekey a Child SA when it is currently rekeying the IKE SA, it **SHOULD** reply with TEMPORARY_FAILURE.	SHOULD	BASIC		[EN.R.P69.L3258.ADD] [SGW.R.P69.L3258.ADD]
	3260	If a peer receives a request to delete a Child SA when it is currently rekeying the IKE SA, it **SHOULD** reply as usual, with a	SHOULD	BASIC		[EN.R.P69.L3260.ADD]

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Delete payload.				[SGW.R.P69.L3260.ADD]
	3265	3. Header and Payload Formats				
	3267	In the tables in this section, some cryptographic primitives and configuration attributes are marked as "UNSPECIFIED". These are items for which there are no known specifications and therefore interoperability is currently impossible. A future specification may describe their use, but until such specification is made, implementations **SHOULD NOT** attempt to use items marked as "UNSPECIFIED" in implementations that are meant to be interoperable.	SHOULD NOT	Not support		Explanation
	3275	3.1. The IKE Header				
	3277	IKE messages use UDP ports 500 and/or 4500, with one IKE message per UDP datagram. Information from the beginning of the packet through the UDP header is largely ignored except that the IP addresses and UDP ports from the headers are reversed and used for return packets.		Not support		Explanation
	3281	When sent on UDP port 500, IKE messages begin immediately following the UDP header.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3282	When sent on UDP port 4500, IKE messages have prepended four octets of zero.		Not support		Explanation
	3283	These four octets of zero are not part of the IKE message and are not included in any of the length fields or checksums defined by IKE. Each IKE message begins with the IKE header, denoted HDR in this document. Following the header are one or more IKE payloads each identified by a "Next Payload" field in the preceding payload. Payloads are identified in the order in which they appear in an IKE message by looking in the "Next Payload" field in the IKE header, and subsequently according to the "Next Payload" field in the IKE payload itself until a "Next Payload" field of zero indicates that no payloads follow.		Not support		Explanation
	3292	If a payload of type "Encrypted" is found, that payload is decrypted and its contents parsed as additional payloads.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3294	An Encrypted payload **MUST** be the last payload in a packet and an Encrypted payload **MUST NOT** contain another Encrypted payload.	MUST MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1
	3298	The responder's SPI in the header identifies an instance of an IKE security association. It is therefore possible for a single instance of IKE to multiplex distinct sessions with multiple peers, including multiple sessions per peer.		Not support		Explanation
	3303	All multi-octet fields representing integers are laid out in big endian order (also known as "most significant byte first", or "network byte order").		Not support		Explanation
	3307	The format of the IKE header is shown in Figure 4.		Not support		Explanation

<table>
<tr><th colspan="2">Section</th><th rowspan="2">Sentence</th><th>RFC</th><th rowspan="2">Test
Requirements</th><th rowspan="2">Target</th><th rowspan="2">Comments</th></tr>
<tr><th>page</th><th>line</th><th>requirement</th></tr>
<tr><td></td><td>3309</td><td>1 2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+
| IKE SA Initiator's SPI
|
|
|
+-+
| IKE SA Responder's SPI
|
|
|
+-+
| Next Payload | MjVer | MnVer | Exchange Type | Flags
|
+-+
| Message ID
|
+-+
| Length
|
+-+

Figure 4: IKE Header Format</td><td></td><td>BASIC</td><td>EN(initiator)
EN(responder)
SGW(initiator)
SGW(responder)</td><td>EN.I.1.1.1.1
EN.I.1.1.1.2
EN.I.1.1.1.3
EN.I.1.2.1.1
EN.I.2.1.1.1
EN.I.2.1.1.2
EN.R.1.1.1.1
EN.R.1.1.1.2
EN.R.1.1.1.3
EN.R.1.2.1.1
EN.R.1.3.1.1
EN.R.2.1.1.1
EN.R.2.1.1.2
SGW.I.1.1.1.1
SGW.I.1.1.1.2
SGW.I.1.1.1.3
SGW.I.1.2.1.1
SGW.I.2.1.1.1
SGW.I.2.1.1.2
SGW.R.1.1.1.1
SGW.R.1.1.1.2
SGW.R.1.1.1.3
SGW.R.1.2.1.1
SGW.R.1.3.1.1
SGW.R.2.1.1.1
SGW.R.2.1.1.2</td></tr>
</table>

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3327	o Initiator's SPI (8 octets) - A value chosen by the initiator to identify a unique IKE security association. This value **MUST NOT** be zero.	MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3331	o Responder's SPI (8 octets) - A value chosen by the responder to identify a unique IKE security association. This value **MUST** be zero in the first message of an IKE Initial Exchange (including repeats of that message including a cookie).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3336	o Next Payload (1 octet) - Indicates the type of payload that immediately follows the header. The format and value of each payload are defined below.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3340	o Major Version (4 bits) - Indicates the major version of the IKE protocol in use. Implementations based on this version of IKE **MUST** set the Major Version to 2.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3342	Implementations based on previous versions of IKE and ISAKMP **MUST** set the Major Version to 1.	MUST	Not support		Out of scope
	3344	Implementations based on this version of IKE **MUST** reject or ignore messages containing a version number greater than 2 with an INVALID_MAJOR_VERSION notification message as described in Section 2.5.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.2 SGW.R.1.1.4.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3349	o Minor Version (4 bits) - Indicates the minor version of the IKE protocol in use. Implementations based on this version of IKE **MUST** set the Minor Version to 0.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3351	They **MUST** ignore the minor version number of received messages.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.1 SGW.R.1.1.4.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3354	o Exchange Type (1 octet) - Indicates the type of exchange being used. This constrains the payloads sent in each message in an exchange.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3356	The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3362	Exchange Type Value --------------------------------- IKE_SA_INIT 34 IKE_AUTH 35 CREATE_CHILD_SA 36 INFORMATIONAL 37		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3369	o Flags (1 octet) - Indicates specific options that are set for the message. Presence of options is indicated by the appropriate bit in the flags field being set. The bits are as follows:		Not support		Explanation
	3373	+-+-+-+-+-+-+-+-+ |X|X|R|V|I|X|X|X| +-+-+-+-+-+-+-+-+		Not support		Explanation
	3377	In the description below, a bit being 'set' means its value is '1', while 'cleared' means its value is '0'.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3378	"X" bits **MUST** be cleared when sending	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3379	and **MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.1 EN.I.1.1.11.2 EN.I.1.2.7.1 EN.R.1.1.11.1 EN.R.1.1.11.2 EN.R.1.2.9.1 EN.R.1.3.3.1 SGW.I.1.1.11.1 SGW.I.1.1.11.2 SGW.I.1.2.7.1 SGW.R.1.1.11.1 SGW.R.1.1.11.2 SGW.R.1.2.9.1 SGW.R.1.3.3.1
	3381	* R (Response) - This bit indicates that this message is a response to a message containing the same message ID. This bit **MUST** be cleared in all request messages	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3383	and **MUST** be set in all responses.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3384	An IKE endpoint **MUST NOT** generate a response to a message that is marked as being a response (with one exception; see Section 2.21.2).	MUST NOT	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3388	* V (Version) - This bit indicates that the transmitter is capable of speaking a higher major version number of the protocol than the one indicated in the major version number field. Implementations of IKEv2 **MUST** clear this bit when sending and **MUST** ignore it in incoming messages.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.3 EN.R.1.1.11.3 SGW.I.1.1.11.3 SGW.R.1.1.11.3
	3394	* I (Initiator) - This bit **MUST** be set in messages sent by the original initiator of the IKE SA	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3395	and **MUST** be cleared in messages sent by the original responder. It is used by the recipient to determine which eight octets of the SPI were generated by the recipient. This bit changes to reflect who initiated the last rekey of the IKE SA.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3401	o Message ID (4 octets, unsigned integer) - Message identifier used to control retransmission of lost packets and matching of requests and responses. It is essential to the security of the protocol because it is used to prevent message replay attacks. See Section 2.1 and Section 2.2.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3407	o Length (4 octets, unsigned integer) - Length of total message (header + payloads) in octets.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3410	3.2. Generic Payload Header				
	3412	Each IKE payload defined in Section 3.3 through Section 3.16 begins with a generic payload header, shown in Figure 5. Figures for each payload below will include the generic payload header, but for brevity the description of each field will be omitted.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3417	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ Figure 5: Generic Payload Header		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3425	The Generic Payload Header fields are defined as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3427	o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0. This field provides a "chaining" capability whereby additional payloads can be added to a message by appending each one to the end of the message and setting the "Next Payload" field of the preceding payload to indicate the new payload's type. An Encrypted payload, which must always be the last payload of a message, is an exception. It contains data structures in the format of additional payloads.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3435	In the header of an Encrypted payload, the Next Payload field is set to the payload type of the first contained payload (instead of 0);		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3438	conversely, the Next Payload field of the last contained payload is set to zero). The payload type values are listed here. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	3446	Next Payload Type Notation Value ---		Not support		Explanation
	3448	No Next Payload 0		Not support		Explanation

Section page	Section line	Sentence	RFC requirement	Test Requirements	Target	Comments
	3449	Security Association SA 33 Key Exchange KE 34 Identification - Initiator IDi 35 Identification - Responder IDr 36 Certificate CERT 37 Certificate Request CERTREQ 38 Authentication AUTH 39 Nonce Ni, Nr 40 Notify N 41 Delete D 42 Vendor ID V 43 Traffic Selector - Initiator TSi 44 Traffic Selector - Responder TSr 45 Encrypted and Authenticated SK 46 Configuration CP 47 Extensible Authentication EAP 48		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3466	(Payload type values 1-32 should not be assigned in the future so that there is no overlap with the code assignments for IKEv1.)		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3470	o Critical (1 bit) - **MUST** be set to zero if the sender wants the recipient to skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload. **MUST** be set to one if the sender wants the recipient to reject this entire message if it does not understand the payload type.	MUST MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2
	3475	**MUST** be ignored by the recipient if the recipient understands the payload type code.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.4 SGW.R.1.1.4.4
	3476	**MUST** be set to zero for payload types defined in this document.	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3477	Note that the critical bit applies to the current payload rather than the "next" payload whose type code appears in the first octet. The reasoning behind not setting the critical bit for payloads defined in this document is that all implementations **MUST** understand all payload types defined in this document and therefore must ignore the Critical bit's value. Skipped payloads are expected to have valid Next Payload and Payload Length fields. See Section 2.5 for more information on this bit.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3487	o RESERVED (7 bits) - **MUST** be sent as zero;	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3487	**MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.1 EN.I.1.1.11.2 EN.I.1.2.7.1 EN.R.1.1.11.1 EN.R.1.1.11.2 EN.R.1.2.9.1 EN.R.1.3.3.1 SGW.I.1.1.11.1 SGW.I.1.1.11.2 SGW.I.1.2.7.1 SGW.R.1.1.11.1 SGW.R.1.1.11.2 SGW.R.1.2.9.1 SGW.R.1.3.3.1
	3490	o Payload Length (2 octets, unsigned integer) - Length in octets of the current payload, including the generic payload header.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3493	Many payloads contain fields marked as "RESERVED". Some payloads in IKEv2 (and historically in IKEv1) are not aligned to 4-octet boundaries.		Not support		Explanation
	3497	3.3. Security Association Payload				
	3499	The Security Association Payload, denoted SA in this document, is used to negotiate attributes of a security association. Assembly of Security Association Payloads requires great peace of mind.		Not support		Explanation
	3501	An SA payload MAY contain multiple proposals. If there is more than one, they MUST be ordered from most preferred to least preferred. Each proposal contains a single IPsec protocol (where a protocol is IKE, ESP, or AH), each protocol MAY contain multiple transforms, and each transform MAY contain multiple attributes.	MAY MUST	ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.6.4 EN.I.1.1.6.6 EN.I.1.2.3.4 EN.I.1.2.3.5 EN.I.1.2.4.4 EN.I.1.2.4.5 SGW.I.1.1.6.4 SGW.I.1.1.6.6 SGW.I.1.2.3.4 SGW.I.1.2.3.5

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.1.2.4.4 SGW.I.1.2.4.5
	3506	When parsing an SA, an implementation **MUST** check that the total Payload Length is consistent with the payload's internal lengths and counts.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.4 EN.R.1.1.6.6 EN.R.1.2.5.4 EN.R.1.2.6.5 EN.R.1.2.6.6 SGW.R.1.1.6.4 SGW.R.1.1.6.6 SGW.R.1.2.5.4 SGW.R.1.2.6.5 SGW.R.1.2.6.6
	3508	Proposals, Transforms, and Attributes each have their own variable length encodings. They are nested such that the Payload Length of an SA includes the combined contents of the SA, Proposal, Transform, and Attribute information. The length of a Proposal includes the lengths of all Transforms and Attributes it contains. The length of a Transform includes the lengths of all Attributes it contains.		Not support		Explanation
	3516	The syntax of Security Associations, Proposals, Transforms, and Attributes is based on ISAKMP; however the semantics are somewhat different. The reason for the complexity and the hierarchy is to allow for multiple possible combinations of algorithms to be encoded in a single SA. Sometimes there is a choice of multiple algorithms, whereas other times there is a combination of algorithms. For example, an initiator might want to propose using ESP with either (3DES and HMAC_MD5) or (AES and HMAC_SHA1).		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3525	One of the reasons the semantics of the SA payload has changed from ISAKMP and IKEv1 is to make the encodings more compact in common cases.		Not support		Explanation
	3529	The Proposal structure contains within it a Proposal Num and an IPsec protocol ID.		Not support		Explanation
	3530	Each structure **MUST** have a proposal number one (1) greater than the previous structure.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.4 EN.I.1.1.6.6 EN.I.1.2.3.5 EN.I.1.2.4.4 EN.R.1.1.6.4 EN.R.1.1.6.6 EN.R.1.2.5.4 EN.R.1.2.6.6 SGW.I.1.1.6.4 SGW.I.1.1.6.6 SGW.I.1.2.3.5 SGW.I.1.2.4.4 SGW.R.1.1.6.4 SGW.R.1.1.6.6 SGW.R.1.2.5.4 SGW.R.1.2.6.6
	3531	The first Proposal in the initiator's SA payload **MUST** have a Proposal Num of one (1).	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3532	One reason to use multiple proposals is to propose both standard crypto ciphers and combined-mode ciphers. Combined-mode ciphers include both integrity and encryption in a single encryption algorithm, and **MUST** either offer no integrity algorithm or a single integrity algorithm of "none", with no integrity algorithm being the RECOMMENDED method.	MUST	Not support		Explanation
	3538	If an initiator wants to propose both combined-mode ciphers and normal ciphers, it must include two proposals: one will have all the combined-mode ciphers, and the other will have all the normal ciphers with the integrity algorithms.		Not support		combined-mode is out of the scope
	3541	For example, one such proposal would have two proposal structures. Proposal 1 is ESP with AES-128, AES-192, and AES-256 bits in Cipher Block Chaining (CBC) mode, with either HMAC-SHA1-96 or XCBC-96 as the integrity algorithm; Proposal 2 is AES-128 or AES-256 in GCM mode with an 8-octet ICV. Both proposals allow but do not require the use of ESN (extended sequence numbers). This can be illustrated as:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3549	SA Payload | +--- Proposal #1 (Proto ID = ESP(3), SPI size = 4, | | 7 transforms, SPI = 0x052357bb) | | | +-- Transform ENCR (Name = ENCR_AES_CBC) | | +-- Attribute (Key Length = 128) | | | +-- Transform ENCR (Name = ENCR_AES_CBC) | | +-- Attribute (Key Length = 192) | | | +-- Transform ENCR (Name = ENCR_AES_CBC) | | +-- Attribute (Key Length = 256) | | | +-- Transform INTEG (Name = AUTH_HMAC_SHA1_96) | +-- Transform INTEG (Name = AUTH_AES_XCBC_96) | +-- Transform ESN (Name = ESNs) | +-- Transform ESN (Name = No ESNs) | +--- Proposal #2 (Proto ID = ESP(3), SPI size = 4, | 4 transforms, SPI = 0x35a1d6f2) | +-- Transform ENCR (Name = AES-GCM with a 8 octet ICV) | +-- Attribute (Key Length = 128) | +-- Transform ENCR (Name = AES-GCM with a 8 octet ICV) | +-- Attribute (Key Length = 256) | +-- Transform ESN (Name = ESNs) +-- Transform ESN (Name = No ESNs)		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3580	Each Proposal/Protocol structure is followed by one or more transform structures. The number of different transforms is generally determined by the Protocol. AH generally has two transforms: Extended Sequence Numbers (ESNs) and an integrity check algorithm. ESP generally has three: ESN, an encryption algorithm, and an integrity check algorithm. IKE generally has four transforms: a Diffie-Hellman group, an integrity check algorithm, a PRF algorithm, and an encryption algorithm.		Not support		Explanation
	3587	For each Protocol, the set of permissible transforms is assigned transform ID numbers, which appear in the header of each transform.		Not support		Explanation
	3591	If there are multiple transforms with the same Transform Type, the proposal is an OR of those transforms. If there are multiple Transforms with different Transform Types, the proposal is an AND of the different groups.		Not support		Explanation
	3594	For example, to propose ESP with (3DES or AES-CBC) and (HMAC_MD5 or HMAC_SHA), the ESP proposal would contain two Transform Type 1 candidates (one for 3DES and one for AEC-CBC) and two Transform Type 3 candidates (one for HMAC_MD5 and one for HMAC_SHA). This effectively proposes four combinations of algorithms. If the initiator wanted to propose only a subset of those, for example (3DES and HMAC_MD5) or (IDEA and HMAC_SHA), there is no way to encode that as multiple transforms within a single Proposal. Instead, the initiator would have to construct two different Proposals, each with two transforms.		Not support		Explanation
	3605	A given transform **MAY** have one or more Attributes. Attributes are necessary when the transform can be used in more than one way, as when an encryption algorithm has a variable key size. The transform would specify the algorithm and the attribute would specify the key size. Most transforms do not have attributes.	MAY	Not support		Not need to test
	3609	A transform **MUST NOT** have multiple attributes of the same type.	MUST NOT	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.6.1 SGW.I.1.1.6.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3610	To propose alternate values for an attribute (for example, multiple key sizes for the AES encryption algorithm), an implementation **MUST** include multiple Transforms with the same Transform Type each with a single Attribute.	MUST	Not support		Only AES with 128bit key is "BASIC". AES with other length keys are out of scope.
	3615	Note that the semantics of Transforms and Attributes are quite different from those in IKEv1. In IKEv1, a single Transform carried multiple algorithms for a protocol with one carried in the Transform and the others carried in the Attributes.		Not support		Explanation
	3620	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | | ~ <Proposals> ~ | | +-+ Figure 6: Security Association Payload		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3632	o Proposals (variable) - One or more proposal substructures.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3634	The payload type for the Security Association Payload is thirty three (33).		Not support		Explanation
	3637	3.3.1. Proposal Substructure				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3639	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | 0 (last) or 2 | RESERVED | Proposal Length | +-+ | Proposal Num | Protocol ID | SPI Size |Num Transforms| +-+ ~ SPI (variable) ~ +-+ | | ~ <Transforms> ~ | | +-+ Figure 7: Proposal Substructure		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3655	o 0 (last) or 2 (more) (1 octet) - Specifies whether this is the last Proposal Substructure in the SA.		BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.6.4 EN.I.1.1.6.6 EN.I.1.2.3.5 EN.I.1.2.4.4 SGW.I.1.1.6.4 SGW.I.1.1.6.6 SGW.I.1.2.3.5 SGW.I.1.2.4.4
	3656	This syntax is inherited from ISAKMP, but is unnecessary because the last Proposal could be identified from the length of the SA. The value (2) corresponds to a Payload Type of Proposal in IKEv1,		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		and the first four octets of the Proposal structure are designed to look somewhat like the header of a Payload.				
	3663	o RESERVED (1 octet) - **MUST** be sent as zero;	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3663	**MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.1 EN.I.1.1.11.2 EN.I.1.2.7.1 EN.R.1.1.11.1 EN.R.1.1.11.2 EN.R.1.2.9.1 EN.R.1.3.3.1 SGW.I.1.1.11.1 SGW.I.1.1.11.2 SGW.I.1.2.7.1 SGW.R.1.1.11.1 SGW.R.1.1.11.2 SGW.R.1.2.9.1 SGW.R.1.3.3.1
	3666	o Proposal Length (2 octets, unsigned integer) - Length of this proposal, including all transforms and attributes that follow.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3669	o Proposal Num (1 octet) - When a proposal is made, the first proposal in an SA payload **MUST** be 1,		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3670	and subsequent proposals **MUST** be one more than the previous proposal (indicating an OR of the two proposals).	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.6.4 EN.I.1.1.6.6 EN.I.1.2.3.5 EN.I.1.2.4.4 SGW.I.1.1.6.4 SGW.I.1.1.6.6 SGW.I.1.2.3.5 SGW.I.1.2.4.4
	3672	When a proposal is accepted, the proposal number in the SA payload **MUST** match the number on the proposal sent that was accepted.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.4 EN.R.1.1.6.6 EN.R.1.2.5.4 EN.R.1.2.6.6 SGW.R.1.1.6.4 SGW.R.1.1.6.6 SGW.R.1.2.5.4 SGW.R.1.2.6.6

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3676	o Protocol ID (1 octet) - Specifies the IPsec protocol identifier for the current negotiation.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3678	The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3683	Protocol Protocol ID ----------------------------------- IKE 1 AH 2 ESP 3		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3689	o SPI Size (1 octet) - For an initial IKE SA negotiation, this field **MUST** be zero;	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3690	the SPI is obtained from the outer header. During subsequent negotiations, it is equal to the size, in octets, of the SPI of the corresponding protocol (8 for IKE, 4 for ESP and AH).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3695	o Num Transforms (1 octet) - Specifies the number of transforms in this proposal.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.3 EN.I.1.1.6.6 EN.R.1.1.6.4 EN.R.1.1.6.6 SGW.I.1.1.6.3 SGW.I.1.1.6.6 SGW.R.1.1.6.4 SGW.R.1.1.6.6

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3698	o SPI (variable) - The sending entity's SPI. Even if the SPI Size is not a multiple of 4 octets, there is no padding applied to the payload. When the SPI Size field is zero, this field is not present in the Security Association payload.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3703	o Transforms (variable) - One or more transform substructures.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3705	3.3.2. Transform Substructure				

<table>
<tr><th colspan="2">Section</th><th rowspan="2">Sentence</th><th>RFC</th><th rowspan="2">Test
Requirements</th><th rowspan="2">Target</th><th rowspan="2">Comments</th></tr>
<tr><th>page</th><th>line</th><th>requirement</th></tr>
<tr><td></td><td>3707</td><td>1 2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+
| 0 (last) or 3 | RESERVED | Transform Length
|
+-+
|Transform Type | RESERVED | Transform ID
|
+-+
|
|
~ Transform Attributes
~
|
|
+-+

Figure 8: Transform Substructure</td><td></td><td>BASIC</td><td>EN(initiator)
EN(responder)
SGW(initiator)
SGW(responder)</td><td>EN.I.1.1.1.1
EN.I.1.1.1.2
EN.I.1.1.1.3
EN.I.1.2.1.1
EN.I.2.1.1.1
EN.I.2.1.1.2
EN.R.1.1.1.1
EN.R.1.1.1.2
EN.R.1.1.1.3
EN.R.1.2.1.1
EN.R.2.1.1.1
EN.R.2.1.1.2
SGW.I.1.1.1.1
SGW.I.1.1.1.2
SGW.I.1.1.1.3
SGW.I.1.2.1.1
SGW.I.2.1.1.1
SGW.I.2.1.1.2
SGW.R.1.1.1.1
SGW.R.1.1.1.2
SGW.R.1.1.1.3
SGW.R.1.2.1.1
SGW.R.2.1.1.1
SGW.R.2.1.1.2</td></tr>
</table>

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3721	o 0 (last) or 3 (more) (1 octet) - Specifies whether this is the last Transform Substructure in the Proposal.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3722	This syntax is inherited from ISAKMP, but is unnecessary because the last transform could be identified from the length of the proposal.		Not support		Explanation
	3725	The value (3) corresponds to a Payload Type of Transform in IKEv1, and the first four octets of the Transform structure are designed to look somewhat like the header of a Payload.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3729	o RESERVED - **MUST** be sent as zero;	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3729	**MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.1 EN.I.1.1.11.2 EN.I.1.2.7.1 EN.R.1.1.11.1 EN.R.1.1.11.2 EN.R.1.2.9.1 EN.R.1.3.3.1 SGW.I.1.1.11.1 SGW.I.1.1.11.2 SGW.I.1.2.7.1 SGW.R.1.1.11.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.11.2 SGW.R.1.2.9.1 SGW.R.1.3.3.1
	3731	o Transform Length - The length (in octets) of the Transform Substructure including Header and Attributes.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3734	o Transform Type (1 octet) - The type of transform being specified in this transform. Different protocols support different transform types. For some protocols, some of the transforms may be optional.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3737	If a transform is optional and the initiator wishes to propose that the transform be omitted, no transform of the given type is included in the proposal.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3739	If the initiator wishes to make use of the transform optional to the responder, it includes a transform substructure with transform ID = 0 as one of the options.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3744	o Transform ID (2 octets) - The specific instance of the transform type being proposed.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3747	The transform type values are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3753	Description Trans. Type Used In -- Encryption Algorithm (ENCR) 1 IKE and ESP Pseudorandom Function (PRF) 2 IKE Integrity Algorithm (INTEG) 3 IKE*, AH, optional in ESP Diffie-Hellman Group (D-H) 4 IKE, optional in AH & ESP Extended Sequence Numbers (ESN) 5 AH and ESP		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3762	(*) Negotiating an integrity algorithm is mandatory for the Encrypted payload format specified in this document. For example, [AEAD] specifies additional formats based on authenticated encryption, in which a separate integrity algorithm is not negotiated.		Not support		Explanation
	3768	For Transform Type 1 (Encryption Algorithm), the Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3774	Name Number Defined In ---		Not support		Explanation
	3776	ENCR_DES_IV64 1 (UNSPECIFIED)		Not support		Explanation
	3777	ENCR_DES 2 (RFC2405), [DES]		Not support		Explanation
	3778	ENCR_3DES 3 (RFC2451)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3779	ENCR_RC5 4 (RFC2451)		Not support		Explanation
	3780	ENCR_IDEA 5 (RFC2451), [IDEA]		Not support		Explanation
	3781	ENCR_CAST 6 (RFC2451)		Not support		Explanation
	3782	ENCR_BLOWFISH 7 (RFC2451)		Not support		Explanation
	3783	ENCR_3IDEA 8 (UNSPECIFIED)		Not support		Explanation
	3784	ENCR_DES_IV32 9 (UNSPECIFIED)		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3785	ENCR_NULL 11 (RFC2410)		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.2 EN.R.1.1.6.2 SGW.I.1.1.6.2 SGW.R.1.1.6.2
	3786	ENCR_AES_CBC 12 (RFC3602)		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3787	ENCR_AES_CTR 13 (RFC3686)		ADVANCED *Only for CHILD_SA	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.2 EN.R.1.1.6.2 SGW.I.1.1.6.2 SGW.R.1.1.6.2
	3789	For Transform Type 2 (Pseudorandom Function), the Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	3795	Name Number Defined In ---		Not support		Explanation
	3797	PRF_HMAC_MD5 1 (RFC2104), [MD5]		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3798	PRF_HMAC_SHA1 2 (RFC2104), [SHA]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3799	PRF_HMAC_TIGER 3 (UNSPECIFIED)		Not support		Explanation
	3801	For Transform Type 3 (Integrity Algorithm), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	3807	Name Number Defined In --		Not support		Explanation
	3809	NONE 0		Not support		Explanation
	3810	AUTH_HMAC_MD5_96 1 (RFC2403)		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3811	AUTH_HMAC_SHA1_96 2 (RFC2404)		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3812	AUTH_DES_MAC 3 (UNSPECIFIED)		Not support		Explanation
	3813	AUTH_KPDK_MD5 4 (UNSPECIFIED)		Not support		Explanation
	3814	AUTH_AES_XCBC_96 5 (RFC3566)		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3816	For Transform Type 4 (Diffie-Hellman Group), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	3822	Name Number Defined in --		Not support		Explanation
	3824	NONE 0		Not support		Explanation
	3825	768-Bit MODP 1 Appendix B 1024-Bit MODP 2 Appendix B		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3827	1536-bit MODP 5 [ADDGROUP]		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3828	2048-bit MODP 14 [ADDGROUP]		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3829	3072-bit MODP 15 [ADDGROUP]		Not support		Explanation
	3830	4096-bit MODP 16 [ADDGROUP]		Not support		Explanation
	3831	6144-bit MODP 17 [ADDGROUP]		Not support		Explanation
	3832	8192-bit MODP 18 [ADDGROUP]		Not support		Explanation
	3834	Although ESP and AH do not directly include a Diffie-Hellman exchange, a Diffie-Hellman group **MAY** be negotiated for the Child SA. This allows the peers to employ Diffie-Hellman in the CREATE_CHILD_SA exchange, providing perfect forward secrecy for the generated Child SA keys.	MAY	Not support		Explanation
	3840	For Transform Type 5 (Extended Sequence Numbers), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	3847	Name Number --		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3849	No Extended Sequence Numbers 0		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3850	Extended Sequence Numbers 1		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3852	Note that an initiator who supports ESNs will usually include two ESN transforms, with values "0" and "1", in its proposals. A proposal containing a single ESN transform with value "1" means		ADVANCED		EN.I.1.1.1.2 SGW.I.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		that using normal (non-extended) sequence numbers is not acceptable.				
	3857	Numerous additional transform types have been defined since the publication of RFC 4306. Please refer to the IANA IKEv2 registry for details.		Not support		Explanation
	3861	3.3.3. Valid Transform Types by Protocol				
	3863	The number and type of transforms that accompany an SA payload are dependent on the protocol in the SA itself. An SA payload proposing the establishment of an SA has the following mandatory and optional transform types.		Not support		Explanation
	3866	A compliant implementation **MUST** understand all mandatory and optional types for each protocol it supports (though it need not accept proposals with unacceptable suites).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3868	A proposal **MAY** omit the optional types if the only value for them it will accept is NONE.	MAY	Not support		Not need to test
	3872	Protocol Mandatory Types Optional Types -- IKE ENCR, PRF, INTEG*, D-H ESP ENCR, ESN INTEG, D-H AH INTEG, ESN D-H		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3878	(*) Negotiating an integrity algorithm is mandatory for the Encrypted payload format specified in this document. For example, [AEAD] specifies additional formats based on authenticated encryption, in which a separate integrity algorithm is not negotiated.		Not support		Explanation
	3884	3.3.4. Mandatory Transform IDs				
	3886	The specification of suites that **MUST** and **SHOULD** be supported for interoperability has been removed from this document because they are likely to change more rapidly than this document evolves.	MUST SHOULD	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3888	At the time of publication of this document, [RFC4307] specifies these suites, but note that it might be updated in the future, and other RFCs might specify different sets of suites.		Not support		Explanation
	3893	An important lesson learned from IKEv1 is that no system should only implement the mandatory algorithms and expect them to be the best choice for all customers.		Not support		Explanation
	3897	It is likely that IANA will add additional transforms in the future, and some users may want to use private suites, especially for IKE where implementations should be capable of supporting different parameters, up to certain size limits.		Not support		Explanation
	3900	In support of this goal, all implementations of IKEv2 **SHOULD** include a management facility that allows specification (by a user or system administrator) of Diffie-Hellman parameters (the generator, modulus, and exponent lengths and values) for new Diffie-Hellman groups.	SHOULD	Not support		No need to test
	3904	Implementations **SHOULD** provide a management interface through which these parameters and the associated transform IDs may be entered (by a user or system administrator), to enable negotiating such groups.	SHOULD	Not support		No need to test
	3909	All implementations of IKEv2 **MUST** include a management facility that enables a user or system administrator to specify the suites that are acceptable for use with IKE.	MUST	Not support		No need to test
	3911	Upon receipt of a payload with a set of transform IDs, the implementation **MUST** compare the transmitted transform IDs against those locally configured via the management controls, to verify that the proposed suite is acceptable based on local policy.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.2.4.1 SGW.R.1.2.4.1
	3915	The implementation **MUST** reject SA proposals that are not authorized by these IKE suite controls.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.2.4.1 SGW.R.1.2.4.1
	3916	Note that cryptographic suites that **MUST** be implemented need not be configured as acceptable to local policy.	MUST	Not support		Internal process
	3920	3.3.5. Transform Attributes				
	3922	Each transform in a Security Association payload may include attributes that modify or complete the specification of the transform. The set of valid attributes depends on the transform.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Currently, only a single attribute type is defined: the Key Length attribute is used by certain encryption transforms with variable-length keys (see below for details).				
	3929	The attributes are type/value pairs and are defined below. Attributes can have a value with a fixed two-octet length or a variable-length value. For the latter, the attribute is encoded as type/length/value.		Not support		Explanation
	3934	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ |A| Attribute Type | AF=0 Attribute Length | |F| | AF=1 Attribute Value | +-+ | AF=0 Attribute Value | | AF=1 Not Transmitted | +-+ Figure 9: Data Attributes		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3946	o Attribute Format (AF) (1 bit) - Indicates whether the data attribute follows the Type/Length/Value (TLV) format or a shortened Type/Value (TV) format. If the AF bit is zero (0), then the attribute uses TLV format; if the AF bit is one (1), the TV format (with two-byte value) is used.		Not support		Explanation
	3952	o Attribute Type (15 bits) - Unique identifier for each type of attribute (see below).		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.6.2
	3955	o Attribute Value (variable length) - Value of the Attribute associated with the Attribute Type. If the AF bit is a zero (0), this field has a variable length defined by the Attribute Length field.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3958	If the AF bit is a one (1), the Attribute Value has a length of 2 octets.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3961	The only currently defined attribute type (Key Length) is fixed length; the variable-length encoding specification is included only for future extensions.		Not support		Explanation
	3963	Attributes described as fixed length **MUST NOT** be encoded using the variable-length encoding unless that length exceeds two bytes.	MUST NOT	Not support		Internal process
	3965	Variable-length attributes **MUST NOT** be encoded as fixed-length even if their value can fit into two octets.	MUST NOT	Not support		Internal process
	3966	NOTE: This is a change from IKEv1, where increased flexibility may have simplified the composer of messages but certainly complicated the parser.		Not support		Explanation
	3971	The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Readers should refer to [IKEV2IANA] for the latest values.				
	3976	Attribute Type Value Attribute Format -- Key Length (in bits) 14 TV		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2
	3980	Values 0-13 and 15-17 were used in a similar context in IKEv1, and should not be assigned except to matching values.		Not support		Explanation
	3983	The Key Length attribute specifies the key length in bits (**MUST** use network byte order) for certain transforms as follows:	MUST	ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.1 EN.I.1.1.6.2 EN.R.1.1.6.1 EN.R.1.1.6.2 SGW.I.1.1.6.1 SGW.I.1.1.6.2 SGW.R.1.1.6.1 SGW.R.1.1.6.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	3986	o The Key Length attribute **MUST NOT** be used with transforms that use a fixed length key.	MUST NOT	ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	3987	For example, this includes ENCR_DES, ENCR_IDEA, and all the Type 2 (Pseudorandom function) and Type 3 (Integrity Algorithm) transforms specified in this document. It is recommended that future Type 2 or 3 transforms do not use this attribute.		Not support		Explanation
	3993	o Some transforms specify that the Key Length attribute **MUST** be always included (omitting the attribute is not allowed, and proposals not containing it **MUST** be rejected). For example, this includes ENCR_AES_CBC and ENCR_AES_CTR.	MUST MUST	Not support		Explanation
	3998	o Some transforms allow variable-length keys, but also specify a default key length if the attribute is not included. For example, these transforms include ENCR_RC5 and ENCR_BLOWFISH.		Not support		ENCR_RC5 and ENCR_BLOWFISH are out of the

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						scope
	4002	Implementation note: To further interoperability and to support upgrading endpoints independently, implementers of this protocol **SHOULD** accept values that they deem to supply greater security.	SHOULD	Not support		Explanation
	4004	For instance, if a peer is configured to accept a variable-length cipher with a key length of X bits and is offered that cipher with a larger key length, the implementation **SHOULD** accept the offer if it supports use of the longer key.	SHOULD	ADVANCED		[Added] newly additional test for responder in RFC 5996
	4010	Support for this capability allows a responder to express a concept of "at least" a certain level of security -- "a key length of _at least_X bits for cipher Y". However, as the attribute is always returned unchanged (see the next section), an initiator willing to accept multiple key lengths has to include multiple transforms with the same Transform Type, each with a different Key Length attribute.		Not support		Explanation
	4017	3.3.6. Attribute Negotiation				
	4019	During security association negotiation initiators present offers to responders.		Not support		Explanation
	4020	Responders **MUST** select a single complete set of parameters from the offers (or reject all offers if none are acceptable).	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.3 EN.R.1.1.6.4 EN.R.1.1.6.5 EN.R.1.1.6.6 EN.R.1.2.5.3 EN.R.1.2.5.4 SGW.R.1.1.6.3 SGW.R.1.1.6.4 SGW.R.1.1.6.5 SGW.R.1.1.6.6 SGW.R.1.2.5.3 SGW.R.1.2.5.4
	4022	If there are multiple proposals, the responder **MUST** choose a single proposal.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.4 EN.R.1.1.6.6 SGW.R.1.1.6.4 SGW.R.1.1.6.6

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4023	If the selected proposal has multiple Transforms with the same type, the responder **MUST** choose a single one.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.3 EN.R.1.1.6.5 SGW.R.1.1.6.3 SGW.R.1.1.6.5
	4025	Any attributes of a selected transform **MUST** be returned unmodified.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.6.3 EN.R.1.1.6.5 SGW.R.1.1.6.3 SGW.R.1.1.6.5
	4026	The initiator of an exchange **MUST** check that the accepted offer is consistent with one of its proposals, and if not **MUST** terminate the exchange.	MUST MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.6.9 EN.I.1.1.6.10 SGW.I.1.1.6.9 SGW.I.1.1.6.10
	4030	If the responder receives a proposal that contains a Transform Type it does not understand, or a proposal that is missing a mandatory Transform Type, it **MUST** consider this proposal unacceptable; however, other proposals in the same SA payload are processed as usual.	MUST	BASIC		[EN.R.P86.L403 0.ADD.1] [SGW.R.P86.L40 30.ADD.1] [EN.R.P86.L403 0.ADD.2] [EN.R.P86.L403 0.ADD.2]
	4034	Similarly, if the responder receives a transform that it does not understand, or one that contains a Transform Attribute it does not understand, it **MUST** consider this transform unacceptable; other transforms with the same Transform Type are processed as usual. This allows new Transform Types and Transform Attributes to be defined in the future.	MUST	BASIC		[EN.R.P86.L403 4.ADD.1] [SGW.R.P86.L40 34.ADD.1] [EN.R.P86.L403 4.ADD.2] [EN.R.P86.L403 4.ADD.2]
	4041	Negotiating Diffie-Hellman groups presents some special challenges. SA offers include proposed attributes and a Diffie-Hellman public number (KE) in the same message.		Not support		Explanation
	4043	If in the initial exchange the initiator offers to use one of several Diffie-Hellman groups, it **SHOULD** pick the one the responder is most likely to accept and include a KE corresponding to that group.	SHOULD	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.1.1 SGW.I.1.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4046	If the responder selects a proposal using a different Diffie-Hellman group (other than NONE), the responder will indicate the correct group in the response and the initiator **SHOULD** pick an element of that group for its KE value when retrying the first message.	SHOULD	SHOULD ADVANCED *Because DH#14 is ADVANCED group.	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.7 EN.R.1.1.6.7 SGW.I.1.1.6.7 SGW.R.1.1.6.7
	4050	It **SHOULD**, however, continue to propose its full supported set of groups in order to prevent a man-in-the-middle downgrade attack.	SHOULD	Not support		Explanation
	4052	If one of the proposals offered is for the Diffie-Hellman group of NONE, and the responder selects that Diffie-Hellman group, then it **MUST** ignore the initiator's KE payload and omit the KE payload from the response.	MUST	Not support		the Diffie-Hellman group of NONE is out of the scope
	4057	3.4. Key Exchange Payload				
	4059	The Key Exchange Payload, denoted KE in this document, is used to exchange Diffie-Hellman public numbers as part of a Diffie-Hellman key exchange. The Key Exchange Payload consists of the IKE generic payload header followed by the Diffie-Hellman public value itself.		Not support		Explanation
	4064	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Diffie-Hellman Group Num | RESERVED | +-+ | | ~ Key Exchange Data ~ |		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		| +-+ Figure 10: Key Exchange Payload Format				
	4078	A key exchange payload is constructed by copying one's Diffie-Hellman public value into the "Key Exchange Data" portion of the payload.		Not support		Explanation
	4080	The length of the Diffie-Hellman public value for modular exponentiation group (MODP) groups **MUST** be equal to the length of the prime modulus over which the exponentiation was performed, prepending zero bits to the value if necessary.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	4085	The Diffie-Hellman Group Num identifies the Diffie-Hellman group in which the Key Exchange Data was computed (see Section 3.3.2).		Not support		Explanation
	4086	This Diffie-Hellman Group Num **MUST** match a Diffie-Hellman Group specified in a proposal in the SA payload that is sent in the same message, and SHOULD match the Diffie-Hellman group in the first group in the first proposal, if such exists.	MUST SHOULD	ADVANCED	Both	EN.R.1.1.6.4 SGW.R.1.1.6.4 [Changed] EN.I.1.1.6.4 SGW.I.1.1.6.4 Observing point is added for initiator to send a packet which is consistent between first group and group number.

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4090	If none of the proposals in that SA payload specifies a Diffie-Hellman Group, the KE payload **MUST NOT** be present.	MUST NOT	BASIC	Both	EN.I.1.1.1.3 EN.R.1.1.1.3 SGW.I.1.1.1.3 SGW.R.1.1.1.3
	4092	If the selected proposal uses a different Diffie-Hellman group (other than NONE), the message **MUST** be rejected with a Notify payload of type INVALID_KE_PAYLOAD. See also Sections 1.2 and 2.7.	MUST	ADVANCED *Because DH#14 is ADVANCED group.	EN(responder) SGW(responder)	EN.R.1.1.6.7 SGW.R.1.1.6.7
	4097	The payload type for the Key Exchange payload is thirty-four (34).		Not support		Explanation
	4099	3.5. Identification Payloads				
	4101	The Identification Payloads, denoted IDi and IDr in this document, allow peers to assert an identity to one another. This identity may be used for policy lookup, but does not necessarily have to match anything in the CERT payload; both fields may be used by an implementation to perform access control decisions.		Not support		Explanation
	4105	When using the ID_IPV4_ADDR/ID_IPV6_ADDR identity types in IDi/IDr payloads, IKEv2 does not require this address to match the address in the IP header of IKEv2 packets, or anything in the TSi/TSr payloads. The contents of IDi/IDr are used purely to fetch the policy and authentication data related to the other party.		Not support		Explanation
	4112	NOTE: In IKEv1, two ID payloads were used in each direction to hold Traffic Selector (TS) information for data passing over the SA. In IKEv2, this information is carried in TS payloads (see Section 3.13).		Not support		Explanation
	4116	The Peer Authorization Database (PAD) as described in RFC 4301 [IPSECARCH] describes the use of the ID payload in IKEv2 and provides a formal model for the binding of identity to policy in addition to providing services that deal more specifically with the details of policy enforcement. The PAD is intended to provide a link between the SPD and the IKE security association management. See Section 4.4.3 of RFC 4301 for more details.		Not support		Explanation
	4124	The Identification Payload consists of the IKE generic payload		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		header followed by identification fields as follows:				
	4127	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | ID Type | RESERVED | +-+ | | ~ Identification Data ~ | | +-+ Figure 11: Identification Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4141	o ID Type (1 octet) - Specifies the type of Identification being used.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4144	o RESERVED - **MUST** be sent as zero;	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4144	**MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.2 EN.R.1.1.11.2 SGW.I.1.1.11.2 SGW.R.1.1.11.2
	4146	o Identification Data (variable length) - Value, as indicated by the Identification Type. The length of the Identification Data is computed from the size in the ID payload header.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4150	The payload types for the Identification Payload are thirty-five (35) for IDi		Not support		Explanation
	4151	and thirty-six (36) for IDr.		Not support		Explanation
	4153	The following table lists the assigned semantics for the Identification Type field. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	4160	ID Type Value ---		Not support		Explanation
	4162	ID_IPV4_ADDR 1 A single four (4) octet IPv4 address.		Not support		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4165	ID_FQDN 2 A fully-qualified domain name string. An example of an ID_FQDN is, "example.com". The string **MUST NOT** contain any terminators (e.g., NULL, CR, etc.). All characters in the ID_FQDN are ASCII; for an "internationalized domain name", the syntax is as defined in [IDNA], for example "xn--tmonesimerkki-bfbb.example.net".	MUST NOT	ADVANCED *Only with RSA-DSS auth.	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4172	ID_RFC822_ADDR 3 A fully-qualified RFC 822 email address string. An example of a ID_RFC822_ADDR is, "jsmith@example.com". The string **MUST NOT** contain any terminators. Because of [EAI], implementations would be wise to treat this field as UTF-8 encoded text, not as pure ASCII.	MUST NOT	ADVANCED *Only with RSA-DSS auth.	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4179	ID_IPV6_ADDR 5 A single sixteen (16) octet IPv6 address.		BASIC (receiving)	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1
				BASIC (sending)	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.1
	4182	ID_DER_ASN1_DN 9 The binary Distinguished Encoding Rules (DER) encoding of an ASN.1 X.500 Distinguished Name [PKIX].		Not support		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4186	ID_DER_ASN1_GN 10 The binary DER encoding of an ASN.1 X.509 GeneralName [PKIX].		Not support		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4189	ID_KEY_ID 11 An opaque octet stream that may be used to pass vendor-specific information necessary to do certain proprietary types of identification.		Not support		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4194	Two implementations will interoperate only if each can generate a type of ID acceptable to the other.		Not support		Explanation
	4195	To assure maximum interoperability, implementations **MUST** be configurable to send at least one of ID_IPV4_ADDR, ID_FQDN, ID_RFC822_ADDR, or ID_KEY_ID,	MUST	Not support *However ID_FQDN and		Not support except ID_IPV6_ADDR, FQDN and

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
				RFC822_ADDR are available only with RSA-DSS auth.		RFC822_ADDR,
	4197	and **MUST** be configurable to accept all of these four types.	MUST	Not support *However ID_FQDN and RFC822_ADDR are available only with RSA-DSS auth.		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4199	Implementations **SHOULD** be capable of generating and accepting all of these types.	SHOULD	Not support *However ID_FQDN and RFC822_ADDR are available only with RSA-DSS auth.		Not support except ID_IPV6_ADDR, FQDN and RFC822_ADDR,
	4200	IPv6-capable implementations **MUST** additionally be configurable to accept ID_IPV6_ADDR. IPv6-only implementations **MAY** be configurable to send only ID_IPV6_ADDR instead of ID_IPV4_ADDR for IP addresses.	MUST MAY	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.2.1.1.1 EN.R.1.1.1.2 EN.R.2.1.1.1 SGW.I.1.1.1.2 SGW.I.2.1.1.1 SGW.R.1.1.1.2 SGW.R.2.1.1.
	4205	EAP [EAP] does not mandate the use of any particular type of identifier, but often EAP is used with Network Access Identifiers (NAIs) defined in [NAI]. Although NAIs look a bit like email addresses (e.g., "joe@example.com"), the syntax is not exactly the same as the syntax of email address in [MAILFORMAT]. For those NAIs that include the realm component, the ID_RFC822_ADDR identification type **SHOULD** be used.	SHOULD	Not support		EAP authentication is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Responder implementations should not attempt to verify that the contents actually conform to the exact syntax given in [MAILFORMAT], but instead should accept any reasonable-looking NAI.				
	4214	For NAIs that do not include the realm component, the ID_KEY_ID identification type **SHOULD** be used.	SHOULD	Not support		EAP authentication is out of the scope
	4217	3.6. Certificate Payload				
	4219	The Certificate Payload, denoted CERT in this document, provides a means to transport certificates or other authentication-related information via IKE.		Not support		Explanation
	4221	Certificate payloads SHOULD be included in an exchange if certificates are available to the sender. The Hash and URL formats of the Certificate payloads should be used in case the peer has indicated an ability to retrieve this information from elsewhere using an HTTP_CERT_LOOKUP_SUPPORTED Notify payload.		Not support		Explanation
	4225	Note that the term "Certificate Payload" is somewhat misleading, because not all authentication mechanisms use certificates and data other than certificates may be passed in this payload.		Not support		Explanation
	4230	The Certificate Payload is defined as follows:		Not support		Explanation
	4232	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Cert Encoding | | +-+-+-+-+-+-+-+-+ | ~ Certificate Data ~ |		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		| +-+ Figure 12:　Certificate Payload Format				
	4245	o　Certificate Encoding (1 octet) - This field indicates the type of certificate or certificate-related information contained in the Certificate Data field.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4247	The values in the following table are only current as of the publication date of RFC 4306.　Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	4253	Certificate Encoding　Value --		Not support		Explanation
	4255	PKCS #7 wrapped X.509 certificate　1　UNSPECIFIED		Not support		Explanation
	4256	PGP Certificate　2　UNSPECIFIED		Not support		Explanation
	4257	DNS Signed Key　3　UNSPECIFIED		Not support		Explanation
	4258	X.509 Certificate - Signature　4		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4259	Kerberos Token　6　UNSPECIFIED		Not support		Explanation
	4260	Certificate Revocation List (CRL)　7		Not support		Explanation
	4261	Authority Revocation List (ARL)　8　UNSPECIFIED		Not support		Explanation
	4262	SPKI Certificate　9　UNSPECIFIED		Not support		Explanation
	4263	X.509 Certificate - Attribute　10　UNSPECIFIED		Not support		Explanation
	4264	Raw RSA Key　11		Not support		Explanation
	4265	Hash and URL of X.509 certificate　12		Not support		Explanation
	4266	Hash and URL of X.509 bundle　13		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4268	o Certificate Data (variable length) - Actual encoding of certificate data. The type of certificate is indicated by the Certificate Encoding field.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4272	The payload type for the Certificate Payload is thirty-seven (37).		Not support		Explanation
	4274	Specific syntax for some of the certificate type codes above is not defined in this document. The types whose syntax is defined in this document are:		Not support		Explanation
	4278	o "X.509 Certificate - Signature" contains a DER-encoded X.509 certificate whose public key is used to validate the sender's AUTH payload.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4280	Note that with this encoding, if a chain of certificates needs to be sent, multiple CERT payloads are used, only the first of which holds the public key used to validate the sender's AUTH payload.		Not support		Explanation
	4285	o "Certificate Revocation List" contains a DER-encoded X.509 certificate revocation list.		Not support		Explanation
	4288	o "Raw RSA Key" contains a PKCS #1 encoded RSA key, that is, a DER-encoded RSAPublicKey structure (see [RSA] and [PKCS1]).		Not support		Explanation
	4291	o Hash and URL encodings allow IKE messages to remain short by replacing long data structures with a 20-octet SHA-1 hash (see [SHA]) of the replaced value followed by a variable-length URL that resolves to the DER-encoded data structure itself.		Not support		Explanation
	4294	This improves efficiency when the endpoints have certificate data cached and makes IKE less subject to DoS attacks that become easier to mount when IKE messages are large enough to require IP fragmentation [DOSUDPPROT].		Not support		Explanation
	4300	The "Hash and URL of a bundle" type uses the following ASN.1 definition for the X.509 bundle:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		

	4303	CertBundle
		{ iso(1) identified-organization(3) dod(6) internet(1)
		security(5) mechanisms(5) pkix(7) id-mod(0)
		id-mod-cert-bundle(34) }
		
		DEFINITIONS EXPLICIT TAGS ::=
		BEGIN
		
		IMPORTS
		Certificate, CertificateList
		FROM PKIX1Explicit88
		{ iso(1) identified-organization(3) dod(6)
		internet(1) security(5) mechanisms(5) pkix(7)
		id-mod(0) id-pkix1-explicit(18) } ;
		
		CertificateOrCRL ::= CHOICE {
		cert [0] Certificate,
		crl [1] CertificateList }
		
		CertificateBundle ::= SEQUENCE OF CertificateOrCRL
		
		END
				Not support		Explanation

	4326	Implementations **MUST** be capable of being configured to send and accept up to four X.509 certificates in support of authentication, and also **MUST** be capable of being configured to send and accept the Hash and URL format (with HTTP URLs).	MUST MUST	Not support		Internal process
	4329	Implementations **SHOULD** be capable of being configured to send and accept Raw RSA keys.	SHOULD	Not support		Internal process
	4330	If multiple certificates are sent, the first certificate **MUST** contain the public key used to sign the AUTH payload. The other certificates may be sent in any order.	MUST	Not support		Difficult to test
	4335	Implementations **MUST** support the HTTP [HTTP] method for hash-and-URL lookup. The behavior of other URL methods [URLS] is not currently specified, and such methods **SHOULD NOT** be used in the absence of a document specifying them.	MUST SHOULD NOT	ADVANCED		[Added] newly additional test for responder in

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						RFC 5996 (Authentication with RSA Digital Signature is ADVANCED)
	4340	3.7. Certificate Request Payload				
	4342	The Certificate Request Payload, denoted CERTREQ in this document, provides a means to request preferred certificates via IKE and can appear in the IKE_INIT_SA response and/or the IKE_AUTH request. Certificate Request payloads **MAY** be included in an exchange when the sender needs to get the certificate of the receiver.	MAY	Not support		Not need to test
	4348	The Certificate Request Payload is defined as follows:		Not support		Explanation
	4350	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Cert Encoding | | +-+-+-+-+-+-+-+-+ | ~ Certification Authority ~ | | +-+ Figure 13: Certificate Request Payload Format		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4363	o Certificate Encoding (1 octet) - Contains an encoding of the type or format of certificate requested. Values are listed in Section 3.6.		ADVANCED	EN(initiator) EN(responder) SGW(initiator)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
					SGW(responder)	SGW.R.1.1.10.3
	4367	o Certification Authority (variable length) - Contains an encoding of an acceptable certification authority for the type of certificate requested.		ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3
	4371	The payload type for the Certificate Request Payload is thirty-eight (38).		Not support		Explanation
	4374	The Certificate Encoding field has the same values as those defined in Section 3.6. The Certification Authority field contains an indicator of trusted authorities for this certificate type. The Certification Authority value is a concatenated list of SHA-1 hashes of the public keys of trusted Certification Authorities (CAs). Each is encoded as the SHA-1 hash of the Subject Public Key Info element (see section 4.1.2.7 of [PKIX]) from each Trust Anchor certificate. The 20-octet hashes are concatenated and included with no other formatting.		Not support		Explanation
	4384	The contents of the "Certification Authority" field are defined only for X.509 certificates, which are types 4, 12, and 13. Other values **SHOULD NOT** be used until standards-track specifications that specify their use are published.	SHOULD NOT	Not support		Explanation
	4389	Note that the term "Certificate Request" is somewhat misleading, in that values other than certificates are defined in a "Certificate" payload and requests for those values can be present in a Certificate Request Payload. The syntax of the Certificate Request payload in such cases is not defined in this document.		Not support		Explanation
	4395	The Certificate Request Payload is processed by inspecting the "Cert Encoding" field to determine whether the processor has any certificates of this type. If so, the "Certification Authority" field is inspected to determine if the processor has any certificates that can be validated up to one of the specified certification authorities. This can be a chain of certificates.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4402	If an end-entity certificate exists that satisfies the criteria specified in the CERTREQ, a certificate or certificate chain SHOULD be sent back to the certificate requestor if the recipient of the CERTREQ: o is configured to use certificate authentication, o is allowed to send a CERT payload, o has matching CA trust policy governing the current negotiation, and o has at least one time-wise and usage-appropriate end-entity certificate chaining to a CA provided in the CERTREQ.	SHOULD	Not support		CERTREQ is "not support"
	4417	Certificate revocation checking must be considered during the chaining process used to select a certificate. Note that even if two peers are configured to use two different CAs, cross-certification relationships should be supported by appropriate selection logic.		Not support		Explanation
	4422	The intent is not to prevent communication through the strict adherence of selection of a certificate based on CERTREQ, when an alternate certificate could be selected by the sender that would still enable the recipient to successfully validate and trust it through trust conveyed by cross-certification, CRLs, or other out-of-band configured means. Thus, the processing of a CERTREQ should be seen as a suggestion for a certificate to select, not a mandated one. If no certificates exist, then the CERTREQ is ignored. This is not an error condition of the protocol. There may be cases where there is a preferred CA sent in the CERTREQ, but an alternate might be acceptable (perhaps after prompting a human operator).		Not support		Explanation
	4434	The HTTP_CERT_LOOKUP_SUPPORTED notification **MAY** be included in any message that can include a CERTREQ payload and indicates that the sender is capable of looking up certificates based	MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		on an HTTP-based URL (and hence presumably would prefer to receive certificate specifications in that format).				
	4440	3.8. Authentication Payload				
	4442	The Authentication Payload, denoted AUTH in this document, contains data used for authentication purposes. The syntax of the Authentication data varies according to the Auth Method as specified below.		Not support		Explanation
	4447	The Authentication Payload is defined as follows:		Not support		Explanation
	4449	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Auth Method | RESERVED | +-+ | | ~ Authentication Data ~ | | +-+ Figure 14: Authentication Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4463	o Auth Method (1 octet) - Specifies the method of authentication used. The types of signatures are listed here.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.I.1.1.10.3 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 EN.R.1.1.10.3 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.I.1.1.10.3 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2 SGW.R.1.1.10.3
	4464	The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	4470	Mechanism Value --		Not support		Explanation
	4472	RSA Digital Signature 1 Computed as specified in Section 2.15 using an RSA private key with RSASSA-PKCS1-v1_5 signature scheme specified in [PKCS1] (implementers should note that IKEv1 used a different method for RSA signatures). To promote interoperability, implementations that support this type **SHOULD** support signatures that use SHA-1 as the hash function and **SHOULD** use SHA-1 as the default hash function when generating signatures. Implementations can use the certificates received from a given	SHOULD SHOULD	ADVANCED	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.10.3 EN.R.1.1.10.3 SGW.I.1.1.10.3 SGW.R.1.1.10.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		peer as a hint for selecting a mutually understood hash function for the AUTH payload signature. Note, however, that the hash algorithm used in the AUTH payload signature doesn't have to be the same as any hash algorithm(s) used in the certificate(s).				
	4486	Shared Key Message Integrity Code 2 Computed as specified in Section 2.15 using the shared key associated with the identity in the ID payload and the negotiated PRF.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4491	DSS Digital Signature 3 Computed as specified in Section 2.15 using a DSS private key (see [DSS]) over a SHA-1 hash.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4495	o Authentication Data (variable length) - see Section 2.15.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.I.1.1.10.3 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.2.1.1.1 EN.R.2.1.1.2 EN.R.1.1.10.3 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.I.1.1.10.3 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.2.1.1.1 SGW.R.2.1.1.2 SGW.R.1.1.10.3
	4497	The payload type for the Authentication Payload is thirty-nine (39).		Not support		Explanation
	4499	3.9. Nonce Payload				
	4501	The Nonce Payload, denoted as Ni and Nr in this document for the initiator's and responder's nonce, respectively, contains random data used to guarantee liveness during an exchange and protect against replay attacks.		Not support		Explanation
	4506	The Nonce Payload is defined as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4508	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | | ~ Nonce Data ~ | | +-+ Figure 15: Nonce Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	4520	o Nonce Data (variable length) - Contains the random data generated by the transmitting entity.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	4523	The payload type for the Nonce Payload is forty (40).		Not support		Explanation
	4525	The size of the Nonce Data **MUST** be between 16 and 256 octets, inclusive.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	4526	Nonce values **MUST NOT** be reused.	MUST NOT	Not support		Difficult to test
	4528	3.10. Notify Payload				
	4530	The Notify Payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify Payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL Exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4538	The Notify Payload is defined as follows:		Not support		Explanation
	4540	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Protocol ID | SPI Size | Notify Message Type | +-+ | | ~ Security Parameter Index (SPI) ~ | | +-+ | | ~ Notification Data ~ | | +-+ Figure 16: Notify Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4558	o Protocol ID (1 octet) - If this notification concerns an existing SA whose SPI is given in the SPI field, this field indicates the type of that SA.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4560	For notifications concerning Child SAs, this field **MUST** contain either	MUST	Not support		Explanation
	4561	(2) to indicate AH	MUST	Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4561	or (3) to indicate ESP.	MUST	Not support		Explanation
	4562	Of the notifications defined in this document, the SPI is included only with INVALID_SELECTORS and REKEY_SA.		Not support		Explanation
	4563	If the SPI field is empty, this field **MUST** be sent as zero	MUST	Not support		Explanation
	4564	and **MUST** be ignored on receipt.	MUST	Not support		Explanation
	4567	o SPI Size (1 octet) - Length in octets of the SPI as defined by the IPsec protocol ID or zero if no SPI is applicable. For a notification concerning the IKE SA, the SPI Size **MUST** be zero and the field must be empty.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4572	o Notify Message Type (2 octets) - Specifies the type of notification message.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4575	o SPI (variable length) - Security Parameter Index.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4577	o Notification Data (variable length) - Status or error data transmitted in addition to the Notify Message Type. Values for this field are type specific (see below).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4581	The payload type for the Notify Payload is forty-one (41).		Not support		Explanation
	4583	3.10.1. Notify Message Types				
	4585	Notification information can be error messages specifying why an SA could not be established. It can also be status data that a process managing an SA database wishes to communicate with a peer process. The table below lists the Notification messages and their corresponding values. The number of different error statuses was greatly reduced from IKEv1 both for simplification and to avoid giving configuration information to probers.		Not support		Explanation
	4593	Types in the range 0 - 16383 are intended for reporting errors.		Not support		Explanation
	4593	An implementation receiving a Notify payload with one of these types that it does not recognize in a response **MUST** assume that	MUST	BASIC	EN(initiator) SGW(initiator)	EN.I.1.1.11.4 SGW.I.1.1.11.4

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		the corresponding request has failed entirely.				
	4596	Unrecognized error types in a request and status types in a request or response **MUST** be ignored, and they should be logged.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.5 EN.R.1.1.11.5 SGW.I.1.1.11.5 SGW.R.1.1.11.5
	4600	Notify payloads with status types **MAY** be added to any message and **MUST** be ignored if not recognized. They are intended to indicate capabilities, and as part of SA negotiation, are used to negotiate non-cryptographic parameters.	MAY MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.5 EN.R.1.1.11.5 SGW.I.1.1.11.5 SGW.R.1.1.11.5
	4605	More information on error handling can be found in Section 2.21.		Not support		Explanation
	4607	The values in the following table are only current as of the publication date of RFC 4306, plus two error types added in this document. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	4613	NOTIFY messages: error types Value --		Not support		Explanation
	4615	UNSUPPORTED_CRITICAL_PAYLOAD 1 See Section 2.5.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.4 SGW.R.1.1.4.4
	4618	INVALID_IKE_SPI 4 See Section 2.21.		Not support		untestable
	4621	INVALID_MAJOR_VERSION 5 See Section 2.5.		BASIC	SGW(initiator) EN(responder)	EN.R.1.1.4.2 SGW.I.1.1.4.2
	4624	INVALID_SYNTAX 7 Indicates the IKE message that was received was invalid because some type, length, or value was out of range or because the request was rejected for policy reasons.		Not support		untestable
	4627	To avoid a DoS attack using forged messages, this status may only be returned for and in an encrypted packet if the message ID and cryptographic checksum were valid.		Not support		Explanation
	4630	To avoid leaking information to someone probing a node, this status **MUST** be sent in response to any error not covered by one of the other status types.	MUST	Not support		untestable

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4633	To aid debugging, more detailed error information should be written to a console or log.		Not support		Internal process
	4636	INVALID_MESSAGE_ID 9 See Section 2.3.		Not support		Explanation
	4639	INVALID_SPI 11 See Section 1.5.		Not support		untestable
	4642	NO_PROPOSAL_CHOSEN 14 None of the proposed crypto suites was acceptable.		BASIC	EN(responder) SGW(responder)	EN.R.1.2.4.1 EN.R.1.2.6.9 SGW.R.1.2.4.1 SGW.R.1.2.6.9
	4643	This can be sent in any case where the offered proposals (including but not limited to SA payload values, USE_TRANSPORT_MODE notify, IPCOMP_SUPPORTED notify) are not acceptable for the responder. This can also be used as "generic" Child SA error when Child SA cannot be created for some other reason. See also Section 2.7.		Not support		Explanation
	4650	INVALID_KE_PAYLOAD 17 See Section 1.2 and 1.3.		ADVANCED *Because DH#14 is ADVANCED group.	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.6.7 EN.R.1.1.6.7 SGW.I.1.1.6.7 SGW.R.1.1.6.7
	4653	AUTHENTICATION_FAILED 24 Sent in the response to an IKE_AUTH message when for some reason the authentication failed. There is no associated data. See also Section 2.21.2.		Not support		Explanation
	4658	SINGLE_PAIR_REQUIRED 34 See Section 2.9.		Not support		Explanation
	4661	NO_ADDITIONAL_SAS 35 See Section 1.3.		Not support		Explanation
	4664	INTERNAL_ADDRESS_FAILURE 36 See Section 3.15.4.		Not support		Explanation
	4667	FAILED_CP_REQUIRED 37 See Section 2.19.		ADVANCED	SGW(responder)	SGW.R.2.1.2.3

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4670	TS_UNACCEPTABLE 38 See Section 2.9.		BASIC	EN(responder) SGW(responder)	EN.R.1.1.7.2 SGW.R.1.1.7.2
	4673	INVALID_SELECTORS 39 **MAY** be sent in an IKE INFORMATIONAL exchange when a node receives an ESP or AH packet whose selectors do not match those of the SA on which it was delivered (and that caused the packet to be dropped).	MAY	Not support		untestable
	4677	The Notification Data contains the start of the offending packet (as in ICMP messages) and the SPI field of the notification is set to match the SPI of the Child SA.		Not support		Explanation
	4681	TEMPORARY_FAILURE 43 See section 2.25.		Not support		Explanation
	4684	CHILD_SA_NOT_FOUND 44 See section 2.25.		Not Support		Explanation
	4689	NOTIFY messages: status types Value ---		Not support		Explanation
	4691	INITIAL_CONTACT 16384 See Section 2.4.		Not support		Explanation
	4694	SET_WINDOW_SIZE 16385 See Section 2.3.		Not support		SET_WINDOW_SIZE is out of the scope
	4697	ADDITIONAL_TS_POSSIBLE 16386 See Section 2.9.		Not support		Explanation
	4700	IPCOMP_SUPPORTED 16387 See Section 2.22.		Not support		Explanation
	4703	NAT_DETECTION_SOURCE_IP 16388 See Section 2.23.		Not support		Explanation
	4706	NAT_DETECTION_DESTINATION_IP 16389 See Section 2.23.		Not support		Explanation
	4709	COOKIE 16390 See Section 2.6.		ADVANCED	EN(initiator) SGW(initiator)	EN.I.1.1.5.1 SGW.I.1.1.5.1
	4712	USE_TRANSPORT_MODE 16391 See Section 1.3.1.		BASIC	EN(initiator) EN(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4715	HTTP_CERT_LOOKUP_SUPPORTED 16392 See Section 3.6.		Not support		Not need to test
	4718	REKEY_SA 16393 See Section 1.3.3.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.1.2.1.1 EN.R.1.2.1.1 SGW.I.1.2.1.1 SGW.R.1.2.1.1
	4721	ESP_TFC_PADDING_NOT_SUPPORTED 16394 See Section 1.3.1.		Not support		Explanation
	4724	NON_FIRST_FRAGMENTS_ALSO 16395 See Section 1.3.1.		Not support		Explanation
	4727	3.11. Delete Payload				
	4729	The Delete Payload, denoted D in this document, contains a protocol-specific security association identifier that the sender has removed from its security association database and is, therefore, no longer valid. Figure 17 shows the format of the Delete Payload.		Not support		Explanation
	4732	It is possible to send multiple SPIs in a Delete payload; however, each SPI **MUST** be for the same protocol.	MUST	Not support		sending test difficult
	4734	Mixing of protocol identifiers **MUST NOT** be performed in the Delete payload.	MUST NOT	Not support		Not need to test
	4735	It is permitted, however, to include multiple Delete payloads in a single INFORMATIONAL exchange where each Delete payload lists SPIs for a different protocol.		Not support		Explanation
	4739	Deletion of the IKE SA is indicated by a protocol ID of 1 (IKE) but no SPIs. Deletion of a Child SA, such as ESP or AH, will contain the IPsec protocol ID of that protocol (2 for AH, 3 for ESP), and the SPI is the SPI the sending endpoint would expect in inbound ESP or AH packets.		Not support		Explanation
	4745	The Delete Payload is defined as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4747	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Protocol ID | SPI Size | Num of SPIs | +-+ | | ~ Security Parameter Index(es) (SPI) ~ | | +-+ Figure 17: Delete Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.6 EN.R.1.1.3.7 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.6 SGW.R.1.1.3.7 SGW.R.1.1.3.8 SGW.R.1.1.3.9
	4761	o Protocol ID (1 octet) - Must be 1 for an IKE SA, 2 for AH, or 3 for ESP.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.6 EN.R.1.1.3.7 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.6 SGW.R.1.1.3.7 SGW.R.1.1.3.8 SGW.R.1.1.3.9

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4764	o SPI Size (1 octet) - Length in octets of the SPI as defined by the protocol ID. It **MUST** be zero for IKE (SPI is in message header) or four for AH and ESP.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.6 EN.R.1.1.3.7 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.6 SGW.R.1.1.3.7 SGW.R.1.1.3.8 SGW.R.1.1.3.9
	4768	o Num of SPIs (2 octets, unsigned integer) - The number of SPIs contained in the Delete payload. The size of each SPI is defined by the SPI Size field.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.6 EN.R.1.1.3.7 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.6 SGW.R.1.1.3.7 SGW.R.1.1.3.8 SGW.R.1.1.3.9
	4772	o Security Parameter Index(es) (variable length) - Identifies the specific security association(s) to delete. The length of this field is determined by the SPI Size and Num of SPIs fields.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.3.9 EN.I.1.1.3.10 EN.R.1.1.3.6 EN.R.1.1.3.7 EN.R.1.1.3.8 EN.R.1.1.3.9 SGW.I.1.1.3.9 SGW.I.1.1.3.10 SGW.R.1.1.3.6 SGW.R.1.1.3.7 SGW.R.1.1.3.8

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.3.9
	4776	The payload type for the Delete Payload is forty-two (42).		Not support		Explanation
	4778	3.12. Vendor ID Payload				
	4780	The Vendor ID Payload, denoted V in this document, contains a vendor-defined constant. The constant is used by vendors to identify and recognize remote instances of their implementations. This mechanism allows a vendor to experiment with new features while maintaining backward compatibility.		Not support		Explanation
	4786	A Vendor ID payload **MAY** announce that the sender is capable of accepting certain extensions to the protocol, or it **MAY** simply identify the implementation as an aid in debugging.	MAY MAY	Not support		Not need to test
	4788	A Vendor ID payload **MUST NOT** change the interpretation of any information defined in this specification	MUST NOT	Not support		Not need to test
	4790	(i.e., the critical bit **MUST** be set to 0).	MUST	Not support		Not need to test
	4791	Multiple Vendor ID payloads **MAY** be sent.	MAY	Not support		Not need to test
	4791	An implementation is not required to send any Vendor ID payload at all.		Not support		Explanation
	4794	A Vendor ID payload may be sent as part of any message. Reception of a familiar Vendor ID payload allows an implementation to make use of private use numbers described throughout this document, such as private payloads, private exchanges, private notifications, etc. Unfamiliar Vendor IDs **MUST** be ignored.	MUST	Not support		Not need to test
	4800	Writers of documents who wish to extend this protocol **MUST** define a Vendor ID payload to announce the ability to implement the extension in the document. It is expected that documents that	MUST	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		gain acceptance and are standardized will be given "magic numbers" out of the Future Use range by IANA, and the requirement to use a Vendor ID will go away.				
	4807	The Vendor ID Payload fields are defined as follows:		Not support		Explanation
	4809	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | | ~ Vendor ID (VID) ~ | | +-+ Figure 18: Vendor ID Payload Format		Not support		Explanation
	4821	o Vendor ID (variable length) - It is the responsibility of the person choosing the Vendor ID to assure its uniqueness in spite of the absence of any central registry for IDs. Good practice is to include a company name, a person name, or some such information. If you want to show off, you might include the latitude and longitude and time where you were when you chose the ID and some random input. A message digest of a long unique string is preferable to the long unique string itself.		Not support		Explanation
	4830	The payload type for the Vendor ID Payload is forty-three (43).		Not support		Explanation
	4832	3.13. Traffic Selector Payload				
	4834	The Traffic Selector Payload, denoted TS in this document, allows peers to identify packet flows for processing by IPsec security services. The Traffic Selector Payload consists of the IKE generic payload header followed by individual traffic selectors as follows:		Not support		Explanation

<table>
<tr><th colspan="2">Section</th><th rowspan="2">Sentence</th><th>RFC</th><th rowspan="2">Test
Requirements</th><th rowspan="2">Target</th><th rowspan="2">Comments</th></tr>
<tr><th>page</th><th>line</th><th>requirement</th></tr>
<tr><td></td><td>4839</td><td>1 2
3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+
| Next Payload |C| RESERVED | Payload Length
|
+-+
| Number of TSs | RESERVED
|
+-+
|
|
~ <Traffic Selectors>
~
|
|
+-+

Figure 19: Traffic Selectors Payload Format</td><td></td><td>BASIC</td><td>EN(initiator)
EN(responder)
SGW(initiator)
SGW(responder)</td><td>EN.I.1.1.1.2
EN.I.1.1.1.3
EN.I.1.2.1.1
EN.I.2.1.1.1
EN.I.2.1.1.2
EN.R.1.1.1.2
EN.R.1.1.1.3
EN.R.1.2.1.1
EN.R.2.1.1.1
EN.R.2.1.1.2
SGW.I.1.1.1.2
SGW.I.1.1.1.3
SGW.I.1.2.1.1
SGW.I.2.1.1.1
SGW.I.2.1.1.2
SGW.R.1.1.1.2
SGW.R.1.1.1.3
SGW.R.1.2.1.1
SGW.R.2.1.1.1
SGW.R.2.1.1.2</td></tr>
<tr><td></td><td>4853</td><td>o Number of TSs (1 octet) - Number of traffic selectors being provided.</td><td></td><td>BASIC</td><td>EN(initiator)
EN(responder)
SGW(initiator)
SGW(responder)</td><td>EN.I.1.1.1.2
EN.I.1.1.1.3
EN.I.1.2.1.1
EN.I.2.1.1.1
EN.I.2.1.1.2
EN.R.1.1.1.2
EN.R.1.1.1.3
EN.R.1.2.1.1
EN.R.2.1.1.1
EN.R.2.1.1.2
SGW.I.1.1.1.2
SGW.I.1.1.1.3
SGW.I.1.2.1.1
SGW.I.2.1.1.1
SGW.I.2.1.1.2</td></tr>
</table>

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4856	o RESERVED - This field **MUST** be sent as zero	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4856	and **MUST** be ignored on receipt.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.11.2 EN.R.1.1.11.2 SGW.I.1.1.11.2 SGW.R.1.1.11.2
	4859	o Traffic Selectors (variable length) - One or more individual traffic selectors.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4862	The length of the Traffic Selector payload includes the TS header and all the traffic selectors.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4865	The payload type for the Traffic Selector payload is forty-four (44) for addresses at the initiator's end of the SA		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4866	and forty-five (45) for addresses at the responder's end.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4869	There is no requirement that TSi and TSr contain the same number of individual traffic selectors. Thus, they are interpreted as follows: a packet matches a given TSi/TSr if it matches at least one of the individual selectors in TSi, and at least one of the individual selectors in TSr.		Not support		Explanation
	4875	For instance, the following traffic selectors:		Not support		Explanation
	4877	TSi = ((17, 100, 198.51.100.66-198.51.100.66), (17, 200, 198.51.100.66-198.51.100.66)) TSr = ((17, 300, 0.0.0.0-255.255.255.255), (17, 400, 0.0.0.0-255.255.255.255))		Not support		Explanation
	4882	would match UDP packets from 198.51.100.66 to anywhere, with any of the four combinations of source/destination ports (100,300), (100,400), (200,300), and (200, 400).		Not support		Explanation
	4886	Thus, some types of policies may require several Child SA pairs. For instance, a policy matching only source/destination ports (100,300) and (200,400), but not the other two combinations, cannot be negotiated as a single Child SA pair.		Not support		Explanation
	4891	3.13.1. Traffic Selector				
	4893	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | TS Type |IP Protocol ID*| Selector Length | +-+ | Start Port* | End Port* | +-+ | | ~ Starting Address* ~ | |		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		+-+ | | ~ Ending Address* ~ | | +-+ Figure 20: Traffic Selector				SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4911	*Note: All fields other than TS Type and Selector Length depend on the TS Type. The fields shown are for TS Types 7 and 8, the only two values currently defined.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4915	o TS Type (one octet) - Specifies the type of traffic selector.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4917	o IP protocol ID (1 octet) - Value specifying an associated IP protocol ID (such as UDP, TCP, and ICMP). A value of zero means that the protocol ID is not relevant to this traffic selector --the SA can carry all protocols.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4922	o Selector Length - Specifies the length of this Traffic Selector substructure including the header.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
○	4925	o Start Port (2 octets, unsigned integer) - Value specifying the smallest port number allowed by this traffic selector. For protocols for which port is undefined (including protocol 0), or if all ports are allowed, this field **MUST** be zero.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4928	ICMP and ICMPv6 Type and Code values, as well as Mobile IP version 6 (MIPv6) mobility header (MH) Type values, are represented in this field as specified in Section 4.4.1.1 of [IPSECARCH]. ICMP Type and Code values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and Code in the least significant eight bits. MIPv6 MH Type values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and the least significant eight bits set to zero.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
○	4938	o End Port (2 octets, unsigned integer) - Value specifying the largest port number allowed by this traffic selector. For protocols for which port is undefined (including protocol 0), or if all ports are allowed, this field **MUST** be 65535.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4941	ICMP and ICMPv6 Type and Code values, as well as MIPv6 MH Type values, are represented in this field as specified in Section 4.4.1.1 of [IPSECARCH]. ICMP Type and Code values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and Code in the least significant eight bits. MIPv6 MH Type values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and the least significant eight bits set to zero.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4951	o Starting Address - The smallest address included in this Traffic Selector (length determined by TS type).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4954	o Ending Address - The largest address included in this Traffic Selector (length determined by TS type).		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4957	Systems that are complying with [IPSECARCH] that wish to indicate "ANY" ports **MUST** set the start port to 0 and the end port to 65535;	MUST	Not support		Difficult to test
	4959	note that according to [IPSECARCH], "ANY" includes "OPAQUE".		Not support		Explanation
	4959	Systems working with [IPSECARCH] that wish to indicate "OPAQUE" ports, but not "ANY" ports, **MUST** set the start port to 65535 and the end port to 0.	MUST	Not support		Difficult to test
	4964	The traffic selector types 7 and 8 can also refer to ICMP or ICMPv6 type and code fields, as well as MH Type fields for the IPv6 mobility header [MIPV6]. Note, however, that neither ICMP nor MIPv6 packets have separate source and destination fields. The method for specifying the traffic selectors for ICMP and MIPv6 is shown by example in Section 4.4.1.3 of [IPSECARCH].		Not support		Explanation
	4971	The following table lists values for the Traffic Selector Type field and the corresponding Address Selector Data. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	4948	TS Type Value --		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4980	TS_IPV4_ADDR_RANGE 7 A range of IPv4 addresses, represented by two four-octet values. The first value is the beginning IPv4 address (inclusive) and the second value is the ending IPv4 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.		Not support		Explanation
	4988	TS_IPV6_ADDR_RANGE 8 A range of IPv6 addresses, represented by two sixteen-octet values. The first value is the beginning IPv6 address (inclusive) and the second value is the ending IPv6 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	4996	3.14. Encrypted Payload				
	4998	The Encrypted Payload, denoted SK{...} in this document, contains other payloads in encrypted form.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	4999	The Encrypted Payload, if present in a message, **MUST** be the last payload in the message. Often, it is the only payload in the message. This payload is also called the "Encrypted and Authenticated" payload.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5004	The algorithms for encryption and integrity protection are negotiated during IKE SA setup, and the keys are computed as specified in Sections 2.14 and 2.18.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	5008	This document specifies the cryptographic processing of Encrypted payloads using a block cipher in CBC mode and an integrity check algorithm that computes a fixed-length checksum over a variable size message.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
○	5011	The design is modeled after the ESP algorithms described in RFCs 2104 [HMAC], 4303 [ESP], and 2451 [ESPCBC]. This document completely specifies the cryptographic processing of IKE data, but those documents should be consulted for design rationale.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	5014	Future documents may specify the processing of Encrypted payloads for other types of transforms, such as counter mode encryption and authenticated encryption algorithms. Peers **MUST NOT** negotiate transforms for which no such specification exists.	MUST NOT	Not support		Explanation
	5020	When an authenticated encryption algorithm is used to protect the IKE SA, the construction of the encrypted payload is different than what is described here. See [AEAD] for more information on authenticated encryption algorithms and their use in ESP.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5025	The payload type for an Encrypted payload is forty-six (46). The Encrypted Payload consists of the IKE generic payload header followed by individual fields as follows:		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5029	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | Initialization Vector | | (length is block size for encryption algorithm) | +-+ ~ Encrypted IKE Payloads ~ + +-+ | | Padding (0-255 octets) | +-+-+-+-+-+-+-+-+ +-+-+-+-+-+-+-+-+ | | Pad Length | +-+ ~ Integrity Checksum Data ~ +-+ Figure 21: Encrypted Payload Format		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5048	o Next Payload - The payload type of the first embedded payload. Note that this is an exception in the standard header format, since the Encrypted payload is the last payload in the message and therefore the Next Payload field would normally be zero. But because the content of this payload is embedded payloads and there was no natural place to put the type of the first one, that type is placed here.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5056	o Payload Length - Includes the lengths of the header, initialization vector (IV), Encrypted IKE Payloads, Padding, Pad Length, and Integrity Checksum Data.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5059	o Initialization Vector - For CBC mode ciphers, the length of the initialization vector (IV) is equal to the block length of the underlying encryption algorithm. Senders **MUST** select a new unpredictable IV for every message; recipients **MUST** accept any value.	MUST MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	5063	The reader is encouraged to consult [MODES] for advice on IV generation. In particular, using the final ciphertext block of the previous message is not considered unpredictable. For modes other than CBC, the IV format and processing is specified in the document specifying the encryption algorithm and mode.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5069	o IKE Payloads are as specified earlier in this section. This field is encrypted with the negotiated cipher.		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5072	o Padding **MAY** contain any value chosen by the sender, and **MUST** have a length that makes the combination of the Payloads, the Padding, and the Pad Length to be a multiple of the encryption block size. This field is encrypted with the negotiated cipher.	MAY MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5077	o Pad Length is the length of the Padding field. The sender **SHOULD** set the Pad Length to the minimum value that makes the combination of the Payloads, the Padding, and the Pad Length a multiple of the block size,	SHOULD	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5080	but the recipient **MUST** accept any length that results in proper alignment. This field is encrypted with the negotiated cipher.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5084	o Integrity Checksum Data is the cryptographic checksum of the entire message starting with the Fixed IKE Header through the Pad Length. The checksum **MUST** be computed over the encrypted message. Its length is determined by the integrity algorithm negotiated.	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1 SGW.R.2.1.1.1 SGW.R.2.1.1.2
	5089	3.15. Configuration Payload				
	5091	The Configuration payload, denoted CP in this document, is used to exchange configuration information between IKE peers. The exchange is for an IRAC to request an internal IP address from an IRAS and to exchange other information of the sort that one would acquire with Dynamic Host Configuration Protocol (DHCP) if the IRAC were directly connected to a LAN.		Not support		Explanation
	5098	The Configuration Payload is defined as follows:		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5100	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | CFG Type | RESERVED | +-+ | | ~ Configuration Attributes ~ | | +-+ Figure 22: Configuration Payload Format		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5114	The payload type for the Configuration Payload is forty-seven (47).		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5116	o CFG Type (1 octet) - The type of exchange represented by the Configuration Attributes. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5123	CFG Type Value -------------------------		Not support		Explanation
	5125	CFG_REQUEST 1		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5126	CFG_REPLY 2		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5127	CFG_SET 3		Not support		Explanation
	5128	CFG_ACK 4		Not support		Explanation
	5130	o RESERVED (3 octets) - **MUST** be sent as zero;	MUST	ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5130	**MUST** be ignored on receipt.	MUST	ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.3 SGW.R.2.1.2.3
	5133	o Configuration Attributes (variable length) - These are type length value (TLV) structures specific to the Configuration Payload and are defined below. There may be zero or more Configuration Attributes in this payload.		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5138	3.15.1. Configuration Attributes				
	5140	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ |R| Attribute Type | Length | +-+ | | ~ Value ~ | | +-+ Figure 23: Configuration Attribute Format		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5152	o Reserved (1 bit) - This bit **MUST** be set to zero	MUST	ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5152	and **MUST** be ignored on receipt.	MUST	ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.3 SGW.R.2.1.2.3
	5155	o Attribute Type (15 bits) - A unique identifier for each of the Configuration Attribute Types.		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5158	o Length (2 octets, unsigned integer) - Length in octets of Value.		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5160	o Value (0 or more octets) - The variable-length value of this Configuration Attribute.		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5161	The following lists the attribute types.		Not support		Explanation
	5163	The values in the following table are only current as of the publication date of RFC 4306 (except INTERNAL_ADDRESS_EXPIRY and INTERNAL_IP6_NBNS which were removed by this document). Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.		Not support		Explanation
	5170	Attribute Type Value Multi-Valued Length --		Not support		Explanation
	5172	INTERNAL_IP4_ADDRESS 1 YES* 0 or 4 octets		Not support		Explanation
	5173	INTERNAL_IP4_NETMASK 2 NO 0 or 4 octets		Not support		Explanation
	5174	INTERNAL_IP4_DNS 3 YES 0 or 4 octets		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5175	INTERNAL_IP4_NBNS 4 YES 0 or 4 octets		Not support		Explanation
	5176	INTERNAL_IP4_DHCP 6 YES 0 or 4 octets		Not support		Explanation
	5177	APPLICATION_VERSION 7 NO 0 or more		Not support		Explanation
	5178	INTERNAL_IP6_ADDRESS 8 YES* 0 or 17 octets		Not support		Explanation
	5179	INTERNAL_IP6_DNS 10 YES 0 or 16 octets		ADVANCED	EN(initiator) SGW(responder)	EN.I.2.1.2.1 EN.I.2.1.2.2 SGW.R.2.1.2.1 SGW.R.2.1.2.2
	5180	INTERNAL_IP6_DHCP 12 YES 0 or 16 octets		Not support		Explanation
	5181	INTERNAL_IP4_SUBNET 13 YES 0 or 8 octets		Not support		Explanation
	5182	SUPPORTED_ATTRIBUTES 14 NO Multiple of 2		Not support		Explanation
	5183	INTERNAL_IP6_SUBNET 15 YES 17 octets		Not support		Explanation
	5185	* These attributes may be multi-valued on return only if multiple values were requested.		Not support		Explanation
	5188	o INTERNAL_IP4_ADDRESS, INTERNAL_IP6_ADDRESS - An address on the internal network, sometimes called a red node address or private address, and **MAY** be a private address on the Internet. In a request message, the address specified is a requested address (or a zero-length address if no specific address is requested). If a specific address is requested, it likely indicates that a previous connection existed with this address and the requestor would like to reuse that address.	MAY	Not support		Not need to test
	5195	With IPv6, a requestor **MAY** supply the low-order address octets it wants to use. Multiple internal addresses **MAY** be requested by requesting multiple internal address attributes.	MAY MAY	Not support		Not need to test
	5198	The responder **MAY** only send up to the number of addresses requested. The INTERNAL_IP6_ADDRESS is made up of two	MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		fields: the first is a 16-octet IPv6 address, and the second is a one-octet prefix-length as defined in [ADDRIPV6].				
	5201	The requested address is valid as long as this IKE SA (or its rekeyed successors) requesting the address is valid. This is described in more detail in Section 3.15.3.		Not support		Explanation
	5206	o INTERNAL_IP4_NETMASK - The internal network's netmask. Only one netmask is allowed in the request and response messages (e.g., 255.255.255.0), and it **MUST** be used only with an INTERNAL_IP4_ADDRESS attribute.	MUST	Not support		IPv4 is out of scope
	5209	INTERNAL_IP4_NETMASK in a CFG_REPLY means roughly the same thing as INTERNAL_IP4_SUBNET containing the same information ("send traffic to these addresses through me"), but also implies a link boundary. For instance, the client could use its own address and the netmask to calculate the broadcast address of the link.		Not support		Explanation
	5214	An empty INTERNAL_IP4_NETMASK attribute can be included in a CFG_REQUEST to request this information (although the gateway can send the information even when not requested). Non-empty values for this attribute in a CFG_REQUEST do not make sense and thus **MUST NOT** be included.	MUST NOT	Not support		Explanation
	5220	o INTERNAL_IP4_DNS, INTERNAL_IP6_DNS - Specifies an address of a DNS server within the network. Multiple DNS servers **MAY** be requested.	MAY	Not support		Not need to test
	5222	The responder **MAY** respond with zero or more DNS server attributes.	MAY	Not support		Not need to test
	5224	o INTERNAL_IP4_NBNS - Specifies an address of a NetBios Name Server (WINS) within the network. Multiple NBNS servers **MAY** be requested.	MAY	Not support		Not need to test
	5226	The responder **MAY** respond with zero or more NBNS server attributes.	MAY	Not support		Not need to test
	5229	o INTERNAL_IP4_DHCP, INTERNAL_IP6_DHCP - Instructs the host to send any internal DHCP requests to the address contained within the attribute. Multiple DHCP servers **MAY** be requested.	MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5231	The responder **MAY** respond with zero or more DHCP server attributes.	MAY	Not support		Not need to test
	5234	o APPLICATION_VERSION - The version or application information of the IPsec host. This is a string of printable ASCII characters that is NOT null terminated.		Not support		Explanation
	5238	o INTERNAL_IP4_SUBNET - The protected sub-networks that this edge-device protects. This attribute is made up of two fields: the first being an IP address and the second being a netmask. Multiple sub-networks **MAY** be requested.	MAY	Not support		Not need to test
	5241	The responder **MAY** respond with zero or more sub-network attributes. This is discussed in more detail in Section 3.15.2.	MAY	Not support		Not need to test
	5245	o SUPPORTED_ATTRIBUTES - When used within a Request, this attribute **MUST** be zero-length and specifies a query to the responder to reply back with all of the attributes that it supports.	MUST	Not support		SUPPORTED_ATTRIBUTES is out of the scope
	5247	The response contains an attribute that contains a set of attribute identifiers each in 2 octets. The length divided by 2 (octets) would state the number of supported attributes contained in the response.		Not support		Explanation
	5253	o INTERNAL_IP6_SUBNET - The protected sub-networks that this edge-device protects. This attribute is made up of two fields: the first is a 16-octet IPv6 address, and the second is a one-octet prefix-length as defined in [ADDRIPV6]. Multiple sub-networks **MAY** be requested.	MAY	Not support		Not need to test
	5257	The responder **MAY** respond with zero or more sub-network attributes. This is discussed in more detail in Section 3.15.2.	MAY	Not support		Not need to test
	5261	Note that no recommendations are made in this document as to how an implementation actually figures out what information to send in a response. That is, we do not recommend any specific method of an IRAS determining which DNS server should be returned to a requesting IRAC.		Not support		Explanation
	5267	The CFG_REQUEST and CFG_REPLY pair allows an IKE endpoint to request information from its peer.		Not support		Explanation
	5268	If an attribute in the CFG_REQUEST Configuration Payload is not zero-length, it is taken as a suggestion for that attribute.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5270	The CFG_REPLY Configuration Payload **MAY** return that value, or a new one. It **MAY** also add new attributes and not include some requested ones.	MAY MAY	Not support		Not need to test
	5272	Unrecognized or unsupported attributes **MUST** be ignored in both requests and responses.	MUST	Not support		Explanation
	5275	The CFG_SET and CFG_ACK pair allows an IKE endpoint to push configuration data to its peer.		Not support		Explanation
	5276	In this case, the CFG_SET Configuration Payload contains attributes the initiator wants its peer to alter. The responder **MUST** return a Configuration Payload if it accepted any of the configuration data	MUST	Not support		CFG-SET/ACK is out of the scope
	5279	and it **MUST** contain the attributes that the responder accepted with zero-length data.	MUST	Not support		CFG-SET/ACK is out of the scope
	5280	Those attributes that it did not accept **MUST NOT** be in the CFG_ACK Configuration Payload.	MUST NOT	Not support		CFG-SET/ACK is out of the scope
	5282	If no attributes were accepted, the responder **MUST** return either an empty CFG_ACK payload or a response message without a CFG_ACK payload.	MUST	Not support		CFG-SET/ACK is out of the scope
	5284	There are currently no defined uses for the CFG_SET/CFG_ACK exchange, though they may be used in connection with extensions based on Vendor IDs.		Not support		Explanation
	5286	An implementation of this specification **MAY** ignore CFG_SET payloads.	MAY	Not support		CFG-SET/ACK is out of the scope
	5289	3.15.2. Meaning of INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET				
	5291	INTERNAL_IP4/6_SUBNET attributes can indicate additional subnets, ones that need one or more separate SAs, that can be reached through the gateway that announces the attributes. INTERNAL_IP4/6_SUBNET attributes may also express the gateway's policy about what traffic should be sent through the gateway; the client can choose whether other traffic (covered by TSr, but not in INTERNAL_IP4/6_SUBNET) is sent through the gateway or directly to the destination. Thus, traffic to the addresses listed in the INTERNAL_IP4/6_SUBNET attributes		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		should be sent through the gateway that announces the attributes. If there are no existing Child SAs whose traffic selectors cover the address in question, new SAs need to be created.				
	5303	For instance, if there are two subnets, 198.51.100.0/26 and 192.0.2.0/24, and the client's request contains the following:		Not support		Explanation
	5306	CP(CFG_REQUEST) = INTERNAL_IP4_ADDRESS() TSi = (0, 0-65535, 0.0.0.0-255.255.255.255) TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)		Not support		Explanation
	5311	then a valid response could be the following (in which TSr and INTERNAL_IP4_SUBNET contain the same information):		Not support		Explanation
	5314	CP(CFG_REPLY) = INTERNAL_IP4_ADDRESS(198.51.100.234) INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192) INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0) TSi = (0, 0-65535, 198.51.100.234-198.51.100.234) TSr = ((0, 0-65535, 198.51.100.0-198.51.100.63), (0, 0-65535, 192.0.2.0-192.0.2.255))		Not support		Explanation
	5322	In these cases, the INTERNAL_IP4_SUBNET does not really carry any useful information.		Not support		Explanation
	5325	A different possible response would have been this:		Not support		Explanation
	5327	CP(CFG_REPLY) = INTERNAL_IP4_ADDRESS(198.51.100.234) INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192) INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)		Not support		Explanation
	5331	TSi = (0, 0-65535, 198.51.100.234-198.51.100.234) TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)		Not support		Explanation
	5334	That response would mean that the client can send all its traffic through the gateway, but the gateway does not mind if the client sends traffic not included by INTERNAL_IP4_SUBNET directly to the destination (without going through the gateway).		Not support		Explanation
	5339	A different situation arises if the gateway has a policy that requires the traffic for the two subnets to be carried in separate		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		SAs. Then a response like this would indicate to the client that if it wants access to the second subnet, it needs to create a separate SA:				
	5345	CP(CFG_REPLY) = INTERNAL_IP4_ADDRESS(198.51.100.234) INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192) INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0) TSi = (0, 0-65535, 198.51.100.234-198.51.100.234) TSr = (0, 0-65535, 198.51.100.0-198.51.100.63)		Not support		Explanation
	5352	INTERNAL_IP4_SUBNET can also be useful if the client's TSr included only part of the address space. For instance, if the client requests the following:		Not support		Explanation
	5356	CP(CFG_REQUEST) = INTERNAL_IP4_ADDRESS() TSi = (0, 0-65535, 0.0.0.0-255.255.255.255) TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)		Not support		Explanation
	5361	then the gateway's response might be:		Not support		Explanation
	5363	CP(CFG_REPLY) = INTERNAL_IP4_ADDRESS(198.51.100.234) INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192) INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0) TSi = (0, 0-65535, 198.51.100.234-198.51.100.234) TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)		Not support		Explanation
	5370	Because the meaning of INTERNAL_IP4_SUBNET/INTERNAL_IP6_SUBNET in CFG_REQUESTs is unclear, they cannot be used reliably in CFG_REQUESTs.		Not support		Explanation
	5374	3.15.3. Configuration payloads for IPv6				
	5376	The configuration payloads for IPv6 are based on the corresponding IPv4 payloads, and do not fully follow the "normal IPv6 way of doing things". In particular, IPv6 stateless autoconfiguration or router advertisement messages are not used, neither is neighbor discovery. Note that there is an additional document that discusses IPv6 configuration in IKEv2,		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		[IPV6CONFIG]. At the present time, it is an experimental document, but there is a hope that with more implementation experience, it will gain the same standards treatment as this document.				
	5386	A client can be assigned an IPv6 address using the INTERNAL_IP6_ADDRESS configuration payload.		Not support		Explanation
	5387	A minimal exchange might look like this:		Not support		Explanation
	5390	CP(CFG_REQUEST) = INTERNAL_IP6_ADDRESS() INTERNAL_IP6_DNS() TSi = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF) TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)		Not support		Explanation
	5396	CP(CFG_REPLY) = INTERNAL_IP6_ADDRESS(2001:DB8:0:1:2:3:4:5/64) INTERNAL_IP6_DNS(2001:DB8:99:88:77:66:55:44) TSi = (0, 0-65535, 2001:DB8:0:1:2:3:4:5 - 2001:DB8:0:1:2:3:4:5) TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)		Not support		Explanation
	5402	The client **MAY** send a non-empty INTERNAL_IP6_ADDRESS attribute in the CFG_REQUEST to request a specific address or interface identifier. The gateway first checks if the specified address is acceptable, and if it is, returns that one. If the address was not acceptable, the gateway attempts to use the interface identifier with some other prefix; if even that fails, the gateway selects another interface identifier.	MAY	Not support		Explanation
	5410	The INTERNAL_IP6_ADDRESS attribute also contains a prefix length field. When used in a CFG_REPLY, this corresponds to the INTERNAL_IP4_NETMASK attribute in the IPv4 case.		Not support		Explanation
	5414	Although this approach to configuring IPv6 addresses is reasonably simple, it has some limitations. IPsec tunnels configured using IKEv2 are not fully featured "interfaces" in the IPv6 addressing architecture sense [ADDRIPV6]. In particular,		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		they do not necessarily have link-local addresses, and this may complicate the use of protocols that assume them, such as [MLDV2].				
	5421	3.15.4. Address Assignment Failures				
	5423	If the responder encounters an error while attempting to assign an IP address to the initiator during the processing of a Configuration Payload, it responds with an INTERNAL_ADDRESS_FAILURE notification. The IKE SA is still created even if the initial Child SA cannot be created because of this failure.		Not support		Explanation
	5427	If this error is generated within an IKE_AUTH exchange, no Child SA will be created. However, there are some more complex error cases.		Not support		Explanation
	5431	If the responder does not support configuration payloads at all, it can simply ignore all configuration payloads. This type of implementation never sends INTERNAL_ADDRESS_FAILURE notifications.		Not support		Explanation
	5434	If the initiator requires the assignment of an IP address, it will treat a response without CFG_REPLY as an error.		ADVANCED	EN(initiator)	EN.I.2.1.2.4
	5437	The initiator may request a particular type of address (IPv4 or IPv6) that the responder does not support, even though the responder supports configuration payloads. In this case, the responder simply ignores the type of address it does not support and processes the rest of the request as usual.		ADVANCED		[SGW.R.P116.L5437.ADD]
	5443	If the initiator requests multiple addresses of a type that the responder supports, and some (but not all) of the requests fail, the responder replies with the successful addresses only. The responder sends INTERNAL_ADDRESS_FAILURE only if no addresses can be assigned.		Not Support		untestable (If SGW has enough IP address, it is difficult for responder to send INTERNAL_ADDRESS_FAILURE)

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5448	If the initiator does not receive the IP address(es) required by its policy, it **MAY** keep the IKE SA up and retry the configuration payload as separate INFORMATIONAL exchange after suitable timeout, or it **MAY** tear down the IKE SA by sending a DELETE payload inside a separate INFORMATIONAL exchange and later retry IKE SA from the beginning after some timeout. Such a timeout should not be too short (especially if the IKE SA is started from the beginning) because these error situations may not be able to be fixed quickly; the timeout should likely be several minutes. For example, an address shortage problem on the responder will probably only be fixed when more entries are returned to the address pool when other clients disconnect or when responder is reconfigured with larger address pool.	MAY MAY	Not support		Explanation
	5462	3.16. Extensible Authentication Protocol (EAP) Payload				
	5464	The Extensible Authentication Protocol Payload, denoted EAP in this document, allows IKE SAs to be authenticated using the protocol defined in RFC 3748 [EAP] and subsequent extensions to that protocol. When using EAP, an appropriate EAP method needs to be selected. Many of these methods have been defined, specifying the protocol's use with various authentication mechanisms. EAP method types are listed in [EAP-IANA]. A short summary of the EAP format is included here for clarity.		Not support		Explanation
	5473	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Next Payload |C| RESERVED | Payload Length | +-+ | | ~ EAP Message ~ | |		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		+-+ Figure 24: EAP Payload Format				
	5485	The payload type for an EAP Payload is forty-eight (48).		Not support		Explanation
	5487	1 2 3 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 +-+ | Code | Identifier | Length | +-+ | Type | Type_Data... +- Figure 25: EAP Message Format		Not support		Explanation
	5497	o Code (1 octet) indicates whether this message is a Request (1), Response (2), Success (3), or Failure (4).		Not support		Explanation
	5500	o Identifier (1 octet) is used in PPP to distinguish replayed messages from repeated ones. Since in IKE, EAP runs over a reliable protocol, it serves no function here. In a response message, this octet **MUST** be set to match the identifier in the corresponding request.	MUST	Not support		EAP is out of the scope
	5506	o Length (2 octets, unsigned integer) is the length of the EAP message and **MUST** be four less than the Payload Length of the encapsulating payload.	MUST	Not support		EAP is out of the scope
	5510	o Type (1 octet) is present only if the Code field is Request (1) or Response (2).		Not support		EAP is out of the scope
	5511	For other codes, the EAP message length **MUST** be four octets and the Type and Type_Data fields **MUST NOT** be present.	MUST MUST NOT	Not support		EAP is out of the scope

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5513	In a Request (1) message, Type indicates the data being requested.		Not support		EAP is out of the scope
	5514	In a Response (2) message, Type **MUST** either be Nak or match the type of the data requested.	MUST	Not support		EAP is out of the scope
	5515	Note that since IKE passes an indication of initiator identity in the first message in the IKE_AUTH exchange, the responder SHOULD NOT send EAP Identity requests (type 1). The initiator **MAY**, however, respond to such requests if it receives them.	MAY	Not support		Explanation
	5521	o Type_Data (Variable Length) varies with the Type of Request and the associated Response. For the documentation of the EAP methods, see [EAP].		Not support		Explanation
	5525	Note that since IKE passes an indication of initiator identity in the first message in the IKE_AUTH exchange, the responder should not send EAP Identity requests.		Not support		Explanation
	5527	The initiator may, however, respond to such requests if it receives them.		Not support		Explanation
	5531	4. Conformance Requirements				
	5533	In order to assure that all implementations of IKEv2 can interoperate, there are "**MUST** support" requirements in addition to those listed elsewhere. Of course, IKEv2 is a security protocol, and one of its major functions is to allow only authorized parties to successfully complete establishment of SAs. So a particular implementation may be configured with any of a number of restrictions concerning algorithms and trusted authorities that will prevent universal interoperability.		Not support		Explanation
	5542	IKEv2 is designed to permit minimal implementations that can interoperate with all compliant implementations. The following are features that can be omitted in a minimal implementation:		Not support		Explanation
	5546	o Ability to negotiate SAs through a NAT and tunnel the resulting ESP SA over UDP.		Not support		Explanation
	5549	o Ability to request (and respond to a request for) a temporary IP address on the remote end of a tunnel.		Not support		Explanation
	5552	o Ability to support EAP-based authentication.		Not support		Explanation
	5554	o Ability to support window sizes greater than one.		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5556	o Ability to establish multiple ESP or AH SAs within a single IKE SA.		Not support		Explanation
	5559	o Ability to rekey SAs.		Not support		Explanation
	5561	To assure interoperability, all implementations **MUST** be capable of parsing all payload types (if only to skip over them) and to ignore payload types that it does not support unless the critical bit is set in the payload header.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.3 SGW.R.1.1.4.3
	5564	If the critical bit is set in an unsupported payload header, all implementations **MUST** reject the messages containing those payloads.	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.1.4.4 SGW.R.1.1.4.4
	5568	Every implementation **MUST** be capable of doing four-message IKE_SA_INIT and IKE_AUTH exchanges establishing two SAs (one for IKE, one for ESP or AH).	MUST	BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.I.1.1.1.2 EN.I.1.1.1.3 EN.I.1.2.1.1 EN.I.2.1.1.1 EN.I.2.1.1.2 EN.R.1.1.1.1 EN.R.1.1.1.2 EN.R.1.1.1.3 EN.R.1.2.1.1 EN.R.1.3.1.1 EN.R.2.1.1.1 EN.R.2.1.1.2 SGW.I.1.1.1.1 SGW.I.1.1.1.2 SGW.I.1.1.1.3 SGW.I.1.2.1.1 SGW.I.2.1.1.1 SGW.I.2.1.1.2 SGW.R.1.1.1.1 SGW.R.1.1.1.2 SGW.R.1.1.1.3 SGW.R.1.2.1.1 SGW.R.1.3.1.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
						SGW.R.2.1.1.1 SGW.R.2.1.1.2
	5570	Implementations **MAY** be initiate-only or respond-only if appropriate for their platform.	MAY	Not support		Not need to test
	5571	Every implementation **MUST** be capable of responding to an INFORMATIONAL exchange,	MUST	BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
	5572	but a minimal implementation **MAY** respond to any request in the INFORMATIONAL exchange with an empty response (note that within the context of an IKE SA, an "empty" message consists of an IKE header followed by an Encrypted payload with no payloads contained in it).	MAY	Not support		Not need to test
	5576	A minimal implementation **MAY** support the CREATE_CHILD_SA exchange only in so far as to recognize requests and reject them with a Notify payload of type NO_ADDITIONAL_SAS.	MAY	Not support		Not need to test
	5579	A minimal implementation need not be able to initiate CREATE_CHILD_SA or INFORMATIONAL exchanges. When an SA expires (based on locally configured values of either lifetime or octets passed), and implementation **MAY** either try to renew it	MAY MAY	Not support		Not need to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		with a CREATE_CHILD_SA exchange or it **MAY** delete (close) the old SA and create a new one.				
	5584	If the responder rejects the CREATE_CHILD_SA request with a NO_ADDITIONAL_SAS notification, the implementation **MUST** be capable of instead deleting the old SA and creating a new one.	MUST	Not support		Difficult to test
	5589	Implementations are not required to support requesting temporary IP addresses or responding to such requests.		Not support		Explanation
	5590	If an implementation does support issuing such requests and its policy requires using temporary IP addresses, it **MUST** include a CP payload in the first message in the IKE_AUTH exchange containing at least a field of type INTERNAL_IP4_ADDRESS or INTERNAL_IP6_ADDRESS. All other fields are optional.	MUST	ADVANCED	EN(initiator)	EN.I.2.1.2.1
	5595	If an implementation supports responding to such requests, it **MUST** parse the CP payload of type CFG_REQUEST in the first message in the IKE_AUTH exchange and recognize a field of type INTERNAL_IP4_ADDRESS or INTERNAL_IP6_ADDRESS.	MUST	ADVANCED	SGW(initiator)	SGW.I.2.1.2.1
	5598	If it supports leasing an address of the appropriate type, it **MUST** return a CP payload of type CFG_REPLY containing an address of the requested type.	MUST	ADVANCED	SGW(initiator)	SGW.I.2.1.2.1
	5600	The responder may include any other related attributes.		Not support		Explanation
	5603	For an implementation to be called conforming to this specification, it **MUST** be possible to configure it to accept the following:	MUST	Not support		Explanation
	5606	o Public Key Infrastructure using X.509 (PKIX) Certificates containing and signed by RSA keys of size 1024 or 2048 bits, where the ID passed is any of ID_KEY_ID, ID_FQDN, ID_RFC822_ADDR, or ID_DER_ASN1_DN.		Not support		Explanation
	5610	o Shared key authentication where the ID passed is any of ID_KEY_ID, ID_FQDN, or ID_RFC822_ADDR.		Not support		Explanation
	5613	o Authentication where the responder is authenticated using PKIX Certificates and the initiator is authenticated using shared key authentication.		Not support		Explanation
	5618	5. Security Considerations				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5620	While this protocol is designed to minimize disclosure of configuration information to unauthenticated peers, some such disclosure is unavoidable. One peer or the other must identify itself first and prove its identity first. To avoid probing, the initiator of an exchange is required to identify itself first, and usually is required to authenticate itself first.		Not support		Explanation
	5625	The initiator can, however, learn that the responder supports IKE and what cryptographic protocols it supports.		Not support		Explanation
	5627	The responder (or someone impersonating the responder) can probe the initiator not only for its identity, but using CERTREQ payloads may be able to determine what certificates the initiator is willing to use.		Not support		Explanation
	5632	Use of EAP authentication changes the probing possibilities somewhat. When EAP authentication is used, the responder proves its identity before the initiator does, so an initiator that knew the name of a valid initiator could probe the responder for both its name and certificates.		Not support		Explanation
	5638	Repeated rekeying using CREATE_CHILD_SA without additional Diffie-Hellman exchanges leaves all SAs vulnerable to cryptanalysis of a single key. Implementers should take note of this fact and set a limit on CREATE_CHILD_SA exchanges between exponentiations. This document does not prescribe such a limit.		Not support		Explanation
	5644	The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. Due to these inputs, it is difficult to determine the strength of a key for any of the defined groups. Diffie-Hellman group number two, when used with a strong random number generator and an exponent no less than 200 bits, is common for use with 3DES.		Not support		Explanation
	5651	Group five provides greater security than group two.		Not support		Explanation
	5656	Group one is for historic purposes only and does not provide sufficient strength except for use with DES, which is also for		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		historic use only. Implementations should make note of these estimates when establishing policy and negotiating security parameters.				
	5658	Note that these limitations are on the Diffie-Hellman groups themselves. There is nothing in IKE that prohibits using stronger groups nor is there anything that will dilute the strength obtained from stronger groups (limited by the strength of the other algorithms negotiated including the PRF). In fact, the extensible framework of IKE encourages the definition of more groups; use of elliptic curve groups may greatly increase strength using much smaller numbers.		Not support		Explanation
	5666	It is assumed that all Diffie-Hellman exponents are erased from memory after use.		Not support		Internal process
	5669	The IKE_SA_INIT and IKE_AUTH exchanges happen before the initiator has been authenticated. As a result, an implementation of this protocol needs to be completely robust when deployed on any insecure network. Implementation vulnerabilities, particularly DoS attacks, can be exploited by unauthenticated peers. This issue is particularly worrisome because of the unlimited number of messages in EAP-based authentication.		Not support		Explanation
	5677	The strength of all keys is limited by the size of the output of the negotiated PRF. For this reason, a PRF whose output is less than 128 bits (e.g., 3DES-CBC) **MUST NOT** be used with this protocol.	MUST NOT	Not support		Difficult to test
	5681	The security of this protocol is critically dependent on the randomness of the randomly chosen parameters. These should be generated by a strong random or properly seeded pseudorandom source (see [RANDOMNESS]).		Not support		Explanation
	5684	Implementers should take care to ensure that use of random numbers for both keys and nonces is engineered in a fashion that does not undermine the security of the keys.		Not support		Explanation
	5688	For information on the rationale of many of the cryptographic design choices in this protocol, see [SIGMA] and [SKEME]. Though the security of negotiated Child SAs does not depend on the strength of the encryption and integrity protection negotiated in the IKE SA, implementations **MUST NOT** negotiate NONE as	MUST NOT	Not support		Difficult to test

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		the IKE integrity protection algorithm or ENCR_NULL as the IKE encryption algorithm.				
	5695	When using pre-shared keys, a critical consideration is how to assure the randomness of these secrets. The strongest practice is to ensure that any pre-shared key contain as much randomness as the strongest key being negotiated. Deriving a shared secret from a password, name, or other low-entropy source is not secure. These sources are subject to dictionary and social-engineering attacks, among others.		Not support		Explanation
	5702	The NAT_DETECTION_*_IP notifications contain a hash of the addresses and ports in an attempt to hide internal IP addresses behind a NAT. Since the IPv4 address space is only 32 bits, and it is usually very sparse, it would be possible for an attacker to find out the internal address used behind the NAT box by trying all possible IP addresses and trying to find the matching hash. The port numbers are normally fixed to 500, and the SPIs can be extracted from the packet. This reduces the number of hash calculations to 2^32. With an educated guess of the use of private address space, the number of hash calculations is much smaller. Designers should therefore not assume that use of IKE will not leak internal address information.		Not support		Explanation
	5714	When using an EAP authentication method that does not generate a shared key for protecting a subsequent AUTH payload, certain man-in-the-middle and server-impersonation attacks are possible [EAPMITM]. These vulnerabilities occur when EAP is also used in protocols that are not protected with a secure tunnel.		Not support		Explanation
	5718	Since EAP is a general-purpose authentication protocol, which is often used to provide single-signon facilities, a deployed IPsec solution that relies on an EAP authentication method that does not generate a shared key (also known as a non-key-generating EAP method) can become compromised due to the deployment of an entirely unrelated application that also happens to use the same non-key-generating EAP method, but in an unprotected fashion. Note that this vulnerability is not limited to just EAP, but can		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		occur in other scenarios where an authentication infrastructure is reused.				
	5727	For example, if the EAP mechanism used by IKEv2 utilizes a token authenticator, a man-in-the-middle attacker could impersonate the web server, intercept the token authentication exchange, and use it to initiate an IKEv2 connection. For this reason, use of non-key-generating EAP methods **SHOULD** be avoided where possible.	SHOULD	Not support		EAP is out of the scope
	5732	Where they are used, it is extremely important that all usages of these EAP methods **SHOULD** utilize a protected tunnel, where the initiator validates the responder's certificate before initiating the EAP authentication.	SHOULD	Not support		EAP is out of the scope
	5735	Implementers should describe the vulnerabilities of using non-key-generating EAP methods in the documentation of their implementations so that the administrators deploying IPsec solutions are aware of these dangers.		Not support		EAP authentication is out of the scope
	5740	An implementation using EAP **MUST** also use a public-key-based authentication of the server to the client before the EAP authentication begins, even if the EAP method offers mutual authentication. This avoids having additional IKEv2 protocol variations and protects the EAP data from active attackers.	MUST	Not support		EAP is out of the scope
	5746	If the messages of IKEv2 are long enough that IP-level fragmentation is necessary, it is possible that attackers could prevent the exchange from completing by exhausting the reassembly buffers. The chances of this can be minimized by using the Hash and URL encodings instead of sending certificates (see Section 3.6). Additional mitigations are discussed in [DOSUDPPROT].		Not support		Explanation
	5753	Admission control is critical to the security of the protocol. For example, trust anchors used for identifying IKE peers should probably be different than those used for other forms of trust, such as those used to identify public web servers. Moreover, although IKE provides a great deal of leeway in defining the security policy		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		for a trusted peer's identity, credentials, and the correlation between them, having such security policy defined explicitly is essential to a secure implementation.				
	5762	5.1. Traffic selector authorization				
	5764	IKEv2 relies on information in the Peer Authorization Database (PAD) when determining what kind of Child SAs a peer is allowed to create. This process is described in Section 4.4.3 of [IPSECARCH]. When a peer requests the creation of an Child SA with some traffic selectors, the PAD must contain "Child SA Authorization Data" linking the identity authenticated by IKEv2 and the addresses permitted for traffic selectors.		Not support		Explanation
	5772	For example, the PAD might be configured so that authenticated identity "sgw23.example.com" is allowed to create Child SAs for 192.0.2.0/24, meaning this security gateway is a valid "representative" for these addresses. Host-to-host IPsec requires similar entries, linking, for example, "fooserver4.example.com" with 198.51.100.66/32, meaning this identity is a valid "owner" or "representative" of the address in question.		Not support		Explanation
	5780	As noted in [IPSECARCH], "It is necessary to impose these constraints on creation of child SAs to prevent an authenticated peer from spoofing IDs associated with other, legitimate peers". In the example given above, a correct configuration of the PAD prevents sgw23 from creating Child SAs with address 198.51.100.66, and prevents fooserver4 from creating Child SAs with addresses from 192.0.2.0/24.		Not support		Explanation
	5788	It is important to note that simply sending IKEv2 packets using some particular address does not imply a permission to create Child SAs with that address in the traffic selectors. For example, even if sgw23 would be able to spoof its IP address as 198.51.100.66, it could not create Child SAs matching fooserver4's traffic.		Not support		Explanation
	5794	The IKEv2 specification does not specify how exactly IP address assignment using configuration payloads interacts with the PAD. Our interpretation is that when a security gateway assigns an		Not support		Explanation

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		address using configuration payloads, it also creates a temporary PAD entry linking the authenticated peer identity and the newly allocated inner address.				
	5801	It has been recognized that configuring the PAD correctly may be difficult in some environments. For instance, if IPsec is used between a pair of hosts whose addresses are allocated dynamically using DHCP, it is extremely difficult to ensure that the PAD specifies the correct "owner" for each IP address. This would require a mechanism to securely convey address assignments from the DHCP server, and link them to identities authenticated using IKEv2.		Not support		Explanation
	5809	Due to this limitation, some vendors have been known to configure their PADs to allow an authenticated peer to create Child SAs with traffic selectors containing the same address that was used for the IKEv2 packets. In environments where IP spoofing is possible (i.e., almost everywhere) this essentially allows any peer to create Child SAs with any traffic selectors. This is not an appropriate or secure configuration in most circumstances. See [H2HIPSEC] for an extensive discussion about this issue, and the limitations of host-to-host IPsec in general.		Not support		Explanation
	5820	6. IANA Considerations				
	5822	[IKEV2] defined many field types and values. IANA has already registered those types and values in [IKEV2IANA], so they are not listed here again.				
	5826	Two items are removed from the IKEv2 Configuration Payload Attribute Types table: INTERNAL_IP6_NBNS and INTERNAL_ADDRESS_EXPIRY.				
	5829	Two new additions to the IKEv2 parameters "NOTIFY MESSAGES - ERROR TYPES" registry are defined here that were not defined in [IKEV2]:				
	5832	43 TEMPORARY_FAILURE 44 CHILD_SA_NOT_FOUND				
	5835	IANA should change the exiting IKEv2 Payload Types table from:				
	5837	46 Encrypted E [IKEv2]				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5839	to				
	5841	46 Encrypted and Authenticated SK [This document]				
	5843	IANA has updated all references to RFC 4306 to point to this document.				
	5847	7. Acknowledgements				
	5849	Many individuals in the IPsecME Working Group were very helpful in contributing ideas and text for this document, as well as in reviewing the clarifications suggested by others.				
	5853	The acknowledgements from the IKEv2 document were:				
	5855	This document is a collaborative effort of the entire IPsec WG. If there were no limit to the number of authors that could appear on an RFC, the following, in alphabetical order, would have been listed: Bill Aiello, Stephane Beaulieu, Steve Bellovin, Sara Bitan, Matt Blaze, Ran Canetti, Darren Dukes, Dan Harkins, Paul Hoffman, John Ioannidis, Charlie Kaufman, Steve Kent, Angelos Keromytis, Tero Kivinen, Hugo Krawczyk, Andrew Krywaniuk, Radia Perlman, Omer Reingold, and Michael Richardson. Many other people contributed to the design. It is an evolution of IKEv1, ISAKMP, and the IPsec DOI, each of which has its own list of authors. Hugh Daniel suggested the feature of having the initiator, in message 3, specify a name for the responder, and gave the feature the cute name "You Tarzan, Me Jane". David Faucher and Valery Smyslov helped refine the design of the traffic selector negotiation.				
	5876	8. References				
	5878	8.1. Normative References				
	5880	[ADDGROUP] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.				
	5885	[ADDRIPV6] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5889	[AEAD] McGrew, D. and D. Black, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", RFC 5282, August 2008.				
	5894	[AESCMACPRF128] Song, J., "The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4615, August 2006.				
	5900	[AESXCBCPRF128] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.				
	5905	[EAP] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.				
	5909	[ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.				
	5913	[ESPCBC] Pereira, R. and R. Adams, "The ESP CBC-Mode Cipher Algorithms", RFC 2451, November 1998.				
	5916	[HTTP] Fielding, et. al., R., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.				
	5919	[IKEV2IANA] "Internet Key Exchange Version 2 (IKEv2) Parameters", <http://www.iana.org/assignments/ikev2-parameters>.				
	5923	[IPSECARCH] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.				
	5927	[MUSTSHOULD] Bradner, S., "Key Words for use in RFCs to indicate Requirement Levels", BCP 14, RFC 2119, March 1997.				
	5931	[PKCS1] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	5935	[PKIX] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.				
	5940	[RFC4307] Schiller, J., "Cryptographic Algorithms for Use in IKEv2", RFC 4307, December 2005.				
	5943	[UDPENCAPS] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.				
	5948	[URLS] Fielding, et. al., R., "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005.				
	5951	8.2. Informative References				
	5953	[AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.				
	5956	[ARCHGUIDEPHIL] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, December 2002.				
	5960	[ARCHPRINC] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.				
	5964	[Clarif] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", RFC 4718, October 2006.				
	5967	[DES] American National Standards Institute, "American National Standard for Information Systems-Data Link Encryption", ANSI X3.106, 1983.				
	5971	[DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n. 6, June 1977.				
	5975	[DIFFSERVARCH] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475.				
	5980	[DIFFSERVFIELD] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		Headers", RFC 2474, December 1998.				
	5986	[DIFFTUNNEL] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.				
	5990	[DOI] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.				
	5993	[DOSUDPPROT] C. Kaufman, R. Perlman, and B. Sommerfeld, "DoS protection for UDP-based protocols", ACM Conference on Computer and Communications Security , October 2003.				
	5998	[DSS] National Institute of Standards and Technology, U.S. Department of Commerce, "Digital Signature Standard", Draft FIPS 186-3, June 2008.				
	6002	[EAI] Abel, Y., "Internationalized Email Headers", RFC 5335, September 2008.				
	6005	[EAP-IANA] "Extensible Authentication Protocol (EAP) Registry: Method Types", <http://www.iana.org/assignments/eap-numbers>.				
	6009	[EAPMITM] N. Asokan, V. Nierni, and K. Nyberg, "Man-in-the-Middle in Tunneled Authentication Protocols", November 2002, <http://eprint.iacr.org/2002/163>.				
	6013	[ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.				
	6016	[EXCHANGEANALYSIS] R. Perlman and C. Kaufman, "Analysis of the IPsec key exchange Standard", WET-ICE Security Conference, MIT , 2001, <http://sec.femto.org/wetice-2001/papers/radia-paper.pdf>.				
	6022	[H2HIPSEC] Aura, T., Roe, M., and A. Mohammed, "Experiences with Host-to-Host IPsec", 13th International Workshop on Security Protocols, Cambridge, UK, April 2005.				
	6027	[HMAC] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		1997.				
	6031	[IDEA] X. Lai, "On the Design and Security of Block Ciphers", ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992.				
	6035	[IDNA] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.				
	6039	[IKEV1] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.				
	6042	[IKEV2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.				
	6045	[IP-COMP] Shacham, A., Monsour, B., Pereira, R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 3173, September 2001.				
	6049	[IPSECARCH-OLD] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.				
	6053	[IPV6CONFIG] Eronen, et. al., P., "IPv6 Configuration in IKEv2", RFC 5739, February 2010.				
	6057	[ISAKMP] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.				
	6061	[MAILFORMAT] Resnick, P., "Internet Message Format", RFC 5322, October 2008.				
	6065	[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.				
	6068	[MIPV6] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.				
	6071	[MLDV2] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.				
	6074	[MOBIKE] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6077	[MODES] National Institute of Standards and Technology, U.S. Department of Commerce, "Recommendation for Block Cipher Modes of Operation", SP 800-38A, 2001.				
	6081	[NAI] Aboba, B., Beadles, M., Eronen, P., and J. Arkko, "The Network Access Identifier", RFC 4282, December 2005.				
	6084	[NATREQ] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.				
	6087	[OAKLEY] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.				
	6090	[PFKEY] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, July 1998.				
	6093	[PHOTURIS] Karn, P. and W. Simpson, "Photuris: Session-Key Management Protocol", RFC 2522, March 1999.				
	6097	[RANDOMNESS] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.				
	6101	[REAUTH] Nir, Y., "Repeated Authentication in Internet Key Exchange (IKEv2) Protocol", RFC 4478, April 2006.				
	6104	[REUSE] Menezes, A. and B. Ustaoglu, "On Reusing Ephemeral Keys In Diffie-Hellman Key Agreement Protocols", December 2008, <http://www.cacr.math.uwaterloo.ca/~ajmeneze/ publications/ephemeral.pdf>.				
	6109	[ROHCV2] Ertekin, et. al., E., "IKEv2 Extensions to Support Robust Header Compression over IPsec (ROHCoIPsec)", draft-ietf-rohc-ikev2-extensions-hcoipsec (work in progress), August 2009.				
	6114	[RSA] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", February 1978.				
	6118	[SHA] National Institute of Standards and Technology, U.S. Department of Commerce, "Secure Hash Standard", FIPS 180-3, October 2008.				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6122	[SIGMA] H. Krawczyk, "SIGMA: the `SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and its Use in the IKE Protocols", Advances in Cryptography - CRYPTO 2003 Proceedings LNCS 2729, 2003, <http:// www.informatik.uni-trier.de/~ley/db/conf/crypto/crypto2003.html>.				
	6129	[SKEME] H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security , 1996.				
	6134	[TRANSPARENCY] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.				
	6139	Appendix A. Summary of changes from IKEv1				
	6141	The goals of this revision to IKE are:				
	6143	1. To define the entire IKE protocol in a single document, replacing RFCs 2407, 2408, and 2409 and incorporating subsequent changes to support NAT Traversal, Extensible Authentication, and Remote Address acquisition;				
	6148	2. To simplify IKE by replacing the eight different initial exchanges with a single four-message exchange (with changes in authentication mechanisms affecting only a single AUTH payload rather than restructuring the entire exchange) see [EXCHANGEANALYSIS];				
	6154	3. To remove the Domain of Interpretation (DOI), Situation (SIT), and Labeled Domain Identifier fields, and the Commit and Authentication only bits;				
	6158	4. To decrease IKE's latency in the common case by making the initial exchange be 2 round trips (4 messages), and allowing the ability to piggyback setup of a Child SA on that exchange;				
	6162	5. To replace the cryptographic syntax for protecting the IKE messages themselves with one based closely on ESP to simplify implementation and security analysis;				
	6166	6. To reduce the number of possible error states by making the protocol reliable (all messages are acknowledged) and sequenced. This allows shortening CREATE_CHILD_SA exchanges from 3 messages to 2;				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6171	7. To increase robustness by allowing the responder to not do significant processing until it receives a message proving that the initiator can receive messages at its claimed IP address;				
	6175	8. To fix cryptographic weaknesses such as the problem with symmetries in hashes used for authentication (documented by Tero Kivinen);				
	6179	9. To specify traffic selectors in their own payloads type rather than overloading ID payloads, and making more flexible the traffic selectors that may be specified;				
	6183	10. To specify required behavior under certain error conditions or when data that is not understood is received in order to make it easier to make future revisions in a way that does not break backward compatibility;				
	6188	11. To simplify and clarify how shared state is maintained in the presence of network failures and DoS attacks; and				
	6191	12. To maintain existing syntax and magic numbers to the extent possible to make it likely that implementations of IKEv1 can be enhanced to support IKEv2 with minimum effort.				
	6196	Appendix B. Diffie-Hellman Groups				
	6198	There are two Diffie-Hellman groups defined here for use in IKE. These groups were generated by Richard Schroeppel at the University of Arizona. Properties of these primes are described in [OAKLEY].				
	6202	The strength supplied by group 1 may not be sufficient for typical uses and is here for historic reasons.				
	6205	Additional Diffie-Hellman groups have been defined in [ADDGROUP].				
	6207	B.1. Group 1 - 768-Bit MODP				
	6209	This group is assigned id 1 (one).				
	6211	The prime is: 2^768 - 2 ^704 - 1 + 2^64 * { [2^638 pi] + 149686 } Its hexadecimal value is:				
	6214	FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08 8A67CC74 020BBEA6 3B139B22 514A0879				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		8E3404DD EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF				
	6219	The generator is 2.				
	6221	B.2. Group 2 - 1024-Bit MODP				
	6223	This group is assigned id 2 (two).				
	6225	The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }. Its hexadecimal value is:				
	6228	FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381 FFFFFFFF FFFFFFFF				
	6235	The generator is 2.				
	6238	Appendix C. Exchanges and Payloads				
	6240	This appendix contains a short summary of the IKEv2 exchanges, and what payloads can appear in which message. This appendix is purely informative; if it disagrees with the body of this document, the other text is considered correct.		Not support		Explanation
	6245	Vendor ID (V) payloads may be included in any place in any message. This sequence here shows what are the most logical places for them.		Not support		Explanation
	6248	C.1. IKE_SA_INIT Exchange				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6250	request --> [N(COOKIE)], SA, KE, Ni, [N(NAT_DETECTION_SOURCE_IP)+, N(NAT_DETECTION_DESTINATION_IP)], [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.1 EN.R.1.1.1.1 SGW.I.1.1.1.1 SGW.R.1.1.1.1
	6256	normal response <-- SA, KE, Nr, (no cookie) [N(NAT_DETECTION_SOURCE_IP), N(NAT_DETECTION_DESTINATION_IP)], [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+], [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.1 SGW.I.1.1.1.2 SGW.R.1.1.1.1
	6262	cookie response <-- N(COOKIE), [V+][N+]		ADVANCED	Both	Initiator
	6265	different Diffie- <-- N(INVALID_KE_PAYLOAD), Hellman group [V+][N+] wanted		ADVANCED	Both	EN.I.1.1.6.7 EN.R.1.1.6.7 SGW.I.1.1.6.7 SGW.R.1.1.6.7
	6269	C.2. IKE_AUTH Exchange without EAP				
	6271	request --> IDi, [CERT+], [N(INITIAL_CONTACT)], [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+], [IDr], AUTH, [CP(CFG_REQUEST)], [N(IPCOMP_SUPPORTED)+], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, TSi, TSr,		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.2 EN.R.1.1.1.2 SGW.I.1.1.1.2 SGW.R.1.1.1.2

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
		[V+][N+]				
	6284	response <-- IDr, [CERT+], AUTH, [CP(CFG_REPLY)], [N(IPCOMP_SUPPORTED)], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, TSi, TSr, [N(ADDITIONAL_TS_POSSIBLE)], [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.1.1.3 EN.R.1.1.1.2 SGW.I.1.1.1.3 SGW.R.1.1.1.2
	6295	error in Child SA <-- IDr, [CERT+], creation AUTH, N(error), [V+][N+]		BASIC	Both	EN.R.1.1.4.4 EN.R.1.1.6.7 EN.R.1.1.7.2 EN.R.1.2.4.1 EN.R.1.2.6.9 SGW.R.1.1.4.4 SGW.R.1.1.6.7 SGW.R.1.1.7.2 SGW.R.1.2.4.1 SGW.R.1.2.6.9
	6300	C.3. IKE_AUTH Exchange with EAP				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6302	first request --> IDi, [N(INITIAL_CONTACT)], [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+], [IDr], [CP(CFG_REQUEST)], [N(IPCOMP_SUPPORTED)+], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, TSi, TSr, [V+][N+]		Not support		Explanation
	6314	first response <-- IDr, [CERT+], AUTH, EAP, [V+][N+]		Not support		Explanation
	6318	/ --> EAP repeat 1..N times | ¥ <-- EAP		Not support		Explanation
	6322	last request --> AUTH		Not support		Explanation
	6324	last response <-- AUTH, [CP(CFG_REPLY)], [N(IPCOMP_SUPPORTED)], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, TSi, TSr, [N(ADDITIONAL_TS_POSSIBLE)], [V+][N+]		Not support		Explanation
	6334	C.4. CREATE_CHILD_SA Exchange for Creating or Rekeying Child SAs				

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6336	request --> [N(REKEY_SA)], [CP(CFG_REQUEST)], [N(IPCOMP_SUPPORTED)+], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, Ni, [KEi], TSi, TSr [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.2 EN.I.1.2.5.2 EN.R.1.2.5.2 EN.R.1.2.7.1 SGW.I.1.2.3.2 SGW.I.1.2.5.2 SGW.R.1.2.5.2 SGW.R.1.2.7.1
	6345	normal response <-- [CP(CFG_REPLY)], [N(IPCOMP_SUPPORTED)], [N(USE_TRANSPORT_MODE)], [N(ESP_TFC_PADDING_NOT_SUPPORTED)], [N(NON_FIRST_FRAGMENTS_ALSO)], SA, Nr, [KEr], TSi, TSr, [N(ADDITIONAL_TS_POSSIBLE)] [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.3.2 EN.I.1.2.5.2 EN.R.1.2.5.2 EN.R.1.2.7.1 SGW.I.1.2.3.2 SGW.I.1.2.5.2 SGW.R.1.2.5.2 SGW.R.1.2.7.1
	6354	error case <-- N(error)		BASIC	Both	EN.R.1.1.6.8 SGW.R.1.1.6.8
	6356	different Diffie-Hellman group wanted <-- N(INVALID_KE_PAYLOAD), [V+][N+]		ADVANCED	Both	EN.I.1.1.6.7 EN.R.1.1.6.7 SGW.I.1.1.6.7 SGW.R.1.1.6.7
	6360	C.5. CREATE_CHILD_SA Exchange for Rekeying the IKE SA				
	6362	request --> SA, Ni, KEi [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.4.2 EN.R.1.2.6.1 SGW.I.1.2.4.2 SGW.R.1.2.6.1
	6365	response <-- SA, Nr, KEr [V+][N+]		BASIC	EN(initiator) EN(responder) SGW(initiator) SGW(responder)	EN.I.1.2.4.2 EN.R.1.2.6.1 SGW.I.1.2.4.2 SGW.R.1.2.6.1

Section		Sentence	RFC	Test	Target	Comments
page	line		requirement	Requirements		
	6368	C.6. INFORMATIONAL Exchange				
	6370	request --> [N+], [D+], [CP(CFG_REQUEST)]		BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1
	6374	response <-- [N+], [D+], [CP(CFG_REPLY)]		BASIC	EN(responder) SGW(responder)	EN.R.1.3.1.1 SGW.R.1.3.1.1