

ICS 33.040.40  
M 32



# 中华人民共和国通信行业标准

YD/T 1897-2009

IKEv2定义在RFC4306，更新于 RFC 5996，本文档则主要是对RFC文档做了中文翻译

## 互联网密钥交换协议（IKEv2）技术要求

Technical requirements of Internet Key Exchange Protocol (IKEv2)

2009-06-15 发布

2009-09-01 实施

中华人民共和国工业和信息化部 发布

## 目 次

前 言.....	II
1 范围.....	1
2 规范性引用文件.....	1
3 术语、定义和缩略语.....	1
4 IKE 基本使用与操作.....	2
5 IKE 协议细节及变化.....	8
6 报头和载荷的格式.....	23
7 一致性要求.....	48
8 安全性考虑.....	49
附录 A (资料性附录) 与 IKE 版本 1 的区别汇总.....	51
附录 B (资料性附录) Diffie-Hellman 组.....	52
参考文献.....	53

## 前　　言

本标准是 IP 安全协议 (IPSec) 系列标准之一，该系列标准的名称及结构预计如下：

1. 《IP 安全协议体系结构》(MOD IETF RFC2401)
2. 《IP 认证头(AH)》(MOD IETF RFC2402)
3. 《IP 封装安全载荷(ESP)》(MOD IETF RFC2406)
4. YD/T 1466-2006 《IP 安全协议 (IPSec) 技术要求》
5. YD/T 1467-2006 《IP 安全协议 (IPSec) 测试方法》
6. 《IP 安全协议 (IPSec) 穿越网络地址翻译 (NAT) 技术要求》
7. 《互联网密钥交换协议 (IKEv2) 技术要求》
8. 《互联网密钥交换协议 (IKEv2) 测试方法》

本标准与《互联网密钥交换协议 (IKEv2) 测试方法》配套使用。

本标准的附录 A、附录 B 均为资料性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：工业和信息化部电信研究院

本标准起草人：谢 玮、刘 述、田慧蓉、马 科、马军峰、高 巍、江浩洁、唐 浩、武 静、  
吴英桦

# 互联网密钥交换协议（IKEv2）技术要求

## 1 范围

本标准规定了动态建立并管理IPsec安全联盟的协议——互联网密钥交换协议（IKEv2）技术要求，包括IKEv2协议使用的命令、交换过程、报头和载荷格式、安全性等方面的要求。

本标准适用于支持互联网密钥交换协议（IKEv2）的设备和网络。

## 2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准。然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

YD/T 1466-2006	IP安全协议（IPSec）技术要求
IETF RFC822 (1982)	ARPA因特网正文信息格式标准
IETF RFC3168 (2001)	到IP的显示拥塞指示其他要求
IETF RFC3513 (2003)	IPv6寻址体系结构
IETF RFC3748 (2004)	扩展认证协议(EAP)
IETF RFC3948 (2005)	IPsec ESP包的UDP封装
IETF RFC4301 (2005)	因特网协议的安全体系结构

## 3 术语、定义和缩略语

### 3.1 术语和定义

YD/T 1466-2006确立的术语和定义适用于本标准。

### 3.2 缩略语

下列缩略语适用于本标准。

AH	Authentication Header	认证头
CA	Certificate Authority	证书授权中心
CBC	Cipher Block Chaining	码块链
ESP	Encapsulating Security Payload	封装安全载荷
DES	Data Encryption Standard	数据加密标准
ECN	Explicit Congestion Notification	显示拥塞指示
HMAC	HASH MAC	散列消息认证码
ICMP	Internet Control Message Protocol	互联网控制消息协议
ICV	Integrity Check Value	完整性校验值
IKE	Internet Key Exchange	互联网密钥交换
IPSec	IP Security	IP安全

MD5	Message Digest 5	消息摘要5
PMTU	Path Maximum Transmission Unit	路径最大传输单元
SA	Security Association	安全联盟
SAD	SA Database	安全联盟数据库
SHA-1	Secure Hash Algorithm-1	安全散列算法-1
SNMP	Simple Network Management Protocol	简单网络管理协议
SPI	Security Parameter Index	安全参数索引
SPD	Security Policy Database	安全策略数据库
TS	Traffic Selector	流量选择器
TCP	Transmission Control Protocol	传输控制协议
UDP	User Datagram Protocol	用户数据报协议

## 4 IKE 基本使用与操作

### 4.1 概述

IPsec 为 IP 数据报提供机密性、数据完整性、接入控制和数据源认证。这些服务通过在 IP 数据报的源和宿主之间维护共享的状态来提供。该状态定义了提供给数据报的特定服务（使用加密算法来提供这些服务）以及作为作为加密算法输入的密钥。

通过手动的方式建立这个共享状态缺乏扩展性。因此，需要一种协议动态建立这个共享状态。本标准就描述了这样的一个协议——互联网密钥交换（IKE）协议的第二版本。

IKE 执行两个参与方之间的相互认证，并建立 IKE 安全联盟（SA），该安全联盟包括共享秘密信息（该信息用于有效建立 ESP 和/或 AH 的 SA），以及 SA 使用的一组加密算法，该算法用于对其承载的流量进行保护。本标准中，术语“算法族”或者“加密算法族”指的是一整套用于保护 SA 的加密算法。发起者通过列举它所支持的能够通过混合匹配方式被组合进“算法族”的算法来建议使用一个或者多个“算法族”。IKE 也能够协商关于 ESP 和/或 AH SA 的 IP 压缩的使用。定义 IKE SA 为“IKE\_SA”。通过 IKE\_SA 获得建立的 ESP 和/或者 AH SA，定义为“CHILD\_SA”。

所有的 IKE 通信都是由消息对组成的：请求和响应。这个消息对被称为“交换”。我们称第一个建立 IKE\_SA 的消息为 IKE\_SA\_INIT 和 IKE\_AUTH 交换，随后的 IKE 交换为 CREATE\_CHILD\_SA 交换或者 INFORMATIONAL 交换。在通常情况下，建立 IKE\_SA 以及第一个 CHILD\_SA，包括一个 IKE\_SA\_INIT 交换和一个 IKE\_AUTH 交换（总共四个消息）。在例外情况下，可能存在多次这些交换。在所有情况下，所有的 IKE\_SA\_INIT 交换都必须在任何其他类型交换之前完成，随后必须完成所有的 IKE\_AUTH 交换，然后任意数目的 CREATE\_CHILD\_SA 和 INFORMATIONAL 交换可以以任意顺序出现。在某些场景中，在 IPsec 端点之间只需要一个单独的 CHILD\_SA，因此这里就不存在额外的交换。随后的交换可以被用于在相同的经过认证的端点对之间建立额外的 CHILD\_SA，以及执行内部事务管理功能。

IKE 消息流总是以一个请求紧跟着一个响应的方式出现。这样请求者可以确保可靠性。如果响应在一个超时间隔内没有被收到，请求者就需要重传这个请求（或者断开链接）。

IKE 会话的第一个请求/响应消息（IKE\_SA\_INIT）为 IKE\_SA 协商安全参数，发送临时随机数

(nonce)，并发送 DiffieHellman 值。

第二个请求/响应（IKE\_AUTH）传输标识符，检验符合两个标识符的安全信息，以及为第一个（通常只有一个）AH 和/或 ESP CHILD\_SA 建立 SA。

随后的交换类型是 CREATE\_CHILD\_SA（创建一个 CHILD\_SA）和 INFORMATIONAL（删除 SA，报告错误条件，或者做其他事务管理）。每个请求要求一个响应。没有载荷的 INFORMATIONAL 请求（不同于通过语法请求的空加密载荷）通常被用于检查存活状态。这些后续的交换只有在初始的交换完成之后才能使用。

在下面的描述中，我们假设没有错误发生。对可能发生错误的数据流的调整在第 5.21 节描述。

## 4.2 使用场景

IKE 可以被用于不同场景下的 ESP 和/或者 AH SA 的协商，每个都具有自己特定的要求。

### 4.2.1 安全网关到安全网关隧道

在图 1 所示的这种场景下，IP 连接的任意一个端点都没有实现 IPsec，但是它们之间的网络节点在部分路径上保护流量。这种保护对于端点是透明的，并且依赖于发送数据报通过隧道端点的普通路由。每一个端点都应该宣告一系列在它“后面”的地址，并且数据报应该按照隧道模式被发送，隧道模式是指内部的 IP 头应该包含实际的端点的 IP 地址。

使用传输模式为什么不行？  
[https://blog.csdn.net/weixin\\_34202952/article/details/92812752](https://blog.csdn.net/weixin_34202952/article/details/92812752)  
 传输模式的特点限制了其能力：传输模式是复用 ip 头，隧道模式是新增 ip 头，ipsec 处理的是 ip 层，不是 tcp 层

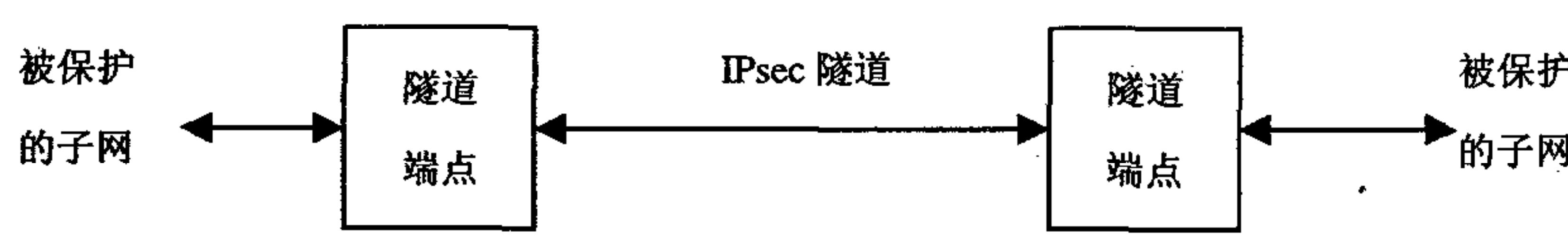


图 1 安全网关到安全网关隧道

### 4.2.2 端点到端点传输

在图 2 所示的这种场景下，两个 IP 连接的端点都实现了 IPsec，遵循 IETF RFC4301 (2005) 中对主机的要求。传输模式通常没有内部 IP 头。如果存在内部 IP 头，内部头的地址应该和外部头的地址相同。被这个 SA 保护的数据包的单一地址对将被协商。这些端点可以给予参与方的 IPsec 认证标识符实现应用层接入控制。该场景能够实现端到端安全，并已经成为互联网的一个指导性原则，同时也是限制网络中固有问题复杂度的方法。尽管这个场景可能不能被完全应用于 IPv4 的互联网，但是它已经在某些使用 IKEv1 的企业网（Intranet）中的特定场景下被成功应用。IKE 应该被广泛地应用于向 IPv6 的过渡中，并应采用 IKEv2。

在这种场景中，可能一个或者两个被保护端点藏在 NAT（网络地址翻译）节点之后，在这种情况下，被隧道封装的数据包采用 UDP 封装，以便 UDP 头中的端口号能够被用于标识藏在 NAT 后面的单独的端点（参见第 5.24）。

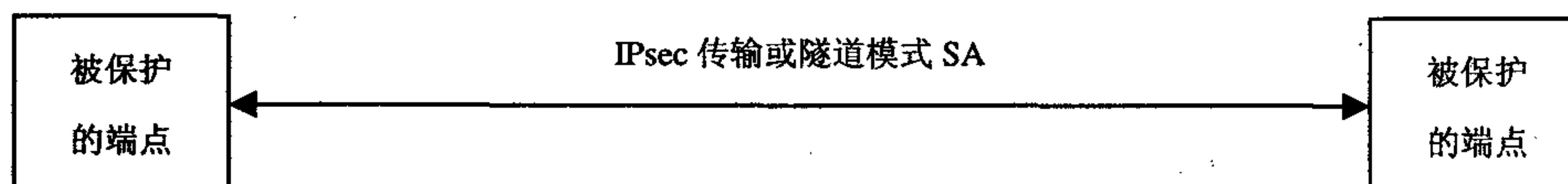


图 2 端点到端点

### 4.2.3 端点到安全网关隧道

在图 3 所示的场景下，被保护的端点（典型例子是正在漫游的可移动计算机）通过 IPsec 保护的隧道连接回它自己的企业网。它可以只利用该隧道访问企业网信息，也可以利用隧道将自己的信息传回企

业网，以便它能够利用企业防火墙的保护来抵御来自互联网的攻击。不管哪种情况，被保护的端点都将在向安全网关获取 IP 地址，使得到它的数据包能够先到安全网关，然后再被隧道封装返回给该端点。这个 IP 地址可以是静态的，也可以由安全网关动态分配。对于后者，本标准包括了一个在 SA 使用期内，发起者向安全网关请求 IP 地址的机制。

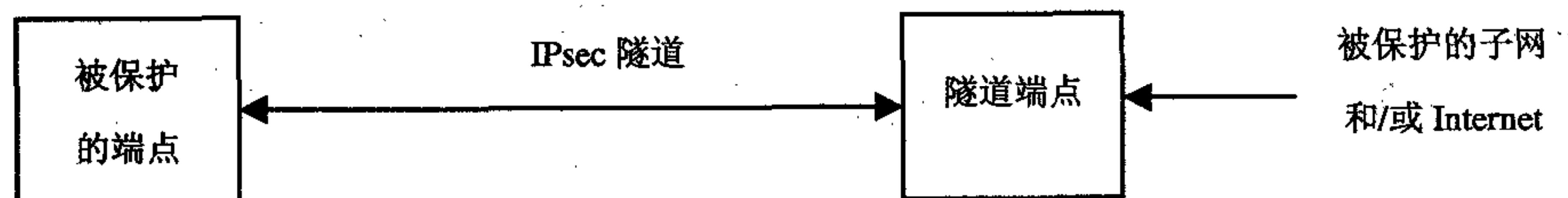


图 3 端点到安全网关隧道

在这个场景下，数据包将使用隧道模式。对于每一个来自被保护的端点的数据包，外部 IP 头将包含同它当前位置一致的源地址（即：将直接获得路由到该端点的流量的地址），而内部 IP 头将包含分配给安全网关的源 IP 地址（即：将获得路由到安全网关的流量的地址，这些流量将被前转给该端点）。外部的目的地址总是安全网关的地址，而内部的目的地址则是数据包的最终目的地址。

在这个场景下，被保护的端点可能隐藏在 NAT 之后。在这种情况下，安全网关看到的 IP 地址同被保护的端点发送的 IP 地址不同，数据包具有一个 UDP 的封装，以便能够被合适地路由。

#### 4.2.4 其他场景

作为上述场景的组合而出现其他场景也是可能的。4.2.1 和 4.2.3 场景组合就是一个明显的例子。一个子网可以通过使用 IPsec 的远程安全网关来完成所有的外部访问，到达子网的数据包将被外部网络路由到安全网关。举例，虚拟地放在 Internet 上具有静态 IP 地址的某家庭网络，实际上，其连接是由 ISP 通过分配一个单独的动态 IP 地址给用户安全网关来提供的（这里的静态 IP 地址和 IPsec 中继由位于其他地方的第三方提供）。

#### 4.3 初始交换

IKE 通信总是从 IKE\_SA\_INIT 和 IKE\_AUTH 交换开始。这些初始交换通常由 4 个消息组成，在某些场景下消息数目可能会增加。所有使用 IKE 的通信都由请求/响应对组成。首先描述基本的交换，然后是一些变化。第一个消息对（IKE\_SA\_INIT）协商加密算法，交换临时随机数和 Diffie-Hellman 交换。

第二个消息对（IKE\_AUTH）认证先前的消息，交换标识符和证书，建立第一个 CHILD\_SA。部分消息通过 IKE\_SA\_INIT 交换建立的密钥进行加密和完整性保护，所以标识符对窃听者是隐藏的，并且在所有消息中的字段都是被认证的。

在表 1 的描述中，列举了消息中包含的载荷的名称。

表 1 消息中载荷的名称

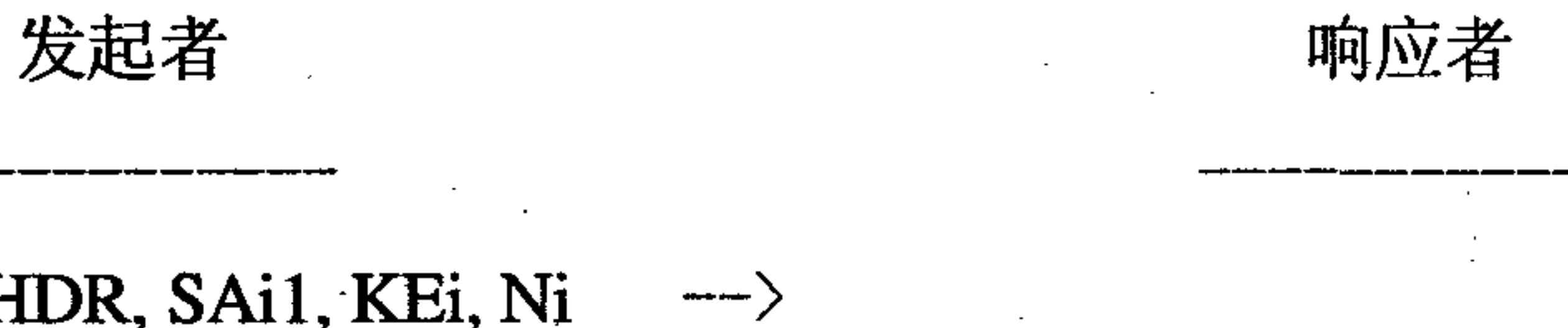
名 称	有效载荷
AUTH	认证
CERT	证书
CERTREQ	证书请求
CP	配置
D	删除
E	加密
EAP	可扩展认证协议
HDR	IKE 头

表 1 (续)

名 称	有效载荷
IDi	发起者标识
IDr	响应者标识
KE	密钥交换
Ni, Nr	临时随机数 (i 表示发起者, r 表示响应者)
N	通知
SA	安全联盟
TSi	流量选择器发起者
TSr	流量选择器响应者
V	厂商 ID

每个载荷的详细内容在第 6 章描述。可选的载荷显示在中括号中，例如[CERTREQ]，表明载荷中包含的证书请求可选。

初始的交换如下所示：



HDR, SAi1, KEi, Ni -->

<-- HDR, SAr1, KEr, Nr, [CERTREQ]

响应者从发起者提供的选择中选择加密族，并且在 SAr1 载荷中表达这种选择，使用 KEr 载荷完成 Diffie-Hellman 交换，并且通过 Nr 载荷发送它的临时随机数。

**IKE\_AUTH阶段** 在这个阶段，每一个参与方都能产生 SKEYSEED，IKE\_SA 的所有密钥都是继承自它。几乎所有后面的消息头都是加密的，且完整性受到保护。用于加密和完整性保护的密钥继承自 SKEYSEED，并且被称为 SK\_e (加密) 和 SK\_a (认证，也称为完整性保护)。SK\_e 和 SK\_a 在每个方向上都分别进行计算。除了继承自 DH 值的，用于 IKE\_SA 保护的 SK\_e 和 SK\_a 密钥外，另外一批 SK\_d 也计算出，并被用于引出将来的 CHILD\_SA 建立过程。符号 SK{...} 显示了那些被加密和被完整性保护的载荷使用了那个方向的 SK\_e 和 SK\_a。

HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,]}

AUTH, SAi2, TSi, TSr} -->

发起者使用 IDi 载荷声称它的身份，证明和 IDi 一致的秘密信息，以及使用 AUTH 载荷保护第一个消息内容的完整性（参见第 6.8 节）。它也可以通过 CERT 载荷发送它的证书，并且在 CERTREQ 载荷中列出它所信任的 CA。如果包含了任何 CERT 载荷，则第一个被提供的证书必须包含用于验证 AUTH 字段的公开密钥。可选的载荷 IDr 使得发起者能够指明它想和哪个响应者的标识符进行对话。这对于正在运行的响应者在使用相同的 IP 地址，但具有多个标识符的主机时，是非常有用的。最后的字段（以 SAi2 开始）在 CREATE\_CHILD\_SA 交换中进行描述。

<-- HDR, SK {IDr, [CERT,] AUTH,  
SAr2, TSi, TSr}

响应者使用 IDr 载荷说明它的标识符，可选发送一个或者多个证书（证书中包含用于验证 AUTH 的

共开密钥), 使用 AUTH 载荷来认证它的标识符和第二个消息的完整性, 通过下节描述的 CREATE\_CHILD\_SA 交换完成 CHILD\_SA 的协商。

消息 3 和 4 的接收必须验证所有被正确计算的签名和 MAC, 验证符合用于产生 AUTH 载荷的密钥的 ID 载荷中的名字。

#### 4.4 CREATE\_CHILD\_SA 交换

这个交换由一个单一的请求/响应对。它可以在初始交换完成之后由 IKE\_SA 的任意一端引发。

紧随初始交换后的所有消息都被使用, 和在 IKE 交换中的前两个消息中协商的加密算法与密钥, 进行加密保护。这些随后的消息使用在第 6.14 节描述的加密载荷的语法。所有的消息都在一个加密载荷中, 即使它们内容为“空的”。

任意一个端点都可以发起 CREATE\_CHILD\_SA 交换, 所以在这章中术语“发起者”指发起交换的端点。

CHILD\_SA 通过发送一个 CREATE\_CHILD\_SA 请求被创建。CREATE\_CHILD\_SA 请求包括一个 KE 载荷, 进行额外的 Diffie-Hellman 交换, 以便能够使 CHILD\_SA 的加密更加健壮。CHILD\_SA 的密钥原料来自在 IKE\_SA 建立期间被建立的 SK\_d 函数, 在 CREATE\_CHILD\_SA 交换进行交换的临时随机数, 以及 Diffie-Hellman 值 (如果在 CREATE\_CHILD\_SA 交换中包含了 KE 载荷)。

当 CHILD\_SA 作为初始交换的一部分时, 第二个 KE 的载荷和临时随机数不需要被发送。来自初始交换的临时随机数被用于计算 CHILD\_SA 的密钥。

CREATE\_CHILD\_SA 请求包括:

发起者

响应者

HDR, SK {[N], SA, Ni, [KEi]},

[TSi, TSr]}

→

发起者在 SA 载荷中发送 SA, 在 Ni 载荷中发送临时随机数, 可选在 KEi 载荷中发送 Diffie-Hellman 值, 以及在 TSi 和 TSr 载荷中发送建议的流量选择器。如果这个 CREATE\_CHILD\_SA 交换是为一个不为 IKE\_SA 的, 已建立的 SA 重新协商密钥时, 第一个 REKEY\_SA 类型的 N 载荷必须表明这是 SA 重新分配密钥。如果这个 CREATE\_CHILD\_SA 交换不是给一个已经存在的 SA 重新分配密钥, 这 N 载荷必须被忽略。如果 SA 包括不同的 Diffie-Hellman 组, KEi 必须是发起者期望接收者接收的组的元素。如果他猜错了, 那么 CREATE\_CHILD\_SA 交换将失败, 并且它将使用不同的 KEi 重试。

紧随着 IKE 头的消息经过加密, 这些消息包括头, 由被 IKE\_SA 协商的加密算法进行完整性保护。

CREATE\_CHILD\_SA 响应包括:

<-- HDR, SK {SA, Nr, [KEr]},

[TSi, TSr]}

响应者在 SA 载荷中回应接收到的 SA (使用相同的消息 ID), 并且如果请求中包含 KEi, 以及被选择的加密族包括那个组, 响应者将 Diffie-Hellman 值包含在 KEr 载荷中。如果响应者选择了不同组的加密族, 他必须拒绝这个请求。发起者应该使用来自响应者选择的组中的 KEi 载荷, 重复发送请求。

TS 载荷中规定了用于送往该 SA 流量的流量选择器, 它可以是 CHILD\_SA 建议的发起者的一个子集。如果这个 CREATE\_CHILD\_SA 轻轻被用于交换 IKE\_SA 密钥, 流量选择器应该被忽略。

#### 4.5 INFORMATIONAL 交换

在 IKE\_SA 操作的不同阶段，对方可以要求互相传送控制信息，而不管特定事件的错误或者错误通知。为了完成这些，IKE 定义了 INFORMATIONAL 交换。INFORMATIONAL 交换必须只在初始交换完成之后出现，并且使用协商的密钥进行加密保护。

属于 IKE\_SA 的控制信息必须在那个 IKE\_SA 下被发送。属于 CHILD\_SA 的控制信息必须在产生他们的 IKE\_SA 的保护下被传送（或者，如果 IKE\_SA 因为重新分配密钥的原因被替代了，应该在 IKE\_SA 的继任者的保护下被传送）。

INFORMATIONAL 交换的信息包含 0 个或者多个错误通知载荷，删除载荷和配置载荷。INFORMATIONAL 交换请求的接收者必须发送响应（否则发送者将认为这些消息在网络传输中丢失了，并重传他们）。这些响应必须是没有载荷的消息。INFORMATIONAL 交换的请求消息也可以不包含任何载荷。这种情况在一个端点验证另一个端点是否存活的情况下出现。

ESP 和 AH 的 SA 总是成对出现的，即每个方向上都存在着一个 SA。当一个 SA 被关闭时，该对中的两个成员都必须被关闭。当 SA 被嵌套时，即相同端点对之间的数据（如果在隧道模式下，包括 IP 头）首先使用 IPComp 封装，然后使用 ESP，最后使用 AH，所有的 SA 都必须被一起删除。任意一个端点必须关闭它的输入 SA，并允许另一个端点关闭另一个 SA。为了删除一个 SA，具有一个或者多个删除载荷的 INFORMATIONAL 交换被发送，该交换中列出了被删除 SA 的 SPI（当它们可能会出现在入站数据包的头中时）。接收者必须关闭指定的 SA。通常，INFORMATIONAL 交换的响应包含删除载荷，该载荷用于去往其他方向的 SA。但是也有一个例外情况。如果碰巧，一组 SA 的两端同时单独决定关闭它们，每一个端点都可能发送一个删除载荷，这样两个方向上的请求可能在网络中一起出现。如果一个节点收到了对 SA 的删除请求，但是它已经发出了删除该 SA 的请求，该节点必须在处理请求时，删除输出的 SA；而在处理响应时，删除输入 SA。在那种情况下，响应不必包含被删除 SA 的删除载荷，因为那样将导致多个删除，并且在理论上可能删除错误的 SA。

节点应将半关闭的连接看作是反常的，并且审计他们应该保持的连接。注意，本标准没有规定时间期，应该由独立的端点来决定应等待多长时间。节点可以拒绝接收来自半关闭连接的数据，但不能单方面的关闭他们，以及重用 SPI。如果连接状态已经变得足够混乱，节点可以关闭 IKE\_SA；这样做将隐含关闭在其之下协商的所有 SA。然后它可以在一个新的 IKE\_SA 下，一个干净的数据库中重建 SA。

INFORMATIONAL 交换定义如下：

发起者	接收者
-----	-----

---

HDR, SK {[N,] [D,] [CP,] ...} -->

<-- HDR, SK {[N,] [D,] [CP,] ...}

INFORMATIONAL 交换的处理由它的载荷构成决定。

#### 4.6 IKE\_SA 之外的 INFORMATIONAL 消息

如果具有不可知的 SPI 加密的 IKE 数据包到达端口 500 或者 4500，可能是因为接收节点最近刚刚宕机，或者状态丢失，也可能因为其他的系统故障或者攻击。如果接收节点具有到数据包源 IP 地址的激活的 IKE\_SA，它可以通过 INFORMATIONAL 交换经由这个 IKE\_SA 发送不正确数据包的通知。如果接收节点不具备这样的 IKE\_SA，它可以发送一个没有加密保护的 INFORMATIONAL 消息到源 IP 地址。该

消息不是 INFORMATIONAL 交换的一部分，接收节点不需要响应它。进行这样的操作可能会形成消息环路。

## 5 IKE 协议细节及变化

### 5.1 概述

尽管可能在 UDP 端口 4500 以稍有不同的格式接收到 IKE 消息，但 IKE 通常在 UDP 端口 500 倾听和发送。由于 UDP 是不可靠的数据包协议，IKE 包含了从传输错误中恢复的规定，包括丢包、数据包重放、数据包伪造。IKE 设计成发生以下情况时还可以工作：(1) 在超时之前，被传输的一系列数据包中至少有一个数据包到达了目的地。(2) 信道还没有充满了重放和伪造的数据，以至于耗尽网络或任一端节点的 CPU 资源。即使没有这些最低的性能要求，IKE 设计成可以明确表示失效的协议。

所有 IKEv2 的实现应能够发送、接收并处理最长 1280 字节的 IKE 消息，并建议能够发送、接收并处理最长 3000 字节的消息。建议 IKEv2 的实现考虑所支持的最大 UDP 消息长度，并且如果通过丢弃一些证书或加密族建议可使得消息长度小于最大值，则可通过该方法使消息变短。可能时，使用“哈希 (Hash) 与 URL”格式，而不是使用证书，可避免大多数问题。但是，要牢记实现及配置，如果只有在 IPsec SA 实现后才能进行 URL 查找，递归问题会导致该技术无法工作。

### 5.2 重传定时器的使用

所有 IKE 消息以请求和响应成对出现。IKE\_SA 的建立通常包含两个请求/响应消息对。一旦 IKE\_SA 建立，安全联盟的任一端都能在任何时候发起请求，在任一给定的时刻，可能有许多传输中的请求和响应。但每一个消息只能被标记为请求或响应，对每一个请求/响应对，安全联盟的一端是发起者，另一端是响应者。

对每一对 IKE 消息，发起者负责超时重传。除非收到重传的请求，响应者从来不应重传响应。在这种情况下，响应者在考虑重传的请求触发响应的重传之外，应忽略重传的请求。发起者应记住每个请求直到收到了相应的响应。响应者应记住每个响应直到收到了一个请求且该请求的序列号大于响应中的序列号与窗口大小的和（见 5.4 节）。

IKE 是一个可靠的协议，因为发起者应重传请求直到收到了相应的回复或者可以确信 IKE 安全联盟失效并丢弃了所有与 IKE\_SA 及协商使用该 IKE\_SA 的任一 CHILD\_SA 相关联的所有状态。

### 5.3 消息 ID 序列号的使用

每一个 IKE 消息都包含一个消息 ID 作为其固定消息头的一部分。该消息 ID 用于请求和响应的匹配，并用来识别重传的消息。

消息 ID 为 32 比特位，对于每一个方向上的第一个 IKE 请求值为 0。最初 IKE\_SA 建立的消息序号总是 0 或者 1。IKE 安全联盟的每一个终端维护两个“当前”消息 ID：将要发起的下一个请求的 ID，期望从对端收到的下一个请求的 ID。这些计数器随着发起或接收到的请求而增加。响应总是与相应的请求具有相同的 ID。这意味着在初始交换之后，每一个整数  $n$  都可能作为消息 ID 在 4 种不同的消息中出现：起始 IKE 发起者的第  $n$  个请求及相应的应答，起始 IKE 响应者的第  $n$  个请求及相应的应答。如果两端使用差异很大的请求序号，那么两个方向的消息 ID 也将会有很大差异。消息头的 I 和 R 比特位指定了消息属于以上 4 种中的哪一种，因为消息的类型是非常明确的。

消息 ID 是加密的，用来防止重放攻击。一种不太可能的情况是消息 ID 增长到太大能填充完 32 比特，此时 IKE\_SA 应被终止。重新建立 IKE\_SA 来重设序列号。

#### 5.4 请求窗口的大小重叠

为了最大化 IKE 的吞吐量, IKE 终端在收到任一个所发起请求的响应之前, 可能会发起多个请求, 如果 IKE 的另一终端已经显示了处理这些请求的能力。为了简单, IKE 的实现可对请求按顺序进行严格的处理, 和/或在发起另一个请求之前, 等待某个请求的响应到达。应遵循一定的规则, 以确保使用不同策略的实现之间可进行互操作。

IKE\_SA 建立之后, 任一端可发起一个或多个请求。这些请求可能依次通过网络。当 IKE 终端有一个未处理的请求时, IKE 端点应准备接收并处理请求以避免出现死锁。建议 IKE 终端在有一个未处理请求的情况下, 准备接收并处理多个请求。

在发送后续的消息之前, IKE 端点应为它发起的每个消息等待响应除非从对端接收到 SET\_WINDOW\_SIZE 的通知报文, 该通知报文说明对端准备维护多个未处理消息的状态以便提高吞吐量。

IKE 终端传送的消息不应超过对端说明的窗口大小。换句话说, 如果响应者说明其窗口大小为 N, 那么当发起者需要发起请求 X 时, 它应等待直到接收到从 X-N 序号起的所有请求的响应。IKE 终端应为它发起的每个请求保留一份拷贝(或能够准确重新生成每一个请求)直到收到了相应的响应。IKE 端点应保留一定数量的以前响应的拷贝(或能够准确重新生成这些响应), 响应的数量应等于它所宣称的窗口的大小, 以备响应丢失且发起者通过重传请求而要求重传响应。

建议所支持窗口大小大于 1 的 IKE 终端要具有处理无序请求的能力, 以此来最大化网络出错或数据包失序时的性能。

#### 5.5 状态同步和连接超时

允许 IKE 终端在任何时刻忘记所有与 IKE\_SA 及相应的 CHILD\_SA 集合关联的所有该终端的状态。这是终端失效或重启时预期会出现的状况。重要的是当终端失效或重新初始化它的状态时, 对端能检测到这些情况并不再继续在已废弃的 SA 发送数据包浪费带宽并使这些数据包掉入黑洞。

由于 IKE 设计为在不考虑来自网络的 DOS 攻击的情况下工作, 终端不应根据任何路由信息(例如, 由于 IKE 被设计为即使来自网络的 DOS 攻击也能运行 ICMP 信息)或收到的没有加密保护的 IKE 信息(例如, 未知 SPI 的通知消息)推断得出其他终端失效的结论。终端只应在以下两种情况可推断得知其他终端失效: 1) 不断尝试连接该节点, 但发生超时且未得到响应。2) 在不同的 IKE\_SA 收到给同一个经过认证实体的加密 INITIAL\_CONTACT 通知。建议终端基于路由信息猜想另一个终端失效并发起请求来核实该终端是否处于激活状态。为了验证另一终端是否处于激活状态, IKE 指定了一个要求给出确认的空的 INFORMATIONAL 消息(类似于所有的 IKE 请求)(需要注意的是, 在 IKE\_SA 的上下文中, 空的消息由 IKE 头及其后面的没有载荷的加密载荷字段组成)。如果最近收到了来自对端的受加密保护的消息, 未经加密保护的通知则可被忽略。这种实现方式应限制基于未加密消息采取行动的频率。

本标准没有涉及重试的次数和超时的长度因为这些不影响互操作性。建议在放弃一个 SA 时, 在经过至少几分钟后, 要最少尝试重传十几次, 但是不同的环境需要不同的规则。重传时间间隔应指数增加以避免网络上的洪泛, 并网络拥塞状况更加严重。如果与某一 IKE\_SA 相关的所有 SA 都只有流出的流量, 那么有必要确认其他端点是否处于激活状态而避免出现黑洞。如果 最近没有从 IKE\_SA 或任一 CHILD\_SA 收到加密消息, 系统需进行生存性验证以防止向失效节点发送消息。在 IKE\_SA 或任一 CHILD\_SA 收到最新的加密消息可以确保 IKE\_SA 及其所有 CHILD\_SA 是处于激活状态的。需要注意的是

这对 IKE 终端的失效模式提出了要求。如果某些失效使得终端不能从所有关联的 SA 接收消息，该实现方式不应继续向任何 SA 发送消息。如果 CHILD\_SA 失效与 SA 中的两方无关，且关联的 IKE\_SA 无法发送出一个删除消息，它们必须重新协商另一个 IKE\_SA。

如果 IKE\_SA 的发起者采取适当的措施，那么一类针对它的拒绝服务攻击是可以避免的。由于在 SA 建立时的最初两个消息是不进行加密保护的，攻击者可以在真正的响应者进行回复之前对发起者的消息进行响应并破坏连接建立。为了防止该情况发生，针对第一个消息，发起者可接收多个响应，把每个响应都当做可能是合法的，并进行响应，然后当它接收到有效的针对任一发起请求的加密响应时，丢弃所有无效的半开放连接。一旦收到一个加密的有效响应，所有后续的响应不管是加密有效的都将被忽略。

需要注意的是，使用这些的规则后，没有理由对 SA 的生存时间进行协商并达成一致。如果基于对 IKE 消息应答的重复缺失，IKE 认为对方失效了，那么 IKE\_SA 及其所有的 CHILD\_SA 将通过删除 IKE\_SA 而建立。

IKE 端点可在任何时候删除非激活状态的 CHILD\_AS 来释放维护其状态的资源。如果一个 IKE 端点选择删除 CHILD\_AS，应发送删除载荷到另一端来通知删除操作。关闭 IKE\_SA 的情况类似。关闭 IKE\_SA 意味着关闭了所有相关的 CHILD\_SA。在这种情况下，建议 IKE 端点发送删除载荷说明它关闭了 IKE\_SA。

## 5.6 版本号和前向兼容性

本标准描述了 IKE 2.0，主版本号是 2，二级版本号是 0。很可能某些实现方式要既支持 1.0 版本，又支持 2.0 版本，并且还有将来的其他版本。

只有当包格式或所需的操作发生了重大的改变，使得旧版本的节点在简单地忽略它所不理解的字段并采取了旧规范中规定的操作，不能与新版本节点进行交互时，才建议增长主版本号。二级版本号代表了新的能力，具有较小二级版本号的节点应忽略二级版本号，而具有较大二级版本号的节点可将该号码用于信息目的。例如，二级版本号可代表处理新定义的通知消息的能力，拥有较大二级版本号的节点会注意到他的对端节点没有理解该消息的能力，因此不发送该消息。

如果终端接收到具有更高主版本号的消息，应丢弃该消息，并建议该终端发送包含它所支持的最高版本的未认证通知消息。如果终端支持主版本号  $n$  和主版本号  $m$ ，则应支持  $n$  和  $m$  之间的所有版本。如果终端接收到了它所支持的主版本的消息，则应利用该版本号进行响应。为了防止两个节点被欺诈而用低于他们所能支持的最高版本的主版本号进行响应，IKE 设计了一个标志来标记节点能支持的更高主版本号。

因此，IKE 头部的主版本号字段代表了消息的版本号，而不是传输者所支持的最高版本号。如果发起者所能支持的版本为  $n, n+1, n+2$ ，而响应者能支持的版本为  $n, n+1$ ，那么他们应协商支持版本  $n+1$ ，发起者会设置标志符说明具有支持更高版本的能力。如果他们错误地（可能由于积极的攻击者发送错误的消息造成）协商使用版本  $n$ ，那么双方会注意到对方可支持更高的版本，并且双方应中断连接，并重新使用版本  $n+1$  建立连接。

同样为了实现前向兼容性，在 2.0 版本的实现中，所有标记为保留的字段应设为 0，且该内容应被 2.0 版本的实现忽略。通过该发现，将来的协议版本可以使用这些字段且确保不理解这些字段的实现可将其忽略。类似地，没有定义的载荷类型是保留以备将来使用的；2.0 版本的实现应跳过这些载荷并忽略其内容。

IKEv2 为每一个载荷头部添加了“紧急”标志使得将来能够灵活的向前兼容。如果设置了紧急字段且载荷类型是不可识别的，那么应拒绝该消息，且针对包含该载荷的 IKE 请求的响应应包含一个通知载荷 UNSUPPORTED\_CRITICAL\_PAYLOAD，来表明不支持所包含的紧急载荷。如果紧急标志没有设置且载荷类型不支持，则应忽略该载荷。

尽管在未来可能添加新的载荷类型，也可能出现在本标准中定义的字段中插入新载荷的情况，实现应以第二节中定义的顺序发送本标准中定义的载荷，且如果载荷是以其他顺序排列的，建议实现拒绝该无效消息。

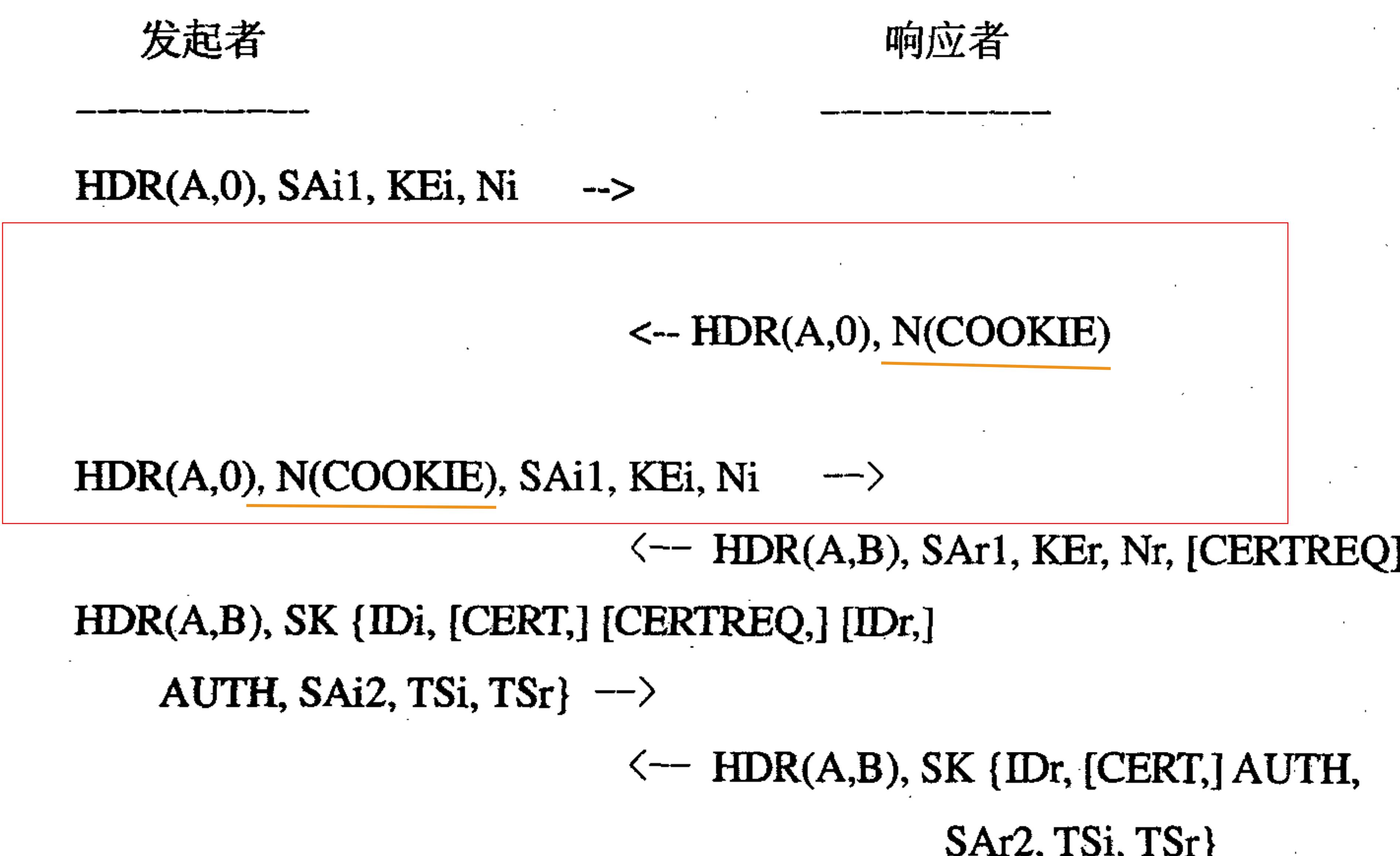
### 5.7 Cookies

互联网安全联盟和密钥管理协议 ISAKMP 确定了消息头包含命名为“cookies”的 2 个 8 字节，在 IKEv1 和 IKEv2 中都使用了该语法，尽管在 IKEv2 中这些字节用作 IKE SPI，且在通知载荷中有单独的字段标志 cookie，在 IKE 包开始部分的，头部中的开始的 2 个 8 字节是作为连接 ID 的。每个终端选择两个 SPI 中的一个并建议选择使其能成为 IKE\_SA 的惟一 ID。值为 0 的 SPI 是特殊的，表明发送者仍不知道远程 SPI 的值。

与 ESP 和 AH 在消息头中仅有接收者的 SPI 不同，在 IKE 中，发送者的 SPI 同样出现在每个消息中。由于由 IKE\_SA 发起者选择的 SPI 总是首先发送，拥有多个 IKE\_SA 的终端，如果想要发现使用它指定的 SPI 的合适的 IKE\_SA，应查看包头的发起者标志位决定它所指定的是第一个还是第二个 8 字节。

在初始 IKE 交换的第一个消息中，发起者不知道响应者的 SPI 值，因此将该字段设置为 0。

为减少受攻击的可能，当监测到有大量的半开放 IKE\_SA 时，建议响应者拒绝最初的 IKE 消息，除非这些消息包含类型为 COOKIE 的通知载荷。建议发送未经保护的 IKE 消息作为响应，该响应包含希望返回的 cookie 数据的 COOKIE 通知消息。收到这类响应的发起者应重试 IKE\_SA\_INIT 消息，该消息包含类型为 COOKIE 的通知载荷，该载荷将响应者提供的 cookie 数据作为第一载荷，其余载荷不变。初始交换过程如下：



除了交换 cookie，最初的两个消息不会影响任何发起者或响应者的状态。特别地，最初的四个消息的消息序列号都是 0，最后的两个消息的消息序列号是 1。A 是发起者指定的 SPI，而 B 是响应者指定的 SPI。

建议 IKE 实现采用这样的方法来进行响应者的 cookie 生成，因为这样的方法使得在第二个 IKE\_SA\_INIT 消息到来时，不需要使用任何保存的状态来识别有效的 cookie。具体产生 cookie 的算法和语法不影响互操作性，因此这里不进行规范。以下是终端如何利用 cookie 来实现有限的 DOS 攻击保护的例子。

一个很好的实现该目标的方法是，响应者按如下方式设置 cookie：

$\text{Cookie} = \langle \text{VersionIDofSecret} \rangle \mid \text{Hash}(\text{Ni} \mid \text{IPi} \mid \text{SPIi} \mid \langle \text{secret} \rangle)$

这里  $\langle \text{secret} \rangle$  是只有响应者知道的随机产生并会周期改变的密文， $\mid$  为连接符。一旦重新产生  $\langle \text{secret} \rangle$ ，则改变  $\langle \text{VersionIDofSecret} \rangle$ 。当 IKE\_SA\_INIT 第二次到达时，则重新计算 cookie 并与接收到的消息的 cookie 进行比较。如果匹配，那么响应者知道该 cookie 是在  $\langle \text{secret} \rangle$  最近一次改变后生成的，且 IPi 应与它第一次看到的源地址相同。结合 SPIi 进行计算确保如果多个 IKE\_SA 并行建立，他们都将得到不同的 cookie（假设发起者选择唯一的 SPIi），结合 Ni 进行哈希确保只看到消息 2 的攻击者不能成功的伪造消息 3。

如果正在连接进行初始化时，为  $\langle \text{secret} \rangle$  选择了一个新的值，而是 IKE\_SA\_INIT 可能被返回，带有新的  $\langle \text{VersionIDofSecret} \rangle$ 。在该情况下，响应者可通过发送具有新 cookie 的其他响应来拒绝该消息或者可保持原来的  $\langle \text{secret} \rangle$  值一段时间并接收任意一方计算的 cookie 值。建议响应者频繁的更换  $\langle \text{secret} \rangle$  的值，尤其在被攻击时。如果响应者在回应了 COOKIE 通知后，值更新了，则发起者发来的带有 COOKIE 的 IKE\_SA\_INIT 消息，还是使用的那个过时的，与当前更新后的不匹配了，在这种情况下，响应者可再次发送 COOKIE 通知（使用新的），也可以（实现为）保持原来的变更前一段时间。

## 5.8 加密算法协商

载荷类型“SA”表明 SA 的 IPsec 协议（IKE，ESP，和/或 AH）的一组选择的建议以及与每个协议相关的加密算法。

一个 SA 载荷可能包含一个或多个建议，每个建议包含一个或多个协议（通常是一个）。每个协议包含一种或多种指定了一个加密算法变换。每种变换包含 0 个或多个属性（只有当变换的 ID 不能完全指定一个加密算法时，才需要属性）。

因为多个值对于多个变换是可接受的，所以设计了一种层次化的结构，在支持的套件的数量较大时，对加密套件建议进行有效的编码。响应者应选择单个套件，该套件可以是遵循如下规则的 SA 建议的任何子集。

每个建议包含一个或多个协议。如果接受了一个建议，SA 响应应包含与建议中的顺序一致的相同的协议。响应者应接受一个单一的建议或拒绝所有的建议并返回一个错误（例如：如果一个单独的建议包含 ESP 和 AH 且该建议被接受，那么 ESP 和 AH 都应被接受。如果 ESP 和 AH 包含在独立的建议里，那么响应者应只接受其中的一个）。

每个 IPsec 协议建议包含一个或多个变换，每个变换包含一个变换类型。可接受的加密套件应完全包含建议中每个类型的一个变换。例如：如果一个 ESP 协议包括变换 ENCR\_3DES, ENCR\_AES w/keysiz 128, ENCR\_AESw/keysiz 256, AUTH\_HMAC\_MD5 和 AUTH\_HMAC\_SHA，那么，可接受的套件应包含一个 ENCR 变换和一个 AUTH 变换。因此，有 6 种组合是可以接受的。

由于发起者在 IKE\_SA\_INIT 中发送了 Diffie-Hellman 值，它一定推测了响应者可能从它列出的支持的组中选择的 Diffie-Hellman 组。如果发起者猜错了，响应者将发送类型为 INVALID\_KE\_PAYLOAD 的通知载荷表明它选择的组。在这种情况下，发起者应重试具有正确 Diffie-Hellman 组的 IKE\_SA\_INIT 消息。而且，发起者应再次提交整套可接受的加密组件，因为拒绝消息是未经认证的，而积极的攻击者可

能欺骗终端协商较弱的加密套件。

## 5.9 重新建立安全联盟

建议 IKE, ESP 和 AH 安全联盟使用的密钥只使用有限的次数，只保护有限的数据。这限制了整个安全联盟的生存时间。当一个安全套件过期时，安全联盟不应再继续使用。如果有需要，可建立新的安全管理。为了替代已经过期的安全联盟，安全联盟的重新建立称之为“密钥重协商”。

考虑到最低限度的 IPsec 实现方式，重建 SA 而不重启整个 IKE\_SA 的能力是可选的。实现方式可在 IKE\_SA 内拒绝所有的 CREATE\_CHILD\_SA。如果一个 SA 已经过期或者将要过期，且使用这里描述的机制的重建企图失败了，实现方式应关闭 IKE\_SA 及其任何相关的 CHILD\_SA，并可重启新的 IKE\_SA 或 CHILD\_SA。建议实现方式支持只重建 SA，由于这样可以提供更好的性能且在传输期间可能会减少丢包的数量。

为了在现有的 IKE\_SA 内重建一个 CHILD\_SA，要创建一个新的、等价的 SA（见第 5.18 节），并当新的建立起来后，删除旧的。为了重建一个 IKE\_SA，要与对端建立一个新的等价的 IKE\_SA（见 5.19 节），该对端是在原有的 IKE\_SA 内。这样建立的 IKE\_SA 继承了所有原始的 IKE\_SA 的 CHILD\_SA。为了维护由旧的 IKE\_SA 创建的 CHILD\_SA，需要对所有控制消息使用新的 IKE\_SA，并删除旧的 IKE\_SA。将自己删除的删除载荷应是在 IKE\_SA 上发送的最后一个请求。

建议主动重建 SA，也就是说，新的 SA 最好在旧的 SA 过期并不可用之前建立。建议在旧的 SA 不可用与新的 SA 建立之间保留足够的时间间隔，以备将流量倒换到新的 SA。

在 IKEv2 中，SA 的每一端负责执行自己的在 SA 的生存时间策略并在需要的时候重建 SA。如果两端有不同的生存时间策略，具有更短生存时间的一端常常是要求重建 SA 的一端。如果一个 SA 已经处于不活动状态很长时间，且如果一端不想在没有流量的时候初始化 SA。当生存时间过期时，该端点可选择关闭 SA 而不是重建。从最近一次 SA 被重建之后，如果已经没有流量，建议端点这么处理。

如果两端有相同的生存时间策略，有可能两端同时发起重建（这将导致冗余的 SA）。为了减少这种情况发生的可能性，建议对重建请求的时间随机调整（在需要重建时，推迟一个随机的时长）。

这种形式的重建会临时导致在同一节点对之间存在多个相似的 SA。当有两个 SA 符合条件接收数据包时，一个节点应从任一个 SA 接收到达的数据包。如果尽管存在冲突，但仍建立了冗余的 SA，建议将某个 SA 由创建该 SA 的终端关闭，该 SA 具有在两次交换中使用的四个临时随机数中最小的临时随机数。

IKEv2 故意允许在普通的终端之间有具有相同流量选择器的并行的 SA。这么处理的一个目的就是为了支持不同 SA 之间的流量 QoS 差异。

对存在的 SA 初始化重建的节点，在新 SA 建立后，删除被替代的 SA。

对 CREATE\_CHILD\_SA 进行响应的响应者应在发送针对创建请求的响应之前准备接收某一 SA 上的消息，因此对于发起者而言没有不确定性。发起者可在一处理完请求后就在某一 SA 上发送。然而，直到接收到并处理了它的 CREATE\_CHILD\_SA 请求的响应，发起者都不能从新创建的 SA 接收消息。那么响应者如何知道什么时候可以在一个新创建的 SA 发送信息呢？

从技术正确性和互操作性的角度来看，响应者只要发送了对 CREATE\_CHILD\_SA 请求的响应就可在 SA 上发送。然而，在某些情况下，这会导致不必要的丢包。因此，实现中可推迟这样的发送。

如果满足以下任一种情况，响应者可确信发起者准备在 SA 上接收消息：(1) 在新的 SA 接收到了加密的有效消息；(2) 新的 SA 重建并替代一个已有的 SA 并且响应者收到了关闭被替换掉的 SA 的 IKE

请求。当重建一个 SA 时，建议响应者持续在旧的 SA 上发送消息直到发生了以上某个事件。当建立一个新的 SA 时，响应者可拒绝在新的 SA 上发送消息，直到它收到了一个消息或发生超时。如果一个发起者从某一 SA 收到了消息，而针对该 SA 它没有收到它发起的 CREATE\_CHILD\_SA 请求的响应，建议将这种情况理解为发生了丢包并重传 CREATE\_CHILD\_SA 请求。如果没有消息队列，为了使响应者确信发起者准备接收消息，那么发起者可在新创建的 SA 上发送一个虚拟的消息。

### 5.10 流量选择器协商

当一个 IPsec 子系统接收到一个 IP 包，且该 IP 包与安全策略数据库 SPD 中的“保护”选择器匹配，则该子系统应使用 IPsec 保护该数据包。如果仍没有 SA 存在，IKE 要创建一个 SA。维护系统的 SPD 不在本标准内规定。

流量选择器 (TS, Traffic Selector) 载荷允许终端交互来自他们 SPD 的信息。TS 载荷指定了将包转发跳过新建立的 SA 的包选择规则。这可以在某些场景下作为一致性检查来确保 SPD 是一致的。另一方面，可指导 SPD 的动态更新。

在建立 CHILD\_AS 对的交换中，每个消息都有两个 TS 载荷。每个 TS 载荷包含一个或多个流量选择器。每个流量选择器由一个地址范围 (IPv4 或 IPv6)，一个端口范围和一个 IP 协议 ID 组成。为了支持 4.2.3 节描述的场景，发起者可要求响应者指定一个 IP 地址，并告诉发起者该地址是什么？

IKEv2 允许响应者在发起者提出的流量中选择一个子集。在两个端点的配置信息都进行了更新，但只有一个端点收到新信息时，这种情况可能发生。由于两个端点可能是由不同的人配置的，因此，可能在一段时间内即使没有错误也会出现不兼容的情况。同时也考虑到了故意设置的不同配置情况，例如当一端配置为依赖于另一端给出最新的列表，将所有地址以隧道传输。

两个 TS 载荷中的第一个是 TSi (Traffic Selector-initiator)，第二个是 TSr (Traffic Selector-responder)。TSi 指定了从 CHILD\_SA 对的发起者转发来的流量的源地址（或者该流量要被转发到什么地方的目的地址）。TSr 指定了要转发给 CHILD\_SA 对的响应者的流量的目的地址（或者该流量是从哪里转发来的源地址）。例如，如果原始的发起者请求创建一个 CHILD\_SA 对，并希望将发起者端的所有来自子网 192.0.1.\* 的流量通过隧道机制传输到响应者端的子网 192.0.2.\*，那么发起者应在每个 TS 载荷包含一个单独的流量选择器。TSi 将指定地址范围 (192.0.1.0-192.0.1.255)，TSr 将指定地址范围 (192.0.2.0-192.0.2.255)。假设接收者接受了该建议，它将返回相同的 TS 载荷。

响应者允许通过选择一个流量的子集来缩小选择范围，例如在集合不会变成空集的情况下，通过消去或缩小流量选择集合一个或多个成员的范围来实现。

对响应者的策略来说，包含多个更小的范围是可行的，这些较小范围都包含在发起者的流量选择器中，每个这些具有较小范围的策略将在不同的 SA 上发送。继续以上的例子，响应者可能有一个策略是将所有这些地址通过隧道机制发送到或接收自发起者，但要求每一个地址对在一个单独协商的 CHILD\_SA。如果响应者没有办法决定该隧道中应包含哪对地址，不得不进行推测或利用 SINGLE\_PAIR\_REQUIRED 状态拒绝该请求。

为了使响应者能够在这种情况下选择合适的范围，如果发起者为某些数据包传送提出建立 SA 要求，建议发起者在每个 TSi 和 TSr 包含一个特定的流量选择器作为第一个流量选择器，该流量选择器包含触发该请求的数据包中的 IP 地址。在上个例子中，发起者将在 TSi 中包含两个流量选择器：第一个包含地址范围 (192.0.1.43-192.0.1.43) 以及数据包中的源端口和 IP 协议，第二个包含有所有端口的地址范围

(192.0.1.0-192.0.1.255) 和 IP 协议。类似地，发起者将在 TSr 中包含两个流量选择器。

如果响应者的策略不允许接受发起者请求中的一整套流量选择器，但允许接受 TSi 和 TSr 中的第一个选择器，那么响应者应将流量选择器的范围缩小到一个子集，该子集包含发起者的一个选择。在这个例子中，响应者可用 TSi 消息进行响应，该 TSi 消息是所有端口的地址范围 (192.0.1.43-192.0.1.43) 和 IP 协议。

如果发起者创建了 CHILD\_SA，不是响应到达的数据包，而是在启动 SA，那么可能发起者没有选择特定的地址用于初始的隧道。在这种情况下，在 TSi 和 TSr 中的第一个值可以是范围而不是特定的值，响应者选择可接受的发起者 TSi 和 TSr 中的一个子集。如果多于一个子集可接受，但他们的合集不可接受，响应者应接受多个子集，可包括类型为 ADDITIONAL\_TS\_POSSIBLE 的通知载荷来表明发起者可能想重试。只有当发起者和响应者的配置不同时，这种情况才可能发生。如果发起者和响应者同意隧道的颗粒度。发起者将不会请求接受者建立超过其接受范围的隧道。这类错误配置将记录在错误日志中。

### 5.11 临时随机数

每个 IKE\_SA\_INIT 消息都包含一个 Nonce。这些数是加密函数的输入。CREATE\_CHILD\_SA 请求和 CREATE\_CHILD\_SA 响应也包含 Nonce。这些 Nonce 是用于为密钥导出技术添加时效性的，该密钥导出技术用于为 CHILD\_SA 获取密钥并确保从 Diffie-Hellman 密钥得出强伪随机比特。用于 IKEv2 的临时随机数应是随机选择的，应至少有 128 比特，应至少为协商的 prf 的一半密钥的长度 (prf 指伪随机函数，是在 IKE 交换期间协商的加密算法)。如果为密钥和临时随机数使用了相同的随机数源，必须小心以确保后一个的使用不会危及前一个的安全。

发送者获知需要nat转换后，则把发送的ike包，用udp封装，并通过4500端口发送，但经过nat/nap转换后，在公网上，就不是这个ip-port了，对于这样的包，要求接收者不能限制接收的端口，否则就接不到这样的包了

### 5.12 地址和端口灵活性

IKE 运行于 UDP 端口 500 和 4500，并默认地为它所运行的 IP 地址上建立 ESP 和 AH 关联。在外部头的 IP 地址和端口不是受自己加密保护的，IKE 被设计能够在即使存在 NAT 转换器的情况下工作。实现方式应接受到达的消息即使源端口不是 500 或 4500，应向接收请求的地址和端口进行响应。实现方式应将接受到请求的地址和端口作为响应的源地址和端口。对于 IPv4 和 IPv6 而言，IKE 功能是一致的。

### 5.13 Diffie-Hellman 指数的重用

为了实现完美的转发保密性，IKE 使用短暂记忆的 Diffie-Hellman 交换来产生密钥生成资源。这意味着一旦一个连接关闭，那么相应的密钥也被忘记了，即使有人记录了连接上的所有数据并能够进入两个终端之间所有的长期密钥，那么他也不能不对会话密钥空间进行暴力搜索而重新生成用于保护会话的密钥。

实现完美的转发保密性要求当连接关闭时，每个终端应不仅忘记用于该连接的密钥，同时忘记可用于重新计算这些密钥的任何信息。特别地，应忘记用于 Diffie-Hellman 计算的密钥以及任何可能在伪随机生成器的状态（机）中持续的状态，这些状态可能被用于重新计算 Diffie-Hellman 密钥。

由于 Diffie-Hellman 指数的计算具有昂贵的计算成本，终端可能会将这些指数重用于多个连接的建立。在重用时有几个合理的策略。终端可以仅仅周期性的选择一个新的指数，尽管如果某些连接持续的时间小于指数的生存时间，这可能导致无法达到完美的转发保密性。或者终端可以跟踪每个连接使用的指数，并只有当相应的连接关闭时，删除指数相关的信息。这使得在进行指数重用时，可以用维护更多状态的代价换取理想的转发保密性。

决定是否重用及何时重用 Diffie-Hellman 指数是一个个人的决定，因为这不影响互操作性。重用指

数的实现可选择记住在过去的交换中其他端点使用的指数，且如果重用了一个指数可避免一半的计算。

### 5.14 密钥生成资源

在 IKE\_SA 的上下文中，协商了四个加密算法：加密算法、完整性保护算法、Diffie-Hellman 组和伪随机生成函数（prf）。伪随机生成函数用于构造所有加密算法的密钥生成资源，这些加密算法既用于 IKE\_SA，也用于 CHILD\_SA。

我们假设每个加密算法和完整性保护算法都使用固定长度的密钥且任何随机选择的这一固定长度的值都是一个合适的密钥。对于可接受变长密钥的算法，固定密钥的长度应作为加密变换（变换）协商的一部分被指定。对于不是所有值都是有效密钥的算法，那么从任意值中获取密钥的算法应在加密变换中被指定。对于基于哈希消息认证码（Hashed Message Authentication Code, HMAC）的完整性保护函数，固定密钥长度是基础哈希函数的输出的长度。当 prf 函数使用变长密钥时及变长的数据并产生固定长度的输出（例如，当使用 HMAC 时）时，那么本标准中的公式适用。当 prf 函数的密钥是固定长度时，除非说明了指明处理过程遵循以下公式，那么有必要对作为密钥而提供的数据进行截取的或用 0 补齐。

密钥生成资源常常是从协商好的 prf 算法的输出得到的。既然所需密钥生成资源的长度大于伪随机生成算法的长度，那我们可以迭代的使用 prf。我们将使用术语 prf+ 来按如下方式描述根据伪随机函数的输入而输出一个伪随机流的函数（这里 I 指串联）。

$$\text{prf}^+(K, S) = T_1 \mid T_2 \mid T_3 \mid T_4 \mid \dots$$

这里：

$$T_1 = \text{prf}(K, S \mid 0x01)$$

$$T_2 = \text{prf}(K, T_1 \mid S \mid 0x02)$$

$$T_3 = \text{prf}(K, T_2 \mid S \mid 0x03)$$

$$T_4 = \text{prf}(K, T_3 \mid S \mid 0x04)$$

要计算所有需要的密钥，需继续以上计算。密钥是从输出的流中不考虑边界获得的。

连接到每一个用于作为 prf 种子的流结尾的常数是一个单独的字节，超出 255 与 prf 输出的长度乘积的 prf+ 在本标准中没有定义。

### 5.15 为 IKE\_SA 生成密钥生成资源

共享的密钥按如下方式计算。称为 SKEYSEED 的数由在 IKE\_SA\_INIT 交换期间交换的临时随机数计算而来，且在此交换期间建立了 Diffie-Hellman 共享密钥。SKEYSEED 用于计算七个其他的密钥：用于为 CHILD\_AS 传送新密钥的 SK\_d，该 CHILD\_AS 是和 IKE\_SA 一起建立的；用于完整性保护算法密钥的 SK\_ai 和 SK\_ar，该完整性保护算法用于对构成后续交换的消息进行鉴权。用于加密（当然也用于解密）所有后续交换的 SK\_ei 和 SK\_er；在产生 AUTH 载荷时使用的 SK\_pi 和 SK\_pr。

SKEYSEED 及其派生的密钥按如下方法计算：

$$\text{SKEYSEED} = \text{prf}(N_i \mid N_r, g^{ir})$$

$$\{SK_d \mid SK_{ai} \mid SK_{ar} \mid SK_{ei} \mid SK_{er} \mid SK_{pi} \mid SK_{pr}\} = \text{prf}^+$$

$$(SKEYSEED, N_i \mid N_r \mid SPI_i \mid SPI_r)$$

（这表明数值 SK\_d, SK\_ai, SK\_ar, SK\_ei, SK\_er, SK\_pi 和 SK\_pr 是按顺序从 prf+ 产生的比特位中生成的）。g<sup>ir</sup> 是来自于短暂的 Diffie-Hellman 交换的共享密钥。g<sup>ir</sup> 是高位在前顺序的一个八位字节流，在需要的时候会补 0 使其长度与模数的长度一致。N<sub>i</sub> 和 N<sub>r</sub> 是去掉了头部的临时随机数。如果协商的 prf

具有固定长度的密钥，且  $N_i$  和  $N_r$  的长度加起来达不到这个长度，那么一半的比特必须来自  $N_i$ ，另一半的比特来自  $N_r$ ，且都是前面的部分。

不同方向的流量使用的是不同的密钥。这些用于保护来自于最初发起者的消息的密钥是  $SK_{ai}$  和  $SK_{ei}$ 。在另一个方向上保护消息的密钥是  $SK_{ar}$  和  $SK_{er}$ 。每个算法使用密钥生成资源的固定长度的比特位，该密钥生成资源是作为算法的一部分被指定的。对于基于密钥哈希的完整性算法，密钥的长度总是等于基础哈希函数输出的长度。

### 5.16 IKE\_SA 的认证

当不使用扩展认证（见 5.17 节）时，通过让每个终端对一组数据进行签名（或使用共享密钥作为密钥的 MAC）来实现对终端的认证。对于响应者而言，被签名的字节以第二个消息头第一个 SPI 的第一个字节开始，以第二个消息的最后一个载荷的最后一个字节结束。之后附加的（为了计算签名）是发起者的临时随机数  $N_i$ （仅仅是值，而不是包含  $N_i$  的载荷）和值  $prf(SK_{pr}, IDr')$ ，这里  $IDr'$  是不包括固定头部的响应者的 ID 载荷。需注意的是不管是临时随机数  $N_i$  还是值  $prf(SK_{pr}, IDr')$  都不传输。类似地，发起者对第一个消息签名，以头部第一个 SPI 的第一个字节开始，以最后一个载荷的最后一个字节结束。之后附加的（为了计算签名）是响应者的临时随机数  $N_r$  和值  $prf(SK_{pr}, IDi')$ 。在上面的计算中， $IDi'$  和  $IDr'$  是不包括固定头部的完整 ID 载荷。对交换安全非常关键的是一端对另一端的临时随机数进行签名。

注意所有的载荷包含在签名下，包括本标准中没有定义的任何载荷类型。如果交换的第一个消息被发送了两次（第二次包含响应者的 cookie 和/或一个不同的 Diffie-Hellman 组），这是经过签名的消息的第二个版本。

可选地，消息 3 和 4 可包含一个证书，或者证书链来提供证据说明用于计算数字签名的密钥属于 ID 载荷中的名字。将使用算法来计算签名或 MAC，这些算法由签名者使用的密钥类型来规定，并由认证载荷中的 Auth 方法字段指定。这里没有要求发起者和响应者使用同样的加密算法来签名。加密算法的选择依赖于他们的密钥类型。特别地，发起者可使用共享密钥，而响应者可使用公共的签名密钥和证书。一般的情况是（并不要求是），如果使用一个共享的密钥进行认证，那么在两个方向上使用相同的密钥。需要注意的是，仅从用户选择的密码得到共享密钥而不考虑与随机性结合这是一种常见的典型的不安全的方法。

这是典型的不安全方法因为用户选择的密码不可能有足够的不可预测性从而能抗字典攻击，且这种攻击不能用认证的方式防止（使用基于密码的认证而进行 bootstrapping 的应用和 IKE\_SA 应使用 5.17 节中的认证方法，该方法是设计来防止离线的字典攻击的）。建议预先共享的密钥应具有与协商的最强的密钥具有一样的不可预测性。在有预先共享密钥的情况下，AUTH 值按如下方式计算：

$AUTH = prf(\text{共享密钥}, \text{"密钥填充"}, <\text{消息字节}>)$

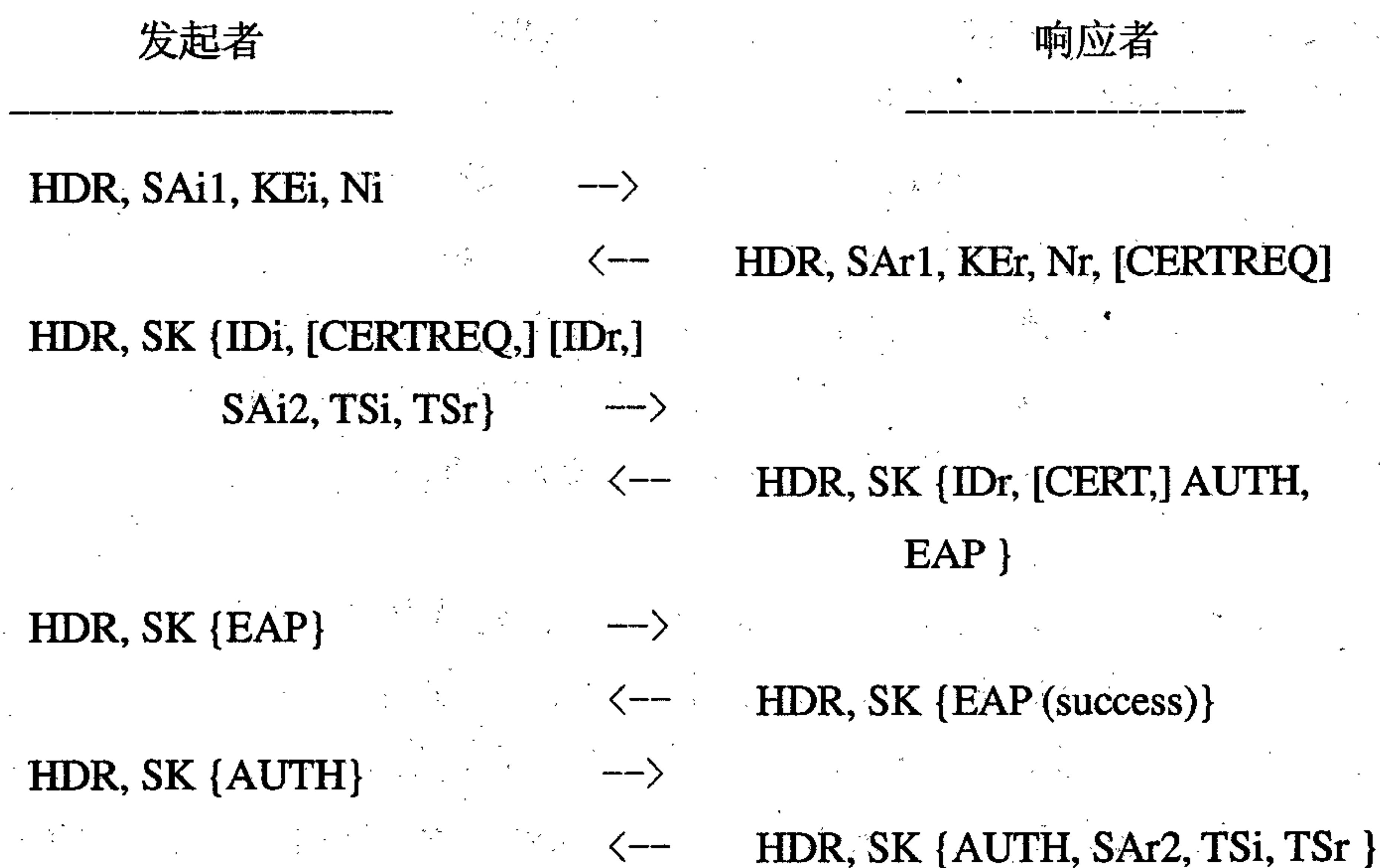
“密钥填充”是没有空结束符的 17 位 ASCII 字符。共享的密钥可以是变长的。填充的字符被添加上去是为了在共享的密钥是从密码获得时，IKE 实现不必以明文存储密码，而可以存储值  $prf(\text{共享密钥}, \text{"密钥填充"})$ ，该值不能作为密码的等价物应用于除 IKEv2 的协议。从以上可以得知，从密码获取共享密钥是不安全的。使用这种结构是因为人们总是会这么用。提供共享密钥的管理接口应接受至少 64 字节的 ASCII 字符，且在将其作为共享密钥之前，不应添加空结束符。该管理接口也应接受共享密钥的十六进制编码。如果指定了算法将编码转换为二进制字符，管理接口可接受其他的编码。如果协商的  $prf$  采用固定长度的密钥，共享的密钥应与该密钥长度相同。

### 5.17 扩展认证协议方法

IKE 除支持使用公开密钥签名和共享密钥的认证外，还支持满足国家商用密码使用的方法。典型地，这些方法是不对称的（为了用户对服务器认证而设计的），且可能不是交互的。基于此，这些协议典型的应用为响应者对发起者认证，且应与一个公开密钥签名联合使用，该签名是基于响应者对发起者认证的。这些方法常与称为“传统认证”的机制相关联。

扩展认证是作为附加的 IKE\_AUTH 交换在 IKE 中实现的，为了初始化 IKE\_SA 应完成该交换。

发起者通过在消息 3 中省去 AUTH 载荷来表明需要使用扩展认证。通过包含一个 IDi 载荷而不是一个 AUTH 载荷，发起者宣布了一个 ID 但还没有验证该 ID。如果响应者愿意使用扩展认证方法，则将在消息 4 中放置一个可扩展认证协议（EAP）载荷，并推迟发送 SAr2, TSi, 和 TSr 直到发起者认证在后续的 IKE\_AUTH 交换中完成。在有最小扩展认证的情况下，最初的 SA 建立将按如下方式出现：



对于生成了作为认证对一方影响的共享密钥的 EAP 方法，该共享密钥应被发起者和响应者用于产生消息 7 和 8 中的 AUTH 载荷，产生时要使用 5.16 节中指定的共享密钥的语法。来自 EAP 的共享密钥来自于 EAP 规范中命名为 MSK 的字段。在 IKE 交换过程中产生的共享密钥不应应用于任何其他目的。

不建议使用没有建立共享密钥的 EAP 方法，相关内容详见安全考虑一节。如果使用了不能产生共享密钥的 EAP 方法，消息 7 和 8 中的 AUTH 载荷应分别使用 SK\_pi 和 SK\_pr 来生成。

建议使用 EAP 的 IKE\_SA 的发起者在响应者发送通知消息和/或重试认证提示时，应具备将初始协议交换扩展为至少 10 次 IKE\_AUTH 交换的能力。一旦由选择的 EAP 认证方法定义的协议交换成功结束，响应者应发送包含 Success 消息的 EAP 载荷。类似地，如果认证失败，响应者应发送包含 Failure 消息的载荷。响应者可在任何时候通过发送包含 Failure 消息的 EAP 载荷来终止 IKE 交换。

遵循这样的扩展交换，紧随包含 EAP Success 消息的载荷，EAP AUTH 载荷应被包含在两类消息中。

### 5.18 为 CHILD\_SA 生成密钥生成资源

一个单独的 CHILD\_SA 是由 IKE\_AUTH 交换创建的，附加的 CHILD\_SA 可选择性的在 CREATE\_CHILD\_SA 交换中创建。他们的密钥生成资源是按以下方式生成的：

$$\text{KEYMAT} = \text{prf} + (\text{SK}_d, \text{Ni} | \text{Nr})$$

如果该请求是第一个创建的 CREATE\_CHILD，那么 Ni 和 Nr 是来自于 IKE\_SA\_INIT 交换的临时随

机数，如果是后续创建的，那么 Ni 和 Nr 是来自于 CREATE\_CHILD\_SA 交换的新的 Ni 和 Nr 值。

对于包含可选 Diffie-Hellman 交换的 CREATE\_CHILD\_SA 交换，可按如下定义计算密钥生成资源：

$$\text{KEYMAT} = \text{prf}(\text{SK\_d}, \text{g}^{\text{ir}}(\text{new}) \mid \text{Ni} \mid \text{Nr})$$

$\text{g}^{\text{ir}}$ （新的）是来自于该 CREATE\_CHILD\_SA 交换（形式为 big endian 顺序的八字节比特流，如果需要将其补齐为模数的长度，则在 high-order 比特位补 0）的短暂 Diffie-Hellman 交换的共享密钥。

一个单独的 CHILD\_SA 协商可能产生多个安全联盟。ESP 和 AH SA 成对存在（各在一个方向上），如果协商将 ESP 和 AH 结合，那么四个 SA 可能在一个单独的 CHILD\_SA 协商中产生。

密钥生成资源应从扩展 KEYMAT 中根据以下顺序获得：

SA 从发起者到响应者传输的数据的所有密钥是于 SA 在反方向上传输之前获得的。

如果协商了多个 IPsec 协议，密钥生成资源是按照封装数据包中协议头出现的顺序获得的。

如果单个协议既有加密，又有认证密钥，那么加密密钥是从 KEYMAT 的第一个八字节获得的，认证密钥是从下一个八字节获得的。

每个加密算法采用固定长度比特位的密钥生成资源，该密钥生成资源是作为算法的一部分被指定的。

利用 CREATE\_CHILD\_SA 重建 IKE\_SA

CREATE\_CHILD\_SA 交换可用于重建一个已有的 IKE\_SA。新的发起者和响应者 SPI 由 SPI 字段提供。当重建一个 IKE\_SA 时，忽略 TS 载荷。新 IKE\_SA 的 SKEYSEED 是利用现有 IKE\_SA 的 SK\_d 按如下规则计算的：

$$\text{SKEYSEED} = \text{prf}(\text{SK\_d}(\text{old}), [\text{g}^{\text{ir}}(\text{new})] \mid \text{Ni} \mid \text{Nr})$$

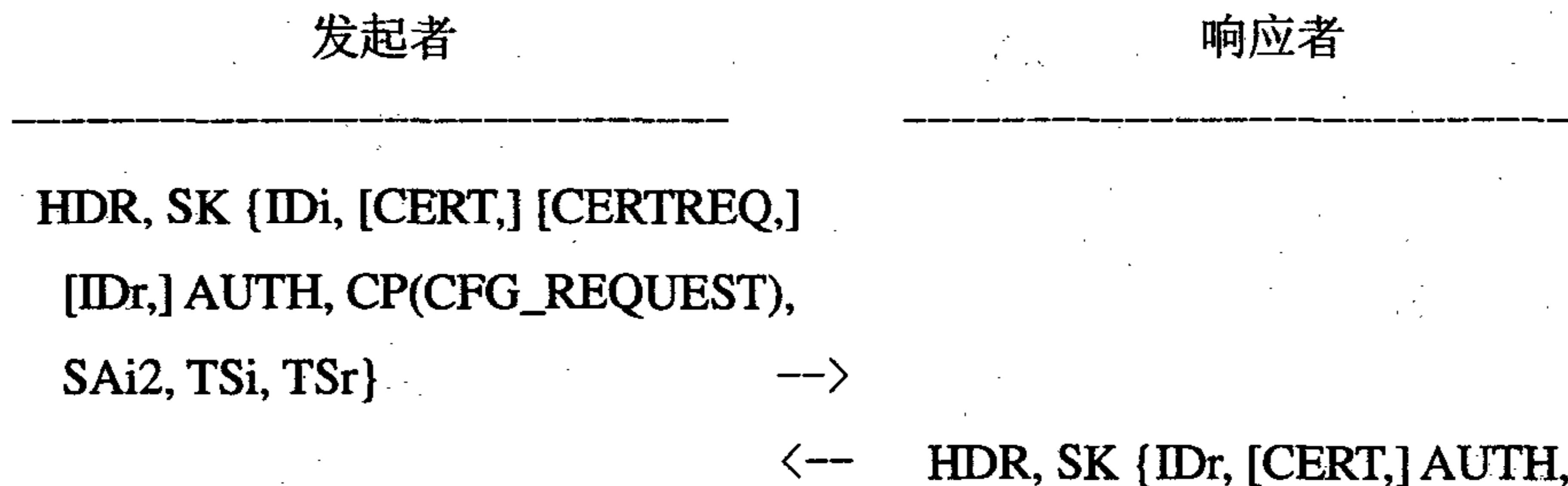
这里  $\text{g}^{\text{ir}}$ （新的）来自于该 CREATE\_CHILD\_SA（形式为 big endian 顺序的八字节字符串，如需将其补齐为模数的长度，则补 0）交换的短暂的 Diffie-Hellman 交换的共享密钥，Ni 与 Nr 是被去掉了头部的两个临时随机数。

新的 IKE\_SA 应将消息计数器重置为 0。

SK\_d, SK\_ai, SK\_ar, SK\_ei, 和 SK\_er 是按照 5.15 节指定的方式从 SKEYSEED 中计算的。

### 5.19 在远程网络请求内部地址

在终端到安全网关的场景中，常发生的情况是终端需要一个网络内部的受安全网关保护的 IP 地址，且需要该地址是动态分配的。可通过包含一个 CP 载荷，将这样一个临时地址的请求包含到创建 CHILD\_SA（在消息 3 中包含隐含的请求）的任何请求中。该函数提供分配给 IPsec 远程接入终端（IRAC）的地址，这里的 IRAC 试图通过隧道接入受 IPsec 远程接入服务器（IRAS）保护的网络。由于 IKE\_AUTH 交换创建了 IKE\_SA 和 CHILD\_SA，因此 IRAC 应在 IKE\_AUTH 交换中请求 IRAS 控制的地址（以及可选的与被保护网络相关的其他信息）。IRAS 可能从许多源中为 IRAC 获得一个地址，例如 DHCP/BOOTP 服务器或它自己的地址池。



CP(CFG\_REPLY), SAr2,  
TSi, TSr}

在所有情况下，CP 载荷应插入到 SA 载荷之前，在有多个 IKE\_AUTH 交换的协议的变体中，CP 载荷应被插入到包含 SA 载荷的消息之前。

CP(CFG\_REQUEST)应包含至少一个 INTERNAL\_ADDRESS 属性 (IPv4 或 IPv6)，但可包含发起者想在响应中返回的很多附加属性。

例如，从发起者到响应者的消息：

CP(CFG\_REQUEST)=  
INTERNAL\_ADDRESS(0.0.0.0)  
INTERNAL\_NETMASK(0.0.0.0)  
INTERNAL\_DNS(0.0.0.0)  
TSi = (0, 0-65535, 0.0.0-255.255.255.255)  
TSr = (0, 0-65535, 0.0.0-255.255.255.255)

注意：流量选择器包含（协议，端口范围，地址范围）。

从响应者到发起者的消息：

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)

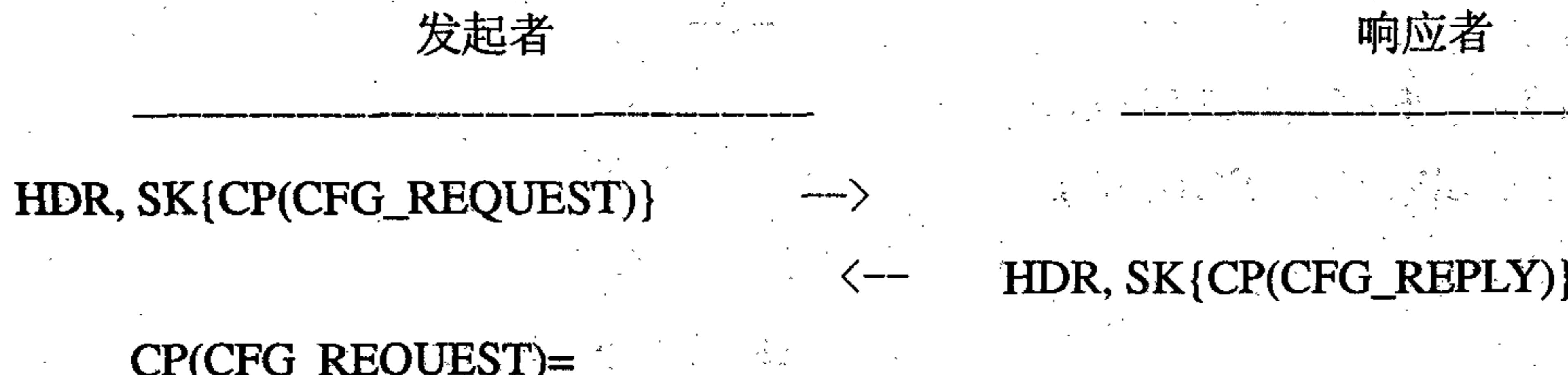
所有返回的值应相互依赖。由以上例子中可知，IRAS 可发送不包含在 CP(CFG\_REQUEST)消息中的其他属性，可忽略它不支持的非强制性属性。

首先没有从发起者收到 CP(CFG\_REQUEST)时，响应者不应发送 CFG\_REPLY 消息，因为如果 IRAC 不能处理该 REPLY，我们不想 IRAS 执行不必要的配置查找。在 IRAS 配置要求 CP 用于给定的身份 IDi，但 IRAC 发送 CP(CFG\_REQUEST)失败的情况下，IRAS 应放弃请求，并以 FAILED\_CP\_REQUIRED 错误终止 IKE 交换。

## 5.20 请求终端版本

希望得知其他终端 IKE 软件版本的 IKE 终端可使用以下的方法。这个例子是在 IKE\_SA 和第一个 CHILD\_SA 创建之后，一个 INFORMATIONAL 交换中的配置请求实例。

IKE 实现可在认证之前甚至认证之后拒绝给出版本信息，来防止被知道安全弱点的一些实现进行攻击。在这种情况下，他应返回一个空字符串或如果不支持 CP 的话，返回一个非 CP 载荷。



```

APPLICATION_VERSION("")  

CP(CFG_REPLY) APPLICATION_VERSION("foobar v1.3beta, (c) Foo Bar  

Inc.")

```

### 5.21 错误处理

在 IKE 处理过程中，可能会出现许多类型的错误。如果由于策略的原因（例如，没有匹配的加密算法），接收到一个格式错误或无法接收的请求，应使用包含通知载荷的响应来说明该错误。如果一个错误发生在 IKE 请求上下文之外（例如，一个节点在不存在的 SPI 上接收 ESP 消息），建议节点初始化一个包含能说明该问题的通知载荷的 INFORMATIONAL 交换。

在加密保护的 IKE\_SA 建立之前发生的错误必须小心的处理。在诊断问题以及对该问题作出响应的需求及避免被基于伪造消息的 DOS 攻击的欺骗的需求中存在一种折衷。

如果节点在它所不知道的 IKE\_SA 上下文之中（也不是开始 IKE\_SA 的请求）在 UDP 端口 500 或 4500 收到一个消息，那这可能是节点最近失效（crash）的结果。如果该消息被标志为响应，节点可审计可疑的事件，但不应进行响应。如果该消息被标记为请求，节点可审计可疑的事件，也可发送响应。如果发送一个响应，那么该响应应发向获得 IKE SPI 和消息 ID 的发送 IP 地址和端口。该响应不应被加密保护且应包含一个通知载荷来说明 INVALID\_IKE\_SPI。

收到这样一个未受保护的通知载荷的节点不应进行响应，不应改变任何现有 SA 的状态。该消息可能是伪造的，或者可能是真正的响应者被欺骗而发送的。建议节点把这样的消息（以及象 ICMP 目的地不可达的网络消息）当作对那个 IP 地址而言，SA 可能存在问题的线索，并建议节点为任何这样的 IKE\_SA 初始化一个生存性检测。建议实现限制生存性检测的频率以避免被欺骗而参与到 DOS 攻击中。

节点从与之存在 IKE\_SA 的 IP 地址接收到一个可疑的消息，则可在该 SA 上的 IKE INFORMATIONAL 交换中发送一个 IKE 通知载荷。接收者不应为此改变任何 SA 的状态，但建议审计事件从而帮助诊断故障。节点应限制对未受保护的消息发送响应消息的频率。

### 5.22 IP 压缩

IP 压缩的使用可作为 CHILD\_SA 建立的一部分进行协商。当 IP 压缩包含每个包中的额外报文头和压缩参数索引（CPI）时，虚拟的“压缩联盟”在包含它的 ESP 或 AHSA 之外失效。当相应的 ESP 或 AH SA 不存在时，压缩关联消失。这不会在任何删除载荷中显式提及。

IP 压缩的协商与 CHILD\_SA 相关的加密参数的协商是分开的。请求 CHILD\_SA 的节点可通过类型为 IPCOMP\_SUPPORTED 的一个或多个通知载荷广播它支持一个或多个压缩算法。响应可使用类型为 IPCOMP\_SUPPORTED 的通知载荷表明它接受一个单一的压缩算法。这些载荷不应出现在不包含 SA 载荷的消息中。

尽管对于接受多个压缩算法以及在 CHILD\_SA 的两个方向上可使用不同的算法仍有讨论，本标准的实现不应接受没有提到的 IPComp 算法，不应接受多于一个算法，且不应使用不是在 CHILD\_SA 建立过程中提及并被接受的算法进行压缩。

将 IPComp 协商从加密参数中分离出来的一个后果是不可能提出多个加密套件，且 IP 压缩只与某些套件一起使用，而不用于其他的加密套件。

### 5.23 NAT 穿越

本节简要的描述什么是 NAT 网关，以及对于 IKE 流量他们是如何工作的。

NAT 被设计对端节点而言是透明的。不管是位于 NAT 之后的节点上的软件，还是互联网上的节点，在通过 NAT 进行通信时，都不需要进行修改。对于某些协议而言，获得这种透明性要比其他协议更难。在数据包的载荷中包含终端 IP 地址的协议就会失效除非 NAT 网络理解该协议并修改内部相关资料及其在头部的相关资料。这样的知识本身是不可靠的，是对网络层次的违背，常会导致一些小的问题。

通过 NAT 打开 IPsec 连接引入了一些特殊的问题。如果连接以传输模式运行，改变数据包的 IP 地址将导致校验和失效，而 NAT 不能修正校验和因为这些是加密保护的。即使在隧道模式，也有路由问题因为透明的转换 AH 和 ESP 数据包的 IP 地址要求 NAT 具有特殊的逻辑，而这种逻辑是启发式的且本质上是不可靠的。因此，IKEv2 可协商对 IKE 和 ESP 数据包进行 UDP 封装。这种编码的有效性稍差但更容易被 NAT 处理。此外，防火墙可通过配置在 UDP 上通过 IPsec 流量而不是 ESP/AH 流量，或者反之亦然。

对于 NAT 而言，将 TCP 或 UDP 的端口和地址进行翻译，并根据进入的数据包的端口号决定哪个内部节点应得到该数据包是普通的工作。基于此，即使 IKE 包应在 UDP 端口 500 进行发送和接收，这些数据包也应能被从任何端口接收，而响应应发送到数据包来的端口。这是因为在穿过 NAT 的时候，端口可能被修改了。类似地，IKE 终端的 IP 地址一般也不包含在 IKE 载荷中，因为载荷是加密保护的，不能被 NAT 透明的修改。

端口 4500 是为 UDP 封装的 ESP 和 IKE 保留使用的。当通过一个 NAT 时，一般最好通过端口 4500 传输 IKE 数据包因为一些老的 NAT 在端口 500 试图在自身不进行 NAT 穿越的终端之间透明的建立 IPsec 连接来灵活的处理 IKE 流量。这类 NAT 可能会阻碍本文档中预想的直接的 NAT 穿越。因此，在自己和对端之间发现了 NAT 的 IPsec 终端应将后续的所有流量都从 NAT 不会进行特殊处理（类似于在 500 端口进行的处理）的 4500 端口发送和接收。

对于支持 NAT 穿越的明确需求列出如下。支持 NAT 穿越是可选的。在本节中，以“必须”列出的需求仅针对支持 NAT 穿越的实现有效。

——IKE 必须在端口 4500 倾听，就像在端口 500 倾听一样，IKE 必须对数据包到达的 IP 和端口进行响应。

——IKE 的发起者和响应者都必须在 IKE\_SA\_INIT 数据包中包含类型为 NAT\_DETECTION\_SOURCE\_IP 和 NAT\_DETECTION\_DESTINATION\_IP 的通知载荷。这些载荷可用于检测在主机之间是否存在 NAT，哪个终端位于 NAT 之后。这些载荷在 IKE\_SA\_INIT 数据包中紧跟在 Ni 和 Nr 载荷（在可选的 CERTREQ 载荷之前）之后。

——如果接收到的 NAT\_DETECTION\_SOURCE\_IP 载荷中没有一个能够匹配源 IP 和端口的哈希，该源 IP 和端口来自于包含该载荷的数据包的 IP 头，这意味着另一端位于 NAT 之后（也就是说，在通信中，原始数据包的原地址被修改以匹配 NAT 转换器）。在这种情况下，该节点应当允许就像后续描述的一样来动态更新另一端的 IP 地址。

——如果接收到的 NAT\_DETECTION\_DESTINATION\_IP 载荷不能与目的地址 IP 和端口的哈希匹配，该目的 IP 和端口来自于包含该载荷的数据包的 IP 头，这意味着该端位于 NAT 之后，在这种情况下，建议该端开始发送在 IETF RFC3948 (2005) 中规定的保持其处于活动状态的数据包。

——IKE 发起者必须检查出现的载荷，如果载荷与发出去的数据包的地址不匹配，IKE 发起者应将未来所有与该 IKE\_SA 相关的 IKE 和 ESP 数据包通过隧道方式发往 UDP 端口 4500。

——为了在 UDP 端口 4500 以隧道方式传输 IKE 数据包，IKE 头部包含四个预先考虑的为 0 的八位字节，且紧跟着 UDP 头。为了在 UDP 端口 4500 以隧道方式传输 ESP 数据包，ESP 数据包紧跟着 UDP 头，由于 ESP 头的开始四个字节包含 SPI，且有效的 SPI 不能为 0，因此 ESP 和 IKE 消息总是可以被区分开来的。

——传输模式 TCP 和 UDP 数据包校验修正所需的原始源和目的 IP 地址从与该交换相关的流量选择器获得。在有 NAT 穿越的情况下，流量选择器必须准确包含一个 IP 地址，该 IP 地址然后被用于原始 IP 地址。

——可能出现的情况是 NAT 转换器决定删除仍处于活动状态的映射（例如，保持活动状态的间隔太长，或者 NAT 转换器重启）。为了从这种情况中恢复，建议不位于 NAT 之后的主机将所有数据包（包括重传的数据包）发往来自于另一端的最后的经过认证有效的数据包的 IP 地址和端口（也就是说动态更新地址）。不建议位于 NAT 之后的主机这么做，因为这打开了 DOS 攻击的可能性。任何认证过的 IKE 数据包或认证过的 UDP 封装的 ESP 数据包都能用于检测发现 IP 地址或端口已经改变了。

——IKE 在移动 IP 中的应用中，可能需要相似的但可能是不一致的处理机制，但这些处理机制在本标准中没有说明。

## 5.24 显示拥塞通知 (ECN)

当 IPsec 隧道按照原来在指定的方式工作时，ECN 用法不适用于外部的 IP 头因为隧道解包过程丢弃了作为网络损失指示的 ECN 拥塞指示。IKEv2 通过要求 ECN 可用于由 IKEv2 创建的所有隧道模式 IPsec SA 的外部的 IP 头中，将这种情形简单化。特别地，对所有由 IKEv2 创建的隧道模式 SA 的隧道封装和解封装应支持 IETF RFC3168 (2001) 中指定的隧道的 ECN 的所有功能选项，且应实现 IETF RFC4301 (2005) 中指定的隧道封装和解封装过程来防止 ECN 拥塞指示的丢弃。

## 6 报头和载荷的格式

### 6.1 IKE 报头

**当为nat穿越时，使用4500端口**

IKE 消息是基于 UDP 传输的，使用 UDP 500 端口或者 4500 端口。IKE 消息最开始的 UDP 信息大部分可以忽略，只需要保留消息返回时需要的 IP 地址以及 UDP 端口号信息。在 UDP 500 端口上发送的 IKE 消息直接跟在 UDP 报头后面，在 UDP 4500 端口上发送的 IKE 消息要保留出“0000”四位字节。这四位字节既不属于 IKE 消息体中的内容，也不包含在 IKE 定义的长度字段或者校验和中。每一条 IKE 消息以 IKE 报头 HDR 作为开始标志。每个 IKE 报头后可以有一个或者多个 IKE 载荷。如果有多个 IKE 载荷，则在每一个 IKE 载荷后有“下一负载”字段进行标识，“下一负载”字段为 0，说明载荷结束。IKE 消息中的载荷按照排列的先后次序依次被处理。如果载荷内容是经过加密的，则需要对载荷进行解密，载荷中的内容会被解析出来，放到另一个载荷内。加密的载荷必须位于 IKE 消息中的最后，而且加密的载荷内容不允许再次加密。

用以表明这是 IKE  
报文，而不是 ESP  
报文：

NON-ESP MRKER

报头中的接收端 SPI 标识出 IKE 安全联盟示例。一个单独的 IKE 示例可以用于与多个对等端建立的多个会话。

所有用多字节字段表示的整数按照高位在前的顺序排列，即最高字节在前面的顺序排列，或者按照网络字节顺序排列。

IKE 报头格式如下：

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 发起者 SPI				
IKE_SA 响应者 SPI				
下一个负载	主版本	次版本	交换类型	标志
消息 ID				
长度				

IKE 报头格式

- IKE\_SA 发起者 SPI (8 字节)：由初始端选择的标识惟一 IKE 安全联盟的数值，数值不能为 0。
- IKE\_SA 响应者 SPI (8 字节)：由响应端选择的标识惟一 IKE 安全联盟的数值。IKE 初始化阶段交换的第一条消息中的 SPI 应该为 0，后续的消息中均不能为 0。
- 下一个负载 (1 字节)：指紧跟在报头后面的载荷类型。
- 主版本 (4 比特)：指正在使用的 IKE 协议的主要版本。当前 IKE 主要协议版本号应当设置为 2。前一版本的 IKE 协议和 ISAKMP 的版本号应该设置为 1。如果版本号大于 2 的 IKE 消息必须被忽略或者拒绝。
- 次版本 (4 比特)：指正在使用的 IKE 协议的次要版本。当前 IKE 协议次要版本号应当设置为 0。必须忽略接收到的消息中 IKE 协议的次要版本号。
- 交换类型 (1 字节)：指正在使用的交换类型。该字段限制消息的载荷内容和交换消息的顺序，见表 2。

表 2

交换类型	值
保留	0~33
IKE_SA_INIT	34
IKE_AUTH	35
CREATE_CHILD_SA	36
INFORMATIONAL	37
IANA 保留	38~239
保留	240~255

——标志 (1 字节)：为消息设置的特定选项。标记字段 (Flags field) 的比特位置位，值为 1，比位清除，值为 0。

- X (保留) (标记字段的第 0~2 位比特)。发送 IKE 消息时，必须对这两个比特位清 0；接收 IKE 消息时，必须忽略这两个比特位。
- I (initiator) (标记字段的第 3 位比特)。IKE\_SA 的起始初始化端在发送消息时必须对该比特位置位；IKE\_SA 的初始响应端在发送消息时必须对该比特位清 0。接收端会根据该比特位决定生成 SPI 的哪 8 位字节。
- V (ersion) (标记字段的第 4 位比特)。该比特位说明传送方能够支持比主要版本号更高的协议主要版本。在发送 IKEv2 消息时，必须对该比特位清 0，在接收 IKEv2 消息时，必须忽略该比特位。

• R (esponse) (标记字段的第 5 位比特)。该比特位说明本消息是对具有相同消息 ID 消息的响应消息。在所有的请求消息中该比特位需要清 0，在所有的响应消息中该比特位需要置位。IKE 端点不能对已经响应的请求消息重复发送响应消息。

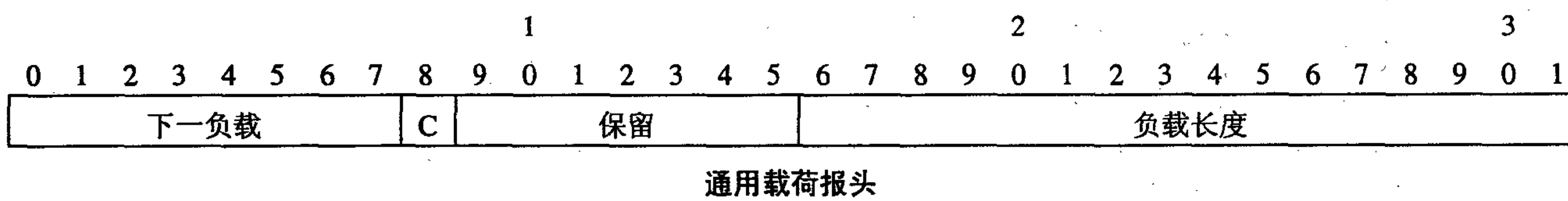
• X (reserverd) (标记字段的第 6~7 位比特)。在发送 IKE 消息时，必须对这两个比特位清 0，在接收 IKE 消息时，必须忽略这两个比特位。

—— 消息 ID (4 字节)：消息标识符，用于控制请求消息和响应消息丢包时重传。消息标识符可以用于防止消息重放攻击，是保障协议安全的一种必要的手段。

—— 长度 (4 字节)：消息体（报头+载荷）的字节总长度。

## 6.2 通用载荷报头

在 6.3 至 6.16 节中定义的 IKE 载荷均使用通用载荷报头作为开始。



通用载荷报头字段定义如下：

—— 下一负载 (1 字节)：标识消息中下一个载荷的类型的标识符。如果当前的载荷是消息中的最后一条载荷，则该字段置 0。该字段提供“chaining”能力，即如果要在消息体末尾增加新的载荷，则将该字段置位，并且在前一个载荷内标识出新载荷的类型。加密的载荷必须位于消息体的末尾，使用另外的载荷格式，/在加密的载荷报头中，下一负载字段设置成消息体中第一个载荷的类型，而不为 0。载荷类型值见表 3。

表 3

下一负载类型	符 号	数 值
No 下一负载		0
保留		1~32
安全联盟	SA	33
密钥交换	Ke	34
标识初始者	IDi	35
标识响应者	IDr	36
证书	CERT	37
证书请求	CERTREQ	38
认证	AUTH	39
随机数	Ni, Nr	40
通知	N	41
删除	D	42
厂商 ID	V	43
TS 发起者	TSi	44
TS 响应者	TSr	45
加密	E	46
配置	CP	47
扩展认证	EAP	48
IANA 保留		49~127
专用保留		128~255

载荷类型值 1~32 不应使用，因此不会与分配给 IKEv1 版本的代码发生重叠。载荷类型值 49~127 保留给 IANA，分配给 IKEv2 使用。载荷类型值 128~255 用于通过彼此认证的组群之间内部使用。

—— 紧急比特（1 比特）：当发送端要求接收端跳过这条载荷或者接收端不能理解在前一条载荷中的下一负载字段中的载荷类型代码，该比特位必须置 0。当发送端希望接收端拒绝整个 IKE 消息，或者接收端不能理解载荷类型时，该比特位必须置 1。当接收端能够理解载荷类型代码，该比特位必须被接收端忽略。如果不对该比特位置位，则协议实现过程中必须理解本规范中定义的所有载荷类型，因此可以忽略该比特位的数值。被跳过的载荷都具备有效的“下一负载”字段和载荷长度字段。

—— 保留（7 比特）：发送端必须将该比特置 0，接收端必须忽略该比特。

—— 负载长度（2 字节）：当前载荷的字节长度，包括通用载荷头。

### 6.3 安全联盟载荷 33

在本标准中，安全联盟载荷被用于协商安全联盟的属性。安全联盟载荷是一个集合，可以包含多个建议，如果有多个建议存在，那么必须按照优先级进行排序。每个建议可能包含多个 IPsec 协议（如密钥交换协议（IKE）、安全载荷封装协议（ESP）、认证头协议（AH）），每个协议可以包含多个转换，每个转换可以包含多个属性。当解析一个安全联盟时，必须要验证整个负载的长度，负载长度必须要与负载内部长度域的值和数量相一致。建议、转换和属性都有各自的可变长度编码，它们彼此嵌套导致安全联盟的负载长度包含了安全联盟、建议、转换和属性信息的组合内容。建议的长度包括它所包含的所有转换和属性的长度。转换的长度包括所包含的所有属性的长度。安全联盟、建议、转换和属性的语法是基于 ISAKMP，但是语义有所不同。由于复杂度和层次化的因素允许在单个安全联盟中定义多个可能的组合算法。有时可以在多个算法中选择，但是在多数情况下则是组合算法。例如，一个发起者可能想建议使用（MD5 AH 和 3DES ESP）或者是（MD5 和 3DES 的 ESP）。

在建议的结构中包含建议的编号（#）和 IPsec 协议标识。每个结构必须具有与先前相同的建议编号（#）或者是比它大 1。第一个建议的编号必须是 1。如果连续的两个结构有相同的编号，意味着建议是由第一个和第二个结构组成。因此认证头和安全载荷封装将会有两个建议结构，一个用于认证头，一个用于安全载荷封装，都具有相同的建议编号 1。认证头或者是安全载荷都有两个建议结构，认证头的编号是 1，安全载荷的编号是 2。

每个建议/协议结构都伴随着一个或者是多个转换结构。不同的转换数量主要是由协议来决定。认证头协议是单个转换：完整性检查算法。安全载荷封装是两个转换：加密算法和完整性检查算法。IKE 包含四个转换：一个 Diffie-Hellman 组，完整性检查算法，Prf（伪随机函数）算法和加密算法。如果一个算法被建议结合加密和完整性检查，那么必须作为一个加密算法提出，而不是一个完整性保护算法。对于每一个协议，允许的转换集合都分配一个转换编号，出现在每个转换的头部。

如果多个转换具有相同的转换类型，建议“或”运算，如果多个转换具有不同的转换类型，建议在不同的组“与”运算。例如，建议安全载荷封装(3DES 或者 IDEA)，(HMAC\_MD5 或者 HMAC\_SHA)，安全载荷封装建议将包含两个转换类型 1（1 个用于 3DES，1 个用于 IDEA），两个转换类型 2（1 个用于 HMAC\_MD5,1 个用于 HMAC\_SHA）。这将提出四种算法组合。如果发起者只想建议其中的一种算法，如（3DES 和 HMAC\_MD5）或者是（IDEA 和 HMAC\_SHA），那么没有办法在单个建议中编码多个转换。发起者将不得不构建两个不同的建议，每个包含两个转换。

一个特定的转换可以有一个或者多个属性。当转换被用于多种方式时，属性是必须的，就如同只预定义了一种属性类型，即 key 长度

个加密算法有一个可变的密钥大小。转换指定算法，属性指定密钥大小。大多数转换不包含属性。转换一定不能有多个相同类型的属性。对于一个属性建议替换值（例如，对 AES 加密算法多个密钥大小），在实现上必须包含多个具有相同转换类型的转换，每个转换具有单一属性。

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
下一负载	C	保留																			
<建议>																					

#### 安全联盟载荷

—— 建议：一个或者多个建议子结构。

安全联盟的载荷类型是 33。

#### 6.3.1 建议子结构

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0 (最后) 或者 2	保留																				
建议 #	协议 ID																				
SPI																					
<变换>																					

#### 建议子结构

—— 0 (最后) 或者 2 (更多) (1 个字节)：指明这是否是安全联盟中最后的建议子结构。这个语法继承于 ISAKMP，但不是必须的，因为最后的建议可以通过安全联盟的长度识别。数值 2 与 IKEv1 中建议的载荷类型符合，建议结构的前 4 个字节类似于载荷的头部。

—— 保留 (1 个字节)：保留，必须赋值为 0，在接收方忽略。

—— 建议长度 (2 字节)：建议的长度，包括所有后继的转换和属性。

—— 建议# (1 字节)：建议编号，在安全联盟载荷中第一个建议的编号必须是 1，后继的建议必须是要么与先前的相同（表明两个建议与运算）或者是比先前的大 1（表明两个建议或运算）。当一个建议被接受，在安全联盟载荷中的所有建议编号必须相同，必须与接收的匹配。

—— 协议 ID (1 字节)：指定当前协商的 IPsec 协议标识，定义的值如下：

- 协议 协议标识
- 保留 0
- 密钥交换 1
- 认证头 2
- 安全载荷封装 3
- 保留 (IANA) 4~200
- 私有用途 201~255

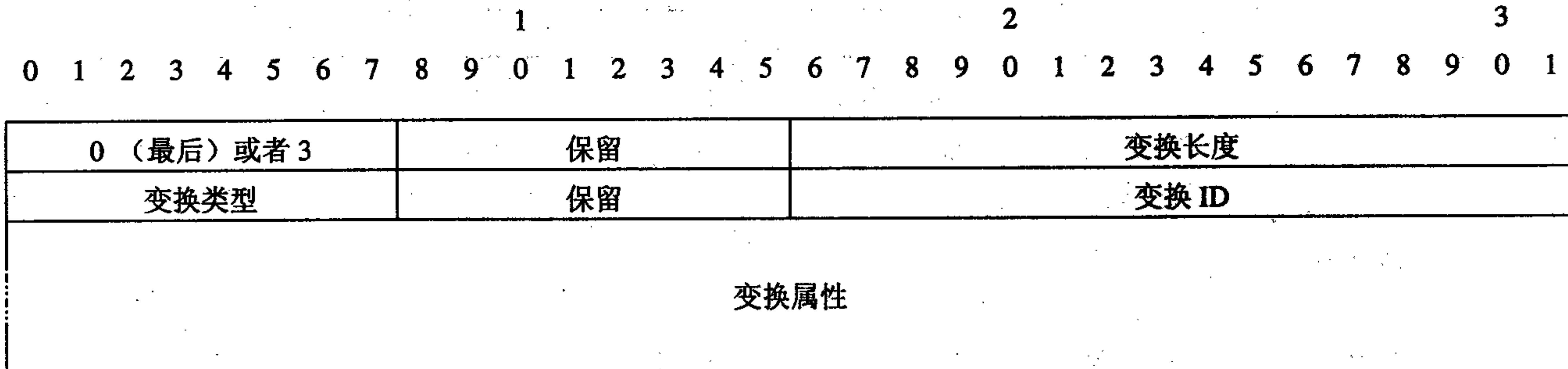
—— SPI 大小 (1 字节)：对于初始的 IKE\_SA 协商，该域必须置成 0；SPI 可以从外层头获得。在后继的协商过程中，对应协议的 SPI 大小相同 (IKE 8 字节，ESP 和 AH 4 字节)。

—— 变换数 (1字节)：指明在该建议中变换的数量。

—— SPI(变长)：发送方的 SPI。假使 SPI 的大小不是 4 字节的整数倍，也不会在负载中增加填充。  
当 SPI 大小域为 0 时，这个域不会在安全联盟载荷中出现。

—— 变换 (变长)：一个或者多个变换子结构。

### 6.3.2 变换子结构



—— 0(最后)或者 3(更多)(1字节)：指明是否这是最后一个变换子结构。该语法继承于 ISAKMP，但不是必须的，因为最后的建议能够通过安全联盟的长度识别。数值 3 与 IKEv1 中的转换负载类型符合，转换结构的前四个字节类似于载荷的头部。

—— 保留 (1字节)：保留，必须置为 0，在接收方必须忽略。

—— 变换长度 (2字节)：转换长度，包含头和属性的转换子结构的长度（按字节长度）。

—— 变换类型 (1字节)：转换类型，在转换结构中指定的转换类型。不同的协议支持不同的转换类型。对于一些协议，其中的一些转换是可选的。如果一个转换是可选的，而发起者希望建议该选项，那么该选项应当被忽略，指定类型的转换不包含在该建议中。如果发起者希望对应答者使用转换选项，它应当包含一个转换标识为 0 的转换子结构作为一个可选项。变换类型的值见表 4。不同变换类型的转换标识分别见表 5~表 9。

—— 变换 ID (2字节)：建议的转换类型的特定实例。

表 4 变换类型 (变换类型) 的值

变换类型	值	用 于
保留	0	
加密算法(ENCR) encrypt	1	密钥交换和安全载荷封装
伪随机函数 (PRF) pseudo random function	2	密钥交换
完整性算法(INTEG) integ	3	密钥交换，认证头，安全载荷封装选项
Diffie-Hellman 组(INTEG) (DH)	4	密钥交换，认证头，安全载荷封装选项
扩展序列号(ESN) extend serial number	5	认证头和安全载荷封装
IANA 保留	6-240	internet assigned numbers authority 互联网数字分配机构
私有用途	241-255	

表 5 变换类型 1 (加密算法) 的转换标识

名 称	编 号
保留	0
ENCR_DES_IV64	1
ENCR_DES	2

表5(续)

名 称	编 号
ENCR_3DES	3
ENCR_RC5	4
ENCR_IDEA	5
ENCR_CAST	6
ENCR_BLOWFISH	7
ENCR_3IDEA	8
ENCR_DES_IV32	9
保留	10
ENCR_NULL	11
ENCR_AES_CBC	12
ENCR_AES_CTR	13

数值 14~1023 为 IANA 保留。数值 1024~65535 用于相互达成一致的实体间的私有使用。

表6 变换类型为2(伪随机函数)的转换标识

名 称	编 号
保留	0
PRF_HMAC_MD5	1
PRF_HMAC_SHA1	2
PRF_HMAC_TIGER	3
PRF_AES128_XCBC	4

数值 5~1023 为 IANA 保留，数值 1024~65535 用于相互达成一致的实体之间。

表7 转换类型3(完整性算法)的转换标识

名 称	编 号
NONE	0
AUTH_HMAC_MD5_96	1
AUTH_HMAC_SHA1_96	2
AUTH_DES_MAC	3
AUTH_KPDK_MD5	4
AUTH_AES_XCBC_96	5

数值 6~1023 为 IANA 保留，数值 1024~65535 用于相互达成一致的实体之间。

表8 转换类型4(Diffie-Hellman)的转换标识

名 称	编 号
NONE	0
附录B中定义	1~2
保留	3~4
[ADDGROUP]中定义	5
IANA 保留	6~13
[ADDGROUP]中定义	14~18
IANA 保留	19~1023
私有使用	1024~65535

表9 转换类型5(扩展序列号)的转换标识

名称	编号
没有扩展序列号	0
扩展序列号	1
保留	2~65535

### 6.3.3 按照协议有效的转换类型

安全联盟载荷的转换类型和数量取决于安全联盟内部协议。安全联盟载荷建议安全联盟的建议具有如下的强制和可选的转换类型。对于所支持的协议必须考虑兼容所有的强制和可选的类型，见表10。如果能够支持的选项值为空，那么可以忽略可选类型（甚至不需要接受那些不可能接收到的选项）。

表10 按照协议有效的转换类型

协议	强制类型	可选类型
IKE	ENCR, PRF, INTEG, D-H	
ESP	ENCR, ESN	INTEG, D-H
AH	INTEG, ESN	D-H

### 6.3.4 强制转换标识

本标准中已经删除了必须或者应当支持的互操作规范，因为这些规范的变化很可能比本标准的演进更快。

从IKEv1规范中我们已经吸取了一个重要的教训，没有系统应当仅仅实现强制算法，而这对于所有的用户而言是最好的选择。例如，在本标准编写的过程中，为适应虚拟专用网应用的需求，一些IKEv1的实现正迁移到AES密码块链模式。一些基于IKEv2的IPsec系统将实现AES，附加的Diffie-Hellman组和附加哈希算法，一些IPsec用户已经对上述列举的算法提出了需求。

在将来IANA会增加一些附加的转换，某些用户可能希望使用私有套间，特别是IKE，在实现上应当能够支持不同的参数，直到明确的大小限制。为实现这个目标，IKEv2的所有实现应当包括管理工具允许对新的DH组规范（用户或者是系统管理员）Diffie-Hellman参数（生成器，模数，指数长度和数值）。在实现上应当提供一个管理接口，通过该接口能够配置这些参数和相关的转换标识（由用户或者是管理员），能够发起协商这些组。

IKEv2的所有实现必须包含一套管理工具使得用户和系统管理员指定IKE能够接受的一组集合。一旦接收到一组带有转换标识的载荷，在实现上必须比较传递中的转换标识和本地通过管理控制配置的转换标识，验证基于本地策略建议的一组选项是能够接受的。在实现上必须拒绝未通过IKE控制套件授权的安全联盟建议。

### 6.3.5 变换属性

在安全联盟载荷中的转换可以包含变化的或者是完整的转换规范属性。这些属性使用类型/数值对来表述。例如，如果一个加密算法有可变长度的密钥，这个密钥长度可以被指定作为一个属性来使用。属性可能是固定2字节长度的一个值或者是可变长度的一个值。可以采用类型/长度/值的形式表述。

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 F	属性类型	AF=0 属性长度 AF=1 属性值
	AF=0 属性值 AF=1 不传送	数据属性

—— 属性类型（2字节）：每个属性类型都有惟一的标识。

—— AF：属性格式比特位是最重要的一个比特。它表明数据属性的格式是类型/长度/值的形式还是类型/值的形式。如果 AF 比特为 0，那么数据属性的格式是类型/长度/数值的形式。如果 AF 位是 1，那么数据属性的格式是类型/值的形式。<sup>TLV</sup>

—— 属性长度（2字节）：按字节表示属性值的长度。当 AF 比特为 1，属性值的长度固定是 2个字节，因此属性长度域将不出现。

—— 属性值（可变长度）：属性值与属性的类型相关。如果 AF 比特为 0，该域由属性长度域来定义。如果 AF 比特为 1，则属性值为 2个字节长度。

需要特别指出的是仅有一个属性类型（密钥长度）被定义，并且是固定长度。可变长度的编码规范将在以后的扩展中定义。本标准中只定义了基于 AES 加密算法、完整性算法和伪随机函数，需要一个属性来指定密钥的宽度。

属性作为基本要素一定不能使用可变长度来编码。可变长度属性一定不能作为基本的要素，即使其长度满足 2个字节。这一点与 IKEv1 不同，在 IKEv1 中尽管增加了灵活性可以简化消息的构造，但是却增加了解析消息的复杂性。不同属性类型的值见表 11。

表 11 不同属性类型的值

属性类型	值
保留	0~13
Key 长度	14
TV 保留	15~17
IANA 保留	18~16383
私有使用	16384~32767

数值 0~13 和 15~17 在 IKEv1 中被应用于相似的上下文，除了匹配数值不应当被分配。数值 18~16383 为 IANA 保留，数值 16384~32767 被应用在相互达成一致的实体之间。

Key 长度（以 bit 计）：当使用可变长度密钥的加密算法时，要使用比特长度来定义密钥的长度（必须使用网络字节顺序）。当指定的加密算法使用固定长度的密钥时，该属性一定不能使用。

### 6.3.6 属性协商

在安全联盟协商阶段，发起者向响应者提出建议。响应者必须从发起者建议的属性中选择一个完整的属性集合（如果没有一个可以接受那么拒绝）。如果有多个建议，响应者必须只选择一个建议编号，并返回与该建议编号相关的所有建议子结构。如果有多个相同类型的转换，响应者必须选择其中的一个。所选择的转换的任何属性都不能被修改。发起者必须检查收到的响应是否与其发出的建议相一致，如果不一致则拒绝接收响应。

协商 Diffie-Hellman 组具有特殊的挑战。安全联盟在相同的消息中提出建议属性和 Diffie-Hellman 公共码（KE）。如果在初始的交互过程中，发起者提出使用多个 Diffie-Hellman 组中的一个组，它应当挑选一个最可能接收的响应者，并且包含与该组相对应的 KE。如果猜测是错的，响应者将会在响应中告诉正确的组，当发起者再次尝试第一个消息时，它应当从该组中挑选一个元素作为 KE 值。然而，它应当继续建议它所有支持的组避免中间人攻击。

实现说明：

确定的协商属性可能会有一个范围或者是有多个可供选择的值。这些包括可变密钥长度的对称密码

的密钥长度。为了进一步的互操作性和支持端点的独立升级，协议的实现应当接受那些能够提供安全保障的值。例如，如果一个对等体被配置成允许接受密钥长度为 X 比特的可变长度密码，那么当提供一个密钥更长的密码时，在实现上应当允许接受这个密码。

支持这种能力要求在是实现上体现“至少”这个概念，即密码 Y 的密钥长度至少是 X 比特。

#### 6.4 密钥交换载荷<sup>34</sup>

密钥交换载荷，在 Diffie-Hellman 密钥交换过程中用于交换 Diffie-Hellman public number。密钥交换载荷包括由 IKE 产生的载荷头部，载荷头部后面是 Diffie-Hellman 公共数值。

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
下一负载	C	保留
DH 组#		负载长度 保留
密钥交换数据		

密钥交换载荷格式

密钥交换载荷中的“密钥交换数据”部分是拷贝过来的 Diffie-Hellman 公开值，Diffie-Hellman 公开值的长度必须与求幂时的素模数相同，如果需要的话还要在数字的前面以 0 比特补足。

DH 组# 定义了一个 Diffie-Hellman 组，密钥交换数据在这个组里进行计算（见 3.3.2 节）。如果选定的方案使用了不同的 Diffie-Hellman 组，需要发送一个通知类型为 INVALID\_KE\_PAYLOAD 的拒绝消息。

密钥交换载荷的载荷类型为 34。

#### 6.5 标识载荷<sup>35/36</sup>

标识载荷（本文中以 IDi 和 IDr 表示），允许一方向其对等端宣告一个身份鉴定。这个身份标识可以用于策略查找，但在证书载荷中不必进行匹配；某一应用可以使用这两个字段来进行接入控制。

注意：在 IKEv1 中，两个 ID 载荷可以用来为 SA 之间两个方向上传递的数据保持流量选择器（Traffic Selector，TS）信息；在 IKEv2 中，这个信息由 TS 载荷来携带（见 6.13 节）

标识载荷包含了由 IKE 产生的载荷头，载荷头后面是鉴定字段。

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
下一负载	C	保留
ID 类型		负载长度 保留
标识数据		

标识载荷格式

- ID 类型（1 个字节）：表明鉴定的类型。
- 保留字段：必须置 0；在收端忽略。
- 负载长度（变长）：由鉴定类型所指定的数值，具体见表 12。鉴定数据的长度由 ID 载荷头的大小来计算 标识载荷的载荷类型为 35（IDi）和 36（IDr）。

表 12 鉴定类型域的已分配类型值

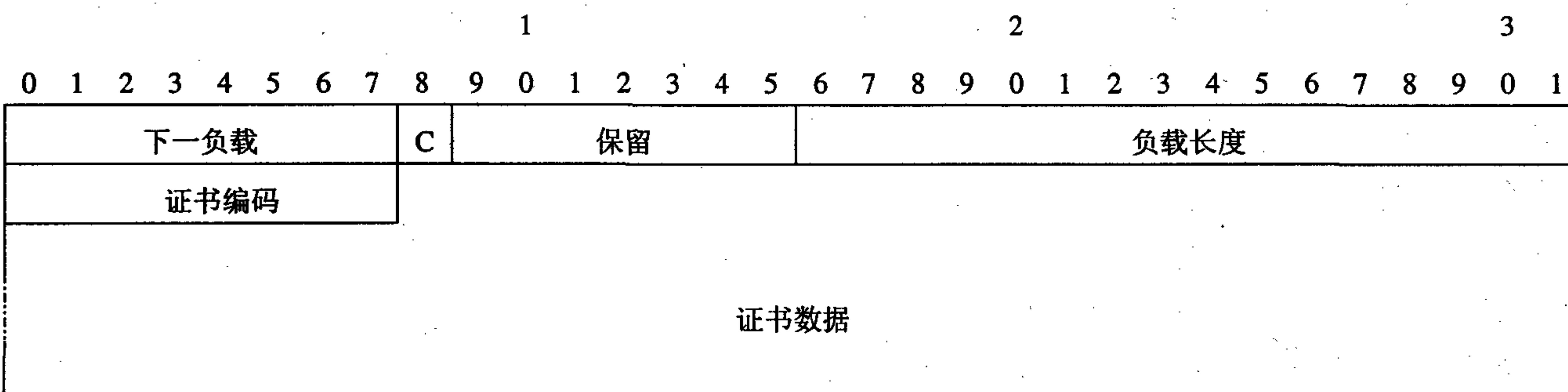
ID 类型	值	说 明
保留	0	
ID_IPV4_ADDR	1	一个 4 字节的 IPv4 地址
ID_FQDN	2	一个完整的域名字符串，例如“example.com”，不能包含任何终止符（NULL，回车符等）
ID_RFC822_ADDR	3	一个完整的符合 IETF RFC822 (1982) 要求的电子邮件地址字符串，例如“jsmith@example.com”。字符串不能包含任何终止符
保留 to IANA	4	为 IANA 保留
ID_IPV6_ADDR	5	一个 16 字节长度的 IPv6 地址
保留 to IANA	6~8	为 IANA 保留
ID_DER ASN1_DN	9	符合判别编码规则的 ASN.1 X.500 判别名
ID_DER ASN1_GN	10	符合判别编码规则的 ASN.1 X.500 全名
ID_KEY_ID	11	一个用于传递厂商信息来的不透名的字节流的标识
保留 to IANA	12~200	为 IANA 保留
保留 for private use	201~255	为私有应用保留

两个应用程序之间只有当产生的 ID 类型可以为对方所接受时才能进行互操作。为了保证最大程度的互操作能力，应用程序必须支持 ID\_IPV4\_ADDR, ID\_FQDN, ID\_RFC822\_ADDR, 或者 ID\_KEY\_ID 4 个鉴定类型之一，并接受所有上述 4 个鉴定类型。应用程序应能够产生或接受所有这些鉴定类型。支持 IPv6 的应用程序必须额外支持 ID\_IPV6\_ADDR。只支持 IPv6 的应用程序可以只发送 ID\_IPV6\_ADDR。

## 6.6 证书载荷 [37](#)

证书载荷（本标准中以 CERT 表示）提供了通过 IKE 传送证书或其他与认证相关信息的手段。如果发送方有可用的证书，证书载荷应该包含在密钥交换过程中，除非对端用 HTTP\_CERT\_LOOKUP\_SUPPORTED 通报载荷指示其可以从其他地方找回这个信息。注意，“证书载荷”这个词可能会产生些误解，因为不是所有认证机制都使用证书，其他不使用证书的数据也可以用这个载荷来传递。

证书载荷的格式如下。



证书载荷格式

—— 证书编码（1 字节）：证书编码，该字段表明了证书数据字段中的证书类型或与证书相关的信息。证书编码字段的值见表 13。

表 13 证书编码字段的值

Cert Encoding	值
保留	0
PKCS #7 封装的 X.509 证书	1
PGP 证书	2
DNS 签名密钥	3
X.509 证书—签名	4
Kerberos 令牌	6
证书撤回列表 (CRL)	7
授权撤回列表 (ARL)	8
SPKI 证书	9
X.509 证书—属性	10
原始 RSA 密钥	11
X.509 证书的哈希和 URL	12
X.509 绑定的哈希和 URL	13
为 IANA 保留	14~200
私有 保留	201~255

——证书数据（变长）：证书数据的实际编码值。证书的类型由证书编码字段指定。

证书载荷的载荷类型值为 37。

上述证书类型编码的详细含义在本标准中不作定义。在本文中进行定义的类型有：

- X.509 证书—签名 (4)：包含一个 DER 编码的 X.509 证书，其公钥用来确认发送方的认证载荷。
- 证书撤回列表 (7)：包含一个 DER 编码的 X.509 证书撤回列表。
- 原始 RSA 密钥 (11)：“包含一个 PKCS #1 编码的 RSA 密钥。
- 哈希和 URL 编码 (12~13)：允许 IKE 消息用原有长数据类型的 SHA-1 哈希值 (20 字节) 代替原有数据以缩短消息的长度，哈希值的后面是一个变长的 URL 指向 DER 编码的数据结构本身。当端点可以进行证书数据的缓存时，较短的消息长度可以提高缓存的效率，并且降低了 IKE 遭受拒绝服务攻击的危险——如果 IKE 消息长度大到需要进行 IP 分片时容易造成拒绝服务攻击。

应用程序必须能够发送并接受 4 个以上的 X.509 证书，并且应配置为发送并接受前两个哈希值和 URL (HTTP URL) 对。应用程序应该能够配置为发送和接受原始 RSA 密钥。如果发送了多个证书，第一个证书必须包含进行认证载荷签名的公钥，其他证书可以以任意的顺序发送。

## 6.7 证书请求载荷 [38](#)

证书请求载荷（本标准中以 CERTREQ 表示），提供了通过 IKE 请求首选证书的方法，这个载荷可以包含在 IKE\_INIT\_SA 回应和/或 IKE\_AUTH 请求消息中。当发送端需要得到接受端的证书时，证书请求载荷可以包含在交换过程中。如果存在多个可信 CA，且 CERT 编码不允许以列表形式发送，那么就需要发送多个证书请求载荷。

证书请求载荷的定义如下：

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	1	2	3
下一负载 证书编码	C	保留	负载长度
		CA	

#### 证书请求载荷格式

- 证书编码（1字节）：证书编码，包含了一个证书请求的类型或格式的编码。编码的取值见 6.6 节。
- CA（变长）：证书鉴权，包含了一个证书请求类型的可接受的 CA 编码。

证书请求载荷的载荷类型为 38。

证书编码字段中证书编码的取值与 6.6 节中的定义相同。CA 字段包含一个对于这个证书类型的可信的授权中心。CA 字段是一个连续列表，列表中为可信的证书授权中心（Certification Authorities, CA）的公钥值的 SHA-1 哈希值。每个值均编码为受信根 CA 证书签名的 Subject Public Key Info 的 SHA-1 哈希值。所有二十字节长哈希值都互相连接，且不包含其他格式。

注意这里的“证书请求”一词有可能会产生误解，它的取值不同于证书载荷中所定义的证书，也不同于将这些值显示在“证书请求载荷”中的请求。在这些场合里所指的“证书请求载荷”在本文中不作定义。

应用程序通过检查证书请求载荷的“证书编码”字段来判断其是否具有改类型的证书。如果有，将检查“证书授权”字段来确定是否具有由一个指定的证书授权者合法授权的证书。这构成了一个认证链。

如果一个终端实体拥有符合 CERTREQ 标准指定的证书，应向证书请求者发回一个证书或证书链，如果 CERTREQ 的接收者：

- 配置为使用证书认证；
- 允许发送 CERT 载荷；
- 与控制当前协商的 CA 置信策略相符，且具有至少一个 time-wise 且使用适当的终端实体。

则在 CERTREQ 中提供链接至一个 CA 的证书。

在用于证书选择的连锁阶段需要证书撤回检查。注意，即使两个对端配置为使用两个不同的 CA，选择逻辑中也应支持交叉证书关系。这样做的目的不是为了防止通过一个严格的基于 CERTREQ 的证书选择的粘合进行通信，当发端可以选择另外一个证书，这个证书在收端可以通过交叉认证、CRLs，或者其他带外配置手段进行成功确认。这样，对一个 CERTREQ 的处理应视作一个选择证书的建议，而不是确定证书的操作。如果没有证书，则 CERTREQ 应该被忽略，这在协议中不作为错误情况。也许会出现这样的情况：在 CERTREQ 中发送了一个优选的 CA，但（也许在提示了操作人员之后）另一个 CA 也可以接受。

#### 6.8 认证载荷 <sup>39</sup>

认证载荷（以 AUTH 表示）包含了用于认证的数据。“认证数据”的含义与下面所述的“认证方式”中的不同。

认证载荷的定义如下：

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		
下一负载	C	保留
认证方式		保留
		认证数据

认证载荷格式

——认证方式 (1 字节)：指定了认证的方式，其值的定义为：

- RSA 数字签名 (1)：以 5.16 节中的方式，使用在一个 PKCS#1 填充的哈希值之上的 RSA 私钥进行计算。
- 共享的密钥消息完整性编码 (2)：以 5.16 节中的方式，使用与在 ID 载荷中的身份以及协商的 prf 功能相关的共享密钥进行计算。
- DSS 数字签名 (3)：以 5.16 节中的方式，使用一个 SHA-1 哈希的 DSS 私钥进行计算
- 数值 0 以及 4: 200 为 IANA 保留。201~255 保留为私有应用。

——认证数据 (变长)：见 5.16 节。

认证载荷的载荷类型值为 39。

### 6.9 临时随机数载荷 40

临时随机数载荷在本标准中分别以 Ni (发起方现时) 和 Nr (响应方现时) 表示，包含了随机数据用以在交换阶段保证系统活跃以及防止重放攻击。

临时随机数载荷定义如下：

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		
下一负载	C	保留
		临时随机数数据

临时随机数载荷格式

——临时随机数数据 (变长)：包含了传送实体产生的随机数据。

临时随机数载荷的载荷类型值为 40。

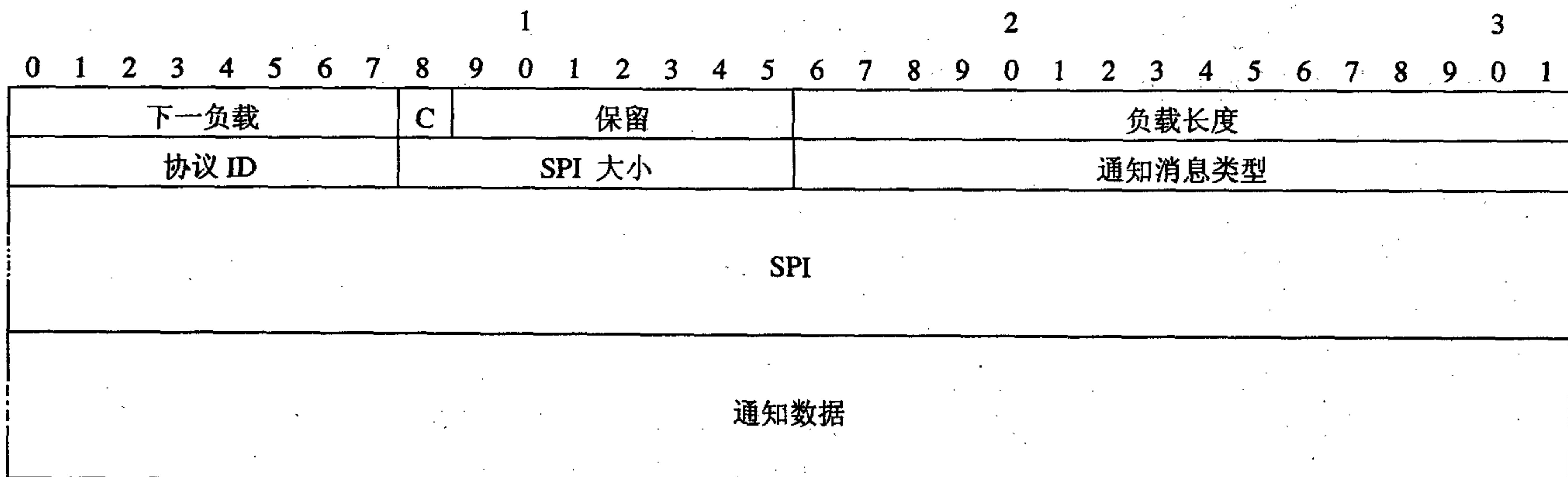
临时随机数载荷的长度必须为 16~256 字节。临时随机数数据不能重用。

### 6.10 通报载荷 41

本标准表示为 N，用于传送消息数据给 IKE 对端，例如错误状态和状态迁移消息。

一个通报载荷可能出现在一条响应消息中（通常是指出请求被拒绝的原因），在一个信息交互 INFORMATIONAL 交换中（用于报告一个不在 IKE 请求中的错误），或者在其他消息中表示发送方的能力或变更请求的意图。

通报载荷的定义如下所示：

**通报载荷格式**

—— 协议 ID（1字节）：协议 ID，如果通告关系到一个已经存在的 SA，这一部分表示该 SA 的类型。“1”代表 IKE\_SA 通告；“2、3”代表通告 IPsec SA，其中“2”表示 AH，“3”表示 ESP；“0”表示此通告与已经存在的 SA 无关，接收方忽略此段信息。这一字段的其他值都由 IANA 保留到未来再进行分配。

—— SPI 大小（1字节）：SPI 大小，此部分表示由 IPsec 协议 ID 定义的 SPI 长度字节数，此部分为 0 则表示没有可用的 SPI。关于 IKE\_SA 的通告此部分必须为 0。

—— 通知 Message 类型（2字节）：通报消息类型，定义通报消息的类型。

—— SPI（可变长度）：安全参数索引。

—— 通知数据（可变长度）：通报数据，关于通报消息类型的补充信息或者错误数据。该字段的数值已经按类型确定（如下所示）。

通告载荷的载荷类型是 41。

### 6.11 通告消息类型

通告消息可以是用于表明为什么不能建立 SA 的错误消息，也可以是管理 SA 数据库的一个进程希望与一个对等进程通信的状态数据。下表列举了通告消息以及与之对应的值，处于简化和避免泄漏配置信息的目的，错误状态的数量与 IKEv1 相比大大减少了。

类型在 0~16383 范围内用于报告错误，具体见表 14。没有响应确认的设备收到此类通告载荷必须认定相应的请求已经完全失败。请求中无法识别的错误类型，以及请求和响应中的状态类型必须忽略，需要记录的内容除外。

携带状态类型的通告载荷可以添加在任何消息中，如果没有被验证则必须忽略，具体见表 15。

这些载荷用于表示能力，并且作为 SA 协商的一部分用于协商非加密的参数。

表 14 通告消息差错类型

NOTIFY MESSAGES-ERROR 类型	值	描述
保留	0	
UNSUPPORTED_CRITICAL_PAYLOAD	1.	如果“紧急”比特位被置位而且载荷类型未识别则发送该错误消息。 通知数据包含一字节的载荷类型
INVALID_IKE_SPI	4	表示收到一条无法识别目的 SPI 的 IKE 消息。这通常意味着接收方已经重启无法识别原有的已存在的 IKE_SA
INVALID_MAJOR_VERSION	5	表示接收方无法处理报头中指明的 IKE 版本。在回复的报头中将会表明接收方所能支持的最接近的版本号码

表 14 (续)

NOTIFY MESSAGES-ERROR 类型	值	描述
INVALID_SYNTAX	7	表示收到的 IKE 消息无效，无效的原因可能是类型，长度，等超过了允许的范围。或者请求因政策原因被拒绝。为防止伪造消息的拒绝服务攻击，这一状态只能由加密的报文带回，要求消息 ID 和加密校验和是正确的。为避免信息从节点泄漏，这一类型必须被发送以代表任意无法被其他状态类型所表示的错误。为了辅助调试，更加详细的错误信息应该记录在控制台或者日志中
INVALID_MESSAGE_ID	9	当 IKE 消息 ID 在所能支持的窗口范围之外时发送此类型。此类通告不可以在一个响应中发送，不可以向无效的请求发送响应，因此，应该通过开始一个信息交换来通知对方，通告的数据由 4 字节的无效消息 ID 来携带。发送这种通告的行为是可选择的，这种类型的通告必须限制速率
INVALID_SPI	11	当节点收到携带无效 SPI 的 ESP 报文或 AH 报文时，可以在 IKE 信息交换中发送无效 SPI。在通告数据中包含无效报文的 SPI。这通常意味着节点已经重启并且已经遗忘了 SA。如果携带这类信息的消息被发送到 IKE_SA 关系对之外，接收者可以将这种现象视为一种已经发生了某种错误的提示（因为这很容易被伪造）
NO_PROPOSAL_CHOSEN	14	建议的加密对都不是可接受的
INVALID_KE_PAYLOAD	17	在 KE 载荷中的 D-H Group# 字段与应答者为进行交互所选择的 group# 不一致。有两个字节的数据与这一通告相关：以 big endian 顺序排列的可接受的 D-H Group#
AUTHENTICATION_FAILED	24	当因为某种原因认证失败时发送此类型通告作为 IKE_AUTH 消息的回应。没有相关的数据
SINGLE_PAIR_REQUIRED	34	这一错误表示 CREATE_CHILD_SA 请求不可接受，原因是发送方只愿意接受流量选项指定一个单一的地址对。要求发起请求的一方只为它希望转发的特定流量请求 SA
NO_ADDITIONAL_SAS	35	这一错误表示 CREATE_CHILD_SA 请求不可接受，原因是响应方在这一 IKE_SA 中不接受额外的 CHILD_SA。一部分小的实现，在初始化 IKE 交互的内容中只接受单一的 CHILD_SA，会拒绝任何添加更多 CHILD_SA 的尝试
INTERNAL_ADDRESS_FAILURE	36	表示响应者在处理一个配置载荷时错误地分配了一个内部地址（例如，内部 IP4 地址或者内部 IP6 地址），如果这个错误在 IKE_AUTH 交互中发生就不会生成 CHILD_SA
FAILED_CP_REQUIRED	37	响应者在期望收到 CP(CFG_REQUEST) 而没有收到的情况下发送此类型，这是与本地配置策略相冲突的。不携带相关的数据
TS_UNACCEPTABLE	38	表示在所提供的流量选项中包括地址、协议、端口都不可接受
INVALID_SELECTORS	39	当节点收到 ESP 或 AH 报文的选项与自身发送 SA 的选项不一致时（这会导致报文被丢弃），可以在 IKE_INFORMATIONAL 交互中发送此类型。通告的数据包含该违反规则报文的报头（像在 ICMP 消息中一样），通告的 SPI 部分被设置与 IPsec SA 的 SPI 匹配
IANA 保留 - Error 类型	40~8191	
私有使用- Errors 类型	8192~16383	

表 15 通告消息状态类型

NOTIFY MESSAGES-STATUS 类型	值	描述												
INITIAL_CONTACT	16384	通告声明这一 IKE_SA 是认证通过的实体之间当前可用的唯一 IKE_SA。当发生崩溃后建立起 IKE_SA 时发送，收到的一方可以利用此信息删除它与认证实体的其他 IKE_SA，而不必等到失效时间。这一通告不可以由可以被复制的实体发送（例如，一个漫游用户的信任状，用户被允许同时从两个远端系统连接到公司的防火墙上）												
SET_WINDOW_SIZE	16385	此通告声明发送方端点有能力同时保持多个活动的交互状态，允许接受到此通告的一方发送在收到第一个回应之前发送多个请求。与一个设置窗口大小通告关联的数据必须是 4 个字节，包含以高位在前方式描述的所能保持的消息数量。窗口大小在初始交互完成之前一直是 1												
ADDITIONAL_TS_POSSIBLE	16386	此通告声明发送的端点缩小了提供的流量选项范围，但是其他流量选项在分离的 SA（见 5.10 节）中仍然可以接受。没有与之关联的数据，通常在包含可接受的选项的消息中作为附加的载荷发送												
IPCOMP_SUPPORTED	16387	<p>此通告可以包含在携带 SA 载荷正在协商 CHILD_SA 的消息中，表示发送方愿意在 SP 上使用 IPComp。与此通告相关的数据包含两字节的 IPComp CPI，其后是一个字节的转换 ID（变换 ID），后面还可选地附加长度和格式由转换 ID 决定的属性内容。提议 SA 的消息可以包含多个支持 IPCOMP 通告以表示可支持多种机制，而接受 SA 的消息只能包含最多一个。</p> <p>转换 ID（变换 ID）目前的定义是：</p> <table> <tr> <td>名字</td> <td>数字</td> </tr> <tr> <td>保留</td> <td>0</td> </tr> <tr> <td>IPCOMP_OUI</td> <td>1</td> </tr> <tr> <td>IPCOMP_DEFLATE</td> <td>2</td> </tr> <tr> <td>IPCOMP_LZS</td> <td>3</td> </tr> <tr> <td>IPCOMP_LZJH</td> <td>4</td> </tr> </table> <p>5~240 由 IANA 保留，241~255 用于相互协商的双方的私有应用</p>	名字	数字	保留	0	IPCOMP_OUI	1	IPCOMP_DEFLATE	2	IPCOMP_LZS	3	IPCOMP_LZJH	4
名字	数字													
保留	0													
IPCOMP_OUI	1													
IPCOMP_DEFLATE	2													
IPCOMP_LZS	3													
IPCOMP_LZJH	4													
NAT_DETECTION_SOURCE_IP	16388	这一通告用于接收方确定源端是否在一个地址翻译黑盒中。与此通告相关的数据是一个 SHA-1 摘要，包括 SPI（以出现在报头中的顺序排列）、IP 地址、发送报文的端口号的内容。如果发送方不知道多个网络附加装置中的哪一个会用于发送报文，一个消息可以有多个此类的通告负载。接收到此通告的一方会将提供的数据与 SPI、源 IP 地址、端口得出的 SHA-1 哈希值比较，如果不匹配这需要使能 NAT 穿越（见 5.24 节）另一种情况是，如果 NAT 穿越不支持，接收方会拒绝连接的尝试												
NAT_DETECTION_DESTINATION_IP	16389	这一通告被接受方用于判断自己是否处于 NAT 黑盒之后。与此通告相关的数据是 SHA-1 摘要，包括 SPI（按照在报头中出现的顺序）、IP 地址、发送报文的端口。接收到此通告的一方会将提供的数据与 SPI、目的 IP 地址、端口得出的 SHA-1 哈希值比较，如果二者不匹配，则表示此端位于 NAT 之后，应该开始发送由 IETF RFC3948（2005）定义的 keepalive 报文。另一种情况是，如果不支持 NAT 穿越，将会拒绝连接的尝试												

表 15 (续)

NOTIFY MESSAGES – STATUS 类型	值	描述
COOKIE	16390	此通告可以包含在一个 IKE_SA_INIT 响应中。它表示请求方应该拷贝这个通告作为第一个载荷后重新发起请求。如果在初始的响应中包含有 COOKIE 通告，那么这个通告必须包含在 IKE_SA_INIT 的重新请求中。与此通告相关的数据长度必须是 1~64 字节
USE_TRANSPORT_MODE	16391	使用传输模式：此通告可以包含在包含有 SA 载荷请求 CHILD_SA 的消息中，它请求 CHILD_SA 使用传输模式而不是隧道模式来生成 SA。如果请求被接受，响应也必须包括使用传输模式的通告。如果响应方拒绝这一请求，CHILD_SA 将会使用隧道模式建立。如果发起方不接受隧道模式，就必须删除该 SA。 注意：除非使用此选项协商传输模式，所有的 CHILD_SA 都默认使用隧道模式。 注意：由 IKEv2 建立的所有的隧道模式 SA 都必须遵循由 IETF RFC4301 (2005) 定义的 ECN 解封装修订
HTTP_CERT_LOOKUP_SUPPORTED	16392	此通告可包含在任意可包含 CERTREQ 载荷的消息中，表示发送方能够查询基于 HTTP 的 URL 所支持的证书（因此可推测出发送方更倾向于接收该格式的证书规格）
REKEY_SA	16393	当有一个 CREATE_CHILD_SA 交互进行的目的是要取代一个已经存在的 ESP 或 AH SA 时，此通告必须包含在该交互中。由 SPI 字段定义 SA 正在被重定义密钥 (rekey)。没有相关联的数据
ESP_TFC_PADDING_NOT_SUPPORTED	16394	不支持 ESP_TFC_填充：此通告声明发送方端点不接收包含流机密填充的报文
NON_FIRST_FRAGMENTS ALSO	16395	用于分片控制，参见 IETF RFC4301 (2005) 解释
由 IANA 保留的- STATUS 类型	16396~40959	
私有应用- STATUS 类型	40960~65535	

## 6.12. 删除载荷 42

删除载荷，在本标准中表示为 D，包含协议指定的安全交互标识，该标识为发送方已经从自己的安全交互数据库中删除的，不再有效。有可能在一个删除载荷中发送多个 SPI，然而，每个 SPI 都必须是对应相同的协议。删除载荷中不能有多个不同协议的标识。当然，在单个信息交换中包含有多个删除载荷，每个删除载荷里的 SPI 协议类型不同是允许的。

IKE\_SA 的删除由协议 ID1 (IKE) 表示，没有 SPI。删除 CHILD\_SA，例如 ESP 或者 AH，需要包含对应的 IPsec 协议 ID (AH 为 2, ESP 为 3)，SPI 应该是发送方节点将要期望的入站 ESP 或 AH 报文的 SPI。

删除载荷的定义如下所示：

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 2 3 4 5 6 7 8 9 0 1
下一负载	C	保留	负载长度
协议 ID	SPI 大小	# of SPIs	Security Parameter Index(es) (SPI)

**删除载荷格式**

- 协议 ID (1 字节)：1 对应 IKE\_SA, 2 对应 AH, 3 对应 ESP。
- SPI 大小 (1 字节)：由协议 ID 定义的 SPI 长度，单位是字节。IKE (SPI 在消息头中) 必须是 0, AH 和 ESP 必须是 4。
- SPI# (2 字节)：删除载荷中包含的 SPI 数量，每个 SPI 的大小由 SPI 大小字段定义。
- SPI (可变长度)：标明要删除的特定安全交互。这一字段的长度由 SPI 大小字段和 SPI# 字段来确定。

该删除载荷的载荷类型为 42。

**6.12.1 厂商 ID 载荷 43**

一个厂商 ID 载荷可以声明发送方能够接受某些协议的扩展，或者可能简单的表示这一个实现是用来辅助进行调试。厂商 ID 载荷不可以改变此规范中任何信息定义的解释（例如，关键比特必须置为 0）。可以发送多个厂商 ID 载荷。厂商 ID 载荷不是一个实现必须发送的。

厂商 ID 可能作为一个消息的一部分发送。接收到熟悉的厂商 ID 载荷使得一个实现可以利用本文中说描述的私有作用数字，包括私有载荷、私有交互、私有通告等等。不熟悉的厂商 ID 必须忽略掉。

想要扩展本协议的互联网文稿作者必须定义一个厂商 ID 载荷，以声明在互联网文稿中可执行该扩展的能力。希望扩展的互联网文稿能够获得接受并且成为标准，能够由 IANA 从留给将来使用的范围内分配一些“幻数”，那么就无需使用厂商 ID 了。

厂商 ID 载荷字段的定义如下：

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 2 3 4 5 6 7 8 9 0 1
下一负载	C	保留	负载长度
厂商 ID (VID)			

**厂商 ID 载荷格式**

- 厂商 ID (可变长度)：选择厂商 ID 的人必须确保在任何 ID 注册机构中该 ID 的惟一性。一个好的实例是包括公司名、个人姓名或者其他。如果你想更显眼，你还可以包含选择该 ID 的地点的经纬度、时间以及一些随机输入。一个长的惟一的字串的摘要比其本身要好。

厂商 ID 载荷的载荷类型是 43。

**6.13 流量选择载荷 44/45**

流量选择载荷，文中以 TS 表示，它允许对等体识别 IPsec 安全业务需要处理的信息流。流量选择载荷

由IKE通用载荷头以及单个的流量选择符组成。

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
下一负载	C	保留
TS 数	保留	
<TS >		

#### 流量选择载荷格式

- TS 数（1字节）：流量选择符的数目。
- 保留：保留，发送时该字段必须置 0，接收时忽略该字段。
- TS（可变长度）：流量选择符，一个或多个流量选择符。

流量选择载荷的长度包括流量选择报头和所有的流量选择符的长度。

流量选择载荷的载荷类型在 SA 的发起端为 44，接收端为 45。

#### 6.13.1 流量选择符

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		
TS 类型	IP 协议 ID*	选择符长度		
起始端口号*	结束端口号*			
起始地址*				
结束地址*				

#### 流量选择符

\* 注：除TS类型和选择符长度之外，其他所有字段都依赖于TS类型。TS类型为7和8的流量选择字段，目前只定义了这两个TS类型值。

- TS 类型（1字节）：指定流量选择符的类型，具体见表 16。
- IP 协议 ID（1字节）：指定相关 IP 协议的 ID 值（如，UDP/TCP/ICMP）。ID 值为 0 表示流量选择符与协议 ID 无关——SA 可以运行所有协议。
- 选择符长度：选择符长度，指定包括报头在内的流量选择符字段的长度。
- 起始端口（2字节）：指定流量选择符允许的最小端口号。对于那些没有定义端口号或者所有端都允许的协议，本字段必须置 0。对 ICMP 协议，这两字节的字段类型和代码当作一个 16 比特的整数（高 8 位为类型，低 8 位为代码）。过滤所用的端口号就基于此字段。
- 结束端口（2字节）：指定流量选择符允许的最大端口号。对于那些没有定义端口号或者所有端口都允许的协议，本字段必须设为 65535。对 ICMP 协议，这两字节的字段类型和代码当作一个 16 比特的整数（高 8 位为类型，低 8 位为代码）。过滤所用的端口号就基于此字段。
- 起始地址，流量选择符中包含的最小地址（长度由 TS 类型决定）。

—— 结束地址，流量选择符中包含的最大地址（长度由 TS 类型决定）。

符合 IETF RFC4301(2005) 的系统为了表示“任意”端口必须将起始端口设为 0，结束端口设为 65535；注意：依据 IETF RFC4301 (2005)，“任意”端口包括“非透明”端口。符合 IETF RFC4301 (2005) 的系统为了表示“非透明”端口而不是“任意”端口，必须将起始端口设为 65535，结束端口设为 0。

表16 已分配的流量选择符类型字段值和相应的地址选择符数据

TS 类型	值	描述
保留	0~6	
TS_IPV4_ADDR_RANGE	7	IPv4 地址的范围，由两个 4 字节值表示。第一个值是 IPv4 起始地址（包括这个地址），第二个值是 IPv4 结束地址（包括这个地址）。包括在这两个地址范围内的所有地址都在列表中
TS_IPV6_ADDR_RANGE	8	IPv6 地址的范围，由两个 16 字节值表示。第一个值是 IPv6 起始地址（包括这个地址），第二个值是 IPv6 结束地址（包括这个地址）。包括在这两个地址范围内的所有地址都在列表中
IANA 保留	9~240	
私有使用	241~255	

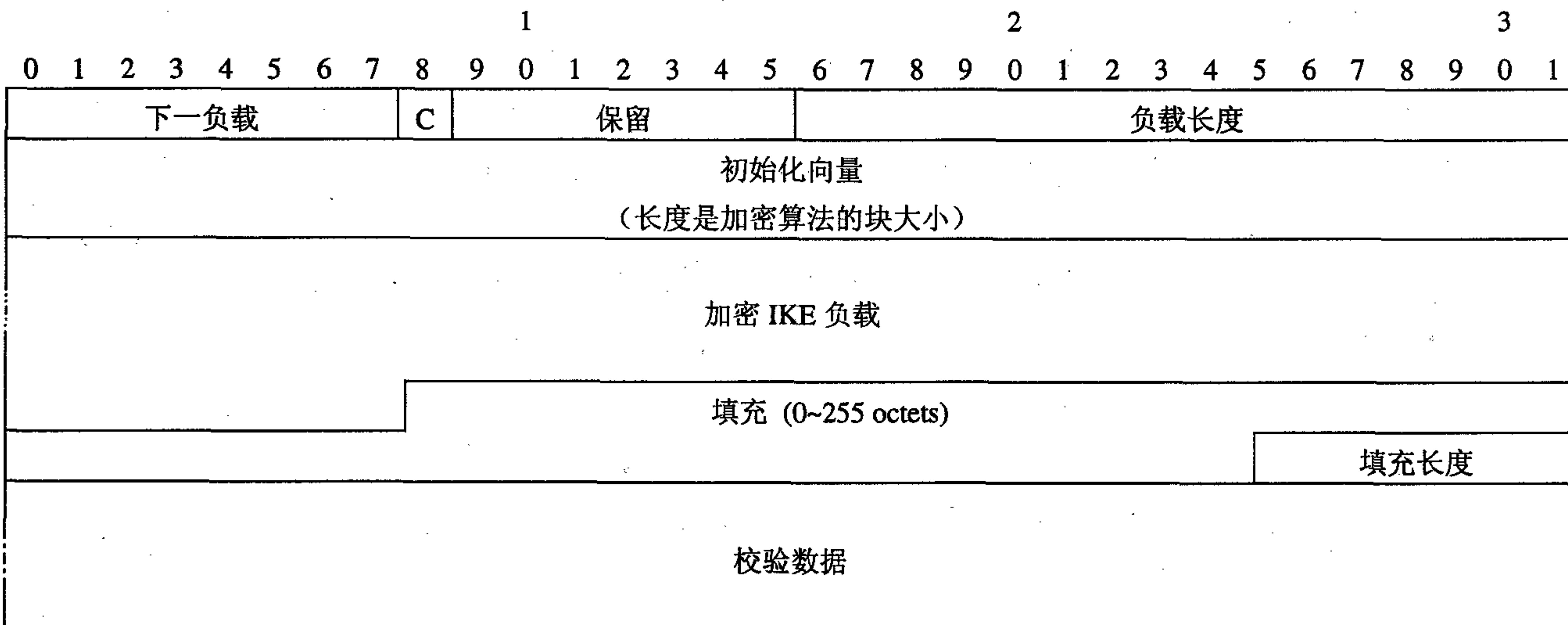
#### 6.14 加密载荷 46

加密载荷，标准中以 SK{...} 或 E 表示，它包含其他载荷的加密格式。如果消息中有加密载荷，则它必须是该消息的最后载荷。加密载荷通常是消息的唯一载荷。

加密和完整性保护算法在 IKE\_SA 建立期间进行协商，密钥的计算在 2.14 和 2.18 节有详细说明。

加密和完整性保护算法使用满足国家商用密码管理的 ESP 算法。ESP 中必须有一个具有固定长度的块密码以及一个可以计算可变长度消息定长校验和的完整性校验算法。

加密载荷的载荷类型是 46。加密载荷由一个 IKE 通用载荷头和接后的各个字段组成，如下所示。



—— 下一负载：下一载荷，第一个内含载荷的载荷类型。注意这是标准报头格式的一个例外：正常来说加密载荷是消息中的最后一个载荷，因此下一载荷字段应置为 0，但由于加密载荷中包含着其他载荷，而这些载荷中的第一个载荷的类型无处可放，于是就放在这个字段中。

—— 负载长度：载荷长度，包含报头、IV、加密后的 IKE 载荷、填充值、填充长度以及完整性校验数据的长度。

—— 初始化向量，一个随机选择的值，其长度等于隐含的加密算法的块长度。接收方必须能够接受任何值。发送方要么独立伪随机的为每条消息选择该值，要么使用前一个发送消息的最后一个密码块。发送方不能对各条消息使用同一个初始化向量值，也不能使用一序列的汉明距离很小的值（例如：顺序排列的数值），同样不能使用从接收消息中得到的密码。

—— 加密 IKE 负载 IKE 载荷在本节前面说明。本字段使用协商后的密码加密。

—— 填充：填充值可以包含发送方选择的任何值，它必须保证载荷、填充值以及填充长度的长度和成为加密块大小的整数倍。本字段使用协商后的密码加密。

—— 填充长度：填充长度是填充值字段的长度。在满足载荷、填充值以及填充长度的长度和为加密块大小整数倍的条件下，发送方应该设置填充长度为最小值，但是接收方必须接受所有能产生正确排列格式的填充长度。本字段使用协商后的密码加密。

—— 完整性校验数据是对整条消息的加密校验和，校验从固定的 IKE 报头开始到填充长度结束。校验和必须对加密后的消息计算得出，其长度由协商后的完整性算法决定。

### 6.15 配置载荷 <sup>47</sup>

配置载荷，文中以CP表示，<sup>client</sup>它用来在IKE对等体之间交换配置信息。<sup>server</sup>交换配置信息是为了让IRAC从IRAS那里请求一个内部IP地址，如果该IRAC直连到LAN，则还可以交换一些可以通过DHCP（动态主机配置协议）得到的其他信息。

配置载荷由CFG\_REQUEST/CFG\_REPLY 或CFG\_SET/CFG\_ACK组成（参考下述载荷描述中的CFG类型）。IKE请求可以选择添加CFG\_REQUEST和CFG\_SET载荷，而IKE应答则要么包含相对应的CFG\_REPLY或CFG\_ACK载荷，要么包含一个携带错误类型的通知载荷，指出请求为何没能正确响应。简化实现中可以有例外，这种情况下CFG\_REQUEST和CFG\_SET载荷可以被忽略，因而必须接收不包含相应的CFG\_REPLY或CFG\_ACK的响应消息，表示请求无法支持。

“CFG\_REQUEST/CFG\_REPLY”允许IKE终端从对等体请求信息。如果CFG\_REQUEST配置载荷中的某个属性字段长度不为0，则它将作为该属性的建议值。CFG\_REPLY配置载荷可以返回该建议属性值，或者返回一个新的属性值，或者还可以增加新的在CFG\_REQUEST配置载荷中没有请求的属性。请求发起者必须忽略它们不能识别的返回属性。

有些属性可以是多值的，在这种情况下，可以发送和/或返回同一个类型的多个属性值。一般来说，当一个属性被请求时，该属性的所有值都会被返回。对某些属性（本规范这一版本中只有内部地址这一个属性）来说，多个请求指的是一个被分配了多值的单个请求。对于这些属性，返回值的个数不应超过请求的个数。

如果CFG\_REQUEST中请求的数据类型不可被识别或不能被支持，应答方不能返回错误类型，而是必须返回一个空的CFG\_REPLY载荷或者一个根本不包含CFG\_REPLY载荷的应答。只有在请求能被识别但不能按照请求执行或者请求格式错误的情况下，应答方才返回错误。

“CFG\_SET/CFG\_ACK”允许一个IKE终端向其对等体推送配置数据。在这种情况下，CFG\_SET配置载荷包含有发起方需要应答方改变的属性。应答方如果接收了配置数据的任何部分，就必须返回一个配置载荷并且在其中包含应答方已接收的数据长度非0的属性。应答方不接受的属性则不能在CFG\_ACK配置载荷中包含。如果应答方一个属性都不接受，那么就必须返回一个空的CFG\_ACK载荷或者返回一个没有CFG\_ACK载荷的应答消息。尽管CFG\_SET/CFG\_ACK交换可用于基于提供者代码扩展相关方面，但目前

还没有对它的详细使用说明。本规范的简化实现可以忽略CFG\_SET载荷。

通过配置载荷进行扩展不应该用于通用管理目的，其主要目的是为了提供一种辅助机制，交换从IRAS到IRAC的IPsec消息。不过它也可以作为一些安全网关（SGW）或者小的网络之间交换信息的有效手段。对于企业管理以及随之而来的信息交换，应该优先使用现有的管理协议例如DHCP，RADIUS，SNMP，或者LDAP等。

配置载荷的格式如下：

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	3																								
下一负载	C	保留	负载长度																																																					
CFG 类型	保留																																																							
配置属性																																																								

配置载荷格式

配置载荷的载荷类型是47。

—— CFG 类型（1个字节）：配置属性所代表的交换类型

CFG类型	值
保留	0
CFG_REQUEST	1
CFG_REPLY	2
CFG_SET	3
CFG_ACK	4

类型值5~127保留在国际互联网地址分配委员会（IANA），值128~255用于相互同意的各方私用。

—— 保留（3个字节）：保留，发送时必须置0，接收时忽略。

—— 配置属性（可变长度）：配置属性，指示配置载荷的类型长度值，它的格式见下节。配置载荷中可能有0或多个配置属性。

### 6.15.1 配置属性

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	3
R	属性类型	长度																														
值																																

配置属性格式

R（1比特）：保留位，发送时该比特位必须置0，接收时忽略。

—— 属性类型（15比特）：属性类型，每个配置属性类型的唯一标识符。

—— 长度（2个字节）：长度，属性值的长度。

—— 值（0或多个字节）：属性值该配置属性的可变长度值。

属性类型的定义具体见表17。

表 17 属性类型（属性类型）定义

属性类型	值	多 值	长 度
保留	0		
INTERNAL_IP4_ADDRESS	1	是*	0或4字节
INTERNAL_IP4_NETMASK	2	否	0或4字节
INTERNAL_IP4_DNS	3	是	0或4字节
INTERNAL_IP4_NBNS	4	是	0或4字节
INTERNAL_ADDRESS_EXPIRY	5	否	0或4字节
INTERNAL_IP4_DHCP	6	是	0或4字节
APPLICATION_VERSION	7	否	0或多个字节
INTERNAL_IP6_ADDRESS	8	是*	0或17字节
保留	9		
INTERNAL_IP6_DNS	10	是	0或16字节
INTERNAL_IP6_NBNS	11	是	0或16字节
INTERNAL_IP6_DHCP	12	是	0或16字节
INTERNAL_IP4_SUBNET	13	是	0或8字节
SUPPORTED_ATTRIBUTES	14	否	2的整数倍字节
INTERNAL_IP6_SUBNET	15	是	17字节

\* 这些属性只有在多个值被请求时才可以返回多个值。

属性类型值16~16383保留在国际因特网地址分配委员会（IANA），值16384~32767用于相互同意的各方私用。

— INTERNAL\_IP4\_ADDRESS, INTERNAL\_IP6\_ADDRESS: 内部网络地址，有时也称为红色节点地址（red node address）或私有地址，可以是一个私有地址。该项用于在请求消息中指定要请求的地址（如果没有要请求的特定地址则该项为 0）。如果某个特定的地址被请求，很可能说明之前已经有过一个与该地址的连接并且发起方要再次使用这个地址。对于 IPv6，发起方可能会给出它想使用的低地址位。通过请求多个内部地址属性，可以请求多个内部地址。应答方可以只发送与请求数目相等的地址个数。INTERNAL\_IP6\_ADDRESS 由两个字段组成：一个 16 字节的 IPv6 地址和一个 1 字节的长度前缀，在 IETF RFC3513（2003）中有定义。

被请求的地址将一直有效，除非内部地址存活期属性中定义的时间到期或者对等体之间已经没有了 IKE\_SA。

— INTERNAL\_IP4\_NETMASK: 内部网络掩码。在请求和应答消息中只允许有一个网络掩码（例如 255.255.255.0），并且该属性必须只能与内部 IP4 地址属性一起使用。

— INTERNAL\_IP4\_DNS, INTERNAL\_IP6\_DNS: 指定网络中的一个 DNS 服务器地址。发起方可以请求多个 DNS 服务器。应答方可以返回 0 或多个 DNS 服务器属性。

— INTERNAL\_IP4\_NBNS, INTERNAL\_IP6\_NBNS: 指定网络中网络输入输出系统名称服务器（WINS）地址。发起方可以请求多个 NBNS 服务器。应答方可以返回 0 或多个 NBNS 服务器属性。

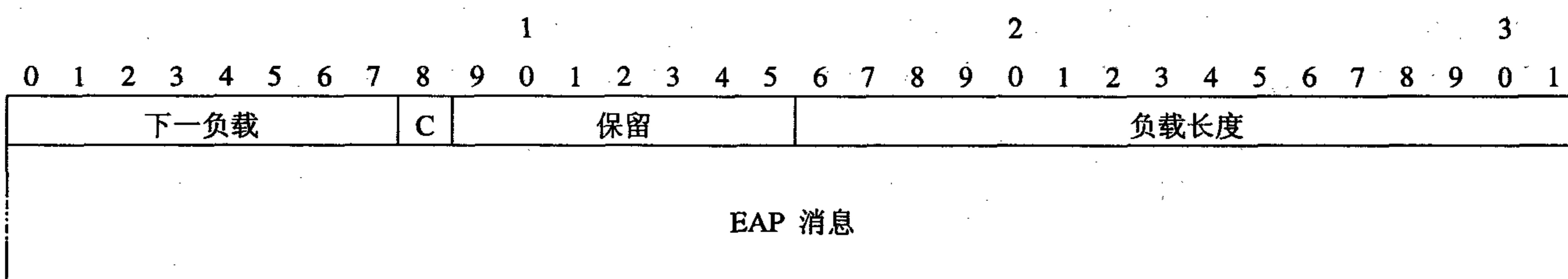
— INTERNAL\_ADDRESS\_EXPIRY: 指定主机能够使用内部 IP 地址的秒数。主机必须在该存活时间到期之前更新 IP 地址。这些属性中只有一个可以出现在应答中。

— INTERNAL\_IP4\_DHCP, INTERNAL\_IP6\_DHCP: 通知主机发送内部 DHCP 请求到该属性包含的地址上。发起方可以请求多个 DHCP 服务器。应答方可以返回 0 或多个 DHCP 服务器属性。

- 应用版本：IPsec 主机的版本或应用信息。它是一个可打印的不以空格结束的 ASCII 字符串。
  - INTERNAL\_IP4\_SUBNET：边缘设备所要保护的子网。该属性由两个字段组成：一个 IP 地址和一个子网掩码。发起方可以请求多个子网。应答方可以返回 0 或多个子网属性。
  - SUPPORTED\_ATTRIBUTES：当在请求中使用时，该属性长度必须置 0，它向应答方查询并请求返回应答方所有支持的属性。应答包含一系列属性标识符，每个标识符 2 个字节。长度除以 2（字节）可以得到应答中包含的所支持的属性个数。
  - INTERNAL\_IP6\_SUBNET：边缘设备所要保护的子网。该属性由两个字段组成：一个 16 字节的 IPv6 地址和一个 1 字节的长度前缀。发起方可以请求多个子网。应答方可以返回 0 或多个子网属性。
- 注意：在一次具体实现中如何确定回复中应发送的信息，本标准不做建议。也就是说，我们不推荐任何一种特定的方法来决定 IRAS 将哪个 DNS 服务器返回给请求的 IRAC。

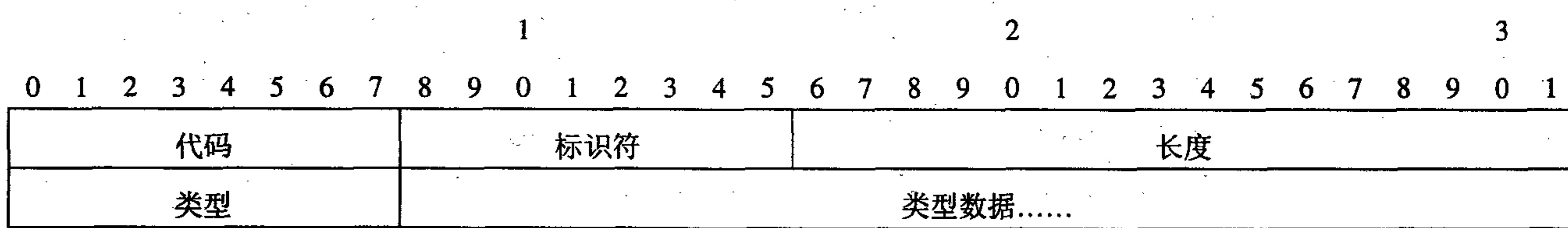
## 6.16 扩展认证协议 (EAP) 载荷

扩展认证协议载荷，文中以 EAP 表示，它允许使用 IETF RFC3748 (2004) 中定义的协议及其后续扩展协议来验证 IKE\_SA。该载荷可接受值的全集在其他地方定义，不过本文包含了 IETF RFC3748 (2004) 的概要，这样使本文在通常情况下可以自成一体。



EAP 载荷格式

EAP 载荷的载荷类型是 48。



EAP 消息格式

- 代码 (1 字节)：指出该消息是请求 (1) 还是应答 (2)，是成功 (3) 还是失败 (4)。
- 标识符 (1 字节)：在 PPP 中用来区分重播消息 (replayed messages) 和重复消息。在 IKE 中，EAP 运行于可靠协议之上，因此标识符在这里没用。在应答消息中，该字节必须置为和相应请求中的标识符一致。其他消息中该字段可以被设置为任何值。
- 长度 (2 字节)：是 EAP 消息的长度，它必须比封装载荷的长度小 4。
- 类型 (1 字节)：只有在代码字段为请求 (1) 或应答 (2) 时出现。对于其他代码，EAP 消息长度必须是四个字节并且类型和类型数据字段不能出现。在请求消息 (1) 中，类型值指出所要求的数据。在应答消息 (2) 中，类型值必须为空 (Nak) 或者与要求的数据类型一致。在 IETF RFC3748 (2004) 中定义了以下类型：

- 1 Identity
- 2 Notification

- 3 Nak (Response Only)
- 4 MD5-Challenge
- 5 One-Time Password (OTP)
- 6 Generic Token Card

——类型数据（可变长度）随请求以及相关的应答类型而变化。

注意：因为IKE在协议的消息3中传递了发起方身份标记，因此应答方不必发送EAP身份请求。然而发起方如果收到身份请求的话，应该作出回应。

## 7 一致性要求

为了所有 IKE2 的实现方式能互操作，除了在本文其他地方列出的要求外，本节还列举了一些必须支持的要求。当然，IKEv2 是一个安全协议，它的主要功能之一就是只允许被授权方建立 SA 成功。因此，可以为一个特定的实现方式配置任意数量的限制条件，这些限制条件与算法有关，也与防止通用互操作的授权机构有关。

设计 IKEv2 是为了让最低限度的实现方式能够与所有相匹配的实现方式互操作。有一些特性是可选的，如果某种实现方式不支持该特性，可以很容易地忽略该特性。这些特性包括：

- 穿越 NAT 协商 SA 的能力以及通过 UDP 以隧道方式传送结果 ESP SA 的能力。
- 在隧道的远端请求得到临时的 IP 地址（或响应请求）的能力。
- 支持各种类型的遗留认证的能力。
- 支持大于一的窗口大小的能力。
- 在一个 IKE SA 内建立多个 ESP 和/或 SA 的能力。
- 重键 SA 的能力。

为了确保互操作性，所有的实现方式必须能够分析所有的载荷类型（如果仅仅为了跳过载荷），除非载荷头中的关键比特被置位，否则忽略不支持的载荷类型。如果在不支持的载荷头中有关键比特被置位，则所有实现方式必须丢弃包含这些载荷的消息。

每种实现方式必须能够处理 IKE\_SA\_INIT 及 IKE\_AUTH 消息的交换，建立两种 SA（一个用于 IKE，一个用于 ESP 和/或 AH）。如果对于其平台适用，实现方式可以是仅发起或仅响应的。每种实现方式必须能够响应 INFORMATIONAL 交换，但最低实现方式可以用一个空的 INFORMATIONAL 应答来响应任意 INFORMATIONAL 消息（注意在一个 IKE\_SA 上下文内，“空”消息的结构为一个 IKE 头，后边跟着一个不包含载荷内容的加密的载荷）。只有在识别请求并且用类型为 NO\_ADDITIONAL\_SAS 的通知载荷拒绝请求时，实现方式可以支持 CREATE\_CHILD\_SA 交换。最低实现方式不必具备发起 CREATE\_CHILD\_SA 或 INFORMATIONAL 交换的能力。（根据本地配置的生存期或经过字节的值）当 SA 过期时，实现方式可以尝试用 CREATE\_CHILD\_SA 交换来更新它，也可以删除（关闭）旧的 SA 并且建立一个新的 SA。如果响应方用 NO\_ADDITIONAL\_SAS 通知来拒绝 CREATE\_CHILD\_SA 请求，则实现方式必须能够关闭旧的 SA 并且建立新的 SA。

不要求实现方式能够请求临时 IP 地址或响应这类请求。如果实现方式确实能够发送这类请求，则必须在消息 3 的 CP 载荷中包含至少一个 INTERNAL\_IP4\_ADDRESS 或 INTERNAL\_IP6\_ADDRESS 类型的字段。所有其他的字段都是可选的。如果实现方式支持对这类请求的响应，则必须分析消息 3 中 CFG\_REQUEST 类型的 CP 载荷，并且识别 INTERNAL\_IP4\_ADDRESS 或 INTERNAL\_IP6\_ADDRESS

类型的字段。如果它能租用适当类型的地址，则必须返回一个类型为 CFG\_REPLY 的 CP 载荷，其中包含所请求的类型的地址。如果响应方有其他相关的属性，则应包含所有这种属性。

除非为了确定 CP 载荷的内容包含一个 INTERNAL\_IP4\_ADDRESS 属性，并且无论发起方是否请求得到地址和相关的属性，都要在响应中将其包含进去，否则最低 IPv4 响应实现方式会忽略 CP 载荷的内容。

最低能力 IPv4 发起方至少会产生一个 CP 载荷，其中仅包含一个 INTERNAL\_IP4\_ADDRESS 属性，并且会分析响应消息，忽略不知如何使用的属性。它必须能够处理的惟一属性就是 INTERNAL\_ADDRESS\_EXPIRY，必须用这个属性来界定 SA 的生存期，除非它可以成功地在到期之前更新租期。最低能力发起方不必具备请求租期更新的能力，并且最低能力响应方不必进行响应。

对于声称符合本规范的实现方式，必须能配置为接受下列信息：

满足国家标准的证书。

共享密钥认证，传送的 ID 为 ID\_KEY\_ID、ID\_FQDN 或者 ID\_RFC822\_ADDR 中的任意一种。

认证，此时响应方使用证书进行认证，并且对发起方的认证使用共享密钥认证方式。

## 8 安全性考虑

本协议是为了最大程度减少向非认证对等体泄漏配置信息的机会而设计的，这种泄漏中有些是不可避免的。两个对等体之间，必须要有一个首先标识自己，并且首先证明自己的身份。为了避免探测，要求交换的发起方首先标识自己，通常要求它首先证实自己。发起方可以获知响应方支持 IKE，以及响应方支持何种加密协议。响应方（或者假扮响应方的某些实体）可以对发起方进行探测，不仅探测它的身份，并且可能利用 CERTREQ 载荷来确定发起方打算使用哪种证书。

使用 EAP 认证一定程度上改变了探测的可能性。使用 EAP 认证时，在发起方证实身份之前，响应方先要证实自己的身份，因此，如果知道了正当的发起方的名字，那么（假扮的）发起方就可以探测响应方的名字和证书。

如果用 CREATE\_CHILD\_SA 重复性地使用密钥，不进行附加的 Diffie-Hellman 交换，任意一个端点使用单一的密钥或者过度使用密钥，就会使所有 SA 容易遭受密码分析学攻击。实现方式应当注意这种事实，对幂指数之间的 CREATE\_CHILD\_SA 交换设置一个界限。本标准对这种界限不做规定。

使用本文定义的任意一组 Diffie-Hellman 交换推导出的密钥的强壮性取决于该组固有的强壮性、使用的指数的大小、使用的随机数发生器提供的熵。由于有这些输入，很难确定任意一个被定义组的密钥的强壮性。使用一个强壮的随机数发生器和一个不小于 200 比特的指数时，Diffie-Hellman 组号 2 通常与 3DES 一起使用。组 5 比组 2 的安全性高。组 1 仅仅是历史上使用的，除了用于 DES 外，没有提供足够的强壮性，而 DES 也仅仅是历史上使用的。实现方式在建立策略和协商安全参数时应该注意这些评价。

注意这些限制是用于 Diffie-Hellman 组自身的。IKE 中并不禁止使用更强壮的组，也不削弱从更强壮的组获得的强壮性（受限于包括 prf 函数在内的其他协商的算法的强壮性）。实际上，IKE 的可扩展框架鼓励定义更多的组；使用更小的数字时，椭圆曲线组的使用可能会大大提高其强壮性。

假定所有的 Diffie-Hellman 指数在使用后都要从存储器中擦除。特别是不允许从寿命长久的秘密信息中推导指数，例如使用后不擦除的伪随机数发生器的种子。

所有密钥的强壮性受限于被协商的 prf 函数的输出的大小。由于这个原因，本协议不允许使用输出小于 128 比特的 prf 函数。

本协议的安全性主要取决于随机选择的参数的随机性。这些参数应当由一个强壮的随机源或适当播种的伪随机源产生。实现者应注意保证两个密钥的随机数的使用，以及临时随机数的生成方式不破坏密钥的安全性。

虽然协商的 CHILD\_SA 的安全性不依赖于在 IKE\_SA 中协商的加密和完整性保护的强壮性，但不允许实现方式将 NONE 协商为 IKE 的完整性保护算法，或将 ENCR\_NULL 协商为 IKE 加密算法。

使用预共享密钥时，关键要考虑如何确保这些秘密信息的随机性。最有力的措施就是保证任何预共享密钥的随机性与协商的最强健的密钥的随机性相同。从口令、名称或其他低熵源中推导共享的秘密信息是不安全的。这些源会遭受字典或社会工程学攻击。

NAT\_DETECTION\_\*\_IP 通知中包含地址和端口的哈希信息，这是为了隐藏 NAT 后边的内部 IP 地址。由于 IPv4 地址长度仅为 32 比特，并且通常十分稀缺，通过对所有可能的 IP 地址进行尝试，并且寻找匹配的哈希信息，攻击者有可能会发现 NAT 设备后边的内部地址。

端口号通常固定为 500 个，而且从包中可以提取 SPI。这就减少了  $2^{32}$  的哈希计算数值。再采用训练有素的方法对专用地址空间进行猜测，哈希计算数值就小很多。因此设计者不应假定使用 IKE 就不会泄漏内部地址信息。

EAP 认证方式不产生用于保护随后的 AUTH 载荷的共享密钥，使用这种方式时，可能会遭受中间人以及假扮服务器的攻击。当 EAP 用在没有安全隧道保护的协议中时，容易遭受这种攻击。由于 EAP 是通用认证协议，通常用于提供单点登入设施，因此采用 EAP 认证方式部署的 IPsec 解决方案的安全性可能会受到威胁。EAP 认证方式不产生共享密钥（也被称作无密钥产生的 EAP 方式），如果部署了一个完全无关的应用，该应用碰巧使用了同样的无密钥产生的 EAP 方式，但采用了无保护的形式，就会危及上述 IPsec 解决方案的安全。注意这种弱点不仅限于 EAP，也可能在重新使用认证基础结构时在其他情况下发生。例如，如果 IKEv2 使用的 EAP 机制使用了一个令牌认证器，中间人攻击者可以假扮网络服务器，阻截令牌认证交换，并用它来发起一个 IKEv2 连接。由于这个原因，应避免将无密钥产生的 EAP 方式用在可能出现这种情况的地方。无论用在哪里，使用 EAP 方式时应该使用被保护的隧道，在发起 EAP 交换之前，发起方要对响应方的证书进行验证，这一点非常重要。实现方式应在其文档中介绍使用无密钥产生的 EAP 方式的脆弱性，以便让部署 IPsec 解决方案的管理者感觉到这些危险的存在。

对于使用 EAP 的实现方式，在 EAP 交换开始之前，即使 EAP 方式提供了双向认证，也必须使用从服务器到客户机的基于公开密钥的认证。这样就避免了附加的 IKEv2 协议变化，并且保护了 EAP 数据，躲避活跃的攻击者。

如果 IKEv2 消息足够长，需要进行 IP 层的分段，攻击者有可能通过耗尽重组缓存器空间而使得交换无法完成。使用 Hash 和 URL 编码而不是发送证书的方式（参见 6.6 节）可以将这种机会减少到最少。

附录 A  
(资料性附录)  
与 IKE 版本 1 的区别汇总

本标准中IKE（版本2）的目标是：

- 1) 在一个单独的文档中定义完整的 IKE 协议以替代 IETF RFC2407 (1998), IETF RFC2408 (1998) 和 IETF RFC2409 (1998)，并包括支持 NAT 穿越、可扩展认证、远程地址获得这些功能而产生的与版本 1 的区别。
- 2) 通过用一个单独的四消息交换（认证机制的改变仅仅是影响了一个单独的 AUTH 载荷，而不是重新构造整个交换）替代八个不同的初始交换，而简化了 IKE；
- 3) 去掉了解释域 (DOI)，状况 (SIT)，以及标记域标识符字段，以及提交和认证比特；
- 4) 为减少 IKE 在通常情况下的延迟，使用了 2 次往返 (4 消息) 的初始交换，并允许在该交换中稍带建立一个 CHILD\_SA；
- 5) 使用大致基于 ESP 的算法替代了用于保护 IKE 消息自身的加密算法，以简化实施和安全分析；
- 6) 通过使协议可靠（所有消息都被确认）并顺序，来减少可能的差错消息的数量。这可以使 CREATE\_CHILD\_SA 交换从 3 个消息缩短到 2 个；
- 7) 增加了鲁棒性，允许响应者不进行明显操作，直到其收到证明初始者能够以其声称的 IP 地址接收消息的消息，并且允许响应者不提交任何状态给交换，直到初始者能够被加密认证；
- 8) 修复了密码的脆弱性，例如用于认证的哈希算法的对称问题；
- 9) 在其自身的载荷类型中定义了流量选择器，取消了过载 ID 载荷，并且流量选择器的定义更加灵活；
- 10) 定义了特定差错条件下或者当接收到不理解的数据时必需的行为，并使将来的修订版本工作更加简单，不必破坏前向兼容性；
- 11) 简化并明确在网络失效以及受到 DOS 攻击的情况下如何维持共享的状态；
- 12) 维持现有语法和语义，以尽可能使得 IKE 版本 1 的实施能够通过最小化的功能增强以支持 IKE 版本 2。

附录 B  
(资料性附录)  
Diffie-Hellman 组

定义了两个用于 IKE 的 Diffie-Hellman 组。这些组是亚利桑那大学的 Richard Schroepel 生成的。

第一个 Diffie-Hellman 组提供的强壮性可能不满足强制实施加密算法的需求，只是由于历史原因还放在了本标准中。

组 1 - 768 比特 MODP

分配该组的 ID 为 1 (一)。

原始数为:  $2^{768} - 2^{704} - 1 + 2^{64} \times \{ [2^{638} \text{ pi}] + 149686 \}$ 。

其十六进制的值为:

FFFFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08  
8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B  
302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9  
A63A3620 FFFFFFFF FFFFFFFF

其生成 (generator) 为 2。

组 2 ~ 1024 比特 MODP

分配该组的 ID 为 2 (二)。

质数为:  $2^{1024} - 2^{960} - 1 + 2^{64} \times \{ [2^{894} \text{ pi}] + 129093 \}$ 。

其十六进制的值为:

FFFFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08  
8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B  
302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9  
A637ED6B 0BFF5CB6 F406B7ED EE386FB8 5A899FA5 AE9F2411 7C4B1FE6  
49286651 ECE65381 FFFFFFFF FFFFFFFF

其生成器为 2。

## 参 考 文 献

- |               |                  |
|---------------|------------------|
| IETF RFC 3439 | 一些互联网体系指导与思想     |
| IETF RFC 4302 | IP认证头            |
| IETF RFC 4303 | IP封装安全载荷（ESP）    |
| IETF RFC 4306 | 互联网密钥交换（IKEv2）协议 |
-

中华人民共和国  
通信行业标准  
互联网密钥交换协议（IKEv2）技术要求

YD/T 1897-2009

\*

人民邮电出版社出版发行  
北京市崇文区夕照寺街 14 号 A 座  
邮政编码：100061  
北京新瑞铭印刷有限公司印刷  
**版权所有 不得翻印**

\*

开本：880×1230 1/16 2009 年 8 月第 1 版

印张：3.75 2009 年 8 月北京第 1 次印刷

字数：101 千字

ISBN 978 - 7 - 115 - 1880/09 - 122

本书如有印装质量问题，请与本社联系 电话：(010)67114922