

Network Working Group Request for Comments:
2409 Category: Standards Track

D. Harkins
d·卡雷尔
Cisco Systems
November 1998

互联网密钥交换(IKE)

本备忘录现状

本文档为互联网社区指定了一个互联网标准跟踪协议，并要求讨论和提出改进建议。关于本协议的标准化状态和状态，请参考当前版本的“互联网官方协议标准”(STD 1)。本备忘录的分发是无限制的。

版权声明

版权所有(C)互联网协会(1998)。版权所有.

目录表

1 文摘	2
2 讨论	2
3 术语和定义	3
3.1 要求术语	3
3.2 符号	3
3.3 完美前向保密	5
3.4 安全协会	5
4 介绍	5
5 交流	8
5.1 数字签名认证	10
5.2 公钥加密认证	12
5.3 一种改进的公钥加密认证方法	13
5.4 预共享密钥认证	16
5.5 快速模式	16
5.6 新群组模式	20
5.7 ISAKMP信息交流	20
6 奥克利组	21
6.1 第一奥克利集团	21
6.2 第二奥克利集团	22
6.3 第三Oakley集团	22
6.4 第四奥克利集团	23
7 有效载荷爆炸完成交换	23
7.1 阶段1带主模式	23
7.2 第二阶段带快速模式	25
8 完美前向保密示例	27
9 实现提示	27

10 安全考虑.....	28
11 IANA 考虑.....	30
12 应答.....	31
13 引用.....	31
附录A.....	33
附录B.....	37
作者的地址.....	40
作者的注意.....	40
完整版权声明.....	41

1. 文摘

ISAKMP ([MSST98]) 提供了一个认证和密钥交换的框架，但没有定义它们。 ISAKMP 设计为独立于密钥交换；也就是说，它被设计成支持多种不同的密钥交换。

Oakley ([Orm96]) 描述了一系列密钥交换——称为“模式”——并详细说明了每种模式提供的服务(例如，密钥的完美前向保密、身份保护和身份验证)。

SKEME ([SKEME]) 描述了一种通用的密钥交换技术，它提供匿名性、可抵免性和快速密钥刷新。

本文档描述了一个协议，该协议使用 Oakley 和 SKEME 的一部分与 ISAKMP 结合使用，以获得用于 ISAKMP 和其他安全关联(如 IETF IPsec DOI 的 AH 和 ESP)的认证密钥材料。

2. 讨论

本备忘录描述了一种混合协议。其目的是以受保护的方式协商并为安全关联提供经过认证的密钥材料。

实现此备忘录的进程可用于协商虚拟专用网络(vpn)，也可用于从远程站点(其IP地址无需事先知道)提供远程用户对安全主机或网络的访问。

支持客户端协商。在客户端模式下，协商方不是正在进行安全关联协商的端点。在客户端模式下使用时，终端各方的身份仍然是隐藏的。

这并没有实现整个Oakley协议，而只是实现其目标所需的一个子集。它不声称符合或遵从整个Oakley协议，也不以任何方式依赖于Oakley协议。

同样，这并没有实现整个SKEME协议，而只是实现了用于身份验证的公钥加密方法及其使用随机数交换快速重设密钥的概念。该协议不以任何方式依赖于SKEME协议。

3. 条款和定义

3.1 要求术语

本文档中出现的关键词“MUST”、“MUSTNOT”、“必需”、“SHOULDNOT”、“SHOULD不”和“MAY”应按[br97]中所述解释。

3.2 符号

以下符号在本备忘录中使用。

HDR是ISAKMP报头，其交换类型为mode。当写为HDR*时，它表示有效负载加密。

SA是具有一个或多个提议的SA协商有效负载。发起者MAY提供多个谈判建议；应答者MUST只回复一个。

<P>_b表示有效载荷<P>的主体——不包括ISAKMP通用有效载荷。

SAi_b是SA有效载荷的整个主体(减去ISAKMP通用标头)——即DOI、情况、发起者提供的所有提议和所有转换。

CKY-I和CKY-R分别是来自ISAKMP报头的发起者和响应者的cookie。

g^x_i 和 g^x_r 分别是发起者和响应者的迪菲-赫尔曼 ([DH]) 公共值。

g^{xy} 是迪菲-赫尔曼共享秘密。

KE是密钥交换有效载荷，它包含在迪菲-赫尔曼交换中交换的公共信息。KE有效载荷的数据没有特定的编码(例如TLV)。

Nx是nonce有效载荷;对于ISAKMP启动器和响应器, x可以分别为:i或r。

IDx是“x”的标识有效载荷。在第一阶段协商中, ISAKMP发起方和响应方分别为“ii”或“ir”;或“ui”或“ur”分别代表第二阶段的用户发起者和响应者。Internet DOI的ID有效载荷格式在[Pip97]中定义。

SIG是签名有效载荷。要签名的数据是特定于交易所的。

CERT是证书有效载荷。

HASH(以及任何导数, 如HASH(2)或HASH_I)是哈希有效载荷。哈希的内容是特定于认证方法的。

Prf(key, msg)是键控伪随机函数——通常是键控哈希函数——用于生成看似伪随机的确定性输出。prf既用于密钥派生, 也用于身份验证(即作为键控MAC)。(见[KBC96])。

SKEYID是一个由秘密材料衍生而来的字符串, 只有交易所中的活跃玩家才知道。

SKEYID_e是ISAKMP SA用来保护其消息机密性的密钥材料。

SKEYID_a是ISAKMP SA用来验证其消息的密钥材料。

SKEYID_d是用于派生非isakmp安全关联密钥的密钥材料。

<x>y表示“x”用密钥“y”加密。

-->表示“发起者到响应者”通信(请求)。

<--表示“响应者到发起者”通信(应答)。

|表示信息的串联——例如X | Y是X与Y的串联。

[x]表示x是可选的。

消息加密(当在ISAKMP报头后面用“*”标记时)MUST在ISAKMP报头之后立即开始。当通信受到保护时，ISAKMP报头之后的所有有效载荷MUST加密。加密密钥是从SKEYID_e生成的，其方式是为每个算法定义的。

3.3 完美前向保密

在备忘录中使用时，完美前向保密(PFS)指的是单个密钥的妥协将只允许访问受单个密钥保护的数据。为了PFS的存在，用于保护数据传输的密钥绝对不能被用来衍生任何额外的密钥，如果用于保护数据传输的密钥是从其他密钥材料中衍生出来的，则该材料绝对不能被用来衍生任何更多的密钥。

该协议为密钥和身份提供了完美前向保密。(章节5.5和章节8)。

3.4 安全协会

安全关联(SA)是一组用于保护信息的策略和密钥。ISAKMP SA是协议中协商对等体用来保护其通信的共享策略和密钥。

4. 介绍

Oakley和SKEME各自定义了一种方法来建立经过身份验证的密钥交换。这包括有效载荷构造、有效载荷携带的信息、它们被处理的顺序以及如何使用它们。

Oakley定义了“模态”，而ISAKMP定义了“相”。两者之间的关系非常直接，IKE将不同的交换呈现为在两个阶段之一中运行的模式。

阶段1是两个ISAKMP对等体建立一个安全的、经过身份验证的通道进行通信的阶段。这被称为ISAKMP安全协会(SA)。“主模式”和“攻击模式”各完成一个阶段1交换。“主模式”和“攻击模式”必须仅在阶段1中使用。

阶段2是安全协会代表服务进行协商的地方，例如IPsec或任何其他需要关键材料和/或参数协商的服务。“快速模式”完成第2阶段交换。“快速模式”必须仅用于阶段2。

“新组模式”实际上并不是第一阶段或第二阶段。它遵循第一阶段，但用于建立一个新的小组，可以在未来的谈判中使用。“新组模式”必须仅在阶段1之后使用。

ISAKMP SA是双向的。也就是说，一旦建立，任何一方都可以启动快速模式、信息模式和新组模式交换。根据基本ISAKMP文档，ISAKMP SA由发起者的cookie和响应者的cookie标识——阶段1交换中各方的角色决定了哪个cookie是发起者的。无论快速模式、信息交换或新组交换的方向如何，阶段1交换建立的cookie命令都将继续识别ISAKMP SA。换句话说，当ISAKMP SA的方向改变时，cookie绝对不能交换位置。

通过使用ISAKMP阶段，可以在必要时实现非常快速的键控。一个阶段1的协商可以用于多个阶段2的协商。另外，单个阶段2协商可以请求多个安全协会。通过这些优化，实现可以看到每个SA少于一次往返，每个SA少于一次DH幂。阶段1的“主模式”提供身份保护。当不需要身份保护时，可以使用“主动模式”来进一步减少往返。下面是开发人员做这些优化的提示。还应该注意的是，使用公钥加密对主动模式交换进行身份验证仍然可以提供身份保护。

该协议本身不定义自己的DOI。在阶段1中建立的ISAKMP SA MAY使用来自非ISAKMP服务的DOI和情况(例如IETF IPSec DOI [Pip97])。在这种情况下，实现MAY选择限制使用ISAKMP SA为相同DOI的服务建立SA。另外，ISAKMP SA MAY在DOI和情形中以零值建立(参见[MSST98]了解这些字段的描述)，在这种情况下，实现可以使用此ISAKMP SA自由地为任何已定义的DOI建立安全服务。如果DOI为零用于建立阶段1 SA，则阶段1中使用的身份有效载荷的语法是在[MSST98]中定义的，而不是来自任何DOI(例如[Pip97])，这可能会进一步扩展身份的语法和语义。

以下属性由IKE使用，并作为ISAKMP安全协会的一部分进行协商。(这些属性仅属于ISAKMP安全协会，而不属于ISAKMP可能代表其他服务进行谈判的任何安全协会。)

- 加密算法
- 哈希算法

- 认证方法
- Diffie-Hellman的组信息。

所有这些属性都是强制性的，MUST协商。此外，可以选择性地协商一个伪随机函数(“prf”)。(目前在本文档中没有定义可协商的伪随机函数。Private use属性值可用于同意方之间的prf协商)。如果“prf”不是协商的，协商散列算法的HMAC(参见[KBC96])版本被用作伪随机函数。其他非强制属性在附录a中描述。选择的哈希算法MUST同时支持本机和HMAC模式。

Diffie-Hellman组MUST使用已定义的组描述(章节6)或通过定义组的所有属性(章节5.6)来指定。组属性(如组类型或素数—参见附录A)绝对不能与先前定义的组(保留的组描述或在新组模式交换结束后建立的私有使用描述)一起提供。

IKE实现MUST支持以下属性值:

- CBC模式下的DES [DES]，具有弱和半弱密钥检查(弱和半弱密钥在[Sch96]中有引用，列在附录a中)，密钥推导参见附录B。

- MD5 [MD5]和SHA [SHA]。
- 预共享密钥认证。

- MODP超过默认组1(见下文)。

另外，IKE实现应该支持:3DES加密;Tiger ([Tiger])用于哈希;数字签名标准，RSA [RSA]签名和认证用RSA公钥加密;MODP组2。IKE实现MAY支持附录A中定义的任何其他加密算法，MAY支持ECP和EC2N组。

此处描述的IKE模式MUST在实现IETF IPsec DOI [Pip97]时实现。其他doi MAY使用这里描述的模式。

5. 交流

建立经过身份验证的密钥交换有两种基本方法:主模式和野蛮模式。每个都从短暂的Diffie-Hellman交换中生成经过身份验证的密钥材料。主模式MUST实现;应该实现野蛮模式。此外, 快速模式MUST作为生成新密钥材料和协商非isakmp安全服务的机制来实现。此外, 新组模式应该作为一种机制来实现, 为Diffie-Hellman交换定义私有组。实现MUSTNOT在交换中切换交换类型。

交换符合标准的ISAKMP有效负载语法、属性编码、消息的超时和重传以及信息性消息——例如, 当提议不可接受或签名验证或解密失败时发送通知响应, 等等。

在阶段1交换中, SA有效载荷MUST在所有其他有效载荷之前。除非另有说明, 任何消息中的ISAKMP有效载荷都不要求按任何特定顺序排列。

在阶段1或阶段2交换中, 在KE有效负载中传递的Diffie-Hellman公共值MUST是协商后的Diffie-Hellman组的长度, 如果有必要, 可以在值前加上零。

nonce有效载荷的长度MUST在8到256字节之间。

主模式是ISAKMP身份保护交换的一个实例:前两个消息协商策略;接下来的两个交换Diffie-Hellman公共值和交换所需的辅助数据(如随机数);最后两条消息验证了Diffie-Hellman交换。作为初始ISAKMP交换的一部分协商的身份验证方法会影响有效载荷的组成, 但不会影响其目的。主模式的XCHG是ISAKMP身份保护。

类似地, 野蛮模式是ISAKMP野蛮交换的一个实例。前两条消息协商策略, 交换交换所需的Diffie-Hellman公共值和辅助数据, 以及身份。另外, 第二条消息对应答者进行身份验证。第三条消息对发起者进行认证, 并提供参与交换的证明。野蛮模式的XCHG是ISAKMP野蛮模式。在ISAKMP SA的保护下, 最终消息MAYNOT发送

如果需要，推迟幂运算，直到完成此交换的协商。侵略性模式的图形描述清晰地显示了最终有效载荷，则不必如此。

IKE中的交换不是开放式的，并且有固定数量的消息。接收证书请求有效载荷绝对不能扩展传输或预期的消息数量。

安全关联协商在攻击模式下受到限制。由于消息构造要求，执行Diffie-Hellman交换的组无法进行协商。此外，不同的认证方法可能会进一步约束属性协商。例如，使用公钥加密的身份验证是无法协商的，当使用修改后的公钥加密方法进行身份验证时，密码和哈希值也无法协商。对于需要IKE的丰富属性协商能力的情况，可能需要使用主模式。

快速模式和新组模式在ISAKMP中没有类比。快速模式和新组模式的XCHG值在附录A中定义。

主模式、野蛮模式和快速模式进行安全关联协商。安全关联提供采用封装在提案有效负载中的转换有效负载的形式，封装在安全关联(SA)有效负载中。如果为阶段1交换(主模式和主动模式)提供多个报价，则它们MUST采取多个转换有效载荷的形式，用于单个SA有效载荷中的单个提案有效载荷。换句话说，对于第一阶段的交换，对于单个SA有效载荷，绝对不能有多个提案有效载荷，并且绝对不能有多个SA有效载荷。本文档不禁止在第二阶段交换的要约上有这种行为。

发起者可以发送给响应者的报价数量没有限制，但一致性实现可能会选择限制它将出于性能原因检查的报价数量。

在安全关联协商期间，发起者向响应者提供潜在安全关联的报价。响应者绝对不能修改任何要约的属性，属性编码除外(参见附录A)。如果交换的发起者注意到属性值已经改变或属性已经从要约中添加或删除，则该响应MUST被拒绝。

主模式或主动模式允许使用四种不同的身份验证方法——数字签名、使用公钥加密的两种身份验证形式或预共享密钥。SKEYID值是为每种身份验证方法单独计算的。

```

For signatures:           SKEYID = prf(Ni_b | Nr_b, g^xy)
For public key encryption: SKEYID = prf(hash(Ni_b | Nr_b), CKY-I |
CKY-R)
For pre-shared keys:      SKEYID = prf(pre-shared-key, Ni_b |
Nr_b)

```

The result of either Main Mode or Aggressive Mode is three groups of authenticated keying material:

```

SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
SKEYID_e = prf(SKEYID, SKEYID_a | a^xv | CKY-I | CKY-R | 2)

```

并就保护进一步通信的策略达成一致。上面的0、1和2的值由一个八位字节表示。用于加密的密钥以特定于算法的方式从SKEYID_e派生(参见附录B)。

为了验证任何一个交换，协议的发起者生成HASH_I，响应者生成HASH_R，其中：

```

HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b) HASH_R = prf
(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b)

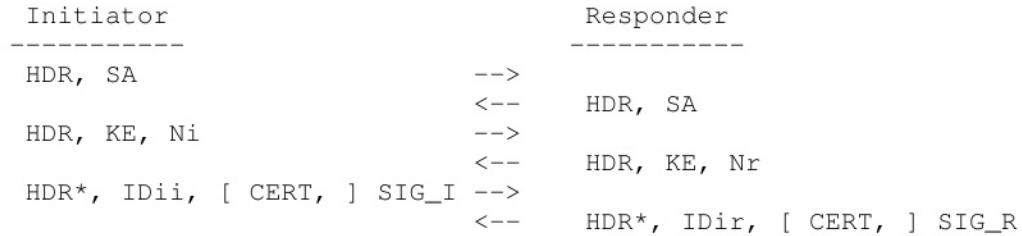
```

对于数字签名的身份验证，HASH_I和HASH_R被签名和验证;对于使用公钥加密或预共享密钥进行身份验证，HASH_I和HASH_R直接对交换进行身份验证。整个ID有效负载(包括ID类型、端口和协议，但不包括通用报头)被散列到HASH_I和HASH_R中。

如上所述，协商的身份验证方法影响阶段1模式的消息内容和使用，但不影响其意图。当使用公钥进行身份验证时，阶段1交换可以通过使用签名或使用公钥加密(如果算法支持的话)来完成。以下是具有不同身份验证选项的阶段1交换。

5.1 通过签名验证的IKE阶段1

使用签名，第二次往返期间交换的辅助信息为随机数;交换通过签署一个相互可获得的哈希值来进行身份验证。采用签名认证的主模式说明如下：



与ISAKMP配合使用签名的野蛮模式描述如下:



在这两种模式中，签名数据SIG_I或SIG_R是分别应用于HASH_I或HASH_R的协商数字签名算法的结果。

一般来说，签名将通过HASH_I和HASH_R进行，如上所述，使用协商好的prf，或者协商好的哈希函数的HMAC版本(如果没有协商好的prf)。但是，如果签名算法与特定的散列算法相关联(例如，DSS仅使用SHA的160位输出定义)，则可以在构建签名时覆盖这一点。在这种情况下，签名将如上所述在HASH_I和HASH_R上进行，除了使用与签名方法相关联的散列算法的HMAC版本。协商后的prf和哈希函数将继续用于所有其他规定的伪随机函数。

由于所使用的哈希算法是已知的，因此没有必要将其OID编码到签名中。此外，PKCS #1中用于RSA签名的oid与本文档中使用的oid之间没有绑定。因此，RSA签名MUST被编码为PKCS #1格式的私钥加密，而不是PKCS #1格式的签名(其中包括哈希算法的OID)。DSS签名MUST编码为r后跟s。

可以选择性地传递一个或多个证书有效载荷。

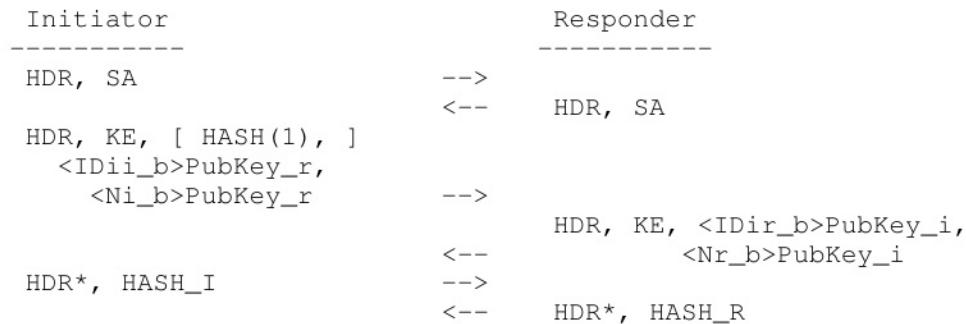
5.2 阶段1通过公钥加密验证

使用公钥加密对交换进行认证，交换的辅助信息是加密的随机数。每一方重建哈希的能力(证明对方解密了nonce)对交换进行了身份验证。

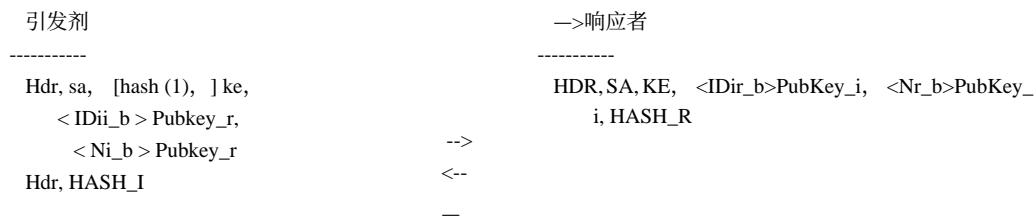
为了执行公钥加密，发起方必须已经拥有响应者的公钥。在响应者有多个公钥的情况下，发起者用来加密辅助信息的证书的哈希值作为第三条消息的一部分被传递。通过这种方式，响应者可以确定使用哪个对应的私钥来解密加密的有效载荷，并保留身份保护。

除了nonce之外，双方的身份(IDii和IDir)也用对方的公钥进行加密。如果身份验证方法是公钥加密，则nonce和身份有效载荷MUST使用另一方的公钥加密。只有有效载荷的主体被加密，有效载荷的报头被保留。

当使用加密进行认证时，Main Mode的定义如下。



采用加密方式认证的野蛮模式描述如下：



其中HASH(1)是发起方用来加密nonce和身份的证书的哈希值(使用协商哈希函数)。

RSA加密MUST以PKCS #1格式编码。虽然只有ID的主体和nonce有效载荷被加密，但加密的数据必须在有效的ISAKMP通用标头之前。有效载荷长度是整个加密有效载荷加上报头的长度。PKCS #1编码允许在解密时确定明文有效载荷的实际长度。

使用加密进行身份验证提供了一种合理的可否认交换。由于每一方都可以完全重建交换的双方，因此没有证据(与数字签名一样)证明对话曾经发生过。此外，由于攻击者不仅必须成功地破坏迪菲-赫尔曼交换，而且必须成功地破坏RSA加密，因此安全性被添加到秘密生成中。这个交换是由[SKEME]驱动的。

请注意，与其他认证方法不同，使用公钥加密的认证允许使用Aggressive Mode进行身份保护。

5.3 使用修改后的公钥加密方式验证阶段1

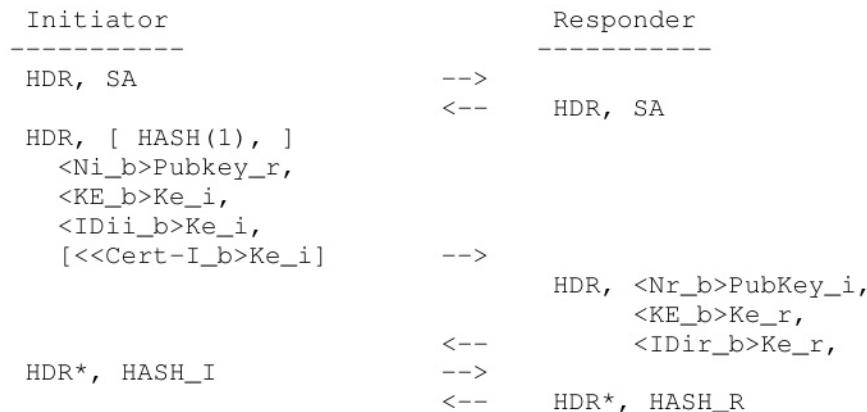
与使用签名的身份验证相比，使用公钥加密的身份验证具有显著的优势(参见上面的5.2节)。不幸的是，这是以4次公钥操作为代价的——两次公钥加密和两次私钥解密。这种身份验证模式保留了使用公钥加密进行身份验证的优点，但只需要进行一半的公钥操作。

在这种模式下，nonce仍然使用对等体的公钥进行加密，但是对等体的身份(以及发送的证书)使用协商的对称加密算法(来自SA有效负载)使用从nonce派生的密钥进行加密。这种解决方案增加了最小的复杂性和状态，但在每一方都节省了两次昂贵的公钥操作。此外，密钥交换有效载荷也使用相同的派生密钥进行加密。这为迪菲-赫尔曼交换的密码分析提供了额外的保护。

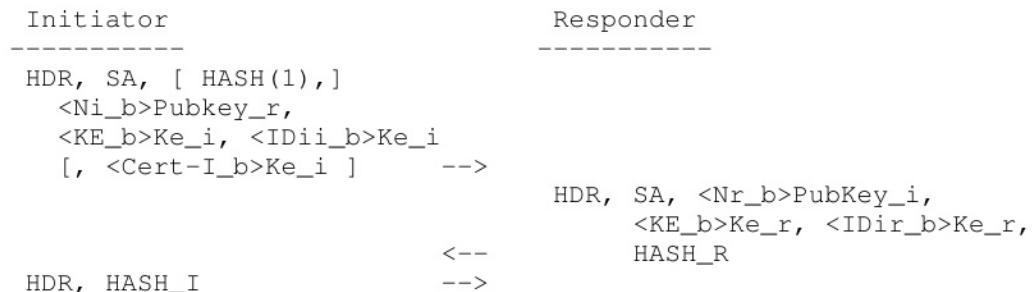
与认证的公钥加密方法(第5.2节)一样，如果响应者有多个包含可用公钥的证书(例如，由于证书限制或算法限制，证书不仅仅用于签名)，则可以发送HASH有效载荷来识别证书。如果HASH

负载MUST是第二次消息交换的第一个负载，并且MUST后跟加密的nonce。如果没有发送HASH有效载荷，则第二个消息交换的第一个有效载荷MUST是加密后的nonce。另外，发起者可以选择性地发送一个证书有效载荷，为响应者提供一个用于响应的公钥。

当使用修订后的加密模式进行身份验证时，Main mode的定义如下。



使用修改后的加密方法认证的野蛮模式描述如下：



其中HASH(1)与5.2节相同。Ke_i和Ke_r是SA有效负载交换中协商的对称加密算法的密钥。只有有效载荷的主体被加密(在公钥和对称操作中都是如此)，通用有效载荷头被保留在清晰的位置。有效载荷长度包括为执行加密而添加的长度。

对称密码密钥由解密后的随机数导出，如下所示。首先计算Ne_i和Ne_r的值：

```

Ne_i = prf(Ni_b, CKY-I)
Ne_r = prf(Nr_b, CKY-R)

```

然后，密钥Ke_i和Ke_r分别从Ne_i和Ne_r中提取，其方法在附录B中描述，用于导出用于协商加密算法的对称密钥。如果协商后的prf输出的长度大于或等于密码的密钥长度要求，则Ke_i和Ke_r分别从Ne_i和Ne_r的最有效位导出。如果Ke_i和Ke_r的期望长度超过prf输出的长度，则通过将prf的结果反复输入自身并将结果连接起来直到达到所需的位数来获得所需的位数。例如，如果协商加密算法需要320位密钥，而prf的输出只有128位，则Ke_i是K中最重要的320位，其中

```

K = K1 | K2 | K3 and
K1 = prf(Ne_i, 0)
K2 = prf(Ne_i, K1)
K3 = prf(Ne_i, K2)

```

为简洁起见，只显示Ke_i的推导；Ke_r是相同的。K1计算中值0的长度为单个八位字节。请注意，Ne_i、Ne_r、Ke_i和Ke_r都是短暂的，使用后MUST丢弃。

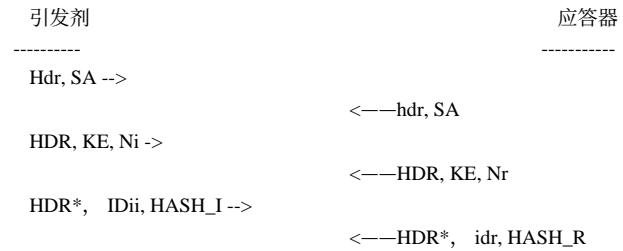
保存可选的HASH有效负载和强制的nonce有效负载的位置上的需求——没有其他有效负载需求。所有载荷——无论以何种顺序——跟随加密的nonce MUST使用Ke_i或Ke_r进行加密，具体取决于方向。

如果使用CBC模式进行对称加密，则初始化向量(IVs)设置如下。用于加密nonce之后的第一个有效载荷的IV设置为0(零)。使用临时对称密码密钥Ke_i加密的后续有效载荷的IV是前一个有效载荷的最后一个密文块。加密后的有效载荷被填充到最接近的块大小。除最后一个字节外，所有填充字节都包含0x00。填充的最后一个字节包含使用的填充字节数，不包括最后一个字节。请注意，这意味着总会有填充。

5.4 阶段1使用预共享密钥进行身份验证

由某些带外机制派生的密钥也可以用于验证交换。这个密钥的实际建立超出了本文的讨论范围。

在做预共享密钥认证时，Main Mode的定义如下：



带有预共享密钥的野蛮模式描述如下：



在主模式下使用预共享密钥身份验证时，密钥只能由对等体的IP地址标识，因为必须在发起者处理idr之前计算HASH_I。野蛮模式允许使用更广泛的预共享密钥标识符。此外，野蛮模式允许双方维护多个不同的预共享密钥，并为特定的交换识别正确的密钥。

5.5 阶段2 -快速模式

快速模式本身并不是一个完整的交换(因为它绑定到阶段1交换)，但它被用作SA协商过程(阶段2)的一部分，用于派生密钥材料并为非isakmp SA协商共享策略。与快速模式一起交换的信息MUST受到ISAKMP SA的保护——即除了ISAKMP报头之外的所有有效载荷都是加密的。在快速模式下，HASH有效载荷MUST紧接ISAKMP报头，SA有效载荷MUST紧接HASH。这个HASH对消息进行身份验证，并提供动态证明。

ISAKMP报头中的消息ID标识特定ISAKMP SA正在进行的快速模式，该快速模式本身由ISAKMP报头中的cookie标识。由于快速模式的每个实例都使用唯一的初始化向量(参见附录B)，因此可以在任何时候基于单个ISAKMP SA同时进行多个快速模式。

快速模式本质上是SA协商和提供重放保护的随机数交换。这些随机数用于生成新的密钥材料，并防止重放攻击生成虚假的安全关联。可以交换可选的密钥交换有效负载，以允许每个快速模式进行额外的Diffie-Hellman交换和幂运算。虽然使用快速模式的密钥交换负载是可选的，但它MUST得到支持。

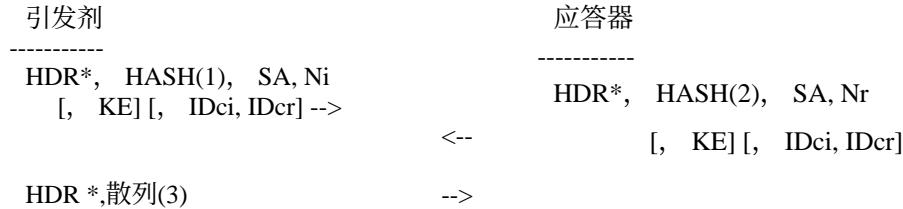
基本快速模式(没有KE有效负载)刷新从阶段1的幂运算派生的键控材料。这并不提供PFS。使用可选的KE有效负载，执行额外的幂运算，并为键控材料提供PFS。

除非在快速模式中指定了客户端标识符，否则在快速模式中协商的安全联盟的身份被隐式地假定为ISAKMP对等体的IP地址，对允许的协议或端口号没有任何隐含的约束。如果ISAKMP代表另一方作为客户谈判代表，则双方的身份MUST先传递为IDci，然后再传递IDcr。本地政策将决定提案是否可以接受指定的身份。如果快速模式响应器不能接受客户端身份(由于策略或其他原因)，则应该发送通知消息类型为INVALID-ID-INFORMATION(18)的Notify有效负载。

在两个对等点之间存在多个隧道的情况下，客户端标识用于标识流量并将其引导到适当的隧道，还用于允许具有不同粒度的唯一和共享安全联盟。

在快速模式期间提出的所有报价在逻辑上是相关的，必须是一致的。例如，如果发送KE有效负载，描述Diffie-Hellman组的属性(参见6.1节和[Pip97])MUST包含在正在协商的每个SA的每个提议的每个转换中。类似地，如果使用客户端标识，它们MUST应用于协商中的每个SA。

快速模式定义如下：



地点:

HASH(1)是来自ISAKMP报头的消息id (M-ID)的prf, 该prf与哈希之后的整个消息相连接, 包括所有有效负载报头, 但不包括为加密添加的任何填充。HASH(2)与HASH(1)相同, 只是发起者的nonce——Ni减去有效载荷头——被添加在M-ID之后, 但在完整的消息之前。将随机数添加到HASH(2)中是为了证明活跃性。HASH(3)——为了活动性——是表示为单个八位字节的值0上的prf, 后面跟着消息id和两个随机数的连接——发起者的随机数后面跟着响者的随机数——减去有效负载头。换句话说, 上述交换的哈希值是:

$$\begin{aligned} \text{HASH}(1) &= \text{prf}(\text{SKEYID_a}, \text{M-ID} \mid \text{SA} \mid \text{Ni} \mid \text{KE} \mid \text{IDci} \mid \text{IDcr}) \\ \text{HASH}(2) &= \text{prf}(\text{SKEYID_a}, \text{M-ID} \mid \text{KE} \mid \text{SA} \mid \text{Nr} \mid \text{IDci} \mid \text{IDcr}) \\ \text{HASH}(3) &= \text{prf}(\text{SKEYID_a}, 0 \mid \text{M-ID} \mid \text{Ni_b} \mid \text{Nr_b}) \end{aligned}$$

除了HASH、SA和可选的ID有效负载之外, 快速模式上没有有效负载排序限制。如果消息中的有效负载顺序与示例中的不同, 或者任何可选的有效负载(例如通知有效负载)已链接到消息, 则HASH(1)和HASH(2)可能与上面的示例不同。

如果不需要PFS, 并且不交换KE有效载荷, 则新的键控材料定义为

$$\text{KEYMAT} = \text{prf}(\text{SKEYID_d}, \text{协议} \mid \text{SPI} \mid \text{Ni_b} \mid \text{Nr_b})。$$

如果需要PFS并且交换KE有效载荷, 则新的键控材料定义为

$$\text{KEYMAT} = \text{prf}(\text{SKEYID_d}, g(\text{qm})^{\text{xy}} \mid \text{protocol} \mid \text{SPI} \mid \text{Ni_b} \mid \text{Nr_b})$$

其中 $g(\text{qm})^{\text{xy}}$ 是快速模式中短暂的Diffie-Hellman交换的共享秘密。

在任何一种情况下, “协议” 和 “SPI” 都来自包含协商转换的ISAKMP提案有效负载。

单个SA协商产生两个安全关联——一个入站和一个出站。每个SA的不同spi(一个由发起方选择，另一个由响应者选择)保证每个方向使用不同的密钥。SA的目的地选择的SPI用于为该SA派生KEYMAT。

对于所需键控材料的数量大于prf提供的数量的情况，KEYMAT通过将prf的结果反馈给自身并连接结果来扩展，直到达到所需的键控材料。换句话说，

```

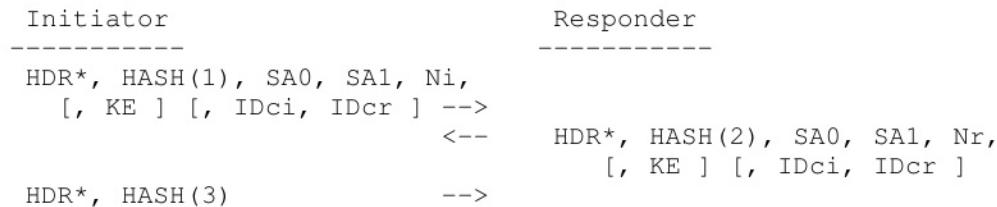
KEYMAT = K1 | K2 | K3 | ...
where
  K1 = prf(SKEYID_d, [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
  K2 = prf(SKEYID_d, K1 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
  K3 = prf(SKEYID_d, K2 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
etc.

```

该密钥材料(无论是否带有PFS，也无论直接派生还是通过连接派生)必须与协商的SA一起使用。由服务来定义如何从键控材料中派生密钥。

在快速模式下短暂的Diffie-Hellman交换的情况下，指数($g(qm)^{xy}$)从当前状态不可恢复地删除，SKEYID_e和SKEYID_a(来自阶段1的协商)继续保护和认证ISAKMP SA，SKEYID_d继续用于派生密钥。

使用快速模式，可以在一个交换器中协商多个SA和密钥，如下所示：



键控材料的推导与单个SA的情况相同。在这种情况下(两个SA有效负载的协商)，结果将是四个安全关联——两个SA各两个。

5.6 新组模式

在建立ISAKMP SA之前， MUST使用新组模式。新组的描述MUST只遵循阶段1的协商。(虽然这不是第2阶段的交换)。



其中， HASH(1)为prf输出， 使用SKEYID_a作为密钥， ISAKMP报头中的message-ID与整个SA提议， 正文和报头连接为数据;HASH(2)是prf的输出， 使用SKEYID_a作为键， ISAKMP报头的消息id与应答连接作为数据。换句话说， 上述交换的哈希值为：

$$\begin{aligned} \text{HASH}(1) &= \text{prf}(\text{SKEYID_a}, \text{ M-ID} \mid \text{SA}) \\ \text{HASH}(2) &= \text{prf}(\text{SKEYID_a}, \text{ M-ID} \mid \text{SA}) \end{aligned}$$

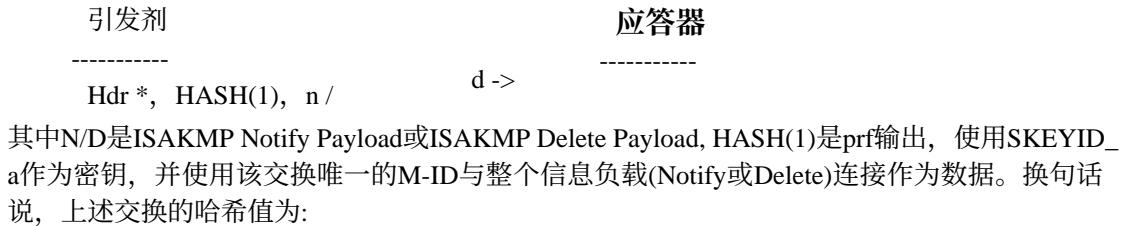
提案将指定组的特征(参见附录A，“属性分配号码”)。私有组的组描述MUST大于或等于 2^{15} 。如果组不可接受，则应答器MUST使用消息类型设置为ATTRIBUTES-NOT-SUPPORTED(13)的Notify有效负载进行应答。

ISAKMP实现可能要求私有组与建立它们的SA一起过期。

组可以直接在SA提案中与主模式进行协商。要做到这一点，组成部分——对于MODP组，类型，素数和发电机;对于EC2N群，其类型、不可约多项式、群生成器1、群生成器2、群曲线a、群曲线B和群顺序——作为SA属性传递(参见附录a)。或者，可以使用新组模式隐藏群的性质，在第一阶段协商时只传递组标识符。

5.7 ISAKMP信息交换

该协议尽可能保护ISAKMP信息交换。一旦建立了ISAKMP安全关联(并且生成了SKEYID_e和SKEYID_a)，ISAKMP信息交换与此协议一起使用时，如下所示：



$$\text{HASH}(1) = \text{prf}(\text{SKEYID}_a, \text{M-ID} \mid \text{N/D})$$

如前所述, ISAKMP报头中的消息ID(在prf计算中使用)是此次交换唯一的, 并且MUST与生成此信息交换的另一个阶段2交换的消息ID不相同。在附录B中描述了与SKEYID_e一起用于加密此消息的初始化向量的推导。

如果在信息交换时还没有建立ISAKMP安全联盟, 则交换是在没有附带HASH负载的情况下以明文方式进行的。

6 奥克利集团

有了IKE, 做迪菲-赫尔曼交换的小组是协商好的。下面定义了四个组——值1到4。这些团体起源于奥克利协议, 因此被称为“奥克利集团”。“Group”的属性类定义在附录a中。所有 2^{15} 及以上的值都用于私有组标识符。有关默认奥克利组强度的讨论, 请参阅下面的安全性注意事项部分。

这些组均由亚利桑那大学的Richard Schroepel生成。这些群的性质在[Orm96]中有描述。

6.1 第一奥克利默认组

奥克利实现MUST支持具有以下素数和生成器的MODP组。这个组被分配id为1(1)。

素数为: $2^{768} - 2^{704} - 1 + 2^{64} * \{[2^{638} \pi] + 149686\}$ 其十六进制值为

生成器为:2。

6.2第二奥克利集团

IKE实现应该支持具有以下素数和生成器的MODP组。这个组被分配id为2(2)。

素数为 $2^{1024} - 2^{960} - 1 + 2^{64} * \{[2^{894} \pi] + 129093\}$ 。其十六进制值为

FFFFFF				6c4c6628b	80 dc1cd1
29024 e08	8 a67cc74	020年bbea6	3 b139b22	514年a0879	8 e3404dd
EF9519B3	CD3A431B	302年b0a6d	F25F1437	4 fe1356d	6 d51c245
E485B576	625年e7ec6	F44C42E9	A637ED6B	0 bff5cb6	F406B7ED
EE386BF8	5 a899fa5	2168C234	7 c4b1fe	49286651	ECE65381
FFFFFF	C90FDAA2				

生成器为2(十进制)

63第三奥克利集团

IKE实现应该支持以下的EC2N组特征。这个组被分配id 3(3)。曲线为基于伽罗瓦场 $GF[2^{155}]$ 。场大小为155。的域的不可约多项式为:

$$Y^2 + XY = X^3 + aX^2 + b$$

字段大小: 155
群素数/不可约多项式:
0 x0800000000000000000000000000000040000000000000000
组发生器一:0x7b
群曲线A: 0 x0
组曲线B: 0 x07338f

群组:0x080000000000000000000000057db5698537193aef944

使用该组时，KE有效载荷中的数据为解(x,y)中的值x，即取随机选择的秘密Ka并计算Ka*P所选择的曲线上的点，其中*为组加法和双操作的重复。P为y坐标等于生成器1和y的曲线点。

由定义方程确定的坐标。曲线方程由群类型和A、B系数隐式已知。y坐标有两个可能的值;任何一个都可以成功使用(双方不需要就选择达成一致)。

6.4 第四奥克利集团

IKE实现应该支持具有以下特征的EC2N组。这个组被分配id为4(4)。曲线基于伽罗瓦场 $GF[2^{185}]$ 。场大小为185。的域的不可约多项式为:

$U^{185} + U^{69} + 1$ 。的

椭圆曲线的方程为:

$$Y^2 + XY \equiv X^3 + aX^2 + b_0$$

字段大小:

185

群素数/不可约多项式·

组发生器1:0x18

0 x0

群曲线A:
组曲线B:

8 NOV

群序:0X01fffffffffffff...dbf2f889b73e484175f94ebc

使用此组时，KE有效载荷中的数据将与使用Oakley组3(三)时相同。

其他组可以使用New Group Mode来定义。这些默认组由亚利桑那大学的Richard Schroeppel生成。这些素数的性质在[Orn96]中有描述。

7. 有效载荷爆炸为一个完整的IKE交换

本节说明如何使用IKE协议。

- 在ISAKMP过程之间建立安全且经过认证的通道(阶段1);和

为IPsec SA生成密钥材料并进行协商(阶段2)。

7.1 阶段1使用主模式

下图展示了双方在第一次往返交换中交换的有效载荷。发起方可能会提出多个提案，回应者必须回复一个。

```

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
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~           ISAKMP Header with XCHG of Main Mode, ~
~           and Next Payload of ISA_SA ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!      0      ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!           Domain of Interpretation !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!           Situation !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!      0      ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!  Proposal #1 ! PROTO_ISAKMP ! SPI size = 0 | # Transforms !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!  ISA_TRANS ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!  Transform #1 ! KEY_OAKLEY | RESERVED2 !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~           preferred SA attributes ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!      0      ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!  Transform #2 ! KEY_OAKLEY | RESERVED2 !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~           alternate SA attributes ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

响应者以实物形式回复，但选择并返回一个变换

提议(ISAKMP SA属性)。

第二次交换包括以下有效载荷:

```

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
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~           ISAKMP Header with XCHG of Main Mode, ~
~           and Next Payload of ISA_KE ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!  ISA_NONCE ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~  D-H Public Value (g^xi from initiator g^xr from responder) ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
!      0      ! RESERVED ! Payload Length !
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~           Ni (from initiator) or Nr (from responder) ~
+-+-+-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

共享密钥SKEYID_e和SKEYID_a现在用于保护和验证所有进一步的通信。注意，SKEYID_e和SKEYID_a都是未经身份验证的。

```

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
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~           ISAKMP Header with XCHG of Main Mode,
~           and Next Payload of ISA_ID and the encryption bit set
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!   ISA_SIG    !   RESERVED    !   Payload Length    !
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~           Identification Data of the ISAKMP negotiator
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!       0        !   RESERVED    !   Payload Length    !
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~           signature verified by the public key of the ID above
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

密钥交换是通过5.1节中描述的签名哈希进行身份验证的。一旦使用作为ISAKMP SA一部分协商的身份验证算法验证了签名，共享密钥SKEYID_e和SKEYID_a就可以被标记为已验证。(为简洁起见，没有交换证书有效载荷)。

7.2 第二阶段使用快速模式

在ISAKMP SA协商的第一轮快速模式中交换以下有效载荷。在这个假想的交换中，ISAKMP谈判者是请求身份验证的其他方的代理。

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
☒           ISAKMP头与快速模式的XCHG,           ☒
☒           ISA_HASH的有效载荷和加密位设置
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!   ISA_SA    !   保留      !   有效载荷长度    !
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
☒           消息的键控散列
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!   ISA_NONCE  !   保留      !   有效载荷长度    !
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!           解读领域
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!           情况
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+
!       0        !   保留      !   有效载荷长度    !
+-+-+-+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

```

! Proposal #1 ! PROTO_IPSEC_AH! SPI size = 4 | # Transforms !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~          SPI (4 octets) ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! ISA_TRANS ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! Transform #1 ! AH_SHA | RESERVED2 !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! other SA attributes !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! 0 ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! Transform #2 ! AH_MD5 | RESERVED2 !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! other SA attributes !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! ISA_ID ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~ nonce ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! ISA_ID ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~ ID of source for which ISAKMP is a client ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! 0 ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~ ID of destination for which ISAKMP is a client ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

where the contents of the hash are described in 5.5 above. The responder replies with a similar message which only contains one transform-- the selected AH transform. Upon receipt, the initiator can provide the key engine with the negotiated security association and the keying material. As a check against replay attacks, the responder waits until receipt of the next message.

```

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
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~          ISAKMP Header with XCHG of Quick Mode, ~
~ Next Payload of ISA_HASH and the encryption bit set ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
! 0 ! RESERVED ! Payload Length !
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
~ hash data ~
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

where the contents of the hash are described in 5.5 above.

8. 完美前向保密示例

该协议可以同时提供密钥和身份的PFS。PFS既可以保护ISAKMP协商对等体的身份，也可以保护协商对等体的身份(如果适用)。

为了提供密钥和所有身份的完美前向保密，双方将执行以下操作：

- o 主模式交换，用于保护ISAKMP对等体的身份。

这将建立一个ISAKMP SA。

- o 快速模式交换，协商其他安全协议保护。

这将在协议的两端建立一个SA。

- o 删除ISAKMP SA及其关联状态。

由于在非isakmp SA中使用的密钥是从单个短暂的迪菲-赫尔曼交换中获得的，因此保留了PFS。

如果在两个对等体之间存在ISAKMP SA，则仅为非isakmp安全关联的密钥提供完美前向保密，不需要进行阶段1交换。只需一个快速模式，其中传递可选的KE有效负载，并执行额外的迪菲-赫尔曼交换。此时，必须从ISAKMP SA中删除从该快速模式派生的状态，如5.5节所述。

9. 实现提示

使用一个ISAKMP阶段1的谈判使得随后的阶段2的谈判非常迅速。只要阶段1状态保持缓存，并且不需要PFS，阶段2就可以在不进行任何幂运算的情况下进行。单个Phase 1可以执行多次Phase 2协商是本地策略问题。该决策将取决于所使用算法的强度和对等系统中的信任水平。

在执行快速模式时，实现可能希望协商一系列sa。通过这样做，他们可以加快“重新键控”。快速模式定义了如何为一系列sa定义KEYMAT。当一个对等方觉得是时候更换sa时，他们只是在规定的范围内使用下一个sa。通过使用一个快速模式协商多个sa(属性相同，spi不同)，可以建立一系列sa。

通常有用的一种优化是在需要对等体之前与它们建立安全关联，以便在需要它们时它们已经到位。这确保了在初始数据传输之前不会因为密钥管理而出现延迟。通过为每个请求的安全关联设置多个安全关联，并缓存那些不立即使用的安全关联，可以很容易地实现此优化。

另外，如果ISAKMP实现被提醒即将需要一个SA(例如，替换即将过期的现有SA)，那么它可以在需要新SA之前建立新的SA。

基本ISAKMP规范描述了协议一方可以将某些活动通知另一方的条件——删除安全关联或响应协议中的某些错误，如签名验证失败或有效载荷解密失败。强烈建议在任何情况下都不要对这些信息交换做出响应。这种情况可能会导致“通知战争”，即由于无法理解消息，导致向无法理解的对等体发出通知，并将自己的通知发回，这也是不被理解的。

10. 安全注意事项

整个备忘录讨论了一种混合协议，将Oakley的部分内容和SKEME的部分内容与ISAKMP结合起来，以一种安全和经过认证的方式为安全关联进行协商并获得密钥材料。

通过使用协商的加密算法来保证机密性。认证通过使用协商的方法来保证：数字签名算法；支持加密的公钥算法；或者，预共享密钥。此交换的机密性和身份验证仅与作为ISAKMP安全协会的一部分协商的属性一样好。

使用快速模式进行多次重密会消耗Diffie-Hellman共享密钥的熵。实现者应该注意到这一点，并对幂次之间的快速模式交换设置限制。本备忘录没有规定这样的限制。

该协议可以实现密钥材料和身份的完全前向保密(PFS)。通过指定Diffie-Hellman组，并在KE有效负载中传递公共值，ISAKMP对等体可以建立密钥的PFS——身份将由SKEYID_e保护，不受ISAKMP SA的保护，因此不受PFS的保护。如果需要密钥材料和身份的PFS，则必须有一个ISAKMP对等体

每个ISAKMP SA只能建立一个非ISAKMP安全联盟(如IPsec安全联盟)。密钥和身份的PFS是通过在建立单个非ISAKMP SA时删除ISAKMP SA(并可选择发出DELETE消息)来实现的。通过这种方式，第一阶段协商与第二阶段协商唯一地绑定在一起，并且在第一阶段协商期间建立的ISAKMP SA永远不会再次使用。

使用此处定义的任何组从Diffie-Hellman交换中派生的密钥的强度取决于组的固有强度、所使用的指数的大小以及所使用的随机数生成器提供的熵。由于这些输入，很难确定任何定义的组的密钥强度。默认的Diffie-Hellman组(编号1)在与强随机数生成器和不小于160位的指数一起使用时足以用于DES。组2到组4提供更高的安全性。在建立策略和协商安全参数时，实现应该注意这些保守估计。

注意，这些限制是针对Diffie-Hellman组本身的。在IKE中没有任何东西禁止使用更强的组，也没有任何东西会稀释从更强的组中获得的强度。实际上，IKE的可扩展框架鼓励定义更多的组;使用椭圆曲线群，使用小得多的数字，将大大增加强度。

对于已定义的组提供的强度不足的情况，可以使用新组模式来交换提供必要强度的Diffie-Hellman组。实现有责任检查所提供的组中的素数并独立地得出强度估计。

假定此交换中的Diffie-Hellman指数在使用后从内存中删除。特别是，这些指数一定不能像伪随机生成器的种子那样，从长期存在的秘密中推导出来。

IKE交换保持运行初始化向量(IV)，其中上一条消息的最后一个密文块是下一条消息的IV。为了防止重传(或带有有效cookie的伪造消息)导致交换不同步，IKE实现不应该更新其正在运行的IV，直到解密的消息通过了基本的完整性检查，并确定实际上推进了IKE状态机——即它不是重传。

虽然主模式的最后一次往返(以及侵略性模式的最后一条消息)是加密的，但严格来说，它并没有经过身份验证。对密文的主动替代攻击可能导致有效载荷损坏。如果这种攻击破坏了强制的有效载荷，它将被认证失败检测到，但如果它破坏了任何可选的有效载荷(例如，链接到主模式交换的最后一条消息上的通知有效载荷)，它可能无法被检测到。

11. IANA的考虑

这份文件包含了许多由IANA维护的“幻数”。本节解释了IANA在每个列表中分配额外编号时使用的标准。

11.1属性类

本协议中协商的属性由其类标识。分配新类的请求必须伴随着一个标准跟踪的RFC，其中描述了该属性的使用。

11.2加密算法类

加密算法类的值定义了在本文中调用时要使用的加密算法。分配新加密算法值的请求必须伴随着对标准跟踪或信息RFC的引用，或者对描述该算法的已发布的加密文献的引用。

11.3哈希算法

哈希算法类的值定义了在本文中调用时要使用的哈希算法。分配新哈希算法值的请求必须伴随着对标准跟踪或信息RFC的引用，或者对描述该算法的已发布的加密文献的引用。由于在IKE中使用HMAC形式的哈希算法的密钥派生和密钥扩展，分配新哈希算法值的请求必须考虑哈希算法本身的密码学属性——例如它的抗碰撞性。

11.4组描述和组类型

组描述类的值标识在Diffie-Hellman交换中使用的组。Group Type类的值定义了组的类型。分配新组的请求必须附带对描述该组的标准跟踪或信息RFC的引用。分配新组的请求

类型必须伴随着对标准跟踪或信息RFC的引用，或者对描述新类型的已发布的密码学或数学文献的引用。

11.5 寿命类型

寿命类型类的值定义了ISAKMP安全关联应用的生命周期类型。分配新生命类型的请求必须附有该类型的单元及其到期的详细描述。

12. 致谢。

本文档是与Hugo Krawczyk、Douglas Maughan、Hilarie Orman、Mark Schertler、Mark Schneider和Jeff Turner密切磋商的结果。它依赖于由他们编写的协议。如果没有他们的兴趣和奉献，这篇文章就不会写出来。

特别感谢Rob Adams, Cheryl Madson, Derrell Piper, Harry Varnis和Elfed Weaver在技术上的投入，鼓励，以及一路上的各种理智检查。

我们还要感谢IPSec工作小组的许多成员，他们在过去的一年里为这个协议的发展做出了贡献。

13. 参考文献。

- [CAST] Adams, C., "The CAST-128 Encryption Algorithm", RFC 2144, May 1997.
- [BLOW] Schneier, B., "The Blowfish Encryption Algorithm", Dr. Dobb's Journal, v. 19, n. 4, April 1994.
- [Bra97] Bradner, S., "Key Words for use in RFCs to indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [DES] ANSI X3.106, "American National Standard for Information Systems-Data Link Encryption", American National Standards Institute, 1983.
- [DH] Diffie, W., and Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, V. IT-22, n. 6, June 1977.

- [DSS] NIST, "Digital Signature Standard", FIPS 186, National Institute of Standards and Technology, U.S. Department of Commerce, May, 1994.
- [IDEA] Lai, X., "On the Design and Security of Block Ciphers," ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992
- [KBC96] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [SKEME] Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.
- [MD5] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, April 1992.
- [MSST98] Maughan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [Orm96] Orman, H., "The Oakley Key Determination Protocol", RFC 2412, November 1998.
- [PKCS1] RSA Laboratories, "PKCS #1: RSA Encryption Standard", November 1993.
- [Pip98] Piper, D., "The Internet IP Security Domain Of Interpretation for ISAKMP", RFC 2407, November 1998.
- [RC5] Rivest, R., "The RC5 Encryption Algorithm", Dr. Dobb's Journal, v. 20, n. 1, January 1995.
- [RSA] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.
- [Sch96] Schneier, B., "Applied Cryptography, Protocols, Algorithms, and Source Code in C", 2nd edition.
- [SHA] NIST, "Secure Hash Standard", FIPS 180-1, National Institute of Standards and Technology, U.S. Department of Commerce, May 1994.
- [TIGER] Anderson, R., and Biham, E., "Fast Software Encryption", Springer LNCS v. 1039, 1996.

附录A

这是DES弱和半弱密钥的列表。
[Sch96]。所有键都以十六进制列出。

这些密钥来自

DES弱密钥	
0101 0101 0101	101
1f1f 1f1f e0e0	E0E0
E0e0 E0e0 1f1f	1 f1f
Fefe fefe0	象皮病

DES半弱密钥	
01fe 01fe 01fe	
1fe0 1fe0 0ef1 0ef1	
01e0 01e0 01f1 01f1	
1ffe 1ffe 0efe 0efe	
011f 011f 010e 010e	
E0fe E0fe f1fe f1fe	

FE01			
E01F	E01F	F10E	F10E
E001	E001	F101	F101
FE1F	FE1F	FE0E	FE0E
1 f01	1 f01	0 e01	0 e01
FE01	FE01	FE01	FEF1

属性分配号码

在第一阶段协商的属性使用以下定义。阶段二属性在适用的DOI规范中定义(例如，IPsec属性在IPsec DOI中定义)，当快速模式包含短暂的Diffie-Hellman交换时，组描述除外。属性类型可以是Basic (B)或Variable- Length (V)，这些属性的编码在基本ISAKMP规范中定义为Type/Value (Basic)和Type/Length/Value (Variable)。

被描述为基本属性的属性绝对不能被编码为变量。可变长度属性MAY编码为基本属性，如果它们的值可以容纳两个八位字节。如果是这种情况，该协议的发起者作为变量(或基本)提供的属性MAY作为基本(或变量)返回给发起者。

属性类

class	value	type
Encryption Algorithm	1	B
Hash Algorithm	2	B
Authentication Method	3	B
Group Description	4	B
Group Type	5	B
Group Prime/Irreducible Polynomial	6	V
Group Generator One	7	V
Group Generator Two	8	V
Group Curve A	9	V
Group Curve B	10	V
Life Type	11	B
Life Duration	12	V
PRF	13	B
Key Length	14	B
Field Size	15	B
Group Order	16	V

17-16383为IANA保留。值16384-32767仅供双方同意的私人使用。

类值

-加密算法	中定义的
des - cbc	1 RFC 2405
IDEA-CBC	2
Blowfish-CBC	3
RC5-R16-B64-CBC	4
3 des - cbc	5
CAST-CBC	6

数值7-65000保留给IANA。65001-65535仅供双方同意的私人使用。

-哈希算法	中定义的
MD5	1 RFC 1321
沙	2个提示180-1
老虎	3参见参考文献[TIGER]

数值4 ~ 65000保留给IANA使用。65001-65535仅供双方同意的私人使用。

-认证方法

pre-shared关键	1
DSS签名	2
RSA签名	3
使用RSA加密	4
修订RSA加密	5

6 ~ 65000保留给IANA。65001 ~ 65535为for
相互同意的双方之间的私人使用。

-组描述

默认的768位MODP组(章节6.1)	1
备用1024位MODP组(第6.2节)	2
GP上的EC2N群[2^155](第6.3节)	3
GP上的EC2N群[2^185](第6.4节)	4

5 ~ 32767为IANA保留。32768-65535仅供双方同意的私人使用。

-组类型

MODP(模幂群)1 ECP (GF上的椭圆曲线群[P]) 2 EC2N (GF上的椭圆曲线群[2^N])

4 ~ 65000保留给IANA。65001-65535仅供双方同意的私人使用。

-生命类型秒-千字节

1
2

3 ~ 65000保留给IANA。65001-65535仅供双方同意的私人使用。对于给定的“生命类型”，“生命持续时间”属性的值定义了SA生命的实际长度——可以是秒数，也可以是受保护的kb数。

—脉冲重复频率

目前没有定义伪随机函数。

1 ~ 65000保留给IANA。65001-65535仅供双方同意的私人使用。

-密钥长度

当使用具有可变长度密钥的加密算法时，此属性以位为单位指定密钥长度。(MUST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，MUST使用此属性。

-字段大小

Diffie-Hellman群的字段大小，以比特为单位。

-组序

椭圆曲线群的群序。注意这个属性的长度取决于字段的大小。

额外的交换定义——XCHG值快速模式

32

新组模式

33

Appendix B

本附录描述了仅在加密ISAKMP消息时使用的加密细节。当服务(如IPSEC转换)利用ISAKMP生成密钥材料时，所有加密算法特定的细节(如密钥和IV生成，填充等)。MUST由该服务定义。ISAKMP并不声称生成适合任何加密算法的密钥。ISAKMP生成请求的密钥材料数量，服务MUST从中生成合适的密钥。诸如弱密钥检查之类的细节是服务的责任。

由于本文件采用的PRF反馈机制，使用协商PRF可能需要扩大PRF输出。例如，如果(虚构的)DOORAK-MAC需要24字节的密钥，但只产生8字节的输出，则输出必须扩展三次才能用作其另一个实例的密钥。PRF的输出通过将PRF的结果反馈到自身以生成连续的块来扩展。这些块被连接起来，直到达到所需的字节数。例如，对于以DOORAK-MAC作为协商PRF的预共享密钥认证：

```
BLOCK1-8 = prf(pre-shared-key, Ni_b | Nr_b)
BLOCK9-16 = prf(pre-share -key, BLOCK1-8 | Ni_b | Nr_b)
BLOCK17-24 = prf(pre-share -key, BLOCK9-16 | Ni_b | Nr_b) and
SKEYID = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
因此，导出SKEYID_d:
```

```
BLOCK1-8 = prf(SKEYID, g^xy | CKY-I | CKY-R | 0) BLOCK9-16 = prf(SKEYID,
BLOCK1-8 | g^xy | CKY-I | CKY-R | 0)
BLOCK17-24 = prf(SKEYID, BLOCK9-16 | g^xy | CKY-I | CKY-R | 0)
SKEYID_d = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

后续的PRF推导也是类似的。

用于保护ISAKMP SA的加密密钥以特定于算法的方式从SKEYID_e中导出。当SKEYID_e不够长，无法提供算法所需的所有必要的密钥材料时，密钥是通过将伪随机函数的结果输入到自身中，将结果连接起来，并取最高的必要位来获得的。

例如，如果(虚构的)算法AKULA需要320位的密钥(并且没有弱密钥检查)，而用于生成SKEYID_e的prf只生成120位的材料，则AKULA的密钥将是Ka的前320位，其中：

和 $Ka = K1 | K2 | K3$

$$\begin{aligned} K1 &= \text{prf} \\ (\text{SKEYID}_e, 0) &K2 = \text{prf} \\ (\text{SKEYID}_e, K1) &K3 = \text{prf} \\ (\text{SKEYID}_e, K2) \end{aligned}$$

其中prf是协商散列函数的HMAC版本的协商后的prf(如果没有协商prf)，0由单个八位字节表示。prf的每个结果提供了120位的材料，总共360位。AKULA将使用这360位字符串的前320位。

在阶段1中，用于CBC模式加密算法的初始化向量(IV材料)的材料是从使用协商散列算法的发起者的公共Diffie-Hellman值和响应者的公共Diffie-Hellman值串联的散列中派生出来的。这仅用于第一个消息。每条消息应该使用包含0x00的字节填充到最接近的块大小。报头中的消息长度MUST包括填充的长度，因为这反映了密文的大小。后续消息MUST使用前一个消息的第一个CBC加密块作为其初始化向量。

在阶段2中，用于快速模式交换的第一条消息的CBC模式加密的初始化向量的材料来自最后阶段1 CBC输出块和使用协商散列算法的阶段2消息id的串联的散列。快速模式交换中后续消息的IV是来自前一个消息的CBC输出块。后续消息的填充和IVs与阶段1一样完成。

ISAKMP SA通过认证后，所有信息交换都使用SKEYID_e加密。这些交换的初始化向量以与快速模式完全相同的方式派生——即，它派生自最后阶段1 CBC输出块和信息交换的ISAKMP报头的消息id的连接的散列(而不是来自可能提示信息交换的消息的消息id)。

注意，最后阶段1的CBC输出块，即最后阶段1消息的加密/解密结果，必须保留在ISAKMP SA状态中，以便为每个快速模式生成唯一的IVs。每个post- phase 1的交换(Quick Modes和

信息交换(information Exchanges)独立生成IVs，以防止两个不同的交换同时启动时IVs不同步。

在所有情况下，都有一个单一的双向密码/IV上下文。让每个快速模式和信息交换保持一个独特的上下文可以防止IVs不同步。

DES-CBC的密钥来源于SKEYID_e的前八(8)个非弱和非半弱字节(参见附录A)。IV是上面导出的IV材料的前8个字节。

IDEA-CBC的键是从SKEYID_e的前16个字节派生出来的。IV是上面导出的IV材料的前八(8)个字节。

Blowfish-CBC的密钥要么是协商的密钥大小，要么是前面提到的伪随机函数反馈方法中导出的密钥的前56(56)个字节(如果没有协商密钥大小)。IV是上面导出的IV材料的前八(8)个字节。

RC5-R16-B64-CBC的密钥是协商的密钥大小，或者密钥的前16(16)个字节(如果没有协商密钥大小)，必要时从上述伪随机函数反馈方法派生。IV是上面导出的IV材料的前八(8)个字节。轮数MUST为16，块大小MUST为64。

3DES-CBC的密钥是前面提到的伪随机函数反馈方法中导出的密钥的前24(24)个字节。3DES-CBC是一种加密-解密-加密操作，使用整个3DES-CBC密钥的前、中、后8个字节。IV是上面导出的IV材料的前八(8)个字节。

CAST-CBC的密钥要么是协商的密钥大小，要么是在上述伪随机函数反馈方法中派生的密钥的前十六(16)个字节。IV是上面导出的IV材料的前八(8)个字节。

对DES-CBC以外算法的支持完全是可选的。部分可选算法可能会受到知识产权索赔。

作者的地址

丹哈金斯
思科系统公司
170 W。塔斯曼博士。
加州圣何塞，95134-1706
美利坚合众国

电话:+1 408 526 4000
电子邮件:dharkins@cisco.com

戴夫·卡雷尔
76 Lippard Ave.
旧金山，加州94131-2947
美利坚合众国

电话:+1 415 337 8469
电子邮件:carrel@ipsec.org

作者的注意

作者鼓励这种混合协议的独立实现和互操作性测试。

完整版权声明

版权所有(C)互联网协会(1998)。版权所有。

本文档及其翻译可以复制并提供给他人，对其进行评论或以其他方式解释或协助其实施的衍生作品可以全部或部分地准备，复制，出版和分发，不受任何限制，只要上述版权声明和本段包含在所有此类副本和衍生作品中。但是，本文档本身不得以任何方式进行修改，例如删除版权声明或对互联网协会或其他互联网组织的引用，除非为制定互联网标准的目的而需要，在这种情况下，必须遵循互联网标准过程中定义的版权程序，或者需要将其翻译成英语以外的语言。

上述有限的许可是永久的，互联网协会或其继承者或受让人不得撤销。

本文件和本文件中包含的信息是按“现状”提供的，互联网协会和互联网工程任务组不承担所有明示或IMPLIED的WARRANTIES，包括但不限于ANY保证，即对本文件中THE的使用NOT侵犯ANY权利或ANY IMPLIED的适销性或适合某一特定目的的WARRANTIES。