5996

原文

RFC 5996: Internet Key Exchange Protocol Version 2 (IKEv2) 中文翻译

URL：https://datatracker.ietf.org/doc/html/rfc5996

标题 ：**RFC 5996**

翻译类型 ：自动生成

Internet Engineering Task Force (IETF)
Request for Comments: 5996
Obsoletes: 4306, 4718
Category: Standards Track
ISSN: 2070-1721

C. Kaufman
Microsoft
P. Hoffman
VPN Consortium
Y. Nir
Check Point
P. Eronen
Independent
September 2010

Internet Engineering Task Force (IETF)
Request for Comments: 5996
Obsoletes: 4306, 4718
Category: Standards Track
ISSN: 2070-1721
C. Kaufman
Microsoft
P. Hoffman
VPN Consortium
Y. Nir
Check Point
P. Eronen
Independent
September 2010

Internet Key Exchange Protocol Version 2 (IKEv2)

Internet 密钥交换协议版本 2（IKEv2）

Abstract

摘要

This document describes version 2 of the Internet Key Exchange (IKE) protocol. IKE is a component of IPsec used for performing mutual authentication and establishing and maintaining Security Associations (SAs). This document replaces and updates RFC 4306, and includes all of the clarifications from RFC 4718.

本文档描述了Internet密钥交换（IKE）协议的版本2。IKE是IPsec的一个组件，用于执行相互身份验证以及建立和维护安全关联（SA）。本文件取代并更新了RFC 4306，包括RFC 4718中的所有澄清。

Status of This Memo

关于下段备忘

This is an Internet Standards Track document.

这是一份互联网标准跟踪文件。

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

本文件是互联网工程任务组（IETF）的产品。它代表了IETF社区的共识。它已经接受了公众审查，并已被互联网工程指导小组（IESG）批准出版。有关互联网标准的更多信息，请参见RFC 5741第2节。

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc5996.

有关本文件当前状态、任何勘误表以及如何提供反馈的信息，请访问http://www.rfc-editor.org/info/rfc5996.

Copyright Notice

版权公告

Table of Contents

目录

1.1.3. Endpoint to Security Gateway in Tunnel Mode 8

1.1.4. Other Scenarios 9

1.2. The Initial Exchanges 9

1.3. The CREATE_CHILD_SA Exchange 13

1.3.1. Creating New Child SAs with the CREATE_CHILD_SA Exchange 14

1.3.2. Rekeying IKE SAs with the CREATE_CHILD_SA Exchange 15

1.3.3. Rekeying Child SAs with the CREATE_CHILD_SA Exchange 16

1.4. The INFORMATIONAL Exchange 17

1.4.1. Deleting an SA with INFORMATIONAL Exchanges 17

1.5. Informational Messages outside of an IKE SA 18

1.6. Requirements Terminology 19

1. Introduction

1. 介绍

IP Security (IPsec) provides confidentiality, data integrity, access control, and data source authentication to IP datagrams. These services are provided by maintaining shared state between the source and the sink of an IP datagram. This state defines, among other things, the specific services provided to the datagram, which cryptographic algorithms will be used to provide the services, and the keys used as input to the cryptographic algorithms.

IP 安全（IPsec）为 IP 数据报提供机密性、数据完整性、访问控制和数据源身份验证。这些服务是通过维护 IP 数据报的源和接收器之间的共享状态来提供的。除其他外，该状态定义提供给数据报的特定服务、将使用哪些加密算法提供服务以及用作加密算法输入的密钥。

Establishing this shared state in a manual fashion does not scale well. Therefore, a protocol to establish this state dynamically is needed. This document describes such a protocol -- the Internet Key Exchange (IKE). Version 1 of IKE was defined in RFCs 2407 [DOI], 2408 [ISAKMP], and 2409 [IKEV1]. IKEv2 replaced all of those RFCs. IKEv2 was defined in [IKEV2] (RFC 4306) and was clarified in [Clarif] (RFC 4718). This document replaces and updates RFC 4306 and RFC 4718. IKEv2 was a change to the IKE protocol that was not backward compatible. In contrast, the current document not only provides a clarification of IKEv2, but makes minimum changes to the IKE protocol. A list of the significant differences between RFC 4306 and this document is given in Section 1.7.

以手动方式建立此共享状态无法很好地扩展。因此，需要一个协议来动态地建立这种状态。本文档描述了这样一种协议——Internet 密钥交换（IKE）。IKE 的版本 1 在 RFCs 2407[DOI]、2408[ISAKMP]和 2409[IKEV1]中定义。IKEv2 替换了所有这些 RFC。IKEv2 在[IKEv2]（RFC 4306）中定义，并在[Clarif]（RFC 4718）中阐明。本文件取代并更新了 RFC 4306 和 RFC 4718。IKEv2 是对 IKE 协议的一个更改，它不向后兼容。相比之下，当前文件不仅对 IKEv2 进行了澄清，而且对 IKE 协议进行了最低限度的修改。第 1.7 节列出了 RFC 4306 和本文件之间的重大差异。

IKE performs mutual authentication between two parties and establishes an IKE security association (SA) that includes shared secret information that can be used to efficiently establish SAs for Encapsulating Security Payload (ESP) [ESP] or Authentication Header (AH) [AH] and a set of cryptographic algorithms to be used by the SAs to protect the traffic that they carry. In this document, the term "suite" or "cryptographic suite" refers to a complete set of algorithms used to protect an

SA. An initiator proposes one or more suites by listing supported algorithms that can be combined into suites in a mix-and-match fashion. IKE can also negotiate use of IP Compression (IPComp) [IP-COMP] in connection with an ESP or AH SA. The SAs for ESP or AH that get set up through that IKE SA we call "Child SAs".

IKE 在双方之间执行相互身份验证，并建立 IKE 安全关联（SA），该关联包括可用于有效建立 SA 以封装安全有效负载（ESP）[ESP]或身份验证头（AH）[AH]的共享机密信息以及 SAs 使用的一组加密算法，以保护其承载的流量。在本文档中，术语“套件”或“加密套件”指用于保护 SA 的一整套算法。发起者通过列出支持的算法来提出一个或多个套件，这些算法可以以混合匹配的方式组合成套件。IKE 还可以协商将 IP 压缩（IPComp）[IP-COMP]用于 ESP 或 AH SA。ESP 或 AH 的 SAs 通过 IKE SA 设置，我们称之为“儿童 SAs”。

All IKE communications consist of pairs of messages: a request and a response. The pair is called an "exchange", and is sometimes called a "request/response pair". The first exchange of messages establishing an IKE SA are called the IKE_SA_INIT and IKE_AUTH exchanges; subsequent IKE exchanges are called the CREATE_CHILD_SA or INFORMATIONAL exchanges. In the common case, there is a single IKE_SA_INIT exchange and a single IKE_AUTH exchange (a total of four messages) to establish the IKE SA and the first Child SA. In exceptional cases, there may be more than one of each of these exchanges. In all cases, all IKE_SA_INIT exchanges MUST complete before any other exchange type, then all IKE_AUTH exchanges MUST

所有 IKE 通信都由成对的消息组成：一个请求和一个响应。该对称为“交换”，有时称为“请求/响应对”。建立 IKE SA 的第一次消息交换称为 IKE_SA_INIT 和 IKE_AUTH 交换；随后的 IKE 交换称为创建子交换或信息交换。在常见情况下，有一个 IKE_SA_INIT 交换和一个 IKE_AUTH 交换（总共四条消息）来建立 IKE SA 和第一个子 SA。在特殊情况下，每个交易所可能不止一个。在所有情况下，所有 IKE_SA_INIT 交换都必须在任何其他交换类型之前完成，然后所有 IKE_AUTH 交换都必须完成

complete, and following that, any number of CREATE_CHILD_SA and INFORMATIONAL exchanges may occur in any order. In some scenarios, only a single Child SA is needed between the IPsec endpoints, and therefore there would be no additional exchanges. Subsequent exchanges MAY be used to establish additional Child SAs between the same authenticated pair of endpoints and to perform housekeeping functions.

完成后，可以按任意顺序进行任意数量的 CREATE_CHILD_SA 和信息交换。在某些场景中，IPsec 端点之间只需要一个子 SA，因此不会有额外的交换。随后的交换可用于在相同的已认证端点对之间建立额外的子 sa，并执行内务管理功能。

An IKE message flow always consists of a request followed by a response. It is the responsibility of the requester to ensure reliability. If the response is not received within a timeout interval, the requester needs to retransmit the request (or abandon the connection).

IKE 消息流始终由请求和响应组成。请求者有责任确保可靠性。如果在超时时间间隔内未收到响应，请求者需要重新传输请求（或放弃连接）。

The first exchange of an IKE session, IKE_SA_INIT, negotiates security parameters for the IKE SA, sends nonces, and sends Diffie-Hellman values.

IKE 会话的第一次交换 IKE_SA_INIT 协商 IKE SA 的安全参数，发送 nonce，并发送 Diffie-Hellman 值。

The second exchange, IKE_AUTH, transmits identities, proves knowledge of the secrets corresponding to the two identities, and sets up an SA for the first (and often only) AH or ESP Child SA (unless there is failure setting up the AH or ESP Child SA, in which case the IKE SA is still established without the Child SA).

第二次交换，IKE_AUTH，传输身份，证明与两个身份对应的机密知识，并为第一个（通常仅）AH 或 ESP 子 SA 设置 SA（除非设置 AH 或 ESP 子 SA 失败，在这种情况下，IKE SA 仍然在没有子 SA 的情况下建立）。

The types of subsequent exchanges are CREATE_CHILD_SA (which creates a Child SA) and INFORMATIONAL (which deletes an SA, reports error conditions, or does other housekeeping). Every request requires a response. An INFORMATIONAL request with no payloads (other than the empty Encrypted payload required by the syntax) is commonly used as a check for liveness. These subsequent exchanges cannot be used until the initial exchanges have completed.

后续交换的类型包括创建子 SA（创建子 SA）和信息交换（删除 SA、报告错误情况或执行其他内务管理）。每个请求都需要响应。没有有效负载的信息性请求（语法要求的空加密有效负载除外）通常用作活动性检查。在初始交换完成之前，不能使用这些后续交换。

In the description that follows, we assume that no errors occur. Modifications to the flow when errors occur are described in Section 2.21.

在下面的描述中，我们假设没有错误发生。第 2.21 节描述了发生错误时对流程的修改。

1.1. Usage Scenarios

1.1. 使用场景

IKE is used to negotiate ESP or AH SAs in a number of different scenarios, each with its own special requirements.

IKE 用于在许多不同的场景中协商 ESP 或 AH SA，每个场景都有自己的特殊要求。

1.1.1. Security Gateway to Security Gateway in Tunnel Mode

1.1.1. 隧道模式下的安全网关到安全网关

```
                +-+-+-+-+-+            +-+-+-+-+-+
                |         | IPsec      |         |
   Protected    |Tunnel   | tunnel     |Tunnel   |     Protected
   Subnet   <-->|Endpoint |<---------->|Endpoint |<--> Subnet
                |         |            |         |
                +-+-+-+-+-+            +-+-+-+-+-+
```

```
                +-+-+-+-+-+            +-+-+-+-+-+
                |         | IPsec      |         |
   Protected    |Tunnel   | tunnel     |Tunnel   |     Protected
   Subnet   <-->|Endpoint |<---------->|Endpoint |<--> Subnet
                |         |            |         |
                +-+-+-+-+-+            +-+-+-+-+-+
```

Figure 1: Security Gateway to Security Gateway Tunnel

图 1：安全网关到安全网关隧道

In this scenario, neither endpoint of the IP connection implements IPsec, but network nodes between them protect traffic for part of the way. Protection is transparent to the endpoints, and depends on ordinary routing to send packets through the tunnel endpoints for processing. Each endpoint would announce the set of addresses "behind" it, and packets would be sent in tunnel mode where the inner IP header would contain the IP addresses of the actual endpoints.

在这种情况下，IP 连接的两个端点都没有实现 IPsec，但它们之间的网络节点在某种程度上保护了通信量。保护对端点是透明的，并且依赖于通过隧道端点发送数据包进行处理的普通路由。每一个端点都会宣布它后面的一组地址，数据包将以隧道模式发送，其中内部 IP 报头将包含实际端点的 IP 地址。

1.1.2. Endpoint-to-Endpoint Transport Mode

1.1.2. 端点到端点传输模式

```
+-+-+-+-+-+                                  +-+-+-+-+-+
|         |              IPsec transport          |         |
|Protected|              or tunnel mode SA        |Protected|
|Endpoint |<---------------------------------------->|Endpoint |
|         |                                  |         |
+-+-+-+-+-+                                  +-+-+-+-+-+
```

```
+-+-+-+-+-+                                  +-+-+-+-+-+
|         |              IPsec transport          |         |
|Protected|              or tunnel mode SA        |Protected|
|Endpoint |<---------------------------------------->|Endpoint |
|         |                                  |         |
+-+-+-+-+-+                                  +-+-+-+-+-+
```

Figure 2: Endpoint to Endpoint

图 2：端点到端点

In this scenario, both endpoints of the IP connection implement IPsec, as required of hosts in [IPSECARCH]. Transport mode will commonly be used with no inner IP header. A single pair of addresses will be negotiated for packets to be protected by this SA. These endpoints MAY implement application-layer access controls based on the IPsec authenticated identities of the participants. This scenario enables the end-to-end security that has been a guiding principle for the Internet since [ARCHPRINC], [TRANSPARENCY], and a method of limiting the inherent problems with complexity in networks noted by [ARCHGUIDEPHIL]. Although this scenario may not be fully applicable to the IPv4 Internet, it has been deployed successfully in specific scenarios within intranets using IKEv1. It should be more broadly enabled during the transition to IPv6 and with the adoption of IKEv2.

在此场景中，IP 连接的两个端点都按照[IPSECARCH]中主机的要求实现 IPsec。传输模式通常在没有内部 IP 报头的情况下使用。将为此 SA 保护的数据包协商一对地址。这些端点可以基于参与者的 IPsec 认证身份实现应用层访问控制。该场景实现了自[ARCHPRINC]、[TRANSPARENCY]以来一直作为互联网指导原则的端到端安全性，以及[ARCHGUIDEPHIL]指出的限制网络复杂性固有问题的方法。尽管此场景可能不完全适用于 IPv4 Internet，但已使用 IKEv1 在内部网内的特定场景中成功部署。在过渡到 IPv6 和采用 IKEv2 的过程中，应该更广泛地启用它。

It is possible in this scenario that one or both of the protected endpoints will be behind a network address translation (NAT) node, in which case the tunneled packets will have to be UDP encapsulated so that port numbers in the UDP headers can be used to identify individual endpoints "behind" the NAT (see Section 2.23).

在这种情况下，一个或两个受保护的端点可能位于网络地址转换（NAT）节点后面，在这种情况下，隧道数据包必须采用UDP封装，以便UDP报头中的端口号可用于标识“在”NAT后面的各个端点（见第2.23节）。

1.1.3. Endpoint to Security Gateway in Tunnel Mode

1.1.3. 隧道模式下的端点到安全网关

```
+-+-+-+-+-+                        +-+-+-+-+-+
|         |         IPsec          |         |     Protected
|Protected|         tunnel         |Tunnel   |     Subnet
|Endpoint |<---------------------->|Endpoint |<--- and/or
|         |                        |         |     Internet
+-+-+-+-+-+                        +-+-+-+-+-+
```

```
+-+-+-+-+-+                        +-+-+-+-+-+
|         |         IPsec          |         |     Protected
|Protected|         tunnel         |Tunnel   |     Subnet
|Endpoint |<---------------------->|Endpoint |<--- and/or
|         |                        |         |     Internet
+-+-+-+-+-+                        +-+-+-+-+-+
```

Figure 3: Endpoint to Security Gateway Tunnel

图3：端点到安全网关隧道

In this scenario, a protected endpoint (typically a portable roaming computer) connects back to its corporate network through an IPsec-protected tunnel. It might use this tunnel only to access information on the corporate network, or it might tunnel all of its traffic back through the corporate network in order to take advantage of protection provided by a corporate firewall against Internet-based attacks. In either case, the protected endpoint will want an IP address associated with the security gateway so that packets returned to it will go to the security gateway and be tunneled back. This IP address may be static or may be dynamically allocated by the security gateway. In support of the latter case, IKEv2 includes a mechanism (namely, configuration payloads) for the initiator to request an IP address owned by the security gateway for use for the duration of its SA.

在此场景中，受保护的端点（通常是便携式漫游计算机）通过受 IPsec 保护的隧道连接回其公司网络。它可能仅使用此隧道访问公司网络上的信息，也可能通过公司网络将其所有通信量隧道返回，以利用公司防火墙提供的保护，抵御基于 Internet 的攻击。在任何一种情况下，受保护的端点都需要一个与安全网关关联的 IP 地址，以便返回给它的数据包将进入安全网关并通过隧道返回。该 IP 地址可以是静态的，也可以由安全网关动态分配。为了支持后一种情况，IKEv2 包括一种机制（即配置有效负载），用于启动器请求安全网关拥有的 IP 地址，以便在其 SA 期间使用。

In this scenario, packets will use tunnel mode. On each packet from the protected endpoint, the outer IP header will contain the source IP address associated with its current location (i.e., the address that will get traffic routed to the endpoint directly), while the inner IP header will contain the source IP address assigned by the security gateway (i.e., the address that will get traffic routed to the security gateway for forwarding to the endpoint). The outer destination address will always be that of the security gateway, while the inner destination address will be the ultimate destination for the packet.

在这种情况下，数据包将使用隧道模式。在来自受保护端点的每个数据包上，外部 IP 报头将包含与其当前位置关联的源 IP 地址（即，将流量直接路由到端点的地址），而内部 IP 报头将包含由安全网关分配的源 IP 地址（即，将流量路由到安全网关以转发到端点的地址）。外部目的地地址将始终是安全网关的地址，而内部目的地地址将是数据包的最终目的地。

In this scenario, it is possible that the protected endpoint will be behind a NAT. In that case, the IP address as seen by the security gateway will not be the same as the IP address sent by the protected

在这种情况下，受保护的端点可能位于 NAT 后面。在这种情况下，安全网关看到的 IP 地址将与受保护网关发送的 IP 地址不同

endpoint, and packets will have to be UDP encapsulated in order to be routed properly. Interaction with NATs is covered in detail in Section 2.23.

端点和数据包必须进行 UDP 封装才能正确路由。第 2.23 节详细介绍了与 NAT 的交互作用。

1.1.4. Other Scenarios

1.1.4. 其他情景

Other scenarios are possible, as are nested combinations of the above. One notable example combines aspects of Sections 1.1.1 and 1.1.3. A subnet may make all external accesses through a remote security gateway using an IPsec tunnel, where the addresses on the subnet are routed to the security gateway by the rest of the Internet. An example would be someone's home network being virtually on the Internet with static IP addresses even though connectivity is provided by an ISP that assigns a single dynamically assigned IP address to the user's security gateway (where the static IP addresses and an IPsec relay are provided by a third party located elsewhere).

其他场景也是可能的，上面的嵌套组合也是如此。一个值得注意的例子结合了第 1.1.1 节和第 1.1.3 节的各个方面。子网可以使用 IPsec 隧道通过远程安全网关进行所有外部访问，其中子网上的地址由 Internet 的其余部分路由到安全网关。例如，即使 ISP 为用户的安全网关分配一个动态分配的 IP 地址（其中静态 IP 地址和 IPsec 中继由位于别处的第三方提供），但某人的家庭网络实际上位于具有静态 IP 地址的 Internet 上。

1.2. The Initial Exchanges

1.2. 最初的交流

Communication using IKE always begins with IKE_SA_INIT and IKE_AUTH exchanges (known in IKEv1 as Phase 1). These initial exchanges normally consist of four messages, though in some scenarios that number can grow. All communications using IKE consist of request/ response pairs. We'll describe the base exchange first, followed by variations. The first pair of messages (IKE_SA_INIT) negotiate cryptographic algorithms, exchange nonces, and do a Diffie-Hellman exchange [DH].

使用 IKE 的通信总是从 IKE_SA_INIT 和 IKE_AUTH 交换开始（在 IKEv1 中称为阶段 1）。这些初始交换通常由四条消息组成，但在某些情况下，这一数字可能会增加。使用 IKE 的所有通信都由请求/响应对组成。我们将首先描述基本交换，然后是变体。第一对消息（IKE_SA_INIT）协商加密算法，交换 nonce，并执行 Diffie-Hellman 交换[DH]。

The second pair of messages (IKE_AUTH) authenticate the previous messages, exchange identities and certificates, and establish the first Child SA. Parts of these messages are encrypted and integrity protected with keys established through the IKE_SA_INIT exchange, so the identities are hidden from eavesdroppers and all fields in all the messages are authenticated. See Section 2.14 for information on how the encryption keys are generated. (A man-in-the-middle attacker who cannot complete the IKE_AUTH exchange can nonetheless see the identity of the initiator.)

第二对消息（IKE_AUTH）对以前的消息进行身份验证，交换身份和证书，并建立第一个子 SA。

这些消息的一部分通过 IKE_SA_INIT 交换建立的密钥进行加密和完整性保护，因此身份对窃听者隐藏，所有消息中的所有字段都经过身份验证。有关如何生成加密密钥的信息，请参见第 2.14 节。（中间人攻击者无法完成 IKE_身份验证交换，但仍可以看到发起人的身份。）

All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the IKE_SA_INIT exchange. These subsequent messages use the syntax of the Encrypted payload described in Section 3.14, encrypted with keys that are derived as described in Section 2.14. All subsequent messages include an Encrypted payload, even if they are referred to in the text as "empty". For the CREATE_CHILD_SA, IKE_AUTH, or INFORMATIONAL exchanges, the message following the header is encrypted and the message including the header is integrity protected using the cryptographic algorithms negotiated for the IKE SA.

初始交换之后的所有消息都使用 IKE_SA_INIT 交换中协商的加密算法和密钥进行加密保护。这些后续消息使用第 3.14 节中描述的加密有效负载的语法，使用第 2.14 节中描述的派生密钥进行加密。所有后续消息都包含加密的有效负载，即使它们在文本中被称为“空”。对于 CREATE_CHILD_SA、IKE_AUTH 或信息交换，头后面的消息将被加密，包括头的消息将使用为 IKE SA 协商的加密算法进行完整性保护。

Every IKE message contains a Message ID as part of its fixed header. This Message ID is used to match up requests and responses, and to identify retransmissions of messages.

每个 IKE 消息都包含一个消息 ID 作为其固定头的一部分。此消息 ID 用于匹配请求和响应，并标识消息的重新传输。

In the following descriptions, the payloads contained in the message are indicated by names as listed below.

在下面的描述中，消息中包含的有效载荷由下面列出的名称表示。

Notation	Payload
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
CP	Configuration
D	Delete
EAP	Extensible Authentication
HDR	IKE header (not a payload)

IDi	Identification - Initiator
IDr	Identification - Responder
KE	Key Exchange
Ni, Nr	Nonce
N	Notify
SA	Security Association
SK	Encrypted and Authenticated
TSi	Traffic Selector - Initiator
TSr	Traffic Selector - Responder
V	Vendor ID

Notation	Payload
AUTH	Authentication
CERT	Certificate
CERTREQ	Certificate Request
CP	Configuration
D	Delete
EAP	Extensible Authentication
HDR	IKE header (not a payload)
IDi	Identification - Initiator
IDr	Identification - Responder
KE	Key Exchange
Ni, Nr	Nonce
N	Notify
SA	Security Association
SK	Encrypted and Authenticated
TSi	Traffic Selector - Initiator
TSr	Traffic Selector - Responder
V	Vendor ID

The details of the contents of each payload are described in section 3. Payloads that may optionally appear will be shown in brackets, such as [CERTREQ]; this indicates that a Certificate Request payload can optionally be included.

第 3 节详细介绍了每个有效载荷的内容。可选择出现的有效载荷将显示在括号中，如 [CERTREQ]；这表示可以选择性地包括证书请求负载。

The initial exchanges are as follows:

初步交流如下：

```
Initiator                         Responder
```

```
----------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
```

```
Initiator                Responder
----------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
```

HDR contains the Security Parameter Indexes (SPIs), version numbers, and flags of various sorts. The SAi1 payload states the cryptographic algorithms the initiator supports for the IKE SA. The KE payload sends the initiator's Diffie-Hellman value. Ni is the initiator's nonce.

HDR 包含各种安全参数索引（SPI）、版本号和标志。SAi1 有效负载说明了启动器支持的 IKE SA 加密算法。KE 有效负载发送启动器的 Diffie-Hellman 值。Ni 是发起者的 nonce。

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

<--HDR、SAr1、KEr、Nr、[CERTREQ]

The responder chooses a cryptographic suite from the initiator's offered choices and expresses that choice in the SAr1 payload, completes the Diffie-Hellman exchange with the KEr payload, and sends its nonce in the Nr payload.

响应者从发起者提供的选项中选择一个加密套件，并在 SAr1 负载中表示该选项，完成与 KEr 负载的 Diffie-Hellman 交换，并在 Nr 负载中发送其 nonce。

At this point in the negotiation, each party can generate SKEYSEED, from which all keys are derived for that IKE SA. The messages that follow are encrypted and integrity protected in their entirety, with the exception of the message headers. The keys used for the encryption and integrity protection are derived from SKEYSEED and are known as SK_e (encryption) and SK_a (authentication, a.k.a. integrity protection); see Sections 2.13 and 2.14 for details on the key derivation. A separate SK_e and SK_a is computed for each direction. In addition to the keys SK_e and SK_a derived from the Diffie-Hellman value for protection of the IKE SA, another quantity SK_d is derived and used for derivation of further keying material for Child SAs. The notation SK { ... } indicates that these payloads are encrypted and integrity protected using that direction's SK_e and SK_a.

在协商的这一点上，每一方都可以生成 skeysed，从中派生出该 IKE SA 的所有密钥。除了消息

头之外，后面的消息都是加密的，并且完整性受到保护。用于加密和完整性保护的密钥来自 SKEYSEED，称为 sku_e（加密）和 SK_a（认证，也称为完整性保护）；有关密钥派生的详细信息，请参见第 2.13 节和第 2.14 节。为每个方向计算单独的 SK_e 和 SK_A。除了从 Diffie-Hellman 值导出的密钥 SK_e 和 SK_a 用于保护 IKE SA 之外，还导出了另一个量 SK_d，并用于派生用于儿童 SA 的进一步键控材料。符号 SK{...}表示使用该方向的 SK_e 和 SK_a 对这些有效载荷进行加密和完整性保护。

HDR, SK {IDi, [CERT,] [CERTREQ,] [IDr,] AUTH, SAi2, TSi, TSr} -->

HDR，SK{IDi，[CERT，][CERTREQ，][IDr，]AUTH，SAi2，TSi，TSr}-->

The initiator asserts its identity with the IDi payload, proves knowledge of the secret corresponding to IDi and integrity protects the contents of the first message using the AUTH payload (see Section 2.15). It might also send its certificate(s) in CERT payload(s) and a list of its trust anchors in CERTREQ payload(s). If any CERT payloads are included, the first certificate provided MUST contain the public key used to verify the AUTH field.

发起者使用 IDi 有效载荷声明其身份，证明其知道与 IDi 相对应的秘密，并且完整性使用认证有效载荷保护第一条消息的内容（参见第 2.15 节）。它还可以在 CERT 有效负载中发送其证书，并在 CERTREQ 有效负载中发送其信任锚的列表。如果包含任何证书有效载荷，则提供的第一个证书必须包含用于验证 AUTH 字段的公钥。

The optional payload IDr enables the initiator to specify to which of the responder's identities it wants to talk. This is useful when the machine on which the responder is running is hosting multiple identities at the same IP address. If the IDr proposed by the initiator is not acceptable to the responder, the responder might use some other IDr to finish the exchange. If the initiator then does not accept the fact that responder used an IDr different than the one that was requested, the initiator can close the SA after noticing the fact.

可选的有效负载 IDr 使发起方能够指定它要与响应方的哪个身份通信。当响应程序运行的计算机在同一 IP 地址上承载多个标识时，这非常有用。如果发起方建议的 IDr 不被响应方接受，响应方可能会使用其他 IDr 来完成交换。如果发起方随后不接受响应方使用的 IDr 与请求的 IDr 不同的事实，则发起方可以在注意到该事实后关闭 SA。

The Traffic Selectors (TSi and TSr) are discussed in Section 2.9.

第 2.9 节讨论了流量选择器（TSi 和 TSr）。

The initiator begins negotiation of a Child SA using the SAi2 payload. The final fields (starting with SAi2) are described in the description of the CREATE_CHILD_SA exchange.

发起方开始使用 SAi2 有效负载协商子 SA。最后的字段（从 SAi2 开始）在 CREATE_CHILD_SA 交换的描述中描述。

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

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

The responder asserts its identity with the IDr payload, optionally sends one or more certificates (again with the certificate containing the public key used to verify AUTH listed first), authenticates its identity and protects the integrity of the second message with the AUTH payload, and completes negotiation of a Child SA with the additional fields described below in the CREATE_CHILD_SA exchange.

响应者用 IDr 有效载荷声明其身份，选择性地发送一个或多个证书（同样，证书包含用于验证首先列出的身份验证的公钥），用身份验证有效载荷验证其身份并保护第二条消息的完整性，并在 CREATE_Child_SA 交换中使用下面描述的其他字段完成子 SA 的协商。

Both parties in the IKE_AUTH exchange MUST verify that all signatures and Message Authentication Codes (MACs) are computed correctly. If either side uses a shared secret for authentication, the names in the ID payload MUST correspond to the key used to generate the AUTH payload.

IKE_身份验证交换的双方必须验证是否正确计算了所有签名和消息身份验证码（MAC）。如果任何一方使用共享密钥进行身份验证，则 ID 有效负载中的名称必须与用于生成身份验证有效负载的密钥相对应。

Because the initiator sends its Diffie-Hellman value in the IKE_SA_INIT, it must guess the Diffie-Hellman group that the responder will select from its list of supported groups. If the initiator guesses wrong, the responder will respond with a Notify payload of type INVALID_KE_PAYLOAD indicating the selected group. In this case, the initiator MUST retry the IKE_SA_INIT with the corrected Diffie-Hellman group. The initiator MUST again propose its full set of acceptable cryptographic suites because the rejection message was unauthenticated and otherwise an active attacker could trick the endpoints into negotiating a weaker suite than a stronger one that they both prefer.

由于发起程序在 IKE_SA_INIT 中发送其 Diffie Hellman 值，因此它必须猜测响应程序将从其支持的组列表中选择的 Diffie Hellman 组。如果发起者猜错了，响应者将使用 INVALID_KEU_payload 类型的 Notify 有效负载进行响应，指示所选组。在这种情况下，启动器必须使用更正的 Diffie-Hellman 组重试 IKE_SA_INIT。发起方必须再次提出其可接受的全套加密套件，因为拒绝消息未经验证，否则主动攻击者可能会诱使端点协商一个较弱的套件，而不是双方都喜欢的较强的套件。

If creating the Child SA during the IKE_AUTH exchange fails for some reason, the IKE SA is still created as usual. The list of Notify message types in the IKE_AUTH exchange that do not prevent an IKE SA from being set up include at least the following: NO_PROPOSAL_CHOSEN, TS_UNACCEPTABLE, SINGLE_PAIR_REQUIRED, INTERNAL_ADDRESS_FAILURE, and FAILED_CP_REQUIRED.

如果在 IKE_身份验证交换过程中创建子 SA 由于某种原因失败，IKE SA 仍将照常创建。IKE_身份验证交换中不阻止设置 IKE SA 的通知消息类型列表至少包括以下内容：未选择建议、TS_不可接受、需要单对、内部地址失败以及需要失败的 CP。

If the failure is related to creating the IKE SA (for example, an AUTHENTICATION_FAILED Notify error message is returned), the IKE SA is not created. Note that although the IKE_AUTH messages are encrypted and integrity protected, if the peer receiving this Notify error message has not yet authenticated the other end (or if the peer fails to authenticate the other end for some reason), the information needs to be treated with caution. More precisely, assuming that the MAC verifies correctly, the sender of the error Notify message is known to be the responder of the IKE_SA_INIT exchange, but the sender's identity cannot be assured.

如果故障与创建 IKE SA 有关（例如，返回身份验证失败通知错误消息），则不会创建 IKE SA。请注意，尽管 IKE_AUTH 消息已加密且完整性受到保护，但如果接收此 Notify 错误消息的对等方尚未对另一端进行身份验证（或者如果对等方由于某种原因未能对另一端进行身份验证），则需要谨慎处理该信息。更准确地说，假设 MAC 正确验证，错误通知消息的发送者已知是 IKE_SA_INIT 交换的响应者，但无法确定发送者的身份。

Note that IKE_AUTH messages do not contain KEi/KEr or Ni/Nr payloads. Thus, the SA payloads in the IKE_AUTH exchange cannot contain Transform Type 4 (Diffie-Hellman group) with any value other than NONE. Implementations SHOULD omit the whole transform substructure instead of sending value NONE.

请注意，IKE_AUTH 消息不包含 KEi/KEr 或 Ni/Nr 有效负载。因此，IKE_AUTH 交换中的 SA 有

效负载不能包含转换类型 4（Diffie Hellman 组），其值不能为 NONE。实现应该省略整个转换子结构，而不是发送值 NONE。

1.3. The CREATE_CHILD_SA Exchange

1.3. 创建子交换

The CREATE_CHILD_SA exchange is used to create new Child SAs and to rekey both IKE SAs and Child SAs. This exchange consists of a single request/response pair, and some of its function was referred to as a Phase 2 exchange in IKEv1. It MAY be initiated by either end of the IKE SA after the initial exchanges are completed.

CREATE_CHILD_SA 交换用于创建新的子 SA，并重新设置 IKE SA 和子 SA 的密钥。此交换由单个请求/响应对组成，其部分功能在 IKEv1 中称为阶段 2 交换。初始交换完成后，它可以由 IKE SA 的任意一端发起。

An SA is rekeyed by creating a new SA and then deleting the old one. This section describes the first part of rekeying, the creation of new SAs; Section 2.8 covers the mechanics of rekeying, including moving traffic from old to new SAs and the deletion of the old SAs. The two sections must be read together to understand the entire process of rekeying.

通过创建新 SA，然后删除旧 SA，可以重新设置 SA 的密钥。本节介绍了密钥更新的第一部分，即新 SA 的创建；第 2.8 节介绍了密钥更新机制，包括将流量从旧 SA 移动到新 SA 以及删除旧 SA。这两个部分必须一起阅读，以了解重新键入的整个过程。

Either endpoint may initiate a CREATE_CHILD_SA exchange, so in this section the term initiator refers to the endpoint initiating this exchange. An implementation MAY refuse all CREATE_CHILD_SA requests within an IKE SA.

任何一个端点都可以启动 CREATE_CHILD_SA 交换，因此在本节中，术语 initiator 指启动此交换的端点。实现可以拒绝 IKE SA 中的所有 CREATE_CHILD_SA 请求。

The CREATE_CHILD_SA request MAY optionally contain a KE payload for an additional Diffie-Hellman exchange to enable stronger guarantees of forward secrecy for the Child SA. The keying material for the Child SA is a function of SK_d established during the establishment of the IKE SA, the nonces exchanged during the CREATE_CHILD_SA exchange, and the Diffie-Hellman value (if KE payloads are included in the CREATE_CHILD_SA exchange).

CREATE_CHILD_SA 请求可以选择性地包含用于额外 Diffie-Hellman 交换的 KE 有效载荷，以

便为 CHILD SA 提供更有力的前向保密性保证。子 SA 的键控材料是在 IKE SA 建立期间建立的 SK_d、在 CREATE_Child_SA 交换期间交换的 nonce 和 Diffie Hellman 值（如果 CREATE_Child_SA 交换中包括 KE 有效载荷）的函数。

If a CREATE_CHILD_SA exchange includes a KEi payload, at least one of the SA offers MUST include the Diffie-Hellman group of the KEi. The Diffie-Hellman group of the KEi MUST be an element of the group the initiator expects the responder to accept (additional Diffie-Hellman groups can be proposed). If the responder selects a proposal using a different Diffie-Hellman group (other than NONE), the responder MUST reject the request and indicate its preferred Diffie-Hellman group in the INVALID_KE_PAYLOAD Notify payload. There are two octets of data associated with this notification: the accepted Diffie-Hellman group number in big endian order. In the case of such a rejection, the CREATE_CHILD_SA exchange fails, and the initiator will probably retry the exchange with a Diffie-Hellman proposal and KEi in the group that the responder gave in the INVALID_KE_PAYLOAD Notify payload.

如果 CREATE_CHILD_SA 交换包含 KEi 有效负载，则至少一个 SA 提供必须包含 KEi 的 Diffie Hellman 组。KEi 的 Diffie-Hellman 组必须是发起方希望响应方接受的组的一个元素（可以提出其他 Diffie-Hellman 组）。如果响应者使用不同的 Diffie-Hellman 组（无组除外）选择提案，则响应者必须拒绝请求，并在无效的有效载荷 Notify 有效载荷中指明其首选的 Diffie-Hellman 组。与此通知关联的数据有两个八位字节：按大端顺序接受的 Diffie-Hellman 组编号。在这种拒绝的情况下，CREATE_CHILD_SA 交换失败，发起方可能会使用 Diffie Hellman 建议和响应方在无效的_KE_有效负载 Notify 有效负载中给出的组中的 KEi 重试交换。

The responder sends a NO_ADDITIONAL_SAS notification to indicate that a CREATE_CHILD_SA request is unacceptable because the responder is unwilling to accept any more Child SAs on this IKE SA. This notification can also be used to reject IKE SA rekey. Some minimal implementations may only accept a single Child SA setup in the context of an initial IKE exchange and reject any subsequent attempts to add more.

响应者发送一个 NO_ADDITIONAL_SAS 通知，指出创建子 SA 请求不可接受，因为响应者不愿意在此 IKE SA 上接受更多子 SA。此通知还可用于拒绝 IKE SA 密钥。一些最小实现可能只接受初始 IKE 交换上下文中的单个子 SA 设置，并拒绝任何后续添加更多的尝试。

1.3.1. Creating New Child SAs with the CREATE_CHILD_SA Exchange

1.3.1. 使用 CREATE_Child_SA 交换创建新的子 SA

A Child SA may be created by sending a CREATE_CHILD_SA request. The

CREATE_CHILD_SA request for creating a new Child SA is:

可以通过发送创建子 SA 请求来创建子 SA。创建新子 SA 的 CREATE_CHILD_SA 请求为：

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SK {SA, Ni, [KEi],
          TSi, TSr}  -->
```

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SK {SA, Ni, [KEi],
          TSi, TSr}  -->
```

The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, optionally a Diffie-Hellman value in the KEi payload, and the proposed Traffic Selectors for the proposed Child SA in the TSi and TSr payloads.

发起方发送 SA 有效载荷中的 SA offer、Ni 有效载荷中的 nonce、KEi 有效载荷中可选的 Diffie-Hellman 值，以及 TSi 和 TSr 有效载荷中建议的子 SA 的建议流量选择器。

The CREATE_CHILD_SA response for creating a new Child SA is:

用于创建新子 SA 的 CREATE_CHILD_SA 响应为：

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

<--HDR，SK{SA，Nr，[KEr]，TSi，TSr}

The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if KEi was included in the request and the selected cryptographic suite includes that group.

如果请求中包含 KEi 且所选加密套件包括该组，则响应者使用 SA 有效负载中的已接受要约和 KEr 有效负载中的 Diffie Hellman 值进行响应（使用相同的消息 ID 进行响应）。

The Traffic Selectors for traffic to be sent on that SA are specified in the TS payloads in the response, which may be a subset of what the initiator of the Child SA proposed.

将在该 SA 上发送的流量的流量选择器在响应中的 TS 有效负载中指定，其可以是子 SA 的发起方

提议的子集。

The USE_TRANSPORT_MODE notification MAY be included in a request message that also includes an SA payload requesting a Child SA. It requests that the Child SA use transport mode rather than tunnel mode for the SA created. If the request is accepted, the response MUST also include a notification of type USE_TRANSPORT_MODE. If the responder declines the request, the Child SA will be established in tunnel mode. If this is unacceptable to the initiator, the initiator MUST delete the SA. Note: Except when using this option to negotiate transport mode, all Child SAs will use tunnel mode.

使用传输模式通知可以包括在请求消息中，该请求消息还包括请求子 SA 的 SA 有效载荷。它要求子 SA 为创建的 SA 使用传输模式而不是隧道模式。如果请求被接受，响应还必须包括类型为 USE\u TRANSPORT\u MODE 的通知。如果响应者拒绝请求，则子 SA 将以隧道模式建立。如果发起者不能接受，发起者必须删除 SA。注意：除使用此选项协商传输模式外，所有子 SA 都将使用隧道模式。

The ESP_TFC_PADDING_NOT_SUPPORTED notification asserts that the sending endpoint will not accept packets that contain Traffic Flow Confidentiality (TFC) padding over the Child SA being negotiated. If neither endpoint accepts TFC padding, this notification is included in both the request and the response. If this notification is included in only one of the messages, TFC padding can still be sent in the other direction.

ESP_TFC_PADDING_NOT_SUPPORTED 通知断言，发送端点将不接受在正在协商的子 SA 上包含流量流机密性（TFC）填充的数据包。如果两个端点都不接受 TFC 填充，则该通知将同时包含在请求和响应中。如果此通知仅包含在一条消息中，则 TFC 填充仍可以向另一个方向发送。

The NON_FIRST_FRAGMENTS_ALSO notification is used for fragmentation control. See [IPSECARCH] for a fuller explanation. Both parties need to agree to sending non-first fragments before either party does so. It is enabled only if NON_FIRST_FRAGMENTS_ALSO notification is included in both the request proposing an SA and the response accepting it. If the responder does not want to send or receive non-first fragments, it only omits NON_FIRST_FRAGMENTS_ALSO notification from its response, but does not reject the whole Child SA creation.

非\u 第一\u 片段\u 通知也用于碎片控制。有关更全面的解释，请参见[IPSECARCH]。在任何一方发送非首件碎片之前，双方必须同意发送非首件碎片。仅当提出 SA 的请求和接受 SA 的响应中都包含非_FIRST _FRAGMENTS _通知时，才会启用此功能。如果响应者不想发送或接收非第一

个片段，它只会从其响应中省略非第一个片段通知，但不会拒绝整个子 SA 创建。

An IPCOMP_SUPPORTED notification, covered in Section 2.22, can also be included in the exchange.

第 2.22 节中介绍的 IPCOMP_支持的通知也可以包含在 exchange 中。

A failed attempt to create a Child SA SHOULD NOT tear down the IKE SA: there is no reason to lose the work done to set up the IKE SA. See Section 2.21 for a list of error messages that might occur if creating a Child SA fails.

创建子 SA 的失败尝试不应破坏 IKE SA：没有理由丢失设置 IKE SA 所做的工作。有关创建子 SA 失败时可能出现的错误消息列表，请参见第 2.21 节。

1.3.2. Rekeying IKE SAs with the CREATE_CHILD_SA Exchange

1.3.2. 使用 CREATE_CHILD_SA 交换重新键入 IKE SA

The CREATE_CHILD_SA request for rekeying an IKE SA is:

重新设置 IKE SA 密钥的 CREATE_CHILD_SA 请求为：

```
Initiator                         Responder
-----------------------------------------------------------------
HDR, SK {SA, Ni, KEi} -->
```

```
Initiator                         Responder
-----------------------------------------------------------------
HDR, SK {SA, Ni, KEi} -->
```

The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, and a Diffie-Hellman value in the KEi payload. The KEi payload MUST be included. A new initiator SPI is supplied in the SPI field of the SA payload. Once a peer receives a request to rekey an IKE SA or sends a request to rekey an IKE SA, it SHOULD NOT start any new CREATE_CHILD_SA exchanges on the IKE SA that is being rekeyed.

启动器发送 SA 有效负载中的 SA offer、Ni 有效负载中的 nonce 和 KEi 有效负载中的 Diffie-Hellman 值。必须包括 KEi 有效载荷。SA 有效负载的 SPI 字段中提供了一个新的启动器 SPI。一旦对等方接收到重新设置 IKE SA 密钥的请求或发送重新设置 IKE SA 密钥的请求，它就不应该在正在重新设置密钥的 IKE SA 上启动任何新的 CREATE_CHILD_SA 交换。

The CREATE_CHILD_SA response for rekeying an IKE SA is:

为 IKE SA 重新设置密钥的 CREATE_CHILD_SA 响应为：

```
                              <--  HDR, SK {SA, Nr, KEr}
```

```
                              <--  HDR, SK {SA, Nr, KEr}
```

The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if the selected cryptographic suite includes that group. A new responder SPI is supplied in the SPI field of the SA payload.

响应者使用 SA 有效负载中的已接受要约以及 KEr 有效负载中的 Diffie Hellman 值（如果所选加密套件包括该组）进行响应（使用相同的消息 ID 进行响应）。SA 有效负载的 SPI 字段中提供了一个新的响应器 SPI。

The new IKE SA has its message counters set to 0, regardless of what they were in the earlier IKE SA. The first IKE requests from both sides on the new IKE SA will have Message ID 0. The old IKE SA retains its numbering, so any further requests (for example, to delete the IKE SA) will have consecutive numbering. The new IKE SA also has its window size reset to 1, and the initiator in this rekey exchange is the new "original initiator" of the new IKE SA.

新的 IKE SA 将其消息计数器设置为 0，而不管它们在早期的 IKE SA 中是什么。来自新 IKE SA 两侧的第一个 IKE 请求将具有消息 ID 0。旧的 IKE SA 保留其编号，因此任何进一步的请求（例如，删除 IKE SA）都将具有连续编号。新 IKE SA 也将其窗口大小重置为 1，并且此密钥交换中的启动器是新 IKE SA 的新“原始启动器”。

Section 2.18 also covers IKE SA rekeying in detail.

第 2.18 节还详细介绍了 IKE SA 密钥更新。

1.3.3. Rekeying Child SAs with the CREATE_CHILD_SA Exchange

1.3.3. 使用 CREATE_Child_SA 交换重新键入子 SA

The CREATE_CHILD_SA request for rekeying a Child SA is:

用于重新键入子 SA 的 CREATE_CHILD_SA 请求为：

```
   Initiator                         Responder
   -------------------------------------------------------------------
```

```
HDR, SK {N(REKEY_SA), SA, Ni, [KEi],
    TSi, TSr}   -->


Initiator                         Responder
-------------------------------------------------------------------
HDR, SK {N(REKEY_SA), SA, Ni, [KEi],
    TSi, TSr}   -->
```

The initiator sends SA offer(s) in the SA payload, a nonce in the Ni payload, optionally a Diffie-Hellman value in the KEi payload, and the proposed Traffic Selectors for the proposed Child SA in the TSi and TSr payloads.

发起方发送SA有效载荷中的SA offer、Ni有效载荷中的nonce、KEi有效载荷中可选的Diffie-Hellman值，以及TSi和TSr有效载荷中建议的子SA的建议流量选择器。

The notifications described in Section 1.3.1 may also be sent in a rekeying exchange. Usually, these will be the same notifications that were used in the original exchange; for example, when rekeying a transport mode SA, the USE_TRANSPORT_MODE notification will be used.

第1.3.1节中所述的通知也可以通过密钥交换发送。通常，这些通知与原始交换中使用的通知相同；例如，当重新键入传输模式SA时，将使用使用传输模式通知。

The REKEY_SA notification MUST be included in a CREATE_CHILD_SA exchange if the purpose of the exchange is to replace an existing ESP or AH SA. The SA being rekeyed is identified by the SPI field in the Notify payload; this is the SPI the exchange initiator would expect in inbound ESP or AH packets. There is no data associated with this Notify message type. The Protocol ID field of the REKEY_SA notification is set to match the protocol of the SA we are rekeying, for example, 3 for ESP and 2 for AH.

如果交换的目的是替换现有ESP或AH SA，则必须在创建子SA交换中包含更新SA通知。通过通知有效载荷中的SPI字段来识别正在被重设密钥的SA；这是exchange启动器在入站ESP或AH数据包中预期的SPI。没有与此通知消息类型关联的数据。重新键入SA通知的协议ID字段设置为与我们正在重新键入的SA的协议匹配，例如，ESP为3，AH为2。

The CREATE_CHILD_SA response for rekeying a Child SA is:

用于重新键入子SA的CREATE_CHILD_SA响应为：

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

<--HDR，SK{SA，Nr，[KEr]，TSi，TSr}

The responder replies (using the same Message ID to respond) with the accepted offer in an SA payload, and a Diffie-Hellman value in the KEr payload if KEi was included in the request and the selected cryptographic suite includes that group.

如果请求中包含 KEi 且所选加密套件包括该组，则响应者使用 SA 有效负载中的已接受要约和 KEr 有效负载中的 Diffie Hellman 值进行响应（使用相同的消息 ID 进行响应）。

The Traffic Selectors for traffic to be sent on that SA are specified in the TS payloads in the response, which may be a subset of what the initiator of the Child SA proposed.

将在该 SA 上发送的流量的流量选择器在响应中的 TS 有效负载中指定，其可以是子 SA 的发起方提议的子集。

1.4. The INFORMATIONAL Exchange

1.4. 信息交流

At various points during the operation of an IKE SA, peers may desire to convey control messages to each other regarding errors or notifications of certain events. To accomplish this, IKE defines an INFORMATIONAL exchange. INFORMATIONAL exchanges MUST ONLY occur after the initial exchanges and are cryptographically protected with the negotiated keys. Note that some informational messages, not exchanges, can be sent outside the context of an IKE SA. Section 2.21 also covers error messages in great detail.

在 IKE SA 的操作期间的不同时刻，对等方可能希望彼此传递关于错误或某些事件的通知的控制消息。为了实现这一点，IKE 定义了一个信息交换。信息交换必须在初始交换之后进行，并使用协商密钥进行加密保护。请注意，某些信息性消息（而不是交换）可以在 IKE SA 的上下文之外发送。第 2.21 节还详细介绍了错误消息。

Control messages that pertain to an IKE SA MUST be sent under that IKE SA. Control messages that pertain to Child SAs MUST be sent under the protection of the IKE SA that generated them (or its successor if the IKE SA was rekeyed).

与 IKE SA 相关的控制消息必须在该 IKE SA 下发送。与子 SA 相关的控制消息必须在生成它们的 IKE SA 的保护下发送（如果 IKE SA 被重新设置密钥，则为其后续消息）。

Messages in an INFORMATIONAL exchange contain zero or more Notification, Delete, and Configuration payloads. The recipient of an INFORMATIONAL exchange request MUST send some response; otherwise, the sender will assume the message was lost in the network and will retransmit it. That response MAY be an empty message. The request message in an INFORMATIONAL exchange MAY also contain no payloads. This is the expected way an endpoint can ask the other endpoint to verify that it is alive.

信息交换中的消息包含零个或多个通知、删除和配置有效负载。信息交换请求的收件人必须发送一些响应；否则，发送方将假定消息在网络中丢失，并将重新传输它。该响应可能是一条空消息。信息交换中的请求消息也可能不包含有效负载。这是端点要求另一个端点验证其是否处于活动状态的预期方式。

The INFORMATIONAL exchange is defined as:

信息交换定义为：

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SK {[N,] [D,]
    [CP,] ...}  -->
                              <--  HDR, SK {[N,] [D,]
                                       [CP], ...}
```

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SK {[N,] [D,]
    [CP,] ...}  -->
                              <--  HDR, SK {[N,] [D,]
                                       [CP], ...}
```

The processing of an INFORMATIONAL exchange is determined by its component payloads.

信息交换的处理由其组件有效负载决定。

1.4.1. Deleting an SA with INFORMATIONAL Exchanges

1.4.1. 使用信息交换删除 SA

ESP and AH SAs always exist in pairs, with one SA in each direction. When an SA is closed, both members of the pair MUST be closed (that is, deleted). Each endpoint

MUST close its incoming SAs and allow the other endpoint to close the other SA in each pair. To delete an SA, an INFORMATIONAL exchange with one or more Delete payloads is

ESP 和 AH SA 始终成对存在，每个方向有一个 SA。当 SA 关闭时，该对的两个成员都必须关闭（即删除）。每个端点必须关闭其传入 SA，并允许另一个端点关闭每对中的另一个 SA。要删除 SA，需要与一个或多个删除有效负载进行信息交换

sent listing the SPIs (as they would be expected in the headers of inbound packets) of the SAs to be deleted. The recipient MUST close the designated SAs. Note that one never sends Delete payloads for the two sides of an SA in a single message. If there are many SAs to delete at the same time, one includes Delete payloads for the inbound half of each SA pair in the INFORMATIONAL exchange.

已发送，列出要删除的 SAs 的 SPI（如入站数据包头中所预期的）。收件人必须关闭指定的 SA。请注意，不会在一条消息中为 SA 的两侧发送删除有效负载。如果同时要删除多个 SA，则信息交换中将包括每个 SA 对的入站一半的删除有效负载。

Normally, the response in the INFORMATIONAL exchange will contain Delete payloads for the paired SAs going in the other direction. There is one exception. If, by chance, both ends of a set of SAs independently decide to close them, each may send a Delete payload and the two requests may cross in the network. If a node receives a delete request for SAs for which it has already issued a delete request, it MUST delete the outgoing SAs while processing the request and the incoming SAs while processing the response. In that case, the responses MUST NOT include Delete payloads for the deleted SAs, since that would result in duplicate deletion and could in theory delete the wrong SA.

通常，信息交换中的响应将包含反向配对 SA 的删除有效载荷。有一个例外。如果碰巧，一组 SA 的两端独立决定关闭它们，则每个 SA 都可能发送一个删除有效负载，并且两个请求可能在网络中交叉。如果节点收到已发出删除请求的 SAs 的删除请求，则必须在处理请求时删除传出 SAs，在处理响应时删除传入 SAs。在这种情况下，响应不得包括已删除 SA 的删除有效载荷，因为这将导致重复删除，并且理论上可能删除错误的 SA。

Similar to ESP and AH SAs, IKE SAs are also deleted by sending an Informational exchange. Deleting an IKE SA implicitly closes any remaining Child SAs negotiated under it. The response to a request that deletes the IKE SA is an empty INFORMATIONAL response.

与 ESP 和 AH SA 类似，IKE SA 也通过发送信息交换来删除。删除 IKE SA 会隐式关闭在其下协

商的所有剩余子 SA。对删除 IKE SA 的请求的响应是空的信息响应。

Half-closed ESP or AH connections are anomalous, and a node with auditing capability should probably audit their existence if they persist. Note that this specification does not specify time periods, so it is up to individual endpoints to decide how long to wait. A node MAY refuse to accept incoming data on half-closed connections but MUST NOT unilaterally close them and reuse the SPIs. If connection state becomes sufficiently messed up, a node MAY close the IKE SA, as described above. It can then rebuild the SAs it needs on a clean base under a new IKE SA.

半封闭的 ESP 或 AH 连接是不正常的，如果它们持续存在，具有审核功能的节点可能应该审核它们的存在。请注意，此规范没有指定时间段，因此由各个端点决定等待多长时间。节点可以拒绝接受半关闭连接上的传入数据，但不能单方面关闭它们并重用 SPI。如果连接状态变得足够混乱，则节点可以如上所述关闭 IKE SA。然后，它可以在新 IKE SA 的干净基础上重建所需的 SA。

1.5. Informational Messages outside of an IKE SA

1.5. IKE SA 外部的信息性消息

There are some cases in which a node receives a packet that it cannot process, but it may want to notify the sender about this situation.

在某些情况下，节点接收到无法处理的数据包，但可能希望将此情况通知发送方。

o If an ESP or AH packet arrives with an unrecognized SPI. This might be due to the receiving node having recently crashed and lost state, or because of some other system malfunction or attack.

o 如果 ESP 或 AH 数据包带有无法识别的 SPI。这可能是由于接收节点最近崩溃并失去状态，或者是由于某些其他系统故障或攻击。

o If an encrypted IKE request packet arrives on port 500 or 4500 with an unrecognized IKE SPI. This might be due to the receiving node having recently crashed and lost state, or because of some other system malfunction or attack.

o 如果加密的 IKE 请求数据包到达端口 500 或 4500 时带有无法识别的 IKE SPI。这可能是由于接收节点最近崩溃并失去状态，或者是由于某些其他系统故障或攻击。

o If an IKE request packet arrives with a higher major version number than the implementation supports.

o 如果 IKE 请求数据包到达时的主版本号高于实现支持的版本号。

In the first case, if the receiving node has an active IKE SA to the IP address from whence the packet came, it MAY send an INVALID_SPI notification of the wayward packet over that IKE SA in an INFORMATIONAL exchange. The Notification Data contains the SPI of the invalid packet. The recipient of this notification cannot tell whether the SPI is for AH or ESP, but this is not important because the SPIs are supposed to be different for the two. If no suitable IKE SA exists, the node MAY send an informational message without cryptographic protection to the source IP address, using the source UDP port as the destination port if the packet was UDP (UDP-encapsulated ESP or AH). In this case, it should only be used by the recipient as a hint that something might be wrong (because it could easily be forged). This message is not part of an INFORMATIONAL exchange, and the receiving node MUST NOT respond to it because doing so could cause a message loop. The message is constructed as follows: there are no IKE SPI values that would be meaningful to the recipient of such a notification; using zero values or random values are both acceptable, this being the exception to the rule in Section 3.1 that prohibits zero IKE Initiator SPIs. The Initiator flag is set to 1, the Response flag is set to 0, and the version flags are set in the normal fashion; these flags are described in Section 3.1.

在第一种情况下，如果接收节点具有到数据包来自的 IP 地址的活动 IKE SA，则它可以在信息交换中通过该 IKE SA 发送任意数据包的无效_SPI 通知。通知数据包含无效数据包的 SPI。此通知的接收者无法判断 SPI 是针对 AH 还是 ESP，但这并不重要，因为 SPI 应该与 AH 或 ESP 不同。如果不存在合适的 IKE SA，则节点可以向源 IP 地址发送无加密保护的信息性消息，如果数据包是 UDP（UDP 封装的 ESP 或 AH），则使用源 UDP 端口作为目标端口。在这种情况下，收件人只应将其用作提示可能存在错误（因为它很容易被伪造）。此消息不是信息交换的一部分，接收节点不得响应此消息，因为这样做可能导致消息循环。消息的构造如下：不存在对此类通知的接收者有意义的 IKE SPI 值；使用零值或随机值都是可以接受的，这是第 3.1 节中禁止零 IKE 启动器 SPI 规则的例外。启动器标志设置为 1，响应标志设置为 0，版本标志以正常方式设置；第 3.1 节描述了这些标志。

In the second and third cases, the message is always sent without cryptographic protection (outside of an IKE SA), and includes either an INVALID_IKE_SPI or an INVALID_MAJOR_VERSION notification (with no notification data). The message is a response message, and thus it is sent to the IP address and port from whence it came with the same IKE SPIs and the Message ID and Exchange Type are copied from the request. The Response flag is set to 1, and the version flags are set in the normal fashion.

在第二和第三种情况下，消息总是在没有加密保护的情况下发送（在 IKE SA 之外），并且包含

无效的 IKE SPI 或无效的主版本通知（没有通知数据）。该消息是一个响应消息，因此它被发送到 IP 地址和端口，从那里它与相同的 IKE SPI 一起出现，并且消息 ID 和交换类型从请求中复制。响应标志设置为 1，版本标志以正常方式设置。

1.6. Requirements Terminology

1.6. 需求术语

Definitions of the primitive terms in this document (such as Security Association or SA) can be found in [IPSECARCH]. It should be noted that parts of IKEv2 rely on some of the processing rules in [IPSECARCH], as described in various sections of this document.

本文档中基本术语（如安全关联或 SA）的定义可在[IPSECARCH]中找到。应该注意的是，IKEv2 的某些部分依赖于[IPSECARCH]中的一些处理规则，如本文件各节所述。

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [MUSTSHOULD].

本文件中的关键词“必须”、“不得”、“必需”、“应”、“不应”、“应”、“不应”、“建议”、“可”和“可选”应按照[必须”中的说明进行解释。

1.7. Significant Differences between RFC 4306 and This Document

1.7. RFC 4306 与本文件之间的重大差异

This document contains clarifications and amplifications to IKEv2 [IKEV2]. Many of the clarifications are based on [Clarif]. The changes listed in that document were discussed in the IPsec Working Group and, after the Working Group was disbanded, on the IPsec mailing list. That document contains detailed explanations of areas that were unclear in IKEv2, and is thus useful to implementers of IKEv2.

本文件包含对 IKEv2[IKEv2]的澄清和详述。许多澄清基于[Clarif]。IPsec 工作组讨论了该文件中列出的更改，工作组解散后，IPsec 邮件列表中也讨论了这些更改。该文档包含对 IKEv2 中不清楚的领域的详细解释，因此对 IKEv2 的实现者很有用。

The protocol described in this document retains the same major version number (2) and minor version number (0) as was used in RFC 4306. That is, the version number is *not* changed from RFC 4306. The small number of technical changes listed here are not expected to affect RFC 4306 implementations that have already been deployed at the time of publication of this document.

本文档中描述的协议保留了 RFC 4306 中使用的相同主版本号（2）和次版本号（0）。也就是说，版本号*未*从 RFC 4306 更改。此处列出的少量技术更改预计不会影响本文档发布时已部署的 RFC 4306 实现。

This document makes the figures and references a bit more consistent than they were in [IKEV2].

本文件使图和参考资料比[IKEV2]中的图和参考资料更加一致。

IKEv2 developers have noted that the SHOULD-level requirements in RFC 4306 are often unclear in that they don't say when it is OK to not obey the requirements. They also have noted that there are MUST-level requirements that are not related to interoperability. This document has more explanation of some of these requirements. All non-capitalized uses of the words SHOULD and MUST now mean their normal English sense, not the interoperability sense of [MUSTSHOULD].

IKEv2 开发人员注意到，RFC4306 中的“应该”级别的要求往往不明确，因为他们没有说什么时候可以不遵守这些要求。他们还注意到，存在与互操作性无关的必须级别需求。本文件对其中一些要求进行了更多解释。这些词的所有非大写用法现在应该也必须是指其正常的英语意义，而不是[MUSTSHOULD]的互操作性意义。

IKEv2 (and IKEv1) developers have noted that there is a great deal of material in the tables of codes in Section 3.10.1 in RFC 4306. This leads to implementers not having all the needed information in the main body of the document. Much of the material from those tables has been moved into the associated parts of the main body of the document.

IKEv2（和 IKEv1）开发者注意到 RFC 4306 第 3.10.1 节中的代码表中有大量材料。这导致实现者在文档主体中没有所有需要的信息。这些表格中的大部分资料都已移至文档主体的相关部分。

This document removes discussion of nesting AH and ESP. This was a mistake in RFC 4306 caused by the lag between finishing RFC 4306 and RFC 4301. Basically, IKEv2 is based on RFC 4301, which does not include "SA bundles" that were part of RFC 2401. While a single packet can go through IPsec processing multiple times, each of these passes uses a separate SA, and the passes are coordinated by the forwarding tables. In IKEv2, each of these SAs has to be created using a separate CREATE_CHILD_SA exchange.

本文件删除了嵌套 AH 和 ESP 的讨论。这是 RFC 4306 中的一个错误，是由于 RFC 4306 和

RFC 4301 之间的延迟造成的。基本上，IKEv2 基于 RFC 4301，它不包括作为 RFC 2401 一部分的“SA 捆绑包”。虽然单个数据包可以多次通过 IPsec 处理，但每个过程都使用单独的 SA，并且这些过程由转发表协调。在 IKEv2 中，必须使用单独的 CREATE_CHILD_SA 交换创建每个 SA。

This document removes discussion of the INTERNAL_ADDRESS_EXPIRY configuration attribute because its implementation was very problematic. Implementations that conform to this document MUST

本文档删除了对内部地址过期配置属性的讨论，因为它的实现非常有问题。符合本文档的实现必须

ignore proposals that have configuration attribute type 5, the old value for INTERNAL_ADDRESS_EXPIRY. This document also removed INTERNAL_IP6_NBNS as a configuration attribute.

忽略配置属性类型为 5（内部地址过期的旧值）的提案。本文档还删除了内部 IP6 作为配置属性。

This document removes the allowance for rejecting messages in which the payloads were not in the "right" order; now implementations MUST NOT reject them. This is due to the lack of clarity where the orders for the payloads are described.

本文档取消了拒绝有效负载顺序不正确的消息的允许；现在，实现不能拒绝它们。这是由于对有效载荷订单的描述不够清晰。

The lists of items from RFC 4306 that ended up in the IANA registry were trimmed to only include items that were actually defined in RFC 4306. Also, many of those lists are now preceded with the very important instruction to developers that they really should look at the IANA registry at the time of development because new items have been added since RFC 4306.

RFC 4306 中最终进入 IANA 注册表的项目列表被修剪为仅包括 RFC 4306 中实际定义的项目。此外，许多列表前面都有一条非常重要的指示，告诉开发人员在开发时确实应该查看 IANA 注册表，因为自 RFC 4306 以来已经添加了新项目。

This document adds clarification on when notifications are and are not sent encrypted, depending on the state of the negotiation at the time.

本文档对何时发送和不发送加密通知进行了说明，具体取决于当时的协商状态。

This document discusses more about how to negotiate combined-mode ciphers.

本文将详细讨论如何协商组合模式密码。

In Section 1.3.2, "The KEi payload SHOULD be included" was changed to be "The KEi payload MUST be included". This also led to changes in Section 2.18.

在第 1.3.2 节中，“应包括 KEi 有效载荷”更改为“必须包括 KEi 有效载荷”。这也导致了第 2.18 节的变更。

In Section 2.1, there is new material covering how the initiator's SPI and/or IP is used to differentiate if this is a "half-open" IKE SA or a new request.

在第 2.1 节中，有新材料介绍了如何使用启动器的 SPI 和/或 IP 来区分这是“半开放”IKE SA 还是新请求。

This document clarifies the use of the critical flag in Section 2.5.

本文件澄清了第 2.5 节中临界标志的使用。

In Section 2.8, "Note that, when rekeying, the new Child SA MAY have different Traffic Selectors and algorithms than the old one" was changed to "Note that, when rekeying, the new Child SA SHOULD NOT have different Traffic Selectors and algorithms than the old one".

在第 2.8 节中，“注意，当重新键入时，新的子 SA 可能具有与旧 SA 不同的流量选择器和算法”更改为“注意，当重新键入时，新的子 SA 不应具有与旧 SA 不同的流量选择器和算法”。

The new Section 2.8.2 covers simultaneous IKE SA rekeying.

新的第 2.8.2 节涵盖了同步 IKE SA 密钥更新。

The new Section 2.9.2 covers Traffic Selectors in rekeying.

新的第 2.9.2 节介绍了重设密钥时的流量选择器。

This document adds the restriction in Section 2.13 that all pseudorandom functions (PRFs) used with IKEv2 MUST take variable-sized keys. This should not affect any implementations because there were no standardized PRFs that have fixed-size keys.

本文件在第 2.13 节中增加了限制，即与 IKEv2 一起使用的所有伪随机函数（PRF）必须采用可

变大小的密钥。这不会影响任何实现，因为没有具有固定大小键的标准 PRF。

Section 2.18 requires doing a Diffie-Hellman exchange when rekeying the IKE_SA. In theory, RFC 4306 allowed a policy where the Diffie-Hellman exchange was optional, but this was not useful (or appropriate) when rekeying the IKE_SA.

第 2.18 节要求在重新键入 IKE_SA 时进行 Diffie-Hellman 交换。理论上，RFC4306 允许一种策略，其中 Diffie-Hellman 交换是可选的，但在为 IKE_SA 重新设置密钥时，这是没有用的（或不合适的）。

Section 2.21 has been greatly expanded to cover the different cases where error responses are needed and the appropriate responses to them.

第 2.21 节已大大扩展，以涵盖需要错误响应的不同情况以及相应的响应。

Section 2.23 clarified that, in NAT traversal, now both UDP-encapsulated IPsec packets and non-UDP-encapsulated IPsec packets need to be understood when receiving.

第 2.23 节阐明，在 NAT 遍历中，现在在接收时需要理解 UDP 封装的 IPsec 数据包和非 UDP 封装的 IPsec 数据包。

Added Section 2.23.1 to describe NAT traversal when transport mode is requested.

增加了第 2.23.1 节，以描述请求传输模式时的 NAT 穿越。

Added Section 2.25 to explain how to act when there are timing collisions when deleting and/or rekeying SAs, and two new error notifications (TEMPORARY_FAILURE and CHILD_SA_NOT_FOUND) were defined.

增加了第 2.25 节，以解释在删除和/或重新设置 SAs 时出现计时冲突时如何采取行动，并定义了两个新的错误通知（临时故障和未找到子故障）。

In Section 3.6, "Implementations MUST support the HTTP method for hash-and-URL lookup. The behavior of other URL methods is not currently specified, and such methods SHOULD NOT be used in the absence of a document specifying them" was added.

在第 3.6 节中，添加了“实现必须支持用于哈希和 URL 查找的 HTTP 方法。当前未指定其他 URL 方法的行为，并且在没有指定这些方法的文档的情况下不应使用这些方法”。

In Section 3.15.3, a pointer to a new document that is related to configuration of IPv6 addresses was added.

在第 3.15.3 节中，添加了指向与 IPv6 地址配置相关的新文档的指针。

Appendix C was expanded and clarified.

对附录 C 进行了扩充和澄清。

2. IKE Protocol Details and Variations

2. IKE 协议的细节和变化

IKE normally listens and sends on UDP port 500, though IKE messages may also be received on UDP port 4500 with a slightly different format (see Section 2.23). Since UDP is a datagram (unreliable) protocol, IKE includes in its definition recovery from transmission errors, including packet loss, packet replay, and packet forgery. IKE is designed to function so long as (1) at least one of a series of retransmitted packets reaches its destination before timing out; and (2) the channel is not so full of forged and replayed packets so as to exhaust the network or CPU capacities of either endpoint. Even in the absence of those minimum performance requirements, IKE is designed to fail cleanly (as though the network were broken).

IKE 通常在 UDP 端口 500 上侦听和发送，但 IKE 消息也可能在 UDP 端口 4500 上以稍微不同的格式接收（参见第 2.23 节）。由于 UDP 是一种数据报（不可靠）协议，IKE 在其定义中包括从传输错误中恢复，包括数据包丢失、数据包重放和数据包伪造。IKE 被设计为在（1）一系列重传分组中的至少一个在超时之前到达其目的地时起作用；和（2）信道中不充满伪造和重放的数据包，从而耗尽任一端点的网络或 CPU 容量。即使在没有这些最低性能要求的情况下，IKE 也被设计为完全失败（就像网络断开一样）。

Although IKEv2 messages are intended to be short, they contain structures with no hard upper bound on size (in particular, digital certificates), and IKEv2 itself does not have a mechanism for

尽管 IKEv2 消息旨在简短，但它们包含的结构在大小上没有硬上限（特别是数字证书），并且 IKEv2 本身没有用于

fragmenting large messages. IP defines a mechanism for fragmentation of oversized UDP messages, but implementations vary in the maximum message size supported. Furthermore, use of IP fragmentation opens an implementation to denial-of-service (DoS) attacks [DOSUDPPROT]. Finally, some NAT and/or firewall implementations may block IP fragments.

分割大消息。IP 定义了一种对超大 UDP 消息进行分段的机制，但实现在支持的最大消息大小上有所不同。此外，IP 碎片的使用为拒绝服务（DoS）攻击[DOSUDPPROT]打开了一个实现。最后，一些 NAT 和/或防火墙实现可能会阻止 IP 片段。

All IKEv2 implementations MUST be able to send, receive, and process IKE messages that are up to 1280 octets long, and they SHOULD be able to send, receive, and process messages that are up to 3000 octets long. IKEv2 implementations need to be aware of the maximum UDP message size supported and MAY shorten messages by leaving out some certificates or cryptographic suite proposals if that will keep messages below the maximum. Use of the "Hash and URL" formats rather than including certificates in exchanges where possible can avoid most problems. Implementations and configuration need to keep in mind, however, that if the URL lookups are possible only after the Child SA is established, recursion issues could prevent this technique from working.

所有 IKEv2 实现必须能够发送、接收和处理长达 1280 个八位字节的 IKE 消息，并且应该能够发送、接收和处理长达 3000 个八位字节的消息。IKEv2 实现需要知道所支持的最大 UDP 消息大小，并且可能会通过省略一些证书或加密套件建议来缩短消息，如果这样会使消息低于最大值。使用“哈希和 URL”格式，而不是尽可能在交换中包含证书，可以避免大多数问题。但是，实现和配置需要记住，如果 URL 查找仅在子 SA 建立之后才可能进行，则递归问题可能会阻止此技术的工作。

The UDP payload of all packets containing IKE messages sent on port 4500 MUST begin with the prefix of four zeros; otherwise, the receiver won't know how to handle them.

在端口 4500 上发送的包含 IKE 消息的所有数据包的 UDP 有效负载必须以四个零的前缀开头；否则，接收者将不知道如何处理它们。

2.1. Use of Retransmission Timers

2.1. 重传定时器的使用

All messages in IKE exist in pairs: a request and a response. The setup of an IKE SA normally consists of two exchanges. Once the IKE SA is set up, either end of the Security Association may initiate requests at any time, and there can be many requests and responses "in flight" at any given moment. But each message is labeled as either a request or a response, and for each exchange, one end of the Security Association is the initiator and the other is the responder.

IKE 中的所有消息成对存在：请求和响应。IKE SA 的设置通常由两个交换机组成。一旦 IKE SA 建立起来，安全关联的任何一端都可以在任何时候发起请求，并且在任何给定时刻都可能有许多请求和响应处于“飞行”状态。但是每个消息都被标记为请求或响应，对于每个交换，安全关联的一端是发起方，另一端是响应方。

For every pair of IKE messages, the initiator is responsible for retransmission in the event of a timeout. The responder MUST never retransmit a response unless it receives a retransmission of the request. In that event, the responder MUST ignore the retransmitted request except insofar as it causes a retransmission of the response. The initiator MUST remember each request until it receives the corresponding response. The responder MUST remember each response until it receives a request whose sequence number is larger than or equal to the sequence number in the response plus its window size (see Section 2.3). In order to allow saving memory, responders are allowed to forget the response after a timeout of several minutes. If the responder receives a retransmitted request for which it has already forgotten the response, it MUST ignore the request (and not, for example, attempt constructing a new response).

对于每对 IKE 消息，发起方负责在超时情况下重新传输。响应者不得重新传输响应，除非它收到请求的重新传输。在这种情况下，响应者必须忽略重新传输的请求，除非它导致响应的重新传输。启动器必须记住每个请求，直到收到相应的响应。响应者必须记住每个响应，直到它收到一个序列号大于或等于响应中的序列号加上其窗口大小的请求（见第 2.3 节）。为了节省内存，允许响应者在超时几分钟后忘记响应。如果响应者接收到已忘记响应的重传请求，则必须忽略该请求（例如，不要尝试构造新响应）。

IKE is a reliable protocol: the initiator MUST retransmit a request until it either receives a corresponding response or deems the IKE SA to have failed. In the latter case, the initiator discards all state associated with the IKE SA and any Child SAs that were negotiated using that IKE SA. A retransmission from the initiator MUST be bitwise identical to the original request. That is, everything starting from the IKE header (the IKE SA initiator's SPI onwards) must be bitwise identical; items before it (such as the IP and UDP headers) do not have to be identical.

IKE 是一种可靠的协议：启动器必须重新传输请求，直到收到相应的响应或认为 IKE SA 失败为止。在后一种情况下，启动器丢弃与 IKE SA 关联的所有状态以及使用该 IKE SA 协商的任何子 SA。发起者的重传必须与原始请求按位相同。也就是说，从 IKE 头开始的所有内容（IKE SA 启动器的 SPI 之后）必须是按位相同的；之前的项目（如 IP 和 UDP 标头）不必相同。

Retransmissions of the IKE_SA_INIT request require some special handling. When a responder receives an IKE_SA_INIT request, it has to determine whether the packet is a retransmission belonging to an existing "half-open" IKE SA (in which case the responder retransmits the same response), or a new request (in which case the responder creates a new IKE SA and sends a fresh response), or it belongs to an existing IKE SA where the IKE_AUTH request has been already received (in which case the responder ignores it).

IKE_SA_INIT 请求的重新传输需要一些特殊处理。当响应者接收到 IKE_SA_INIT 请求时，它必须确定该分组是属于现有“半开放”IKE SA 的重传（在这种情况下，响应者重传相同的响应）还是新请求（在这种情况下，响应者创建新的 IKE SA 并发送新的响应），或者它属于已经收到 IKE_AUTH 请求的现有 IKE SA（在这种情况下，响应者会忽略它）。

It is not sufficient to use the initiator's SPI and/or IP address to differentiate between these three cases because two different peers behind a single NAT could choose the same initiator SPI. Instead, a robust responder will do the IKE SA lookup using the whole packet, its hash, or the Ni payload.

仅使用启动器的 SPI 和/或 IP 地址来区分这三种情况是不够的，因为单个 NAT 后面的两个不同对等方可以选择相同的启动器 SPI。相反，健壮的响应程序将使用整个数据包、其哈希或 Ni 有效负载执行 IKE SA 查找。

The retransmission policy for one-way messages is somewhat different from that for regular messages. Because no acknowledgement is ever sent, there is no reason to gratuitously retransmit one-way messages. Given that all these messages are errors, it makes sense to send them only once per "offending" packet, and only retransmit if further offending packets are received. Still, it also makes sense to limit retransmissions of such error messages.

单向消息的重传策略与常规消息的重传策略有些不同。因为从未发送过确认，所以没有理由无偿重新传输单向消息。考虑到所有这些消息都是错误，每个“违规”数据包只发送一次消息是有意义的，并且只有在收到更多违规数据包时才重新传输。不过，限制此类错误消息的重新传输也是有意义的。

2.2. Use of Sequence Numbers for Message ID

2.2. 对消息 ID 使用序列号

Every IKE message contains a Message ID as part of its fixed header. This Message ID is used to match up requests and responses and to identify retransmissions of messages. Retransmission of a message MUST use the same Message ID as the original message.

每个 IKE 消息都包含一个消息 ID 作为其固定头的一部分。此消息 ID 用于匹配请求和响应，并标识消息的重新传输。消息的重新传输必须使用与原始消息相同的消息 ID。

The Message ID is a 32-bit quantity, which is zero for the IKE_SA_INIT messages (including retries of the message due to responses such as COOKIE and INVALID_KE_PAYLOAD), and incremented for each subsequent exchange. Thus, the first pair of IKE_AUTH messages will have an ID of 1, the second (when EAP is used) will be 2, and so on. The Message ID is reset to zero in the new IKE SA after the IKE SA is rekeyed.

消息 ID 是一个 32 位的数量，对于 IKE_SA_INIT 消息（包括由于 COOKIE 和无效的_KE_负载等响应而重试消息），它为零，并且对于每个后续的交换递增。因此，第一对 IKE_AUTH 消息的 ID 为 1，第二对（使用 EAP 时）的 ID 为 2，依此类推。在 IKE SA 重新设置密钥后，新 IKE SA 中的消息 ID 重置为零。

Each endpoint in the IKE Security Association maintains two "current" Message IDs: the next one to be used for a request it initiates and the next one it expects to see in a request from the other end. These counters increment as requests are generated and received. Responses always contain the same Message ID as the corresponding request. That means that after the initial exchange, each integer n may appear as the Message ID in four distinct messages: the nth request from the original IKE initiator, the corresponding response, the nth request from the original IKE responder, and the corresponding response. If the two ends make a very different number of requests, the Message IDs in the two directions can be very different. There is no ambiguity in the messages, however, because the Initiator and Response flags in the message header specify which of the four messages a particular one is.

IKE 安全关联中的每个端点维护两个“当前”消息 ID：下一个用于它发起的请求，下一个用于它期望在另一端的请求中看到的。这些计数器随着请求的生成和接收而增加。响应始终包含与相应请求相同的消息 ID。这意味着在初始交换之后，每个整数 n 可以作为消息 ID 出现在四个不同的消息中：来自原始 IKE 发起方的第 n 个请求、相应的响应、来自原始 IKE 响应方的第 n 个请求以及相应的响应。如果两端发出的请求数量非常不同，则两个方向上的消息 ID 可能会非常不同。但是，消息中没有歧义，因为消息头中的启动器和响应标志指定了四条消息中的哪一条是特定消息。

Throughout this document, "initiator" refers to the party who initiated the exchange being described. The "original initiator" always refers to the party who initiated the exchange that resulted in the current IKE SA. In other words, if the "original responder" starts rekeying the IKE SA, that party becomes the "original initiator" of the new IKE SA.

在本文件中，“发起人”指发起所述交易的一方。“原始发起人”始终指发起导致当前 IKE SA 的交换的一方。换句话说，如果“原始响应者”开始为 IKE SA 重新键入密钥，则该方成为新 IKE SA 的“原始发起人”。

Note that Message IDs are cryptographically protected and provide protection against message replays. In the unlikely event that Message IDs grow too large to fit in 32 bits, the IKE SA MUST be closed or rekeyed.

请注意，消息 ID 受到加密保护，并提供防止消息重播的保护。如果消息 ID 变得太大而无法容纳 32 位，则必须关闭或重新设置 IKE SA 密钥。

2.3. Window Size for Overlapping Requests

2.3. 重叠请求的窗口大小

The SET_WINDOW_SIZE notification asserts that the sending endpoint is capable of keeping state for multiple outstanding exchanges, permitting the recipient to send multiple requests before getting a response to the first. The data associated with a SET_WINDOW_SIZE notification MUST be 4 octets long and contain the big endian representation of the number of messages the sender promises to keep. The window size is always one until the initial exchanges complete.

SET_WINDOW_SIZE 通知声明发送端点能够保持多个未完成交换的状态，允许接收方在获得对第一个交换的响应之前发送多个请求。与 SET_WINDOW_SIZE 通知关联的数据必须为 4 个八位字节长，并且包含发送方承诺保留的消息数量的大端表示。在初始交换完成之前，窗口大小始终为一。

An IKE endpoint MUST wait for a response to each of its messages before sending a subsequent message unless it has received a SET_WINDOW_SIZE Notify message from its peer informing it that the peer is prepared to maintain state for multiple outstanding messages in order to allow greater throughput.

IKE 端点在发送后续消息之前必须等待对其每条消息的响应，除非它已从其对等方接收到一条 SET_WINDOW_SIZE Notify 消息，通知其对等方已准备好维护多条未完成消息的状态，以便允许更大的吞吐量。

After an IKE SA is set up, in order to maximize IKE throughput, an IKE endpoint MAY issue multiple requests before getting a response to any of them, up to the limit set by its peer's SET_WINDOW_SIZE. These requests may pass one another over the network. An IKE endpoint MUST be prepared to accept and process a request while it

在设置 IKE SA 之后，为了最大限度地提高 IKE 吞吐量，IKE 端点可以在获得对其中任何一个请求的响应之前发出多个请求，直到其对等方的 set_WINDOW_大小设置的限制。这些请求可以通过网络相互传递。IKE 端点必须准备好接受和处理请求，同时

has a request outstanding in order to avoid a deadlock in this situation. An IKE endpoint may also accept and process multiple requests while it has a request outstanding.

有一个未完成的请求，以避免在这种情况下出现死锁。当 IKE 端点有一个未完成的请求时，它也可以接受和处理多个请求。

An IKE endpoint MUST NOT exceed the peer's stated window size for transmitted IKE requests. In other words, if the responder stated its window size is N, then when the initiator needs to make a request X, it MUST wait until it has received responses to all requests up through request X-N. An IKE endpoint MUST keep a copy of (or be able to regenerate exactly) each request it has sent until it receives the corresponding response. An IKE endpoint MUST keep a copy of (or be able to regenerate exactly) the number of previous responses equal to its declared window size in case its response was lost and the initiator requests its retransmission by retransmitting the request.

IKE 端点对于传输的 IKE 请求不得超过对等方规定的窗口大小。换句话说，如果响应者声明其窗口大小为 N，则当发起方需要发出请求 X 时，它必须等待直到通过请求 X-N 收到所有请求的响应。IKE 端点必须保留其发送的每个请求的副本（或能够准确地重新生成），直到收到相应的响应。IKE 端点必须保留（或能够准确地重新生成）先前响应数量的副本，该数量等于其声明的窗口大小，以防其响应丢失，并且启动器通过重新传输请求来请求其重新传输。

An IKE endpoint supporting a window size greater than one ought to be capable of processing incoming requests out of order to maximize performance in the event of network failures or packet reordering.

支持大于 1 的窗口大小的 IKE 端点应该能够处理无序的传入请求，以便在发生网络故障或数据包重新排序时最大限度地提高性能。

The window size is normally a (possibly configurable) property of a particular implementation, and is not related to congestion control (unlike the window size in TCP, for example). In particular, what the responder should do when it receives a

SET_WINDOW_SIZE notification containing a smaller value than is currently in effect is not defined. Thus, there is currently no way to reduce the window size of an existing IKE SA; you can only increase it. When rekeying an IKE SA, the new IKE SA starts with window size 1 until it is explicitly increased by sending a new SET_WINDOW_SIZE notification.

窗口大小通常是特定实现的一个属性（可能是可配置的），与拥塞控制无关（例如，与 TCP 中的窗口大小不同）。特别是，未定义响应程序在收到包含小于当前有效值的 SET_WINDOW_SIZE 通知时应执行的操作。因此，目前没有办法减小现有 IKE SA 的窗口大小；你只能增加它。在为 IKE SA 重新设置密钥时，新的 IKE SA 从窗口大小 1 开始，直到通过发送新的 SET_window_size 通知而显式增大为止。

The INVALID_MESSAGE_ID notification is sent when an IKE Message ID outside the supported window is received. This Notify message MUST NOT be sent in a response; the invalid request MUST NOT be acknowledged. Instead, inform the other side by initiating an INFORMATIONAL exchange with Notification data containing the four-octet invalid Message ID. Sending this notification is OPTIONAL, and notifications of this type MUST be rate limited.

接收到支持窗口外的 IKE 消息 ID 时，将发送无效消息 ID 通知。此通知消息不得在响应中发送；不能确认无效的请求。相反，通过与包含四个八位字节无效消息 ID 的通知数据进行信息交换来通知另一方。发送此通知是可选的，并且此类型的通知必须是速率受限的。

2.4. State Synchronization and Connection Timeouts

2.4. 状态同步和连接超时

An IKE endpoint is allowed to forget all of its state associated with an IKE SA and the collection of corresponding Child SAs at any time. This is the anticipated behavior in the event of an endpoint crash and restart. It is important when an endpoint either fails or reinitializes its state that the other endpoint detect those conditions and not continue to waste network bandwidth by sending packets over discarded SAs and having them fall into a black hole.

允许 IKE 端点随时忘记与 IKE SA 关联的所有状态以及相应子 SA 的集合。这是端点崩溃和重新启动时的预期行为。当一个端点出现故障或重新初始化其状态时，另一个端点必须检测到这些情况，并且不要通过在丢弃的 SA 上发送数据包并使其落入黑洞而继续浪费网络带宽，这一点很重要。

The INITIAL_CONTACT notification asserts that this IKE SA is the only IKE SA currently active between the authenticated identities. It MAY be sent when an IKE SA is established after a crash, and the recipient MAY use this information to delete any other IKE SAs it has to the same authenticated identity without waiting for a timeout. This notification MUST NOT be sent by an entity that may be replicated (e.g., a roaming user's credentials where the user is allowed to connect to the corporate firewall from two remote systems at the same time). The INITIAL_CONTACT notification, if sent, MUST be in the first IKE_AUTH request or response, not as a separate exchange afterwards; receiving parties MAY ignore it in other messages.

初始_联系人通知断言此 IKE SA 是在经过身份验证的身份之间当前处于活动状态的唯一 IKE SA。它可以在崩溃后建立 IKE SA 时发送，并且接收者可以使用此信息删除其具有相同身份验证的任何其他 IKE SA，而无需等待超时。此通知不得由可复制的实体发送（例如，允许用户同时从两个远程系统连接到公司防火墙的漫游用户凭据）。初始联系人通知（如果发送）必须在第一个 IKE_认证请求或响应中，而不是在之后作为单独的交换；接收方可以在其他消息中忽略它。

Since IKE is designed to operate in spite of DoS attacks from the network, an endpoint MUST NOT conclude that the other endpoint has failed based on any routing information (e.g., ICMP messages) or IKE messages that arrive without cryptographic protection (e.g., Notify messages complaining about unknown SPIs). An endpoint MUST conclude that the other endpoint has failed only when repeated attempts to contact it have gone unanswered for a timeout period or when a cryptographically protected INITIAL_CONTACT notification is received on a different IKE SA to the same authenticated identity. An endpoint should suspect that the other endpoint has failed based on routing information and initiate a request to see whether the other endpoint is alive. To check whether the other side is alive, IKE specifies an empty INFORMATIONAL message that (like all IKE requests) requires an acknowledgement (note that within the context of an IKE SA, an "empty" message consists of an IKE header followed by an Encrypted payload that contains no payloads). If a cryptographically protected (fresh, i.e., not retransmitted) message has been received from the other side recently, unprotected Notify messages MAY be ignored. Implementations MUST limit the rate at which they take actions based on unprotected messages.

由于 IKE 被设计为即使来自网络的 DoS 攻击也能运行，因此端点不得基于任何路由信息（例如 ICMP 消息）或到达时没有加密保护的 IKE 消息（例如，通知消息抱怨未知 SPI）断定另一个端点发生故障。一个端点必须得出结论，只有在重复尝试联系另一个端点的超时时间内无人应答时，或者在不同 IKE SA 上接收到具有相同身份验证的受密码保护的初始\u 联系通知时，另一个端点才失败。端点应根据路由信息怀疑另一个端点已失败，并启动请求以查看另一个端点是否处于活动状态。为了检查另一方是否处于活动状态，IKE 指定一条空的信息性消息，该消息（与所有

IKE 请求一样）需要确认（注意，在 IKE SA 的上下文中，“空”消息由 IKE 头和不包含有效负载的加密有效负载组成）。如果最近已从另一方接收到加密保护（新的，即未重新传输）消息，则可能会忽略未保护的通知消息。实现必须限制基于未受保护的消息采取操作的速率。

The number of retries and length of timeouts are not covered in this specification because they do not affect interoperability. It is suggested that messages be retransmitted at least a dozen times over a period of at least several minutes before giving up on an SA, but different environments may require different rules. To be a good network citizen, retransmission times MUST increase exponentially to avoid flooding the network and making an existing congestion situation worse. If there has only been outgoing traffic on all of the SAs associated with an IKE SA, it is essential to confirm liveness of the other endpoint to avoid black holes. If no cryptographically protected messages have been received on an IKE SA or any of its Child SAs recently, the system needs to perform a liveness check in order to prevent sending messages to a dead peer. (This is sometimes called "dead peer detection" or "DPD", although it

重试次数和超时长度不在本规范中，因为它们不影响互操作性。建议在放弃 SA 之前，在至少几分钟的时间内至少重新传输十几次消息，但不同的环境可能需要不同的规则。要成为一个好的网络公民，重传时间必须成倍增加，以避免网络泛滥，并使现有的拥塞情况恶化。如果与 IKE SA 关联的所有 SA 上只有传出流量，则必须确认另一个端点的活动性以避免黑洞。如果 IKE SA 或其任何子 SA 最近未收到任何受加密保护的消息，则系统需要执行活动性检查，以防止向死对等方发送消息。（这有时被称为“死点检测”或“DPD”，尽管

is really detecting live peers, not dead ones.) Receipt of a fresh cryptographically protected message on an IKE SA or any of its Child SAs ensures liveness of the IKE SA and all of its Child SAs. Note that this places requirements on the failure modes of an IKE endpoint. An implementation needs to stop sending over any SA if some failure prevents it from receiving on all of the associated SAs. If a system creates Child SAs that can fail independently from one another without the associated IKE SA being able to send a delete message, then the system MUST negotiate such Child SAs using separate IKE SAs.

在 IKE SA 或其任何子 SA 上接收新的加密保护消息可确保 IKE SA 及其所有子 SA 的活跃性。请注意，这对 IKE 端点的故障模式提出了要求。如果某个故障阻止实现在所有相关 SA 上接收，则实现需要停止通过任何 SA 发送。如果系统创建的子 SA 彼此独立失败，而关联的 IKE SA 无法发送删除消息，则系统必须使用单独的 IKE SA 协商此类子 SA。

There is a DoS attack on the initiator of an IKE SA that can be avoided if the initiator takes the proper care. Since the first two messages of an SA setup are not cryptographically protected, an attacker could respond to the initiator's message before the genuine responder and poison the connection setup attempt. To prevent this, the initiator MAY be willing to accept multiple responses to its first message, treat each as potentially legitimate, respond to it, and then discard all the invalid half-open connections when it receives a valid cryptographically protected response to any one of its requests. Once a cryptographically valid response is received, all subsequent responses should be ignored whether or not they are cryptographically valid.

IKE SA 的启动器上存在 DoS 攻击，如果启动器采取适当措施，则可以避免该攻击。由于 SA 设置的前两条消息没有加密保护，攻击者可能会在真正的响应者之前响应启动器的消息，并毒害连接设置尝试。为了防止这种情况，发起方可能愿意接受对其第一条消息的多个响应，将每个响应视为可能合法的，对其进行响应，然后在接收到对其任何一个请求的有效加密保护响应时丢弃所有无效的半开连接。一旦接收到加密有效的响应，则应忽略所有后续响应，无论它们是否加密有效。

Note that with these rules, there is no reason to negotiate and agree upon an SA lifetime. If IKE presumes the partner is dead, based on repeated lack of acknowledgement to an IKE message, then the IKE SA and all Child SAs set up through that IKE SA are deleted.

请注意，根据这些规则，没有理由就 SA 生命周期进行协商和达成一致。如果 IKE 基于反复缺少对 IKE 消息的确认而假定伙伴已死亡，则删除 IKE SA 和通过该 IKE SA 设置的所有子 SA。

An IKE endpoint may at any time delete inactive Child SAs to recover resources used to hold their state. If an IKE endpoint chooses to delete Child SAs, it MUST send Delete payloads to the other end notifying it of the deletion. It MAY similarly time out the IKE SA. Closing the IKE SA implicitly closes all associated Child SAs. In this case, an IKE endpoint SHOULD send a Delete payload indicating that it has closed the IKE SA unless the other endpoint is no longer responding.

IKE 端点可以随时删除非活动子 SA，以恢复用于保持其状态的资源。如果 IKE 端点选择删除子 SA，它必须向另一端发送删除有效负载，通知其删除。类似地，它可能会使 IKE SA 超时。关闭 IKE SA 将隐式关闭所有关联的子 SA。在这种情况下，除非另一个端点不再响应，否则 IKE 端点应发送一个删除有效负载，指示它已关闭 IKE SA。

2.5. Version Numbers and Forward Compatibility

2.5. 版本号和向前兼容性

This document describes version 2.0 of IKE, meaning the major version number is 2 and the minor version number is 0. This document is a replacement for [IKEV2]. It is likely that some implementations will want to support version 1.0 and version 2.0, and in the future, other versions.

本文档描述了 IKE 的 2.0 版，即主版本号为 2，次版本号为 0。本文件取代[IKEV2]。有些实现可能希望支持 1.0 版和 2.0 版，将来还可能支持其他版本。

The major version number should be incremented only if the packet formats or required actions have changed so dramatically that an older version node would not be able to interoperate with a newer version node if it simply ignored the fields it did not understand and took the actions specified in the older specification. The minor version number indicates new capabilities, and MUST be ignored by a node with a smaller minor version number, but used for informational purposes by the node with the larger minor version number. For example, it might indicate the ability to process a newly defined Notify message type. The node with the larger minor version number would simply note that its correspondent would not be able to understand that message and therefore would not send it.

只有当数据包格式或所需操作发生了巨大变化，以至于旧版本节点无法与新版本节点进行互操作时（如果它忽略了不理解的字段并采取了旧规范中指定的操作），主版本号才应增加。次要版本号表示新功能，次要版本号较小的节点必须忽略该功能，但次要版本号较大的节点用于提供信息。例如，它可能指示处理新定义的通知消息类型的能力。较小版本号较大的节点只会注意到其对应者无法理解该消息，因此不会发送该消息。

If an endpoint receives a message with a higher major version number, it MUST drop the message and SHOULD send an unauthenticated Notify message of type INVALID_MAJOR_VERSION containing the highest (closest) version number it supports. If an endpoint supports major version n, and major version m, it MUST support all versions between n and m. If it receives a message with a major version that it supports, it MUST respond with that version number. In order to prevent two nodes from being tricked into corresponding with a lower major version number than the maximum that they both support, IKE has a flag that indicates that the node is capable of speaking a higher major version number.

如果端点接收到具有更高主版本号的消息，则必须删除该消息，并应发送类型为 INVALID_major_version 的未经验证的通知消息，其中包含其支持的最高（最接近）版本号。

如果端点支持主版本 n 和主版本 m，则它必须支持 n 和 m 之间的所有版本。如果它收到一条包含其支持的主要版本的消息，则必须使用该版本号进行响应。为了防止两个节点被欺骗，使其对应的主版本号低于它们都支持的最大值，IKE 有一个标志，指示该节点能够说出更高的主版本号。

Thus, the major version number in the IKE header indicates the version number of the message, not the highest version number that the transmitter supports. If the initiator is capable of speaking versions n, n+1, and n+2, and the responder is capable of speaking versions n and n+1, then they will negotiate speaking n+1, where the initiator will set a flag indicating its ability to speak a higher version. If they mistakenly (perhaps through an active attacker sending error messages) negotiate to version n, then both will notice that the other side can support a higher version number, and they MUST break the connection and reconnect using version n+1.

因此，IKE 头中的主版本号表示消息的版本号，而不是发送器支持的最高版本号。如果发起者能够讲 n、n+1 和 n+2 版本，而响应者能够讲 n 和 n+1 版本，则他们将协商讲 n+1 版本，发起者将设置一个标志，指示其讲更高版本的能力。如果他们错误地（可能是通过主动攻击者发送错误消息）协商到版本 n，那么双方都会注意到对方可以支持更高的版本号，并且他们必须断开连接并使用版本 n+1 重新连接。

Note that IKEv1 does not follow these rules, because there is no way in v1 of noting that you are capable of speaking a higher version number. So an active attacker can trick two v2-capable nodes into speaking v1. When a v2-capable node negotiates down to v1, it should note that fact in its logs.

请注意，IKEv1 不遵循这些规则，因为在 v1 中无法注意到您能够说出更高的版本号。因此，主动攻击者可以诱使两个支持 v2 的节点说出 v1。当一个支持 v2 的节点向下协商到 v1 时，它应该在其日志中注意这一事实。

Also, for forward compatibility, all fields marked RESERVED MUST be set to zero by an implementation running version 2.0, and their content MUST be ignored by an implementation running version 2.0 ("Be conservative in what you send and liberal in what you receive" [IP]). In this way, future versions of the protocol can use those fields in a way that is guaranteed to be ignored by implementations that do not

此外，为了向前兼容，运行版本 2.0 的实现必须将所有标记为保留的字段设置为零，运行版本 2.0 的实现必须忽略这些字段的内容（“发送内容要保守，接收内容要自由”[IP]）。通过这种方式，协议的未来版本可以以一种保证不会被不使用这些字段的实现忽略的方式使用这些字段

understand them. Similarly, payload types that are not defined are reserved for future use; implementations of a version where they are undefined MUST skip over those payloads and ignore their contents.

理解他们。类似地，未定义的有效载荷类型保留供将来使用；未定义的版本的实现必须跳过这些有效负载并忽略其内容。

IKEv2 adds a "critical" flag to each payload header for further flexibility for forward compatibility. If the critical flag is set and the payload type is unrecognized, the message MUST be rejected and the response to the IKE request containing that payload MUST include a Notify payload UNSUPPORTED_CRITICAL_PAYLOAD, indicating an unsupported critical payload was included. In that Notify payload, the notification data contains the one-octet payload type. If the critical flag is not set and the payload type is unsupported, that payload MUST be ignored. Payloads sent in IKE response messages MUST NOT have the critical flag set. Note that the critical flag applies only to the payload type, not the contents. If the payload type is recognized, but the payload contains something that is not (such as an unknown transform inside an SA payload, or an unknown Notify Message Type inside a Notify payload), the critical flag is ignored.

IKEv2 为每个有效负载头添加了一个“关键”标志，以进一步提高向前兼容性的灵活性。如果设置了 critical（关键）标志且无法识别有效负载类型，则必须拒绝该消息，并且对包含该有效负载的 IKE 请求的响应必须包括 Notify payload UNSUPPORTED_critical_有效负载，指示包含了不受支持的关键有效负载。在该通知有效负载中，通知数据包含一个八位字节的有效负载类型。如果未设置临界标志且有效负载类型不受支持，则必须忽略该有效负载。IKE 响应消息中发送的有效负载不得设置临界标志。请注意，临界标志仅适用于有效负载类型，而不适用于内容。如果有效负载类型已识别，但有效负载包含未识别的内容（如 SA 有效负载内的未知转换，或 Notify 有效负载内的未知 Notify 消息类型），则忽略临界标志。

Although new payload types may be added in the future and may appear interleaved with the fields defined in this specification, implementations SHOULD send the payloads defined in this specification in the order shown in the figures in Sections 1 and 2; implementations MUST NOT reject as invalid a message with those payloads in any other order.

尽管将来可能会添加新的有效载荷类型，并且可能与本规范中定义的字段交叉出现，但实现应按照第 1 节和第 2 节中图中所示的顺序发送本规范中定义的有效载荷；实现不能以任何其他顺序拒绝具有这些有效负载的消息。

2.6. IKE SA SPIs and Cookies

2.6. IKE SA SPIs 和饼干

The initial two eight-octet fields in the header, called the "IKE SPIs", are used as a connection identifier at the beginning of IKE packets. Each endpoint chooses one of the two SPIs and MUST choose them so as to be unique identifiers of an IKE SA. An SPI value of zero is special: it indicates that the remote SPI value is not yet known by the sender.

报头中最初的两个八位字节字段（称为“IKE SPI”）用作 IKE 数据包开头的连接标识符。每个端点选择两个 SPI 中的一个，并且必须选择它们，以便成为 IKE SA 的唯一标识符。SPI 值为零是特殊的：它表示发送方尚未知道远程 SPI 值。

Incoming IKE packets are mapped to an IKE SA only using the packet's SPI, not using (for example) the source IP address of the packet.

传入的 IKE 数据包仅使用数据包的 SPI 映射到 IKE SA，而不使用（例如）数据包的源 IP 地址。

Unlike ESP and AH where only the recipient's SPI appears in the header of a message, in IKE the sender's SPI is also sent in every message. Since the SPI chosen by the original initiator of the IKE SA is always sent first, an endpoint with multiple IKE SAs open that wants to find the appropriate IKE SA using the SPI it assigned must look at the Initiator flag in the header to determine whether it assigned the first or the second eight octets.

与 ESP 和 AH 不同，在 ESP 和 AH 中，只有收件人的 SPI 出现在消息头中，在 IKE 中，发件人的 SPI 也会在每条消息中发送。由于 IKE SA 的原始启动器选择的 SPI 始终首先发送，因此，如果要使用分配的 SPI 查找适当的 IKE SA，则打开多个 IKE SA 的端点必须查看头中的启动器标志，以确定其分配的是第一个八位字节还是第二个八位字节。

In the first message of an initial IKE exchange, the initiator will not know the responder's SPI value and will therefore set that field to zero. When the IKE_SA_INIT exchange does not result in the creation of an IKE SA due to INVALID_KE_PAYLOAD, NO_PROPOSAL_CHOSEN, or COOKIE (see Section 2.6), the responder's SPI will be zero also in the response message. However, if the responder sends a non-zero responder SPI, the initiator should not reject the response for only that reason.

在初始 IKE 交换的第一条消息中，发起方将不知道响应方的 SPI 值，因此将该字段设置为零。当 IKE_SA_INIT 交换由于无效的 IKE_有效负载、未选择提案或 COOKIE（参见第 2.6 节）而未导致 IKE SA 的创建时，响应者的 SPI 在响应消息中也将为零。但是，如果响应程序发送非零响应程序

SPI，则发起程序不应仅出于该原因拒绝响应。

Two expected attacks against IKE are state and CPU exhaustion, where the target is flooded with session initiation requests from forged IP addresses. These attacks can be made less effective if a responder uses minimal CPU and commits no state to an SA until it knows the initiator can receive packets at the address from which it claims to be sending them.

针对 IKE 的两种预期攻击是状态攻击和 CPU 耗尽攻击，其中来自伪造 IP 地址的会话启动请求充斥目标。如果响应程序使用最少的 CPU，并且在知道启动器可以在其声称发送数据包的地址接收数据包之前，不向 SA 提交任何状态，则这些攻击可能会降低效率。

When a responder detects a large number of half-open IKE SAs, it SHOULD reply to IKE_SA_INIT requests with a response containing the COOKIE notification. The data associated with this notification MUST be between 1 and 64 octets in length (inclusive), and its generation is described later in this section. If the IKE_SA_INIT response includes the COOKIE notification, the initiator MUST then retry the IKE_SA_INIT request, and include the COOKIE notification containing the received data as the first payload, and all other payloads unchanged. The initial exchange will then be as follows:

当响应程序检测到大量半开放 IKE SA 时，它应该使用包含 COOKIE 通知的响应来响应 IKE_SA_INIT 请求。与此通知关联的数据长度必须在 1 到 64 个八位字节之间（包括 1 到 64 个八位字节），其生成将在本节后面介绍。如果 IKE_SA_INIT 响应包含 COOKIE 通知，则启动器必须重试 IKE_SA_INIT 请求，并将包含接收数据的 COOKIE 通知作为第一个有效负载，所有其他有效负载保持不变。初始交换将如下所示：

```
Initiator                         Responder
-------------------------------------------------------------------
HDR(A,0), SAi1, KEi, Ni  -->
                            <--  HDR(A,0), N(COOKIE)
HDR(A,0), N(COOKIE), SAi1,
    KEi, Ni  -->
                            <--  HDR(A,B), SAr1, KEr,
                                     Nr, [CERTREQ]
HDR(A,B), SK {IDi, [CERT,]
    [CERTREQ,] [IDr,] AUTH,
    SAi2, TSi, TSr}  -->
                            <--  HDR(A,B), SK {IDr, [CERT,]
                                     AUTH, SAr2, TSi, TSr}
```

```
Initiator                         Responder
-------------------------------------------------------------------
HDR(A,0), SAi1, KEi, Ni  -->
                          <--  HDR(A,0), N(COOKIE)
HDR(A,0), N(COOKIE), SAi1,
    KEi, Ni  -->
                          <--  HDR(A,B), SAr1, KEr,
                                   Nr, [CERTREQ]
HDR(A,B), SK {IDi, [CERT,]
    [CERTREQ,] [IDr,] AUTH,
    SAi2, TSi, TSr}  -->
                          <--  HDR(A,B), SK {IDr, [CERT,]
                                   AUTH, SAr2, TSi, TSr}
```

The first two messages do not affect any initiator or responder state except for communicating the cookie. In particular, the message sequence numbers in the first four messages will all be zero and the message sequence numbers in the last two messages will be one. 'A' is the SPI assigned by the initiator, while 'B' is the SPI assigned by the responder.

前两条消息不影响任何发起方或响应方的状态，但与 cookie 通信除外。特别是，前四条消息中的消息序列号将全部为零，最后两条消息中的消息序列号将为一。”“A”是发起者分配的 SPI，“B”是响应者分配的 SPI。

An IKE implementation can implement its responder cookie generation in such a way as to not require any saved state to recognize its valid cookie when the second IKE_SA_INIT message arrives. The exact algorithms and syntax used to generate cookies do not affect interoperability and hence are not specified here. The following is an example of how an endpoint could use cookies to implement limited DoS protection.

IKE 实现可以在第二条 IKE_SA_INIT 消息到达时，以不需要任何保存的状态来识别其有效 cookie 的方式实现其响应器 cookie 生成。用于生成 cookie 的确切算法和语法不会影响互操作性，因此此处未指定。以下是端点如何使用 cookie 实现有限的 DoS 保护的示例。

A good way to do this is to set the responder cookie to be:

执行此操作的一个好方法是将响应者 cookie 设置为：

```
Cookie = <VersionIDofSecret> | Hash(Ni | IPi | SPIi | <secret>)
Cookie = <VersionIDofSecret> | Hash(Ni | IPi | SPIi | <secret>)
```

where <secret> is a randomly generated secret known only to the responder and periodically changed and | indicates concatenation. <VersionIDofSecret> should be changed whenever <secret> is regenerated. The cookie can be recomputed when the IKE_SA_INIT arrives the second time and compared to the cookie in the received message. If it matches, the responder knows that the cookie was generated since the last change to <secret> and that IPi must be the same as the source address it saw the first time. Incorporating SPIi into the calculation ensures that if multiple IKE SAs are being set up in parallel they will all get different cookies (assuming the initiator chooses unique SPIi's). Incorporating Ni in the hash ensures that an attacker who sees only message 2 can't successfully forge a message 3. Also, incorporating SPIi in the hash prevents an attacker from fetching one cookie from the other end, and then initiating many IKE_SA_INIT exchanges all with different initiator SPIs (and perhaps port numbers) so that the responder thinks that there are a lot of machines behind one NAT box that are all trying to connect.

其中，<secret>是一个随机生成的秘密，只有响应者知道，并定期更改，|表示串联<只要重新生成<secret>，就应该更改 VersionIDofSecret>。当 IKE_SA_INIT 第二次到达并与接收到的消息中的 cookie 进行比较时，可以重新计算 cookie。如果匹配，响应者知道 cookie 是在上次更改为<secret>后生成的，并且 IPi 必须与第一次看到的源地址相同。将 SPIi 合并到计算中可确保，如果并行设置多个 IKE SA，它们将获得不同的 cookie（假设启动器选择唯一的 SPIi）。将 Ni 合并到哈希中可确保仅看到消息 2 的攻击者无法成功伪造消息 3。此外，在散列中加入 SPIi 可以防止攻击者从另一端获取一个 cookie，然后启动许多 IKE_SA_INIT 交换，所有交换都使用不同的启动器 SPI（可能还有端口号），因此响应者认为在一个 NAT 盒后面有很多机器都在尝试连接。

If a new value for <secret> is chosen while there are connections in the process of being initialized, an IKE_SA_INIT might be returned with other than the current <VersionIDofSecret>. The responder in that case MAY reject the message by sending another response with a new cookie or it MAY keep the old value of <secret> around for a short time and accept cookies computed from either one. The responder should not accept cookies indefinitely after <secret> is changed, since that would defeat part of the DoS protection. The responder should change the value of <secret> frequently, especially if under attack.

如果在初始化过程中存在连接时为<secret>选择了一个新值，则 IKE_SA_INIT 可能会返回当前<VersionIDofSecret>以外的值。在这种情况下，响应者可以通过发送另一个带有新 cookie 的

响应来拒绝消息，或者它可以在短时间内保留<secret>的旧值，并接受从其中任何一个计算出的 cookie。响应程序不应在<secret>更改后无限期地接受 cookie，因为这将破坏部分 DoS 保护。响应者应经常更改<secret>的值，尤其是在受到攻击时。

When one party receives an IKE_SA_INIT request containing a cookie whose contents do not match the value expected, that party MUST ignore the cookie and process the message as if no cookie had been included; usually this means sending a response containing a new cookie. The initiator should limit the number of cookie exchanges it tries before giving up, possibly using exponential back-off. An

当一方收到 IKE_SA_INIT 请求，其中包含内容与预期值不匹配的 cookie 时，该方必须忽略该 cookie，并像未包含 cookie 一样处理该消息；通常这意味着发送包含新 cookie 的响应。发起者应该限制放弃之前尝试的 cookie 交换次数，可能使用指数退避。一

attacker can forge multiple cookie responses to the initiator's IKE_SA_INIT message, and each of those forged cookie replies will cause two packets to be sent: one packet from the initiator to the responder (which will reject those cookies), and one response from responder to initiator that includes the correct cookie.

攻击者可以伪造多个对发起方 IKE_SA_INIT 消息的 cookie 响应，每个伪造的 cookie 响应将导致发送两个数据包：一个数据包从发起方发送到响应方（响应方将拒绝这些 cookie），另一个响应方发送到包含正确 cookie 的发起方。

A note on terminology: the term "cookies" originates with Karn and Simpson [PHOTURIS] in Photuris, an early proposal for key management with IPsec, and it has persisted. The Internet Security Association and Key Management Protocol (ISAKMP) [ISAKMP] fixed message header includes two eight-octet fields called "cookies", and that syntax is used by both IKEv1 and IKEv2, although in IKEv2 they are referred to as the "IKE SPI" and there is a new separate field in a Notify payload holding the cookie.

术语说明：“cookies”一词起源于 PHOTURIS 中的 Karn 和 Simpson[PHOTURIS]，PHOTURIS 是 IPsec 密钥管理的早期建议，并且一直存在。Internet 安全关联和密钥管理协议（ISAKMP）[ISAKMP]固定消息头包括两个称为“cookies”的八位字节字段，该语法由 IKEv1 和 IKEv2 使用，尽管在 IKEv2 中它们被称为“IKE SPI”，并且在保存 cookie 的 Notify 有效载荷中有一个新的单独字段。

2.6.1. Interaction of COOKIE and INVALID_KE_PAYLOAD

2.6.1. COOKIE 与无效的_KE_负载的交互

There are two common reasons why the initiator may have to retry the IKE_SA_INIT exchange: the responder requests a cookie or wants a different Diffie-Hellman group than was included in the KEi payload. If the initiator receives a cookie from the responder, the initiator needs to decide whether or not to include the cookie in only the next retry of the IKE_SA_INIT request, or in all subsequent retries as well.

发起方可能必须重试 IKE_SA_INIT 交换有两个常见原因：响应方请求 cookie 或想要不同于 KEi 有效负载中包含的 Diffie Hellman 组。如果发起方从响应方接收到 cookie，则发起方需要决定是否仅在 IKE_SA_INIT 请求的下一次重试中，或在所有后续重试中包含 cookie。

If the initiator includes the cookie only in the next retry, one additional round trip may be needed in some cases. An additional round trip is needed also if the initiator includes the cookie in all retries, but the responder does not support this. For instance, if the responder includes the KEi payloads in cookie calculation, it will reject the request by sending a new cookie.

如果启动器仅在下一次重试中包含 cookie，则在某些情况下可能需要一次额外的往返。如果发起方在所有重试中都包含 cookie，但响应方不支持，则还需要额外的往返。例如，如果响应程序在 cookie 计算中包含 KEi 有效负载，它将通过发送新 cookie 来拒绝请求。

If both peers support including the cookie in all retries, a slightly shorter exchange can happen.

如果两个对等方都支持在所有重试中包含 cookie，则可能会发生稍短的交换。

```
Initiator                   Responder
-----------------------------------------------------------
HDR(A,0), SAi1, KEi, Ni -->
                      <-- HDR(A,0), N(COOKIE)
HDR(A,0), N(COOKIE), SAi1, KEi, Ni  -->
                      <-- HDR(A,0), N(INVALID_KE_PAYLOAD)
HDR(A,0), N(COOKIE), SAi1, KEi', Ni -->
                      <-- HDR(A,B), SAr1, KEr, Nr
```

```
Initiator                   Responder
-----------------------------------------------------------
HDR(A,0), SAi1, KEi, Ni -->
                      <-- HDR(A,0), N(COOKIE)
HDR(A,0), N(COOKIE), SAi1, KEi, Ni  -->
```

```
                       <-- HDR(A,0), N(INVALID_KE_PAYLOAD)
   HDR(A,0), N(COOKIE), SAi1, KEi', Ni -->
                       <-- HDR(A,B), SAr1, KEr, Nr
```

Implementations SHOULD support this shorter exchange, but MUST NOT fail if other implementations do not support this shorter exchange.

实现应支持此较短的交换，但如果其他实现不支持此较短的交换，则不能失败。

2.7. Cryptographic Algorithm Negotiation

2.7. 密码算法协商

The payload type known as "SA" indicates a proposal for a set of choices of IPsec protocols (IKE, ESP, or AH) for the SA as well as cryptographic algorithms associated with each protocol.

称为“SA”的有效负载类型表示为 SA 选择一组 IPsec 协议（IKE、ESP 或 AH）以及与每个协议相关联的加密算法的建议。

An SA payload consists of one or more proposals. Each proposal includes one protocol. Each protocol contains one or more transforms -- each specifying a cryptographic algorithm. Each transform contains zero or more attributes (attributes are needed only if the Transform ID does not completely specify the cryptographic algorithm).

SA 有效负载由一个或多个方案组成。每项提案包括一项议定书。每个协议都包含一个或多个转换—每个转换指定一个加密算法。每个变换包含零个或多个属性（仅当变换 ID 未完全指定加密算法时才需要属性）。

This hierarchical structure was designed to efficiently encode proposals for cryptographic suites when the number of supported suites is large because multiple values are acceptable for multiple transforms. The responder MUST choose a single suite, which may be any subset of the SA proposal following the rules below.

这种层次结构设计用于在支持的套件数量较大时有效地编码加密套件的建议，因为多个转换可以接受多个值。响应者必须选择单个套件，可以是 SA 提案的任何子集，遵循以下规则。

Each proposal contains one protocol. If a proposal is accepted, the SA response MUST contain the same protocol. The responder MUST accept a single proposal or reject them all and return an error. The error is given in a notification of type

NO_PROPOSAL_CHOSEN.

每项提案包含一项议定书。如果提案被接受，SA 响应必须包含相同的协议。响应者必须接受单个提议或拒绝所有提议并返回错误。错误出现在类型为 NO_PROPOSAL_Selected 的通知中。

Each IPsec protocol proposal contains one or more transforms. Each transform contains a Transform Type. The accepted cryptographic suite MUST contain exactly one transform of each type included in the proposal. For example: if an ESP proposal includes transforms ENCR_3DES, ENCR_AES w/keysize 128, ENCR_AES w/keysize 256, AUTH_HMAC_MD5, and AUTH_HMAC_SHA, the accepted suite MUST contain one of the ENCR_ transforms and one of the AUTH_ transforms. Thus, six combinations are acceptable.

每个 IPsec 协议提案都包含一个或多个转换。每个变换都包含一个变换类型。接受的加密套件必须仅包含提案中包含的每种类型的一个转换。例如：如果 ESP 方案包括转换 ENCR_3DES、ENCR_AES w/keysize 128、ENCR_AES w/keysize 256、AUTH_HMAC_MD5 和 AUTH_HMAC_SHA，则接受的套件必须包含一个 ENCR_转换和一个 AUTH_转换。因此，可以接受六种组合。

If an initiator proposes both normal ciphers with integrity protection as well as combined-mode ciphers, then two proposals are needed. One of the proposals includes the normal ciphers with the integrity algorithms for them, and the other proposal includes all the combined-mode ciphers without the integrity algorithms (because combined-mode ciphers are not allowed to have any integrity algorithm other than "none").

如果发起者同时提出具有完整性保护的普通密码和组合模式密码，则需要两种方案。其中一个方案包括具有完整性算法的普通密码，另一个方案包括不具有完整性算法的所有组合模式密码（因为组合模式密码不允许具有除“无”之外的任何完整性算法）。

2.8. Rekeying

2.8. 重新键入

IKE, ESP, and AH Security Associations use secret keys that should be used only for a limited amount of time and to protect a limited amount of data. This limits the lifetime of the entire Security Association. When the lifetime of a Security Association expires, the Security Association MUST NOT be used. If there is demand, new

IKE、ESP 和 AH 安全关联使用的密钥应仅在有限的时间内使用，并保护有限的数据量。这限制了

整个安全关联的生存期。当安全关联的生存期到期时，不得使用该安全关联。如果有需求，新的

Security Associations MAY be established. Reestablishment of Security Associations to take the place of ones that expire is referred to as "rekeying".

可以建立安全协会。重新建立安全关联以取代过期的安全关联称为“密钥更新”。

To allow for minimal IPsec implementations, the ability to rekey SAs without restarting the entire IKE SA is optional. An implementation MAY refuse all CREATE_CHILD_SA requests within an IKE SA. If an SA has expired or is about to expire and rekeying attempts using the mechanisms described here fail, an implementation MUST close the IKE SA and any associated Child SAs and then MAY start new ones. Implementations may wish to support in-place rekeying of SAs, since doing so offers better performance and is likely to reduce the number of packets lost during the transition.

为了实现最小的 IPsec 实现，可以选择在不重新启动整个 IKE SA 的情况下为 SA 重新设置密钥。实现可以拒绝 IKE SA 中的所有 CREATE_CHILD_SA 请求。如果 SA 已过期或即将过期，并且使用此处描述的机制重新设置密钥的尝试失败，则实现必须关闭 IKE SA 和任何关联的子 SA，然后可以启动新的子 SA。实现可能希望支持 SAs 的就地密钥更新，因为这样做可以提供更好的性能，并可能减少转换过程中丢失的数据包数量。

To rekey a Child SA within an existing IKE SA, create a new, equivalent SA (see Section 2.17 below), and when the new one is established, delete the old one. Note that, when rekeying, the new Child SA SHOULD NOT have different Traffic Selectors and algorithms than the old one.

要在现有 IKE SA 中为子 SA 重新设置密钥，请创建一个新的等效 SA（见下文第 2.17 节），并在建立新 SA 后删除旧 SA。请注意，在重新设置密钥时，新的子 SA 不应具有与旧 SA 不同的流量选择器和算法。

To rekey an IKE SA, establish a new equivalent IKE SA (see Section 2.18 below) with the peer to whom the old IKE SA is shared using a CREATE_CHILD_SA within the existing IKE SA. An IKE SA so created inherits all of the original IKE SA's Child SAs, and the new IKE SA is used for all control messages needed to maintain those Child SAs. After the new equivalent IKE SA is created, the initiator deletes the old IKE SA, and the Delete payload to delete itself MUST be the last request sent over the old IKE SA.

要重新设置 IKE SA 的密钥，请与使用现有 IKE SA 中的 CREATE_CHILD_SA 共享旧 IKE SA 的

对等方建立一个新的等效 IKE SA（见下文第 2.18 节）。这样创建的 IKE SA 继承了原始 IKE SA 的所有子 SA，新 IKE SA 用于维护这些子 SA 所需的所有控制消息。创建新的等效 IKE SA 后，启动器将删除旧 IKE SA，并且要删除自身的删除有效负载必须是通过旧 IKE SA 发送的最后一个请求。

SAs should be rekeyed proactively, i.e., the new SA should be established before the old one expires and becomes unusable. Enough time should elapse between the time the new SA is established and the old one becomes unusable so that traffic can be switched over to the new SA.

应主动更新 SA，即新 SA 应在旧 SA 到期且无法使用之前建立。在建立新 SA 和旧 SA 变得不可用之间应经过足够的时间，以便将流量切换到新 SA。

A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If an SA has been inactive for a long time and if an endpoint would not initiate the SA in the absence of traffic, the endpoint MAY choose to close the SA instead of rekeying it when its lifetime expires. It can also do so if there has been no traffic since the last time the SA was rekeyed.

IKEv1 和 IKEv2 之间的区别在于，在 IKEv1 中，SA 的寿命是协商的。在 IKEv2 中，SA 的每一端都负责在 SA 上实施自己的生存期策略，并在必要时重新设置 SA 的密钥。如果两端具有不同的生存期策略，则生存期较短的一端将始终是请求密钥更新的一端。如果 SA 长时间处于非活动状态，并且如果端点在没有通信量的情况下不会启动 SA，则端点可以选择关闭 SA，而不是在 SA 的生存期到期时重新设置其密钥。如果自上次重新设置 SA 密钥以来没有流量，它也可以这样做。

Note that IKEv2 deliberately allows parallel SAs with the same Traffic Selectors between common endpoints. One of the purposes of this is to support traffic quality of service (QoS) differences among the SAs (see [DIFFSERVFIELD], [DIFFSERVARCH], and Section 4.1 of [DIFFTUNNEL]). Hence unlike IKEv1, the combination of the endpoints and the Traffic Selectors may not uniquely identify an SA between those endpoints, so the IKEv1 rekeying heuristic of deleting SAs on the basis of duplicate Traffic Selectors SHOULD NOT be used.

请注意，IKEv2 故意允许在公共端点之间使用相同流量选择器的并行 SA。其目的之一是支持 SA 之间的业务服务质量（QoS）差异（参见[DIFFSERVFIELD]、[DIFFSERVARCH]和

[DIFFTUNNEL]第 4.1 节）。因此，与 IKEv1 不同，端点和流量选择器的组合可能不会唯一地标识那些端点之间的 SA，因此不应使用基于重复流量选择器删除 SA 的 IKEv1 密钥更新启发式。

There are timing windows -- particularly in the presence of lost packets -- where endpoints may not agree on the state of an SA. The responder to a CREATE_CHILD_SA MUST be prepared to accept messages on an SA before sending its response to the creation request, so there is no ambiguity for the initiator. The initiator MAY begin sending on an SA as soon as it processes the response. The initiator, however, cannot receive on a newly created SA until it receives and processes the response to its CREATE_CHILD_SA request. How, then, is the responder to know when it is OK to send on the newly created SA?

存在定时窗口——特别是在存在丢失数据包的情况下——其中端点可能不同意 SA 的状态。CREATE_CHILD_SA 的响应者必须准备好接受 SA 上的消息，然后再发送其对创建请求的响应，因此启动器没有歧义。发起方可以在处理响应后立即开始在 SA 上发送。但是，在启动器接收并处理对其创建子 SA 请求的响应之前，它无法在新创建的 SA 上接收。那么，响应者如何知道何时可以发送新创建的 SA?

From a technical correctness and interoperability perspective, the responder MAY begin sending on an SA as soon as it sends its response to the CREATE_CHILD_SA request. In some situations, however, this could result in packets unnecessarily being dropped, so an implementation MAY defer such sending.

从技术正确性和互操作性的角度来看，只要响应者发送了对 CREATE_CHILD_SA 请求的响应，就可以开始在 SA 上发送。然而，在某些情况下，这可能会导致不必要地丢弃数据包，因此实现可能会推迟此类发送。

The responder can be assured that the initiator is prepared to receive messages on an SA if either (1) it has received a cryptographically valid message on the other half of the SA pair, or (2) the new SA rekeys an existing SA and it receives an IKE request to close the replaced SA. When rekeying an SA, the responder continues to send traffic on the old SA until one of those events occurs. When establishing a new SA, the responder MAY defer sending messages on a new SA until either it receives one or a timeout has occurred. If an initiator receives a message on an SA for which it has not received a response to its CREATE_CHILD_SA request, it interprets that as a likely packet loss and retransmits the CREATE_CHILD_SA request. An initiator MAY send a dummy ESP message on a newly created ESP SA if it has no messages queued in order to assure the responder that the initiator is ready to receive messages.

如果（1）在 SA 对的另一半上接收到加密有效的消息，或者（2）新 SA 对现有 SA 重新加密，并且接收到关闭替换 SA 的 IKE 请求，则可以确保响应者准备在 SA 上接收消息。在为 SA 重新设置密钥时，响应者继续在旧 SA 上发送通信量，直到发生其中一个事件。建立新 SA 时，响应者可能会延迟在新 SA 上发送消息，直到收到消息或超时为止。如果启动器在 SA 上接收到一条消息，但尚未收到对其 CREATE_CHILD_SA 请求的响应，则会将其解释为可能的数据包丢失，并重新传输 CREATE_CHILD_SA 请求。如果新创建的 ESP SA 上没有消息排队，则启动器可以发送虚拟 ESP 消息，以确保响应者启动器已准备好接收消息。

2.8.1. Simultaneous Child SA Rekeying

2.8.1. 同时子 SA 密钥更新

If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered (delayed by a random amount of time after the need for rekeying is noticed).

如果两端具有相同的生存期策略，则两者可能同时启动密钥更新（这将导致冗余 SA）。为了降低发生这种情况的概率，应该抖动重新键入请求的时间（在注意到需要重新键入后延迟随机时间）。

This form of rekeying may temporarily result in multiple similar SAs between the same pairs of nodes. When there are two SAs eligible to receive packets, a node MUST accept incoming packets through either SA. If redundant SAs are created though such a collision, the SA created with the lowest of the four nonces used in the two exchanges SHOULD be closed by the endpoint that created it. "Lowest" means an octet-by-octet comparison (instead of, for instance, comparing the nonces as large integers). In other words, start by comparing the first octet; if they're equal, move to the next octet, and so on. If you reach the end of one nonce, that nonce is the lower one. The node that initiated the surviving rekeyed SA should delete the replaced SA after the new one is established.

这种形式的密钥更新可能会暂时导致同一对节点之间出现多个类似的 SA。当有两个 SA 有资格接收数据包时，节点必须通过任一 SA 接收传入数据包。如果通过这种冲突创建了冗余 SA，则创建该 SA 的端点应关闭使用两个交换中使用的四个 nonce 中最低的 nonce 创建的 SA。“最低”是指一个八位字节一个八位字节的比较（而不是，例如，将 nonce 作为大整数进行比较）。换句话说，从比较第一个八位组开始；如果它们相等，则移动到下一个八位组，依此类推。如果到达一个 nonce 的末尾，则该 nonce 是较低的一个。在建立新 SA 后，启动幸存的重新密钥 SA 的节点应删除替换的 SA。

The following is an explanation on the impact this has on implementations. Assume that hosts A and B have an existing Child SA pair with SPIs (SPIa1,SPIb1), and both start rekeying it at the same time:

下面解释了这对实现的影响。假设主机 A 和主机 B 具有一个具有 SPI（SPIa1、SPIb1）的现有子 SA 对，并且两者同时开始对其重新设置密钥：

```
Host A                          Host B
-----------------------------------------------------------------
send req1: N(REKEY_SA,SPIa1),
    SA(..,SPIa2,..),Ni1,..  -->
                             <--  send req2: N(REKEY_SA,SPIb1),
                                     SA(..,SPIb2,..),Ni2
recv req2 <--
```

```
Host A                          Host B
-----------------------------------------------------------------
send req1: N(REKEY_SA,SPIa1),
    SA(..,SPIa2,..),Ni1,..  -->
                             <--  send req2: N(REKEY_SA,SPIb1),
                                     SA(..,SPIb2,..),Ni2
recv req2 <--
```

At this point, A knows there is a simultaneous rekeying happening. However, it cannot yet know which of the exchanges will have the lowest nonce, so it will just note the situation and respond as usual.

在这一点上，A 知道有一个同步的密钥更新正在发生。然而，它还不知道哪一家交易所的市盈率最低，因此它只会注意到情况并像往常一样做出反应。

```
send resp2: SA(..,SPIa3,..),
     Nr1,..  -->
                             -->  recv req1
```

```
send resp2: SA(..,SPIa3,..),
     Nr1,..  -->
                             -->  recv req1
```

Now B also knows that simultaneous rekeying is going on. It responds as usual.

现在 B 也知道同步重新键入正在进行。它的反应和往常一样。

```
                           <--  send resp1: SA(..,SPIb3,..),
                                  Nr2,..
  recv resp1 <--
                           -->  recv resp2
```

```
                           <--  send resp1: SA(..,SPIb3,..),
                                  Nr2,..
  recv resp1 <--
                           -->  recv resp2
```

At this point, there are three Child SA pairs between A and B (the old one and two new ones). A and B can now compare the nonces. Suppose that the lowest nonce was Nr1 in message resp2; in this case, B (the sender of req2) deletes the redundant new SA, and A (the node that initiated the surviving rekeyed SA), deletes the old one.

此时，A 和 B 之间有三个子 SA 对（旧的和两个新的）。A 和 B 现在可以比较 nonce。假设消息 resp2 中的最低 nonce 为 Nr1；在这种情况下，B（req2 的发送方）删除冗余的新 SA，A（启动幸存的密钥更新 SA 的节点）删除旧 SA。

```
  send req3: D(SPIa1) -->
                            <--  send req4: D(SPIb2)
                            -->  recv req3
                            <--  send resp3: D(SPIb1)
  recv req4 <--
  send resp4: D(SPIa3) -->
```

```
  send req3: D(SPIa1) -->
                            <--  send req4: D(SPIb2)
                            -->  recv req3
                            <--  send resp3: D(SPIb1)
  recv req4 <--
  send resp4: D(SPIa3) -->
```

The rekeying is now finished.

重新键入现在已完成。

However, there is a second possible sequence of events that can happen if some packets are lost in the network, resulting in retransmissions. The rekeying begins as usual, but A's first packet (req1) is lost.

但是，如果某些数据包在网络中丢失，导致重新传输，则可能会发生第二个可能的事件序列。密钥更新照常开始，但 A 的第一个数据包（req1）丢失。

```
Host A                  Host B
-----------------------------------------------------------------
send req1: N(REKEY_SA,SPIa1),
   SA(..,SPIa2,..),
   Ni1,..  -->  (lost)
                  <--  send req2: N(REKEY_SA,SPIb1),
                        SA(..,SPIb2,..),Ni2
recv req2 <--
send resp2: SA(..,SPIa3,..),
   Nr1,.. -->
                  -->  recv resp2
                  <--  send req3: D(SPIb1)
recv req3 <--
send resp3: D(SPIa1) -->
                  -->  recv resp3
```

```
Host A                  Host B
-----------------------------------------------------------------
send req1: N(REKEY_SA,SPIa1),
   SA(..,SPIa2,..),
   Ni1,..  -->  (lost)
                  <--  send req2: N(REKEY_SA,SPIb1),
                        SA(..,SPIb2,..),Ni2
recv req2 <--
send resp2: SA(..,SPIa3,..),
   Nr1,.. -->
                  -->  recv resp2
                  <--  send req3: D(SPIb1)
recv req3 <--
send resp3: D(SPIa1) -->
                  -->  recv resp3
```

From B's point of view, the rekeying is now completed, and since it has not yet received A's req1, it does not even know that there was simultaneous rekeying. However, A will continue retransmitting the message, and eventually it will reach B.

从 B 的角度来看，重新键入现在已经完成，而且由于它还没有收到 A 的 req1，它甚至不知道同时进行了重新键入。但是，A 将继续重新传输消息，最终它将到达 B。

```
  resend req1 -->
                        -->  recv req1


  resend req1 -->
                        -->  recv req1
```

To B, it looks like A is trying to rekey an SA that no longer exists; thus, B responds to the request with something non-fatal such as CHILD_SA_NOT_FOUND.

对 B 来说，看起来 A 试图重新输入一个不再存在的 SA；因此，B 用一些非致命的东西来响应请求，例如 CHILD_SA_NOT_FOUND。

```
                        <--  send resp1: N(CHILD_SA_NOT_FOUND)
  recv resp1 <--


                        <--  send resp1: N(CHILD_SA_NOT_FOUND)
  recv resp1 <--
```

When A receives this error, it already knows there was simultaneous rekeying, so it can ignore the error message.

当 A 收到此错误时，它已经知道同时进行了密钥更新，因此可以忽略错误消息。

2.8.2. Simultaneous IKE SA Rekeying

2.8.2. 同步 IKE SA 密钥更新

Probably the most complex case occurs when both peers try to rekey the IKE_SA at the same time. Basically, the text in Section 2.8 applies to this case as well; however, it is important to ensure that the Child SAs are inherited by the correct IKE_SA.

最复杂的情况可能发生在两个对等方同时尝试为 IKE_SA 重新设置密钥时。基本上，第 2.8 节中的文本也适用于本案例；但是，必须确保子 SA 由正确的 IKE_SA 继承。

The case where both endpoints notice the simultaneous rekeying works the same way as with Child SAs. After the CREATE_CHILD_SA exchanges, three IKE SAs exist between A and B: the old IKE SA and two new IKE SAs. The new IKE SA containing the lowest nonce SHOULD be deleted by the node that created it, and the other surviving new IKE SA MUST inherit all the Child SAs.

两个端点都注意到同时重设密钥的情况与子 SA 的工作方式相同。在 CREATE_CHILD_SA 交换之后，A 和 B 之间存在三个 IKE SA：旧的 IKE SA 和两个新的 IKE SA。包含最低 nonce 的新 IKE SA 应由创建它的节点删除，而另一个幸存的新 IKE SA 必须继承所有子 SA。

In addition to normal simultaneous rekeying cases, there is a special case where one peer finishes its rekey before it even notices that other peer is doing a rekey. If only one peer detects a simultaneous rekey, redundant SAs are not created. In this case, when the peer that did not notice the simultaneous rekey gets the request to rekey the IKE SA that it has already successfully rekeyed, it SHOULD return TEMPORARY_FAILURE because it is an IKE SA that it is currently trying to close (whether or not it has already sent the delete notification for the SA). If the peer that did notice the simultaneous rekey gets the delete request from the other peer for the old IKE SA, it knows that the other peer did not detect the simultaneous rekey, and the first peer can forget its own rekey attempt.

除了正常的同步密钥更新情况外，还有一种特殊情况，即一个对等方在注意到另一个对等方正在进行密钥更新之前完成其密钥更新。如果只有一个对等方检测到同时重设密钥，则不会创建冗余 SA。在这种情况下，当未注意到同步密钥更新的对等方获得对其已成功密钥更新的 IKE SA 进行密钥更新的请求时，它应返回临时_FAILURE，因为它当前正试图关闭一个 IKE SA（无论它是否已发送 SA 的删除通知）。如果确实注意到同时重新密钥的对等方从另一对等方获取了旧 IKE SA 的删除请求，则它知道另一对等方没有检测到同时重新密钥，并且第一对等方可以忘记其自己的重新密钥尝试。

```
Host A                    Host B
------------------------------------------------------------------
send req1:
    SA(..,SPIa1,..),Ni1,.. -->
                    <-- send req2: SA(..,SPIb1,..),Ni2,..
                    --> recv req1
                    <-- send resp1: SA(..,SPIb2,..),Nr2,..
recv resp1 <--
send req3: D() -->
                    --> recv req3
```

```
Host A                    Host B
------------------------------------------------------------------
```

```
send req1:
     SA(..,SPIa1,..),Ni1,.. -->
                        <-- send req2: SA(..,SPIb1,..),Ni2,..
                        --> recv req1
                        <-- send resp1: SA(..,SPIb2,..),Nr2,..
recv resp1 <--
send req3: D() -->
                        --> recv req3
```

At this point, host B sees a request to close the IKE_SA. There's not much more to do than to reply as usual. However, at this point host B should stop retransmitting req2, since once host A receives resp3, it will delete all the state associated with the old IKE_SA and will not be able to reply to it.

此时，主机 B 看到关闭 IKE_SA 的请求。没有什么比照常回答更多的了。但是，此时主机 B 应停止重新传输 req2，因为一旦主机 A 接收到 resp3，它将删除与旧 IKE_SA 关联的所有状态，并且将无法回复该状态。

<-- send resp3: ()

<--发送响应 3:（）

The TEMPORARY_FAILURE notification was not included in RFC 4306, and support of the TEMPORARY_FAILURE notification is not negotiated.

RFC 4306 中未包含临时_故障通知，也未协商对临时_故障通知的支持。

Thus, older peers that implement RFC 4306 but not this document may receive these notifications. In that case, they will treat it the same as any other unknown error notification, and will stop the exchange. Because the other peer has already rekeyed the exchange, doing so does not have any ill effects.

因此，实现 RFC 4306 而非本文档的较老对等方可能会收到这些通知。在这种情况下，他们会将其视为任何其他未知错误通知，并停止交换。因为另一个对等方已经为交换重新设置了密钥，所以这样做不会产生任何不良影响。

2.8.3. Rekeying the IKE SA versus Reauthentication

2.8.3. 重新键入 IKE SA 与重新验证

Rekeying the IKE SA and reauthentication are different concepts in IKEv2. Rekeying the IKE SA establishes new keys for the IKE SA and resets the Message ID counters, but it does not authenticate the parties again (no AUTH or EAP payloads are involved).

在 IKEv2 中，为 IKE SA 重新键入密钥和重新验证是不同的概念。为 IKE SA 重新设置密钥为 IKE SA 建立新密钥并重置消息 ID 计数器，但不会再次对各方进行身份验证（不涉及身份验证或 EAP 有效负载）。

Although rekeying the IKE SA may be important in some environments, reauthentication (the verification that the parties still have access to the long-term credentials) is often more important.

尽管在某些环境中，为 IKE SA 重新键入密钥可能很重要，但重新验证（验证各方仍然可以访问长期凭据）通常更为重要。

IKEv2 does not have any special support for reauthentication. Reauthentication is done by creating a new IKE SA from scratch (using IKE_SA_INIT/IKE_AUTH exchanges, without any REKEY_SA Notify payloads), creating new Child SAs within the new IKE SA (without REKEY_SA Notify payloads), and finally deleting the old IKE SA (which deletes the old Child SAs as well).

IKEv2 对重新验证没有任何特殊支持。通过从头创建新的 IKE SA（使用 IKE_SA_INIT/IKE_AUTH 交换，不使用任何 REKEY_SA Notify 有效载荷），在新 IKE SA 内创建新的子 SA（不使用 REKEY_SA Notify 有效载荷），最后删除旧的 IKE SA（这也会删除旧的子 SA），可以完成重新身份验证。

This means that reauthentication also establishes new keys for the IKE SA and Child SAs. Therefore, while rekeying can be performed more often than reauthentication, the situation where "authentication lifetime" is shorter than "key lifetime" does not make sense.

这意味着重新验证还为 IKE SA 和子 SA 建立新密钥。因此，虽然可以比重新认证更频繁地执行密钥更新，但是“认证生存期”比“密钥生存期”短的情况没有意义。

While creation of a new IKE SA can be initiated by either party (initiator or responder in the original IKE SA), the use of EAP and/or Configuration payloads means in practice that reauthentication has to be initiated by the same party as the original IKE SA. IKEv2 does not currently allow the responder to request reauthentication in this case; however, there are extensions that add this functionality such as [REAUTH].

虽然新 IKE SA 的创建可以由任何一方（原始 IKE SA 中的发起方或响应方）发起，但 EAP 和/或配置有效载荷的使用实际上意味着重新验证必须由与原始 IKE SA 相同的一方发起。在这种情况下，IKEv2 当前不允许响应者请求重新验证；但是，有些扩展添加了此功能，如[REAUTH]。

2.9. Traffic Selector Negotiation

2.9. 交通选择器协商

When an RFC4301-compliant IPsec subsystem receives an IP packet that matches a "protect" selector in its Security Policy Database (SPD), the subsystem protects that packet with IPsec. When no SA exists yet, it is the task of IKE to create it. Maintenance of a system's SPD is outside the scope of IKE, although some implementations might update their SPD in connection with the running of IKE (for an example scenario, see Section 1.1.3).

当符合 RFC4301 的 IPsec 子系统接收到与其安全策略数据库（SPD）中的“保护”选择器匹配的 IP 数据包时，该子系统使用 IPsec 保护该数据包。当尚不存在 SA 时，IKE 的任务是创建 SA。系统 SPD 的维护不在 IKE 的范围内，尽管一些实现可能会在 IKE 运行时更新其 SPD（有关示例场景，请参阅第 1.1.3 节）。

Traffic Selector (TS) payloads allow endpoints to communicate some of the information from their SPD to their peers. These must be communicated to IKE from the SPD (for example, the PF_KEY API [PFKEY] uses the SADB_ACQUIRE message). TS payloads specify the selection criteria for packets that will be forwarded over the newly set up SA. This can serve as a consistency check in some scenarios to assure that the SPDs are consistent. In others, it guides the dynamic update of the SPD.

流量选择器（TS）有效负载允许端点将一些信息从其 SPD 传输到其对等方。这些信息必须从 SPD 传送到 IKE（例如，PF_密钥 API[PFKEY]使用 SADB_获取消息）。TS 有效负载指定将通过新设置的 SA 转发的数据包的选择标准。在某些情况下，这可以作为一致性检查，以确保 SPD 的一致性。在其他情况下，它指导 SPD 的动态更新。

Two TS payloads appear in each of the messages in the exchange that creates a Child SA pair. Each TS payload contains one or more Traffic Selectors. Each Traffic Selector consists of an address range (IPv4 or IPv6), a port range, and an IP protocol ID.

两个 TS 有效负载出现在创建子 SA 对的 exchange 中的每条消息中。每个 TS 有效负载包含一个或多个流量选择器。每个流量选择器由地址范围（IPv4 或 IPv6）、端口范围和 IP 协议 ID 组成。

The first of the two TS payloads is known as TSi (Traffic Selector-initiator). The second is known as TSr (Traffic Selector-responder). TSi specifies the source address of traffic forwarded from (or the destination address of traffic forwarded to) the initiator of the Child SA pair. TSr specifies the destination address of the traffic forwarded to (or the source address of the traffic forwarded from) the responder of the Child SA pair. For example, if the original initiator requests the creation of a Child SA pair, and wishes to tunnel all traffic from subnet 198.51.100.* on the initiator's side to subnet 192.0.2.* on the responder's side, the initiator would include a single Traffic Selector in each TS payload. TSi would specify the address range (198.51.100.0 - 198.51.100.255) and TSr would specify the address range (192.0.2.0 - 192.0.2.255). Assuming that proposal was acceptable to the responder, it would send identical TS payloads back.

两个 TS 有效负载中的第一个称为 TSi（流量选择器启动器）。第二种称为 TSr（流量选择器响应器）。TSi 指定从子 SA 对的启动器转发的流量的源地址（或转发到子 SA 对的流量的目标地址）。TSr 指定转发到子 SA 对的响应者的通信量的目标地址（或转发自子 SA 对的通信量的源地址）。例如，如果原始启动器请求创建子 SA 对，并希望通过隧道将所有流量从启动器侧的子网 198.51.100.*传输到响应方侧的子网 192.0.2.*，则启动器将在每个 TS 有效负载中包含一个流量选择器。TSi 将指定地址范围（198.51.100.0-198.51.100.255），TSr 将指定地址范围（192.0.2.0-192.0.2.255）。假设响应者可以接受该建议，它将发送相同的 TS 有效负载。

IKEv2 allows the responder to choose a subset of the traffic proposed by the initiator. This could happen when the configurations of the two endpoints are being updated but only one end has received the new information. Since the two endpoints may be configured by different people, the incompatibility may persist for an extended period even in the absence of errors. It also allows for intentionally different configurations, as when one end is configured to tunnel all addresses and depends on the other end to have the up-to-date list.

IKEv2 允许响应者选择发起者提议的通信量的子集。当两个端点的配置正在更新，但只有一端接收到新信息时，可能会发生这种情况。由于两个端点可能由不同的人配置，因此即使在没有错误的情况下，不兼容性也可能持续很长一段时间。它还允许有意不同的配置，例如一端配置为隧道所有地址，并依赖另一端拥有最新列表。

When the responder chooses a subset of the traffic proposed by the initiator, it narrows the Traffic Selectors to some subset of the initiator's proposal (provided the set does not become the null set). If the type of Traffic Selector proposed is unknown, the responder ignores that Traffic Selector, so that the unknown type is not returned in the narrowed set.

当响应者选择发起者建议的流量子集时，它会将流量选择器缩小到发起者建议的某个子集（前提是该集合不会变为空集合）。如果建议的流量选择器类型未知，响应者将忽略该流量选择器，以便在缩小的集合中不返回未知类型。

To enable the responder to choose the appropriate range in this case, if the initiator has requested the SA due to a data packet, the initiator SHOULD include as the first Traffic Selector in each of TSi and TSr a very specific Traffic Selector including the addresses in the packet triggering the request. In the example, the initiator would include in TSi two Traffic Selectors: the first containing the address range (198.51.100.43 - 198.51.100.43) and the source port and IP protocol from the packet and the second containing (198.51.100.0 - 198.51.100.255) with all ports and IP protocols. The initiator would similarly include two Traffic Selectors in TSr. If the initiator creates the Child SA pair not in response to an arriving packet, but rather, say, upon startup, then there may be no specific addresses the initiator prefers for the initial tunnel over any other. In that case, the first values in TSi and TSr can be ranges rather than specific values.

在这种情况下，为了使响应者能够选择适当的范围，如果发起方由于数据分组而请求 SA，则发起方应在每个 TSi 和 TSr 中包括一个非常特定的流量选择器作为第一个流量选择器，该流量选择器包括触发请求的分组中的地址。在该示例中，启动器将在 TSi 中包括两个流量选择器：第一个包含地址范围（198.51.100.43-198.51.100.43）和来自数据包的源端口和 IP 协议，第二个包含所有端口和 IP 协议（198.51.100.0-198.51.100.255）。发起方将类似地在 TSr 中包括两个流量选择器。如果启动器创建子 SA 对不是响应到达的数据包，而是（比如）在启动时创建，则启动器可能不喜欢初始隧道的特定地址。在这种情况下，TSi 和 TSr 中的第一个值可以是范围，而不是特定值。

The responder performs the narrowing as follows:

响应者按如下方式执行变窄：

o If the responder's policy does not allow it to accept any part of the proposed Traffic Selectors, it responds with a TS_UNACCEPTABLE Notify message.

o 如果响应者的策略不允许其接受提议的流量选择器的任何部分，则响应者将发送一条 TS_UNACCEPTABLE Notify 消息。

o If the responder's policy allows the entire set of traffic covered by TSi and TSr, no narrowing is necessary, and the responder can return the same TSi and TSr values.

o 如果响应者的策略允许 TSi 和 TSr 覆盖的整个流量集，则无需缩小，响应者可以返回相同的

TSi 和 TSr 值。

o If the responder's policy allows it to accept the first selector of TSi and TSr, then the responder MUST narrow the Traffic Selectors to a subset that includes the initiator's first choices. In this example above, the responder might respond with TSi being (198.51.100.43 - 198.51.100.43) with all ports and IP protocols.

o 如果响应者的策略允许其接受 TSi 和 TSr 的第一个选择器，则响应者必须将流量选择器缩小到包含启动器的第一个选择的子集。在上面的这个示例中，响应程序可能会使用 TSi（198.51.100.43-198.51.100.43）和所有端口和 IP 协议进行响应。

o If the responder's policy does not allow it to accept the first selector of TSi and TSr, the responder narrows to an acceptable subset of TSi and TSr.

o 如果响应者的策略不允许其接受 TSi 和 TSr 的第一个选择器，则响应者将缩小到 TSi 和 TSr 的可接受子集。

When narrowing is done, there may be several subsets that are acceptable but their union is not. In this case, the responder arbitrarily chooses one of them, and MAY include an ADDITIONAL_TS_POSSIBLE notification in the response. The ADDITIONAL_TS_POSSIBLE notification asserts that the responder narrowed the proposed Traffic Selectors but that other Traffic Selectors would also have been acceptable, though only in a separate SA. There is no data associated with this Notify type. This case will occur only when the initiator and responder are configured differently from one another. If the initiator and responder agree on the granularity of tunnels, the initiator will never request a tunnel wider than the responder will accept.

当缩小范围时，可能有几个子集是可接受的，但它们的联合是不可接受的。在这种情况下，响应者任意选择其中一个，并且可以在响应中包括附加的可能通知。附加的可能通知断言响应者缩小了提议的流量选择器，但也可以接受其他流量选择器，尽管仅在单独的 SA 中。没有与此通知类型关联的数据。只有当启动器和响应程序的配置不同时，才会发生这种情况。如果发起者和响应者在隧道的粒度上达成一致，发起者将永远不会请求比响应者所能接受的更宽的隧道。

It is possible for the responder's policy to contain multiple smaller ranges, all encompassed by the initiator's Traffic Selector, and with the responder's policy being that each of those ranges should be sent over a different SA. Continuing the example above, the responder might have a policy of being willing to tunnel those addresses to and from the initiator, but might require that each address pair be on a separately negotiated Child SA. If the initiator didn't generate its request based on the packet, but (for example) upon startup, there would not be the very specific first Traffic Selectors helping the responder to select the correct range. There would be no way for the responder to determine which pair of addresses should be included in this tunnel, and it would have to make a guess or reject the request with a SINGLE_PAIR_REQUIRED Notify message.

响应者的策略可能包含多个较小的范围，所有范围都由启动器的流量选择器包含，并且响应者的策略是，这些范围中的每一个都应通过不同的 SA 发送。继续上面的示例，响应者可能具有愿意将这些地址隧道到发起方和从发起方传出的策略，但可能要求每个地址对位于单独协商的子 SA 上。如果发起者没有根据数据包生成请求，但是（例如）在启动时，将不会有非常特定的第一个流量选择器帮助响应者选择正确的范围。响应程序将无法确定此隧道中应包含哪对地址，它必须猜测或使用一条所需的通知消息拒绝请求。

The SINGLE_PAIR_REQUIRED error indicates that a CREATE_CHILD_SA request is unacceptable because its sender is only willing to accept Traffic Selectors specifying a single pair of addresses. The requestor is expected to respond by requesting an SA for only the specific traffic it is trying to forward.

SINGLE_PAIR_REQUIRED 错误表示 CREATE_CHILD_SA 请求不可接受，因为其发送方只愿意接受指定一对地址的流量选择器。请求者应通过仅针对其试图转发的特定流量请求 SA 进行响应。

Few implementations will have policies that require separate SAs for each address pair. Because of this, if only some parts of the TSi and TSr proposed by the initiator are acceptable to the responder, responders SHOULD narrow the selectors to an acceptable subset rather than use SINGLE_PAIR_REQUIRED.

很少有实现具有要求每个地址对使用单独 SA 的策略。因此，如果只有发起者提出的 TSi 和 TSr 的某些部分可被响应者接受，则响应者应将选择器缩小到可接受的子集，而不是使用所需的单个对。

2.9.1. Traffic Selectors Violating Own Policy

2.9.1. 违反自己策略的流量选择器

When creating a new SA, the initiator needs to avoid proposing Traffic Selectors that violate its own policy. If this rule is not followed, valid traffic may be dropped. If you use decorrelated policies from [IPSECARCH], this kind of policy violations cannot happen.

创建新 SA 时，启动器需要避免提出违反其自身策略的流量选择器。如果不遵守此规则，则可能

会丢弃有效的通信量。如果使用[IPSECARCH]中的解相关策略，则不会发生此类策略冲突。

This is best illustrated by an example. Suppose that host A has a policy whose effect is that traffic to 198.51.100.66 is sent via host B encrypted using AES, and traffic to all other hosts in 198.51.100.0/24 is also sent via B, but must use 3DES. Suppose also that host B accepts any combination of AES and 3DES.

一个例子最好地说明了这一点。假设主机 A 有一个策略，其效果是到 198.51.100.66 的流量通过使用 AES 加密的主机 B 发送，到 198.51.100.0/24 中所有其他主机的流量也通过 B 发送，但必须使用 3DES。还假设主机 B 接受 AES 和 3DE 的任意组合。

If host A now proposes an SA that uses 3DES, and includes TSr containing (198.51.100.0-198.51.100.255), this will be accepted by host B. Now, host B can also use this SA to send traffic from 198.51.100.66, but those packets will be dropped by A since it requires the use of AES for this traffic. Even if host A creates a new SA only for 198.51.100.66 that uses AES, host B may freely continue to use the first SA for the traffic. In this situation,

如果主机 A 现在建议使用 3DES 的 SA，并且包含 TSr（198.51.100.0-198.51.100.255），这将被主机 B 接受。现在，主机 B 也可以使用此 SA 从 198.51.100.66 发送流量，但这些数据包将被 A 丢弃，因为它需要对该流量使用 AES。即使主机 A 仅为 198.51.100.66 创建使用 AES 的新 SA，主机 B 也可以自由地继续为流量使用第一个 SA。在这种情况下,，

when proposing the SA, host A should have followed its own policy, and included a TSr containing ((198.51.100.0- 198.51.100.65),(198.51.100.67-198.51.100.255)) instead.

在提议 SA 时，主机 A 应遵循其自己的政策，并包含一份 TSr，其中包含（（198.51.100.0-198.51.100.65），（198.51.100.67-198.51.100.255））。

In general, if (1) the initiator makes a proposal "for traffic X (TSi/TSr), do SA", and (2) for some subset X' of X, the initiator does not actually accept traffic X' with SA, and (3) the initiator would be willing to accept traffic X' with some SA' (!=SA), valid traffic can be unnecessarily dropped since the responder can apply either SA or SA' to traffic X'.

通常，如果（1）发起者提出“针对流量 X（TSi/TSr），do SA”的建议，（2）对于 X 的某些子集 X'，发起者实际上不接受 SA 的流量 X'，并且（3）发起者愿意接受带有某些 SA'（！=SA）的流量 X'，则，由于响应者可以将 SA 或 SA‘应用于流量 X’，有效流量可能会被不必要地丢弃。

2.10. Nonces

2.10. 临时

The IKE_SA_INIT messages each contain a nonce. These nonces are used as inputs to cryptographic functions. The CREATE_CHILD_SA request and the CREATE_CHILD_SA response also contain nonces. These nonces are used to add freshness to the key derivation technique used to obtain keys for Child SA, and to ensure creation of strong pseudorandom bits from the Diffie-Hellman key. Nonces used in IKEv2 MUST be randomly chosen, MUST be at least 128 bits in size, and MUST be at least half the key size of the negotiated pseudorandom function (PRF). However, the initiator chooses the nonce before the outcome of the negotiation is known. Because of that, the nonce has to be long enough for all the PRFs being proposed. If the same random number source is used for both keys and nonces, care must be taken to ensure that the latter use does not compromise the former.

IKE_SA_INIT 消息每个都包含一个 nonce。这些 nonce 用作加密函数的输入。CREATE_CHILD_SA 请求和 CREATE_CHILD_SA 响应也包含 nonce。这些 nonce 用于为用于获取子 SA 密钥的密钥派生技术添加新鲜度，并确保从 Diffie-Hellman 密钥创建强伪随机位。IKEv2 中使用的 nonce 必须随机选择，大小必须至少为 128 位，并且必须至少为协商伪随机函数（PRF）密钥大小的一半。但是，发起者在知道协商结果之前选择 nonce。因此，nonce 必须足够长，以满足所有提议的 PRF。如果密钥和 nonce 使用相同的随机数源，则必须注意确保后者的使用不会损害前者。

2.11. Address and Port Agility

2.11. 地址和端口灵活性

IKE runs over UDP ports 500 and 4500, and implicitly sets up ESP and AH associations for the same IP addresses over which it runs. The IP addresses and ports in the outer header are, however, not themselves cryptographically protected, and IKE is designed to work even through Network Address Translation (NAT) boxes. An implementation MUST accept incoming requests even if the source port is not 500 or 4500, and MUST respond to the address and port from which the request was received. It MUST specify the address and port at which the request was received as the source address and port in the response. IKE functions identically over IPv4 or IPv6.

IKE 在 UDP 端口 500 和 4500 上运行，并隐式地为其运行的相同 IP 地址设置 ESP 和 AH 关联。然而，外部报头中的 IP 地址和端口本身并没有加密保护，IKE 设计为即使通过网络地址转换（NAT）盒也能工作。即使源端口不是 500 或 4500，实现也必须接受传入请求，并且必须响应

接收请求的地址和端口。它必须指定接收请求的地址和端口作为响应中的源地址和端口。IKE 在 IPv4 或 IPv6 上的功能相同。

2.12. Reuse of Diffie-Hellman Exponentials

2.12. Diffie-Hellman 指数的重用

IKE generates keying material using an ephemeral Diffie-Hellman exchange in order to gain the property of "perfect forward secrecy". This means that once a connection is closed and its corresponding keys are forgotten, even someone who has recorded all of the data from the connection and gets access to all of the long-term keys of

IKE 使用短暂的 Diffie-Hellman 交换生成密钥材料，以获得“完美前向保密”的特性。这意味着，一旦一个连接被关闭，其对应的密钥被遗忘，即使是记录了该连接的所有数据并访问该连接的所有长期密钥的人

the two endpoints cannot reconstruct the keys used to protect the conversation without doing a brute force search of the session key space.

如果不对会话密钥空间进行强制搜索，这两个端点无法重建用于保护会话的密钥。

Achieving perfect forward secrecy requires that when a connection is closed, each endpoint MUST forget not only the keys used by the connection but also any information that could be used to recompute those keys.

要实现完美的前向保密性，需要在连接关闭时，每个端点不仅必须忘记连接使用的密钥，还必须忘记可用于重新计算这些密钥的任何信息。

Because computing Diffie-Hellman exponentials is computationally expensive, an endpoint may find it advantageous to reuse those exponentials for multiple connection setups. There are several reasonable strategies for doing this. An endpoint could choose a new exponential only periodically though this could result in less-than-perfect forward secrecy if some connection lasts for less than the lifetime of the exponential. Or it could keep track of which exponential was used for each connection and delete the information associated with the exponential only when some corresponding connection was closed. This would allow the exponential to be reused without losing perfect forward secrecy at the cost of maintaining more state.

由于计算 Diffie-Hellman 指数在计算上非常昂贵，端点可能会发现将这些指数用于多个连接设

置是有利的。有几种合理的策略可以做到这一点。端点只能周期性地选择一个新的指数，但如果某些连接持续时间少于指数的生命周期，则这可能会导致不完美的前向保密性。或者，它可以跟踪每个连接使用的是哪个指数，并且只有在某些对应的连接关闭时才删除与指数相关的信息。这将允许在不丢失完美的前向保密性的情况下重用指数，而代价是维护更多的状态。

Whether and when to reuse Diffie-Hellman exponentials are private decisions in the sense that they will not affect interoperability. An implementation that reuses exponentials MAY choose to remember the exponential used by the other endpoint on past exchanges and if one is reused to avoid the second half of the calculation. See [REUSE] for a security analysis of this practice and for additional security considerations when reusing ephemeral Diffie-Hellman keys.

是否以及何时重用 Diffie-Hellman 指数是私人决定，因为它们不会影响互操作性。重用指数的实现可能会选择记住另一个端点在过去的交换中使用的指数，如果重用一个端点，则可以避免计算的后半部分。有关此实践的安全性分析以及重用短暂 Diffie-Hellman 密钥时的其他安全注意事项，请参阅[重用]。

2.13. Generating Keying Material

2.13. 生成键控材料

In the context of the IKE SA, four cryptographic algorithms are negotiated: an encryption algorithm, an integrity protection algorithm, a Diffie-Hellman group, and a pseudorandom function (PRF). The PRF is used for the construction of keying material for all of the cryptographic algorithms used in both the IKE SA and the Child SAs.

在 IKE SA 的上下文中，协商了四种加密算法：加密算法、完整性保护算法、Diffie-Hellman 群和伪随机函数（PRF）。PRF 用于为 IKE SA 和子 SA 中使用的所有加密算法构造密钥材料。

We assume that each encryption algorithm and integrity protection algorithm uses a fixed-size key and that any randomly chosen value of that fixed size can serve as an appropriate key. For algorithms that accept a variable-length key, a fixed key size MUST be specified as part of the cryptographic transform negotiated (see Section 3.3.5 for the definition of the Key Length transform attribute). For algorithms for which not all values are valid keys (such as DES or 3DES with key parity), the algorithm by which keys are derived from arbitrary values MUST be specified by the cryptographic transform.

我们假设每个加密算法和完整性保护算法都使用固定大小的密钥，并且任意随机选择的固定大小的值都可以作为适当的密钥。对于接受可变长度密钥的算法，必须将固定密钥大小指定为协商的

加密转换的一部分（有关密钥长度转换属性的定义，请参见第 3.3.5 节）。对于并非所有值都是有效密钥的算法（如具有密钥奇偶性的 DES 或 3DES），密码转换必须指定从任意值派生密钥的算法。

For integrity protection functions based on Hashed Message Authentication Code (HMAC), the fixed key size is the size of the output of the underlying hash function.

对于基于哈希消息身份验证码（HMAC）的完整性保护函数，固定密钥大小是基础哈希函数输出的大小。

It is assumed that PRFs accept keys of any length, but have a preferred key size. The preferred key size MUST be used as the length of SK_d, SK_pi, and SK_pr (see Section 2.14). For PRFs based on the HMAC construction, the preferred key size is equal to the length of the output of the underlying hash function. Other types of PRFs MUST specify their preferred key size.

假定 PRF 接受任意长度的密钥，但具有首选密钥大小。首选密钥大小必须用作 SK_d、SK_pi 和 SK_pr 的长度（见第 2.14 节）。对于基于 HMAC 构造的 PRF，首选密钥大小等于基础哈希函数输出的长度。其他类型的 PRF 必须指定其首选密钥大小。

Keying material will always be derived as the output of the negotiated PRF algorithm. Since the amount of keying material needed may be greater than the size of the output of the PRF, the PRF is used iteratively. The term "prf+" describes a function that outputs a pseudorandom stream based on the inputs to a pseudorandom function called "prf".

键控材料将始终作为协商 PRF 算法的输出导出。由于所需的键控材料量可能大于 PRF 输出的大小，所以重复使用 PRF。术语“prf+”描述了一种函数，该函数基于称为“prf”的伪随机函数的输入输出伪随机流。

In the following, | indicates concatenation. prf+ is defined as:

在下文中，|表示串联。prf+定义为：

prf+ (K,S) = T1 | T2 | T3 | T4 | ...

prf+（K，S）=T1 | T2 | T3 | T4 |。。。

where: T1 = prf (K, S | 0x01) T2 = prf (K, T1 | S | 0x02) T3 = prf (K, T2 | S | 0x03) T4 = prf (K, T3 | S | 0x04) ...

式中：T1=prf（K，S | 0x01）T2=prf（K，T1 | S | 0x02）T3=prf（K，T2 | S | 0x03）T4=prf（K，T3 | S | 0x04）。。。

This continues until all the material needed to compute all required keys has been output from prf+. The keys are taken from the output string without regard to boundaries (e.g., if the required keys are a 256-bit Advanced Encryption Standard (AES) key and a 160-bit HMAC key, and the prf function generates 160 bits, the AES key will come from T1 and the beginning of T2, while the HMAC key will come from the rest of T2 and the beginning of T3).

这将一直持续到从 prf+输出计算所有所需关键帧所需的所有材质。密钥取自输出字符串而不考虑边界（例如，如果所需密钥是 256 位高级加密标准（AES）密钥和 160 位 HMAC 密钥，并且 prf 函数生成 160 位，则 AES 密钥将来自 T1 和 T2 的开头，而 HMAC 密钥将来自 T2 的其余部分和 T3 的开头）。

The constant concatenated to the end of each prf function is a single octet. The prf+ function is not defined beyond 255 times the size of the prf function output.

连接到每个 prf 函数末尾的常量是一个八位字节。prf+函数的定义不超过 prf 函数输出大小的 255 倍。

2.14. Generating Keying Material for the IKE SA

2.14. 为 IKE SA 生成关键帧材质

The shared keys are computed as follows. A quantity called SKEYSEED is calculated from the nonces exchanged during the IKE_SA_INIT exchange and the Diffie-Hellman shared secret established during that exchange. SKEYSEED is used to calculate seven other secrets: SK_d used for deriving new keys for the Child SAs established with this

共享密钥的计算如下所示。称为 Skeysed 的量是根据 IKE_SA_INIT 交换期间交换的 nonce 和在该交换期间建立的 Diffie Hellman 共享秘密计算的。SKEYSEED 用于计算其他七个秘密：SK_d 用于为使用此密钥建立的子 SA 派生新密钥

IKE SA; SK_ai and SK_ar used as a key to the integrity protection algorithm for authenticating the component messages of subsequent exchanges; SK_ei and SK_er used for encrypting (and of course decrypting) all subsequent exchanges; and SK_pi and SK_pr, which are used when generating an AUTH payload. The lengths of SK_d, SK_pi, and SK_pr MUST be the preferred key length of the PRF agreed upon.

艾克萨；SK_ai 和 SK_ar 用作完整性保护算法的密钥，用于验证后续交换的组件消息；用于加密（当然还有解密）所有后续交换的 SK_ei 和 SK_er；以及 SK_pi 和 SK_pr，它们在生成身份验证有效负载时使用。SK_d、SK_pi 和 SK_pr 的长度必须是商定的 PRF 的首选密钥长度。

SKEYSEED and its derivatives are computed as follows:

SKEYSEED 及其衍生物的计算如下：

```
SKEYSEED = prf(Ni | Nr, g^ir)
```

```
SKEYSEED = prf(Ni | Nr, g^ir)
```

```
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr }
             = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr )
```

```
{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr }
             = prf+ (SKEYSEED, Ni | Nr | SPIi | SPIr )
```

(indicating that the quantities SK_d, SK_ai, SK_ar, SK_ei, SK_er, SK_pi, and SK_pr are taken in order from the generated bits of the prf+). g^ir is the shared secret from the ephemeral Diffie-Hellman exchange. g^ir is represented as a string of octets in big endian order padded with zeros if necessary to make it the length of the modulus. Ni and Nr are the nonces, stripped of any headers. For historical backward-compatibility reasons, there are two PRFs that are treated specially in this calculation. If the negotiated PRF is AES-XCBC-PRF-128 [AESXCBCPRF128] or AES-CMAC-PRF-128 [AESCMACPRF128], only the first 64 bits of Ni and the first 64 bits of Nr are used in calculating SKEYSEED, but all the bits are used for input to the prf+ function.

（指示量 SK_d、SK_ai、SK_ar、SK_ei、SK_er、SK_pi 和 SK_pr 是按顺序从 prf+的生成位获取的）。g^ir 是短暂的 Diffie-Hellman 交换的共享秘密。g^ir 表示为一个以大端顺序排列的八位字节串，如果需要，用零填充，以使其成为模的长度。Ni 和 Nr 是无任何标题的 nonce。出于历史向后兼容性的原因，有两个 PRF 在此计算中被特别处理。如果协商的 PRF 为 AES-XCBC-PRF-128[AESXCBCPRF128]或 AES-CMAC-PRF-128[AESCMCAPRF128]，则在计算 SKEYSED 时仅使用 Ni 的前 64 位和 Nr 的前 64 位，但所有位都用于 PRF+函数的输入。

The two directions of traffic flow use different keys. The keys used to protect messages from the original initiator are SK_ai and SK_ei. The keys used to protect messages in the other direction are SK_ar and SK_er.

交通流的两个方向使用不同的键。用于保护来自原始启动器的消息的密钥是 sku ai 和 sku ei。用于保护另一个方向的消息的键是 SK_-ar 和 SK_-er。

2.15. Authentication of the IKE SA

2.15. IKE SA 的身份验证

When not using extensible authentication (see Section 2.16), the peers are authenticated by having each sign (or MAC using a padded shared secret as the key, as described later in this section) a block of data. In these calculations, IDi' and IDr' are the entire ID payloads excluding the fixed header. For the responder, the octets to be signed start with the first octet of the first SPI in the header of the second message (IKE_SA_INIT response) and end with the last octet of the last payload in the second message. Appended to this (for the purposes of computing the signature) are the initiator's nonce Ni (just the value, not the payload containing it), and the value prf(SK_pr, IDr'). Note that neither the nonce Ni nor the value prf(SK_pr, IDr') are transmitted. Similarly, the initiator signs the first message (IKE_SA_INIT request), starting with the first octet of the first SPI in the header and ending with the last

当不使用可扩展身份验证（参见第 2.16 节）时，通过让每个签名（或使用填充共享密钥作为密钥的 MAC，如本节后面所述）具有一个数据块来对对等方进行身份验证。在这些计算中，IDi'和 IDr'是不包括固定标头的整个 ID 有效载荷。对于响应者，要签名的八位字节以第二条消息（IKE_SA_INIT response）报头中第一个 SPI 的第一个八位字节开始，并以第二条消息中最后一个有效负载的最后一个八位字节结束。附加在这之后（为了计算签名的目的）是启动器的 nonce Ni（只是值，而不是包含它的有效负载）和值 prf（SK_pr，IDr'）。注意，既不传输 nonce Ni 也不传输值 prf（SK_pr，IDr'）。类似地，发起者对第一条消息（IKE_SA_INIT request）进行签名，从报头中第一个 SPI 的第一个八位组开始，以最后一个八位组结束

octet of the last payload. Appended to this (for purposes of computing the signature) are the responder's nonce Nr, and the value prf(SK_pi, IDi'). It is critical to the security of the exchange that each side sign the other side's nonce.

最后一个有效载荷的八位字节。在此之后（为了计算签名的目的）附加了响应者的 nonce Nr 和值 prf（SK_pi，IDi'）。每一方签署另一方的临时证书对交易所的安全至关重要。

The initiator's signed octets can be described as:

启动器的签名八位字节可以描述为：

```
InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage1 = RealIKEHDR | RestOfMessage1
NonceRPayload = PayloadHeader | NonceRData
InitiatorIDPayload = PayloadHeader | RestOfInitIDPayload
RestOfInitIDPayload = IDType | RESERVED | InitIDData
MACedIDForI = prf(SK_pi, RestOfInitIDPayload)
```

```
InitiatorSignedOctets = RealMessage1 | NonceRData | MACedIDForI
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage1 = RealIKEHDR | RestOfMessage1
NonceRPayload = PayloadHeader | NonceRData
InitiatorIDPayload = PayloadHeader | RestOfInitIDPayload
RestOfInitIDPayload = IDType | RESERVED | InitIDData
MACedIDForI = prf(SK_pi, RestOfInitIDPayload)
```

The responder's signed octets can be described as:

响应者的签名八位字节可以描述为：

```
ResponderSignedOctets = RealMessage2 | NonceIData | MACedIDForR
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage2 = RealIKEHDR | RestOfMessage2
NonceIPayload = PayloadHeader | NonceIData
ResponderIDPayload = PayloadHeader | RestOfRespIDPayload
RestOfRespIDPayload = IDType | RESERVED | RespIDData
MACedIDForR = prf(SK_pr, RestOfRespIDPayload)
```

```
ResponderSignedOctets = RealMessage2 | NonceIData | MACedIDForR
GenIKEHDR = [ four octets 0 if using port 4500 ] | RealIKEHDR
RealIKEHDR =  SPIi | SPIr |  . . . | Length
RealMessage2 = RealIKEHDR | RestOfMessage2
NonceIPayload = PayloadHeader | NonceIData
ResponderIDPayload = PayloadHeader | RestOfRespIDPayload
RestOfRespIDPayload = IDType | RESERVED | RespIDData
MACedIDForR = prf(SK_pr, RestOfRespIDPayload)
```

Note that all of the payloads are included under the signature, including any payload types not defined in this document. If the first message of the exchange is sent multiple times (such as with a responder cookie and/or a different Diffie-Hellman group), it is the latest version of the message that is signed.

请注意，所有有效载荷都包含在签名下，包括本文档中未定义的任何有效载荷类型。如果 exchange 的第一条消息被多次发送（例如使用响应者 cookie 和/或不同的 Diffie Hellman 组），则签名的是该消息的最新版本。

Optionally, messages 3 and 4 MAY include a certificate, or certificate chain providing evidence that the key used to compute a digital signature belongs to the name in the ID payload. The signature or MAC will be computed using algorithms dictated by the type of key used by the signer, and specified by the Auth Method field in the Authentication payload. There is no requirement that the initiator and responder sign with the same cryptographic algorithms. The choice of cryptographic algorithms depends on the type of key each has. In particular, the initiator may be using a shared key while the responder may have a public signature key and certificate. It will commonly be the case (but it is not required) that, if a shared secret is used for authentication, the same key is used in both directions.

可选地，消息 3 和 4 可以包括证书或证书链，该证书或证书链提供用于计算数字签名的密钥属于 ID 有效载荷中的名称的证据。签名或 MAC 将使用由签名者使用的密钥类型指定的算法进行计算，并由身份验证有效负载中的 Auth Method 字段指定。不要求发起者和响应者使用相同的加密算法签名。加密算法的选择取决于每种算法的密钥类型。具体地，发起方可以使用共享密钥，而响应方可以具有公共签名密钥和证书。通常情况下（但不是必需的），如果共享密钥用于身份验证，则在两个方向上使用相同的密钥。

Note that it is a common but typically insecure practice to have a shared key derived solely from a user-chosen password without incorporating another source of randomness. This is typically insecure because user-chosen passwords are unlikely to have sufficient unpredictability to resist dictionary attacks and these attacks are not prevented in this authentication method. (Applications using password-based authentication for bootstrapping and IKE SA should use the authentication method in Section 2.16, which is designed to prevent off-line dictionary attacks.) The pre-shared key needs to contain as much unpredictability as the strongest key being negotiated. In the case of a pre-shared key, the AUTH value is computed as:

请注意，一种常见但通常不安全的做法是，仅从用户选择的密码派生共享密钥，而不包含其他随机性来源。这通常是不安全的，因为用户选择的密码不太可能具有足够的不可预测性来抵抗字典攻击，并且这种身份验证方法无法防止这些攻击。（使用基于密码的身份验证进行引导和 IKE SA

的应用程序应使用第 2.16 节中的身份验证方法，该方法旨在防止离线字典攻击。）预共享密钥需要包含与协商的最强密钥一样多的不可预测性。对于预共享密钥，AUTH 值的计算如下：

```
For the initiator:
   AUTH = prf( prf(Shared Secret, "Key Pad for IKEv2"),
                    <InitiatorSignedOctets>)
For the responder:
   AUTH = prf( prf(Shared Secret, "Key Pad for IKEv2"),
                    <ResponderSignedOctets>)
```

```
For the initiator:
   AUTH = prf( prf(Shared Secret, "Key Pad for IKEv2"),
                    <InitiatorSignedOctets>)
For the responder:
   AUTH = prf( prf(Shared Secret, "Key Pad for IKEv2"),
                    <ResponderSignedOctets>)
```

where the string "Key Pad for IKEv2" is 17 ASCII characters without null termination. The shared secret can be variable length. The pad string is added so that if the shared secret is derived from a password, the IKE implementation need not store the password in cleartext, but rather can store the value prf(Shared Secret,"Key Pad for IKEv2"), which could not be used as a password equivalent for protocols other than IKEv2. As noted above, deriving the shared secret from a password is not secure. This construction is used because it is anticipated that people will do it anyway. The management interface by which the shared secret is provided MUST accept ASCII strings of at least 64 octets and MUST NOT add a null terminator before using them as shared secrets. It MUST also accept a hex encoding of the shared secret. The management interface MAY accept other encodings if the algorithm for translating the encoding to a binary string is specified.

其中字符串“IKEv2 的键盘”为 17 个 ASCII 字符，无空终止。共享秘密可以是可变长度的。添加 pad 字符串，以便如果共享密钥来自密码，则 IKE 实现不需要以明文形式存储密码，而是可以存储值 prf（共享密钥，“IKEv2 的键盘”），该值不能用作 IKEv2 以外协议的等效密码。如上所述，从密码导出共享秘密是不安全的。之所以使用这种结构，是因为预计人们无论如何都会这样做。提供共享机密的管理接口必须接受至少 64 个八位字节的 ASCII 字符串，并且在将其用作共享机密之前不得添加空终止符。它还必须接受共享秘密的十六进制编码。如果指定了将编码转换为二进制字符串的算法，则管理接口可以接受其他编码。

There are two types of EAP authentication (described in Section 2.16), and each type uses different values in the AUTH computations shown above. If the EAP method is key-generating, substitute master session key (MSK) for the shared secret in the computation. For non-key-generating methods, substitute SK_pi and SK_pr, respectively, for the shared secret in the two AUTH computations.

有两种类型的 EAP 身份验证（如第 2.16 节所述），每种类型在上面所示的身份验证计算中使用不同的值。如果 EAP 方法是密钥生成，则在计算中用主会话密钥（MSK）替换共享密钥。对于非密钥生成方法，在两次身份验证计算中分别用 SK_pi 和 SK_pr 替换共享密钥。

2.16. Extensible Authentication Protocol Methods

2.16. 可扩展认证协议方法

In addition to authentication using public key signatures and shared secrets, IKE supports authentication using methods defined in RFC 3748 [EAP]. Typically, these methods are asymmetric (designed for a user authenticating to a server), and they may not be mutual. For this reason, these protocols are typically used to authenticate the initiator to the responder and MUST be used in conjunction with a public-key-signature-based authentication of the responder to the initiator. These methods are often associated with mechanisms referred to as "Legacy Authentication" mechanisms.

除了使用公钥签名和共享秘密进行身份验证外，IKE 还支持使用 RFC 3748[EAP]中定义的方法进行身份验证。通常，这些方法是不对称的（专为向服务器进行身份验证的用户而设计），并且它们可能不是相互的。因此，这些协议通常用于向响应者验证启动器，并且必须与基于公钥签名的响应者向启动器验证结合使用。这些方法通常与称为“遗留身份验证”机制的机制相关联。

While this document references [EAP] with the intent that new methods can be added in the future without updating this specification, some simpler variations are documented here. [EAP] defines an authentication protocol requiring a variable number of messages. Extensible Authentication is implemented in IKE as additional IKE_AUTH exchanges that MUST be completed in order to initialize the IKE SA.

虽然本文档引用[EAP]的目的是在未来可以添加新方法，而无需更新本规范，但此处记录了一些更简单的变化。[EAP]定义了需要可变数量消息的身份验证协议。可扩展身份验证在 IKE 中实现，作为初始化 IKE SA 必须完成的附加 IKE_身份验证交换。

An initiator indicates a desire to use EAP by leaving out the AUTH payload from the first message in the IKE_AUTH exchange. (Note that the AUTH payload is required for non-EAP authentication, and is thus not marked as optional in the rest of this document.) By including an IDi payload but not an AUTH payload, the initiator has declared an identity but has not proven it. If the responder is willing to use an EAP method, it will place an Extensible Authentication Protocol (EAP) payload in the response of the IKE_AUTH exchange and defer sending SAr2, TSi, and TSr until initiator authentication is complete in a subsequent IKE_AUTH exchange. In the case of a minimal EAP method, the initial SA establishment will appear as follows:

发起者通过在 IKE_身份验证交换的第一条消息中省略身份验证有效负载来表示希望使用 EAP。（请注意，非 EAP 身份验证需要验证有效负载，因此在本文档的其余部分中未标记为可选。）通过包含 IDi 有效负载而非验证有效负载，发起方已声明身份，但尚未证明身份。如果响应者愿意使用 EAP 方法，它将在 IKE_认证交换的响应中放置可扩展认证协议（EAP）有效负载，并延迟发送 SAr2、TSi 和 TSr，直到在后续 IKE_认证交换中完成启动器认证。在最小 EAP 方法的情况下，初始 SA 建立如下所示：

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
                          <--  HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK {IDi, [CERTREQ,]
    [IDr,] SAi2,
    TSi, TSr}  -->
                          <--  HDR, SK {IDr, [CERT,] AUTH,
                                 EAP }
HDR, SK {EAP}  -->
                          <--  HDR, SK {EAP (success)}
HDR, SK {AUTH}  -->
                          <--  HDR, SK {AUTH, SAr2, TSi, TSr }
```

```
Initiator                         Responder
-------------------------------------------------------------------
HDR, SAi1, KEi, Ni  -->
                          <--  HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK {IDi, [CERTREQ,]
    [IDr,] SAi2,
    TSi, TSr}  -->
                          <--  HDR, SK {IDr, [CERT,] AUTH,
                                 EAP }
HDR, SK {EAP}  -->
                          <--  HDR, SK {EAP (success)}
HDR, SK {AUTH}  -->
                          <--  HDR, SK {AUTH, SAr2, TSi, TSr }
```

As described in Section 2.2, when EAP is used, each pair of IKE SA initial setup messages will have their message numbers incremented; the first pair of AUTH messages will have an ID of 1, the second will be 2, and so on.

如第 2.2 节所述，当使用 EAP 时，每对 IKE SA 初始设置消息将增加其消息编号；第一对身份验证消息的 ID 为 1，第二对为 2，依此类推。

For EAP methods that create a shared key as a side effect of authentication, that shared key MUST be used by both the initiator and responder to generate AUTH payloads in messages 7 and 8 using the syntax for shared secrets specified in Section 2.15. The shared key from EAP is the field from the EAP specification named MSK. This shared key generated during an IKE exchange MUST NOT be used for any other purpose.

对于创建共享密钥作为身份验证副作用的 EAP 方法，启动器和响应程序必须使用该共享密钥，以使用第 2.15 节中指定的共享机密语法在消息 7 和消息 8 中生成身份验证有效载荷。EAP 中的共享密钥是 EAP 规范中名为 MSK 的字段。IKE 交换期间生成的此共享密钥不得用于任何其他目的。

EAP methods that do not establish a shared key SHOULD NOT be used, as they are subject to a number of man-in-the-middle attacks [EAPMITM] if these EAP methods are used in other protocols that do not use a server-authenticated tunnel. Please see the Security Considerations section for more details. If EAP methods that do not generate a shared key are used, the AUTH payloads in messages 7 and 8 MUST be generated using SK_pi and SK_pr, respectively.

不应使用未建立共享密钥的 EAP 方法，因为如果这些 EAP 方法用于不使用服务器身份验证隧道的其他协议中，它们会受到许多中间人攻击[EAPMITM]。有关更多详细信息，请参阅安全注意事项部分。如果使用不生成共享密钥的 EAP 方法，则必须分别使用 SK_pi 和 SK_pr 生成消息 7 和 8 中的身份验证有效负载。

The initiator of an IKE SA using EAP needs to be capable of extending the initial protocol exchange to at least ten IKE_AUTH exchanges in the event the responder sends notification messages and/or retries the authentication prompt. Once the protocol exchange defined by the chosen EAP authentication method has successfully terminated, the responder MUST send an EAP payload containing the Success message. Similarly, if the authentication method has failed, the responder MUST send an EAP payload containing the Failure message. The responder MAY at any time terminate the IKE exchange by sending an EAP payload containing the

Failure message.

如果响应者发送通知消息和/或重试身份验证提示，则使用 EAP 的 IKE SA 的启动器需要能够将初始协议交换扩展到至少十个 IKE_认证交换。一旦所选 EAP 身份验证方法定义的协议交换成功终止，响应者必须发送包含成功消息的 EAP 有效负载。类似地，如果身份验证方法失败，响应者必须发送包含失败消息的 EAP 有效负载。响应者可随时通过发送包含故障消息的 EAP 有效载荷来终止 IKE 交换。

Following such an extended exchange, the EAP AUTH payloads MUST be included in the two messages following the one containing the EAP Success message.

在这种扩展交换之后，EAP AUTH 有效负载必须包含在包含 EAP Success 消息的消息之后的两条消息中。

When the initiator authentication uses EAP, it is possible that the contents of the IDi payload is used only for Authentication, Authorization, and Accounting (AAA) routing purposes and selecting which EAP method to use. This value may be different from the identity authenticated by the EAP method. It is important that policy lookups and access control decisions use the actual authenticated identity. Often the EAP server is implemented in a separate AAA server that communicates with the IKEv2 responder. In this case, the authenticated identity, if different from that in the IDi payload, has to be sent from the AAA server to the IKEv2 responder.

当发起方身份验证使用 EAP 时，IDi 有效负载的内容可能仅用于身份验证、授权和记帐（AAA）路由目的以及选择要使用的 EAP 方法。此值可能不同于 EAP 方法验证的标识。重要的是，策略查找和访问控制决策使用实际的经过身份验证的身份。EAP 服务器通常在与 IKEv2 响应程序通信的单独 AAA 服务器中实现。在这种情况下，如果与 IDi 有效负载中的身份不同，则必须将经过身份验证的身份从 AAA 服务器发送到 IKEv2 响应程序。

2.17. Generating Keying Material for Child SAs

2.17. 为子 SAs 生成关键帧材质

A single Child SA is created by the IKE_AUTH exchange, and additional Child SAs can optionally be created in CREATE_CHILD_SA exchanges. Keying material for them is generated as follows:

单个子 SA 由 IKE_AUTH 交换创建，并且可以选择在 CREATE_Child_SA 交换中创建其他子 SA。它们的键控材质生成如下：

KEYMAT = prf+(SK_d, Ni | Nr)

KEYMAT=prf+（SK|d，Ni|Nr）

Where Ni and Nr are the nonces from the IKE_SA_INIT exchange if this request is the first Child SA created or the fresh Ni and Nr from the CREATE_CHILD_SA exchange if this is a subsequent creation.

其中，如果该请求是创建的第一个子SA，则Ni和Nr是来自IKE_SA_INIT交换的nonce；如果是后续创建，则Ni和Nr是来自CREATE_Child_SA交换的新Ni和Nr。

For CREATE_CHILD_SA exchanges including an optional Diffie-Hellman exchange, the keying material is defined as:

对于包括可选Diffie-Hellman交换的CREATE_CHILD_SA交换，键控材质定义为：

```
KEYMAT = prf+(SK_d, g^ir (new) | Ni | Nr )
```

```
KEYMAT = prf+(SK_d, g^ir (new) | Ni | Nr )
```

where g^ir (new) is the shared secret from the ephemeral Diffie-Hellman exchange of this CREATE_CHILD_SA exchange (represented as an octet string in big endian order padded with zeros in the high-order bits if necessary to make it the length of the modulus).

其中，g^ir（new）是来自此CREATE_CHILD_SA交换的短暂Diffie-Hellman交换的共享秘密（表示为一个以大端顺序排列的八位组字符串，如果需要，在高阶位中填充零以使其成为模的长度）。

A single CHILD_SA negotiation may result in multiple Security Associations. ESP and AH SAs exist in pairs (one in each direction), so two SAs are created in a single Child SA negotiation for them. Furthermore, Child SA negotiation may include some future IPsec protocol(s) in addition to, or instead of, ESP or AH (for example, ROHC_INTEG as described in [ROHCV2]). In any case, keying material for each Child SA MUST be taken from the expanded KEYMAT using the following rules:

单个子_SA协商可能导致多个安全关联。ESP和AH SA成对存在（每个方向一个），因此在一个子SA协商中为它们创建两个SA。此外，子SA协商可能包括一些未来的IPsec协议，作为ESP或AH的补充或替代（例如，如[ROHCV2]中所述的ROHC_INTEG）。在任何情况下，必须使用以下规则从扩展的键盘垫中获取每个子SA的键控材料：

o All keys for SAs carrying data from the initiator to the responder are taken before SAs going from the responder to the initiator.

o 将数据从发起方传送到响应方的 SAs 的所有密钥在 SAs 从响应方传送到发起方之前获取。

o If multiple IPsec protocols are negotiated, keying material for each Child SA is taken in the order in which the protocol headers will appear in the encapsulated packet.

o 如果协商了多个 IPsec 协议，则按照协议头在封装数据包中出现的顺序获取每个子 SA 的密钥材料。

o If an IPsec protocol requires multiple keys, the order in which they are taken from the SA's keying material needs to be described in the protocol's specification. For ESP and AH, [IPSECARCH] defines the order, namely: the encryption key (if any) MUST be taken from the first bits and the integrity key (if any) MUST be taken from the remaining bits.

o 如果 IPsec 协议需要多个密钥，则需要在协议规范中描述从 SA 的密钥材料中获取密钥的顺序。对于 ESP 和 AH，[IPSECARCH]定义了顺序，即：加密密钥（如果有）必须从第一位获取，完整性密钥（如果有）必须从其余位获取。

Each cryptographic algorithm takes a fixed number of bits of keying material specified as part of the algorithm, or negotiated in SA payloads (see Section 2.13 for description of key lengths, and Section 3.3.5 for the definition of the Key Length transform attribute).

每种加密算法都采用作为算法一部分指定的或在 SA 有效载荷中协商的固定数量的密钥材料位（密钥长度描述见第 2.13 节，密钥长度转换属性定义见第 3.3.5 节）。

2.18. Rekeying IKE SAs Using a CREATE_CHILD_SA Exchange

2.18. 使用 CREATE_CHILD_SA 交换重新键入 IKE SA

The CREATE_CHILD_SA exchange can be used to rekey an existing IKE SA (see Sections 1.3.2 and 2.8). New initiator and responder SPIs are supplied in the SPI fields in the Proposal structures inside the Security Association (SA) payloads (not the SPI fields in the IKE header). The TS payloads are omitted when rekeying an IKE SA. SKEYSEED for the new IKE SA is computed using SK_d from the existing IKE SA as follows:

CREATE_CHILD_SA 交换可用于为现有 IKE SA 重新设置密钥（参见第 1.3.2 和 2.8 节）。新的

发起方和响应方 SPI 在安全关联（SA）有效负载内的建议结构中的 SPI 字段中提供（而不是 IKE 头中的 SPI 字段）。在为 IKE SA 重新设置密钥时，TS 有效载荷被忽略。使用来自现有 IKE SA 的 SK_d 计算新 IKE SA 的 skeysed，如下所示：

```
  SKEYSEED = prf(SK_d (old), g^ir (new) | Ni | Nr)
```

```
  SKEYSEED = prf(SK_d (old), g^ir (new) | Ni | Nr)
```

where g^ir (new) is the shared secret from the ephemeral Diffie-Hellman exchange of this CREATE_CHILD_SA exchange (represented as an octet string in big endian order padded with zeros if necessary to make it the length of the modulus) and Ni and Nr are the two nonces stripped of any headers.

其中，g^ir（new）是来自此 CREATE_CHILD_SA 交换的短暂 Diffie-Hellman 交换的共享秘密（表示为一个以大端顺序填充零的八位字节字符串，如果有必要，使其成为模的长度），Ni 和 Nr 是从任何头中剥离的两个 nonce。

The old and new IKE SA may have selected a different PRF. Because the rekeying exchange belongs to the old IKE SA, it is the old IKE SA's PRF that is used to generate SKEYSEED.

新旧 IKE SA 可能选择了不同的 PRF。由于密钥交换属于旧 IKE SA，因此用于生成 Skeysed 的是旧 IKE SA 的 PRF。

The main reason for rekeying the IKE SA is to ensure that the compromise of old keying material does not provide information about the current keys, or vice versa. Therefore, implementations MUST perform a new Diffie-Hellman exchange when rekeying the IKE SA. In other words, an initiator MUST NOT propose the value "NONE" for the Diffie-Hellman transform, and a responder MUST NOT accept such a proposal. This means that a successful exchange rekeying the IKE SA always includes the KEi/KEr payloads.

对 IKE SA 重新设置密钥的主要原因是确保旧密钥材料的泄露不会提供有关当前密钥的信息，反之亦然。因此，在为 IKE SA 重新设置密钥时，实现必须执行新的 Diffie-Hellman 交换。换句话说，发起者不能为 Diffie-Hellman 转换提出值“NONE”，响应者也不能接受这样的建议。这意味着成功的 IKE SA 密钥交换始终包括 KEi/KEr 有效载荷。

The new IKE SA MUST reset its message counters to 0.

新 IKE SA 必须将其消息计数器重置为 0。

SK_d, SK_ai, SK_ar, SK_ei, and SK_er are computed from SKEYSEED as specified in Section 2.14, using SPIi, SPIr, Ni, and Nr from the new exchange, and using the new IKE SA's PRF.

SK_d、SK_ai、SK_ar、SK_ei 和 SK_er 根据第 2.14 节规定的 skyseed，使用新交换的 SPIi、SPIr、Ni 和 Nr，并使用新 IKE SA 的 PRF 进行计算。

2.19. Requesting an Internal Address on a Remote Network

2.19. 请求远程网络上的内部地址

Most commonly occurring in the endpoint-to-security-gateway scenario, an endpoint may need an IP address in the network protected by the security gateway and may need to have that address dynamically

在端点到安全网关场景中最常见的情况是，端点可能需要由安全网关保护的网络中的 IP 地址，并且可能需要动态地拥有该地址

assigned. A request for such a temporary address can be included in any request to create a Child SA (including the implicit request in message 3) by including a CP payload. Note, however, it is usual to only assign one IP address during the IKE_AUTH exchange. That address persists at least until the deletion of the IKE SA.

分配。通过包括 CP 有效载荷，可以在创建子 SA 的任何请求（包括消息 3 中的隐式请求）中包括对这种临时地址的请求。但是，请注意，在 IKE_身份验证交换期间通常只分配一个 IP 地址。该地址至少在 IKE SA 删除之前一直存在。

This function provides address allocation to an IPsec Remote Access Client (IRAC) trying to tunnel into a network protected by an IPsec Remote Access Server (IRAS). Since the IKE_AUTH exchange creates an IKE SA and a Child SA, the IRAC MUST request the IRAS-controlled address (and optionally other information concerning the protected network) in the IKE_AUTH exchange. The IRAS may procure an address for the IRAC from any number of sources such as a DHCP/BOOTP (Bootstrap Protocol) server or its own address pool.

此函数为试图通过隧道进入受 IPsec 远程访问服务器（IRAS）保护的网络的 IPsec 远程访问客户端（IRAC）提供地址分配。由于 IKE_认证交换创建 IKE SA 和子 SA，IRAC 必须在 IKE_认证交换中请求 IRAS 控制的地址（以及可选的其他有关受保护网络的信息）。IRA 可以从任何数量的源（如 DHCP/BOOTP（引导协议）服务器或其自己的地址池）获取 IRAC 的地址。

```
Initiator                         Responder
-------------------------------------------------------------------
 HDR, SK {IDi, [CERT,]
    [CERTREQ,] [IDr,] AUTH,
    CP(CFG_REQUEST), SAi2,
    TSi, TSr}  -->
                               <--  HDR, SK {IDr, [CERT,] AUTH,
                                      CP(CFG_REPLY), SAr2,
                                      TSi, TSr}
```

```
Initiator                         Responder
-------------------------------------------------------------------
 HDR, SK {IDi, [CERT,]
    [CERTREQ,] [IDr,] AUTH,
    CP(CFG_REQUEST), SAi2,
    TSi, TSr}  -->
                               <--  HDR, SK {IDr, [CERT,] AUTH,
                                      CP(CFG_REPLY), SAr2,
                                      TSi, TSr}
```

In all cases, the CP payload MUST be inserted before the SA payload. In variations of the protocol where there are multiple IKE_AUTH exchanges, the CP payloads MUST be inserted in the messages containing the SA payloads.

在所有情况下，CP 有效负载必须在 SA 有效负载之前插入。在存在多个 IKE_AUTH 交换的协议变体中，必须将 CP 有效负载插入包含 SA 有效负载的消息中。

CP(CFG_REQUEST) MUST contain at least an INTERNAL_ADDRESS attribute (either IPv4 or IPv6) but MAY contain any number of additional attributes the initiator wants returned in the response.

CP（CFG_请求）必须至少包含一个内部_地址属性（IPv4 或 IPv6），但可以包含启动器希望在响应中返回的任何数量的附加属性。

For example, message from initiator to responder:

例如，从发起方到响应方的消息：

```
CP(CFG_REQUEST)=
  INTERNAL_ADDRESS()
TSi = (0, 0-65535,0.0.0.0-255.255.255.255)
TSr = (0, 0-65535,0.0.0.0-255.255.255.255)
```

```
   CP(CFG_REQUEST)=
     INTERNAL_ADDRESS()
   TSi = (0, 0-65535,0.0.0.0-255.255.255.255)
   TSr = (0, 0-65535,0.0.0.0-255.255.255.255)
```

NOTE: Traffic Selectors contain (protocol, port range, address range).

注意：流量选择器包含（协议、端口范围、地址范围）。

Message from responder to initiator:

从响应者到启动器的消息：

```
   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)
```

```
   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)
```

All returned values will be implementation dependent. As can be seen in the above example, the IRAS MAY also send other attributes that were not included in CP(CFG_REQUEST) and MAY ignore the non-mandatory attributes that it does not support.

所有返回值都将取决于实现。从上述示例中可以看出，IRA 还可以发送 CP（CFG_请求）中未包含的其他属性，并且可以忽略其不支持的非强制性属性。

The responder MUST NOT send a CFG_REPLY without having first received a CP(CFG_REQUEST) from the initiator, because we do not want the IRAS to perform an unnecessary configuration lookup if the IRAC cannot process the REPLY.

在未首先从发起方收到 CP（CFG_请求）之前，响应方不得发送 CFG_回复，因为如果 IRAC 无法处理回复，我们不希望 IRA 执行不必要的配置查找。

In the case where the IRAS's configuration requires that CP be used for a given identity IDi, but IRAC has failed to send a CP(CFG_REQUEST), IRAS MUST fail the request, and terminate the Child SA creation with a FAILED_CP_REQUIRED error. The FAILED_CP_REQUIRED is not fatal to the IKE SA; it simply causes the Child SA creation to fail. The initiator can fix this by later starting a new Configuration payload request. There is no associated data in the FAILED_CP_REQUIRED error.

如果 IRAS 的配置要求 CP 用于给定的标识 IDi，但 IRAC 未能发送 CP（CFG_请求），IRAS 必须使请求失败，并终止子 SA 创建，并出现失败的_CP_REQUIRED 错误。失败的 CP_对 IKE SA 来说不是致命的；它只会导致子 SA 创建失败。启动器可以通过稍后启动新的配置有效负载请求来解决此问题。失败的\u CP\u 必需错误中没有相关数据。

2.20. Requesting the Peer's Version

2.20. 请求对等方的版本

An IKE peer wishing to inquire about the other peer's IKE software version information MAY use the method below. This is an example of a configuration request within an INFORMATIONAL exchange, after the IKE SA and first Child SA have been created.

希望查询另一对等方的 IKE 软件版本信息的 IKE 对等方可以使用以下方法。这是创建 IKE SA 和第一个子 SA 后信息交换中的配置请求示例。

An IKE implementation MAY decline to give out version information prior to authentication or even after authentication in case some implementation is known to have some security weakness. In that case, it MUST either return an empty string or no CP payload if CP is not supported.

IKE 实现可能会拒绝在身份验证之前甚至在身份验证之后提供版本信息，以防已知某些实现存在某些安全弱点。在这种情况下，如果不支持 CP，则必须返回空字符串或不返回 CP 有效负载。

```
Initiator                  Responder
-------------------------------------------------------------------
HDR, SK{CP(CFG_REQUEST)}  -->
                       <--  HDR, SK{CP(CFG_REPLY)}


Initiator                  Responder
```

```
   ---------------------------------------------------------------
   HDR, SK{CP(CFG_REQUEST)}  -->
                               <--  HDR, SK{CP(CFG_REPLY)}
```

CP(CFG_REQUEST)= APPLICATION_VERSION("")

CP（CFG_请求）=应用程序_版本（“”）

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

CP（CFG_回复）应用程序_版本（“foobar v1.3beta，（c）foobar Inc.））

2.21. Error Handling

2.21. 错误处理

There are many kinds of errors that can occur during IKE processing. The general rule is that if a request is received that is badly formatted, or unacceptable for reasons of policy (such as no matching cryptographic algorithms), the response contains a Notify payload indicating the error. The decision whether or not to send such a response depends whether or not there is an authenticated IKE SA.

IKE 处理过程中可能会发生多种错误。一般规则是，如果收到的请求格式不正确，或者由于策略原因（例如没有匹配的加密算法）而不可接受，则响应包含一个指示错误的 Notify 有效负载。是否发送此类响应的决定取决于是否存在经过身份验证的 IKE SA。

If there is an error parsing or processing a response packet, the general rule is to not send back any error message because responses should not generate new requests (and a new request would be the only way to send back an error message). Such errors in parsing or processing response packets should still cause the recipient to clean up the IKE state (for example, by sending a Delete for a bad SA).

如果解析或处理响应数据包时出错，一般规则是不发回任何错误消息，因为响应不应生成新请求（而新请求将是发回错误消息的唯一方式）。解析或处理响应数据包时的此类错误仍应导致收件人清除 IKE 状态（例如，通过发送对坏 SA 的删除）。

Only authentication failures (AUTHENTICATION_FAILED and EAP failure) and malformed messages (INVALID_SYNTAX) lead to a deletion of the IKE SA without requiring an explicit INFORMATIONAL exchange carrying a Delete payload. Other error conditions MAY require such an exchange if policy dictates that this is needed.

If the exchange is terminated with EAP Failure, an AUTHENTICATION_FAILED notification is not sent.

只有身份验证失败（身份验证失败和 EAP 失败）和格式错误的消息（无效的语法）会导致删除 IKE SA，而不需要进行带有删除负载的显式信息交换。如果策略规定需要这样做，则其他错误条件可能需要这样的交换。如果交换因 EAP 失败而终止，则不会发送身份验证失败通知。

2.21.1. Error Handling in IKE_SA_INIT

2.21.1. IKE_SA_INIT 中的错误处理

Errors that occur before a cryptographically protected IKE SA is established need to be handled very carefully. There is a trade-off between wanting to help the peer to diagnose a problem and thus responding to the error and wanting to avoid being part of a DoS attack based on forged messages.

在建立受加密保护的 IKE SA 之前发生的错误需要非常小心地处理。在希望帮助对等方诊断问题从而响应错误和希望避免成为基于伪造消息的 DoS 攻击的一部分之间，存在一种权衡。

In an IKE_SA_INIT exchange, any error notification causes the exchange to fail. Note that some error notifications such as COOKIE, INVALID_KE_PAYLOAD or INVALID_MAJOR_VERSION may lead to a subsequent successful exchange. Because all error notifications are completely unauthenticated, the recipient should continue trying for some time before giving up. The recipient should not immediately act based on the error notification unless corrective actions are defined in this specification, such as for COOKIE, INVALID_KE_PAYLOAD, and INVALID_MAJOR_VERSION.

在 IKE_SA_INIT 交换中，任何错误通知都会导致交换失败。请注意，一些错误通知（如 COOKIE、无效的\u KE\u 负载或无效的\u 主\u 版本）可能会导致后续的成功交换。因为所有错误通知都是完全未经验证的，所以收件人应该在放弃之前继续尝试一段时间。收件人不应立即根据错误通知采取行动，除非本规范中定义了纠正措施，例如 COOKIE、无效的_KE_有效负载和无效的_主要_版本。

2.21.2. Error Handling in IKE_AUTH

2.21.2. IKE_AUTH 中的错误处理

All errors that occur in an IKE_AUTH exchange, causing the authentication to fail for whatever reason (invalid shared secret, invalid ID, untrusted certificate issuer, revoked or expired certificate, etc.) SHOULD result in an AUTHENTICATION_FAILED notification. If the error occurred on the responder, the notification is returned in the protected response, and is usually the only payload in that response. Although the IKE_AUTH messages are encrypted and integrity protected, if the peer receiving this notification has not authenticated the other end yet, that peer needs to treat the information with caution.

IKE_身份验证交换中发生的所有错误，无论出于何种原因（无效的共享机密、无效的 ID、不受信任的证书颁发者、吊销或过期的证书等），都会导致身份验证失败通知。如果错误发生在响应程序上，则通知将在受保护的响应中返回，并且通常是该响应中的唯一有效负载。虽然 IKE_AUTH 消息是加密的，并且完整性受到保护，但是如果接收此通知的对等方尚未对另一端进行身份验证，则该对等方需要谨慎处理该信息。

If the error occurs on the initiator, the notification MAY be returned in a separate INFORMATIONAL exchange, usually with no other payloads. This is an exception for the general rule of not starting new exchanges based on errors in responses.

如果错误发生在启动器上，则通知可能会在单独的信息交换中返回，通常没有其他有效负载。这是一个例外的一般规则，不启动新的交流的基础上的错误的反应。

Note, however, that request messages that contain an unsupported critical payload, or where the whole message is malformed (rather than just bad payload contents), MUST be rejected in their entirety, and MUST only lead to an UNSUPPORTED_CRITICAL_PAYLOAD or INVALID_SYNTAX Notification sent as a response. The receiver should not verify the payloads related to authentication in this case.

但是，请注意，如果请求消息包含不受支持的关键负载，或者整个消息的格式不正确（而不仅仅是错误的负载内容），则必须全部拒绝，并且只能导致不受支持的关键负载或作为响应发送的无效语法通知。在这种情况下，接收方不应验证与身份验证相关的有效负载。

If authentication has succeeded in the IKE_AUTH exchange, the IKE SA is established; however, establishing the Child SA or requesting configuration information may still fail. This failure does not automatically cause the IKE SA to be deleted. Specifically, a responder may include all the payloads associated with authentication (IDr, CERT, and AUTH) while sending error notifications for the piggybacked exchanges (FAILED_CP_REQUIRED, NO_PROPOSAL_CHOSEN, and so on), and the initiator MUST NOT fail the authentication because of this. The initiator MAY, of course, for reasons of policy later delete such an IKE SA.

如果在 IKE_认证交换中认证成功，则建立 IKE SA；但是，建立子 SA 或请求配置信息仍可能失败。此故障不会自动导致删除 IKE SA。具体地说，响应者可以包括与身份验证（IDr、CERT 和

AUTH）相关联的所有有效负载，同时为搭载的交换发送错误通知（需要失败的\u CP\u、没有选择的\u 建议\u 等），并且启动器不得因此而使身份验证失败。当然，出于策略的原因，发起者稍后可以删除这样的 IKE SA。

In an IKE_AUTH exchange, or in the INFORMATIONAL exchange immediately following it (in case an error happened when processing a response to IKE_AUTH), the UNSUPPORTED_CRITICAL_PAYLOAD, INVALID_SYNTAX, and AUTHENTICATION_FAILED notifications are the only ones to cause the IKE SA to be deleted or not created, without a Delete payload. Extension documents may define new error notifications with these semantics, but MUST NOT use them unless the peer has been shown to understand them, such as by using the Vendor ID payload.

在 IKE_AUTH 交换中，或在紧接其后的信息交换中（如果在处理对 IKE_AUTH 的响应时发生错误），只有不受支持的_CRITICAL_负载、无效的_语法和身份验证失败的通知会导致 IKE SA 被删除或不创建，而没有删除负载。扩展文档可以使用这些语义定义新的错误通知，但除非对等方已经被证明理解它们，否则不能使用它们，例如通过使用供应商 ID 负载。

2.21.3. Error Handling after IKE SA is Authenticated

2.21.3. IKE SA 经过身份验证后的错误处理

After the IKE SA is authenticated, all requests having errors MUST result in a response notifying about the error.

IKE SA 经过身份验证后，所有有错误的请求都必须产生一个通知错误的响应。

In normal situations, there should not be cases where a valid response from one peer results in an error situation in the other peer, so there should not be any reason for a peer to send error messages to the other end except as a response. Because sending such error messages as an INFORMATIONAL exchange might lead to further errors that could cause loops, such errors SHOULD NOT be sent. If errors are seen that indicate that the peers do not have the same state, it might be good to delete the IKE SA to clean up state and start over.

在正常情况下，不应该存在来自一个对等方的有效响应导致另一个对等方出现错误的情况，因此除了作为响应，对等方不应该有任何理由向另一端发送错误消息。由于以信息交换方式发送此类错误消息可能会导致进一步的错误，从而导致循环，因此不应发送此类错误。如果发现错误表明对等点不具有相同的状态，那么最好删除 IKE SA 以清理状态并重新开始。

If a peer parsing a request notices that it is badly formatted (after it has passed the message authentication code checks and window checks) and it returns an

INVALID_SYNTAX notification, then this error notification is considered fatal in both peers, meaning that the IKE SA is deleted without needing an explicit Delete payload.

如果解析请求的对等方注意到其格式不正确（在通过消息身份验证代码检查和窗口检查后），并返回无效的_语法通知，则此错误通知在两个对等方中都被视为致命的，这意味着在不需要显式删除负载的情况下删除 IKE SA。

2.21.4. Error Handling Outside IKE SA

2.21.4. IKE SA 外部的错误处理

A node needs to limit the rate at which it will send messages in response to unprotected messages.

节点需要限制其发送消息以响应未受保护的消息的速率。

If a node receives a message on UDP port 500 or 4500 outside the context of an IKE SA known to it (and the message is not a request to start an IKE SA), this may be the result of a recent crash of the node. If the message is marked as a response, the node can audit the suspicious event but MUST NOT respond. If the message is marked as a request, the node can audit the suspicious event and MAY send a response. If a response is sent, the response MUST be sent to the IP address and port from where it came with the same IKE SPIs and the Message ID copied. The response MUST NOT be cryptographically protected and MUST contain an INVALID_IKE_SPI Notify payload. The INVALID_IKE_SPI notification indicates an IKE message was received with an unrecognized destination SPI; this usually indicates that the recipient has rebooted and forgotten the existence of an IKE SA.

如果节点在其已知的 IKE SA 上下文之外的 UDP 端口 500 或 4500 上接收到消息（并且该消息不是启动 IKE SA 的请求），这可能是节点最近崩溃的结果。如果消息标记为响应，则节点可以审核可疑事件，但不得响应。如果消息标记为请求，则节点可以审核可疑事件并发送响应。如果发送了响应，则必须将响应发送到 IP 地址和端口，该 IP 地址和端口来自相同的 IKE SPI 和复制的消息 ID。响应必须不受加密保护，并且必须包含无效的\u IKE\u SPI Notify 有效负载。无效的_IKE_SPI 通知表示接收到带有无法识别的目标 SPI 的 IKE 消息；这通常表示收件人已重新启动并忘记 IKE SA 的存在。

A peer receiving such an unprotected Notify payload MUST NOT respond and MUST NOT change the state of any existing SAs. The message might be a forgery or might be a response that a genuine correspondent was tricked into sending. A node should treat such a message (and also a network message like ICMP destination unreachable) as a hint that there might be problems with SAs to that IP address and should initiate a liveness check for any such IKE SA. An implementation SHOULD limit the frequency of such tests to avoid being tricked into participating in a DoS attack.

接收此类未受保护的 Notify 有效负载的对等方不得响应，也不得更改任何现有 SA 的状态。这条消息可能是伪造的，也可能是一位真正的通讯员被骗发送的回复。节点应将此类消息（以及类似 ICMP destination unreachable 的网络消息）视为指向该 IP 地址的 SA 可能存在问题的提示，并应启动任何此类 IKE SA 的活动性检查。实现应限制此类测试的频率，以避免被骗参与 DoS 攻击。

If an error occurs outside the context of an IKE request (e.g., the node is getting ESP messages on a nonexistent SPI), the node SHOULD initiate an INFORMATIONAL exchange with a Notify payload describing the problem.

如果在 IKE 请求的上下文之外发生错误（例如，节点在不存在的 SPI 上获取 ESP 消息），则节点应启动信息交换，并使用描述问题的 Notify 有效负载。

A node receiving a suspicious message from an IP address (and port, if NAT traversal is used) with which it has an IKE SA SHOULD send an IKE Notify payload in an IKE INFORMATIONAL exchange over that SA. The recipient MUST NOT change the state of any SAs as a result, but may wish to audit the event to aid in diagnosing malfunctions.

从具有 IKE SA 的 IP 地址（和端口，如果使用 NAT 遍历）接收可疑消息的节点应通过该 SA 在 IKE 信息交换中发送 IKE Notify 有效负载。接收者不得因此改变任何 SAs 的状态，但可能希望审核事件以帮助诊断故障。

2.22. IPComp

2.22. IP 压缩

Use of IP Compression [IP-COMP] can be negotiated as part of the setup of a Child SA. While IP Compression involves an extra header in each packet and a compression parameter index (CPI), the virtual "compression association" has no life outside the ESP or AH SA that contains it. Compression associations disappear when the corresponding ESP or AH SA goes away. It is not explicitly mentioned in any Delete payload.

IP 压缩[IP-COMP]的使用可以作为子 SA 设置的一部分进行协商。虽然 IP 压缩涉及每个数据包中的额外报头和压缩参数索引（CPI），但虚拟“压缩关联”在包含它的 ESP 或 AH SA 之外没有生

命。当相应的 ESP 或 AH SA 消失时，压缩关联消失。在任何删除有效负载中都没有明确提到它。

Negotiation of IP Compression is separate from the negotiation of cryptographic parameters associated with a Child SA. A node requesting a Child SA MAY advertise its support for one or more compression algorithms through one or more Notify payloads of type IPCOMP_SUPPORTED. This Notify message may be included only in a message containing an SA payload negotiating a Child SA and indicates a willingness by its sender to use IPComp on this SA. The response MAY indicate acceptance of a single compression algorithm with a Notify payload of type IPCOMP_SUPPORTED. These payloads MUST NOT occur in messages that do not contain SA payloads.

IP 压缩的协商独立于与子 SA 相关联的加密参数的协商。请求子 SA 的节点可以通过一个或多个类型为 IPCOMP_SUPPORTED 的 Notify 有效载荷来公布其对一个或多个压缩算法的支持。此通知消息只能包含在包含与子 SA 协商的 SA 有效负载的消息中，并表示其发送方愿意在此 SA 上使用 IPComp。响应可能表示接受支持 IPCOMP_类型的通知有效负载的单个压缩算法。这些有效负载不得出现在不包含 SA 有效负载的消息中。

The data associated with this Notify message includes a two-octet IPComp CPI followed by a one-octet Transform ID optionally followed by attributes whose length and format are defined by that Transform ID. A message proposing an SA may contain multiple IPCOMP_SUPPORTED notifications to indicate multiple supported algorithms. A message accepting an SA may contain at most one.

与此通知消息相关联的数据包括两个八位字节的 IPComp CPI，后跟一个八位字节的转换 ID（可选），后跟由该转换 ID 定义长度和格式的属性。建议 SA 的消息可能包含多个 IPComp 受支持的通知，以指示多个受支持的算法。接受 SA 的消息最多可以包含一个。

The Transform IDs are listed here. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

此处列出了变换 ID。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

```
Name            Number   Defined In
-------------------------------------
IPCOMP_OUI        1
```

IPCOMP_DEFLATE	2	RFC 2394
IPCOMP_LZS	3	RFC 2395
IPCOMP_LZJH	4	RFC 3051

Name	Number	Defined In
IPCOMP_OUI	1	
IPCOMP_DEFLATE	2	RFC 2394
IPCOMP_LZS	3	RFC 2395
IPCOMP_LZJH	4	RFC 3051

Although there has been discussion of allowing multiple compression algorithms to be accepted and to have different compression algorithms available for the two directions of a Child SA, implementations of this specification MUST NOT accept an IPComp algorithm that was not proposed, MUST NOT accept more than one, and MUST NOT compress using an algorithm other than one proposed and accepted in the setup of the Child SA.

尽管已经讨论过允许接受多个压缩算法，并允许在子 SA 的两个方向上使用不同的压缩算法，但本规范的实现不得接受未提出的 IPComp 算法，也不得接受多个，并且不得使用在子 SA 设置中提出并接受的算法以外的算法进行压缩。

A side effect of separating the negotiation of IPComp from cryptographic parameters is that it is not possible to propose multiple cryptographic suites and propose IP Compression with some of them but not others.

将 IPComp 的协商与加密参数分离的一个副作用是，不可能提出多个加密套件，也不可能提出使用其中一些套件的 IP 压缩，而不是其他套件。

In some cases, Robust Header Compression (ROHC) may be more appropriate than IP Compression. [ROHCV2] defines the use of ROHC with IKEv2 and IPsec.

在某些情况下，健壮的报头压缩（ROHC）可能比 IP 压缩更合适。[ROHCV2]定义了 ROHC 与 IKEv2 和 IPsec 的结合使用。

2.23. NAT Traversal

2.23. 内网互联

Network Address Translation (NAT) gateways are a controversial subject. This section briefly describes what they are and how they are likely to act on IKE traffic.

Many people believe that NATs are evil and that we should not design our protocols so as to make them work better. IKEv2 does specify some unintuitive processing rules in order that NATs are more likely to work.

网络地址转换（NAT）网关是一个有争议的话题。本节简要描述了它们是什么以及它们可能如何对 IKE 流量起作用。许多人认为 NAT 是邪恶的，我们不应该设计我们的协议来让它们更好地工作。IKEv2 确实指定了一些非直观的处理规则，以便 NAT 更容易工作。

NATs exist primarily because of the shortage of IPv4 addresses, though there are other rationales. IP nodes that are "behind" a NAT have IP addresses that are not globally unique, but rather are assigned from some space that is unique within the network behind the NAT but that are likely to be reused by nodes behind other NATs. Generally, nodes behind NATs can communicate with other nodes behind the same NAT and with nodes with globally unique addresses, but not with nodes behind other NATs. There are exceptions to that rule. When those nodes make connections to nodes on the real Internet, the

NAT 的存在主要是因为 IPv4 地址的短缺，尽管还有其他原因。NAT“后面”的 IP 节点的 IP 地址不是全局唯一的，而是从 NAT 后面的网络中唯一但可能被其他 NAT 后面的节点重用的某个空间分配的。通常，NAT 后面的节点可以与同一 NAT 后面的其他节点以及具有全局唯一地址的节点通信，但不能与其他 NAT 后面的节点通信。这条规则也有例外。当这些节点连接到真实 Internet 上的节点时

NAT gateway "translates" the IP source address to an address that will be routed back to the gateway. Messages to the gateway from the Internet have their destination addresses "translated" to the internal address that will route the packet to the correct endnode.

NAT 网关将 IP 源地址“转换”为将路由回网关的地址。从 Internet 发送到网关的消息将其目标地址“转换”为内部地址，该地址将数据包路由到正确的端节点。

NATs are designed to be "transparent" to endnodes. Neither software on the node behind the NAT nor the node on the Internet requires modification to communicate through the NAT. Achieving this transparency is more difficult with some protocols than with others. Protocols that include IP addresses of the endpoints within the payloads of the packet will fail unless the NAT gateway understands the protocol and modifies the internal references as well as those in the headers. Such knowledge is inherently unreliable, is a network layer violation, and often results in subtle problems.

NAT 被设计为对端节点“透明”。NAT 后面的节点上的软件和 Internet 上的节点都不需要修改才能通过 NAT 进行通信。某些协议比其他协议更难实现这种透明度。除非 NAT 网关理解协议并修改内部引用以及报头中的引用，否则包含数据包有效负载内端点 IP 地址的协议将失败。这些知识本质上是不可靠的，是违反网络层的，并且常常导致微妙的问题。

Opening an IPsec connection through a NAT introduces special problems. If the connection runs in transport mode, changing the IP addresses on packets will cause the checksums to fail and the NAT cannot correct the checksums because they are cryptographically protected. Even in tunnel mode, there are routing problems because transparently translating the addresses of AH and ESP packets requires special logic in the NAT and that logic is heuristic and unreliable in nature. For that reason, IKEv2 will use UDP encapsulation of IKE and ESP packets. This encoding is slightly less efficient but is easier for NATs to process. In addition, firewalls may be configured to pass UDP-encapsulated IPsec traffic but not plain, unencapsulated ESP/AH or vice versa.

通过 NAT 打开 IPsec 连接会带来特殊问题。如果连接在传输模式下运行，更改数据包上的 IP 地址将导致校验和失败，NAT 无法更正校验和，因为它们受到加密保护。即使在隧道模式下，也存在路由问题，因为透明地转换 AH 和 ESP 数据包的地址需要 NAT 中的特殊逻辑，并且该逻辑本质上是启发式的和不可靠的。因此，IKEv2 将使用 IKE 和 ESP 数据包的 UDP 封装。这种编码效率稍低，但 NAT 更容易处理。此外，防火墙可以配置为通过 UDP 封装的 IPsec 通信，但不能通过普通的、未封装的 ESP/AH，反之亦然。

It is a common practice of NATs to translate TCP and UDP port numbers as well as addresses and use the port numbers of inbound packets to decide which internal node should get a given packet. For this reason, even though IKE packets MUST be sent to and from UDP port 500 or 4500, they MUST be accepted coming from any port and responses MUST be sent to the port from whence they came. This is because the ports may be modified as the packets pass through NATs. Similarly, IP addresses of the IKE endpoints are generally not included in the IKE payloads because the payloads are cryptographically protected and could not be transparently modified by NATs.

NAT 的常见做法是转换 TCP 和 UDP 端口号以及地址，并使用入站数据包的端口号来决定哪个内部节点应获得给定数据包。因此，即使 IKE 数据包必须发送到 UDP 端口 500 或 4500，也必须接受来自任何端口的数据包，并且必须将响应发送到它们来自的端口。这是因为当数据包通过 NAT 时，端口可能会被修改。类似地，IKE 端点的 IP 地址通常不包括在 IKE 有效负载中，因为有效负载受到加密保护，并且不能被 NAT 透明地修改。

Port 4500 is reserved for UDP-encapsulated ESP and IKE. An IPsec endpoint that discovers a NAT between it and its correspondent (as described below) MUST send all subsequent traffic from port 4500, which NATs should not treat specially (as they might with port 500).

端口 4500 保留给 UDP 封装的 ESP 和 IKE。如果 IPsec 端点在其与对应方（如下所述）之间发现 NAT，则必须从端口 4500 发送所有后续流量，而 NAT 不应特别处理这些流量（如端口 500）。

An initiator can use port 4500 for both IKE and ESP, regardless of whether or not there is a NAT, even at the beginning of IKE. When either side is using port 4500, sending ESP with UDP encapsulation is not required, but understanding received UDP-encapsulated ESP packets

启动器可以将端口 4500 用于 IKE 和 ESP，无论是否存在 NAT，甚至在 IKE 开始时也是如此。当任何一方使用端口 4500 时，不需要发送带有 UDP 封装的 ESP，但需要理解接收到的 UDP 封装的 ESP 数据包

is required. UDP encapsulation MUST NOT be done on port 500. If Network Address Translation Traversal (NAT-T) is supported (that is, if NAT_DETECTION_*_IP payloads were exchanged during IKE_SA_INIT), all devices MUST be able to receive and process both UDP-encapsulated ESP and non-UDP-encapsulated ESP packets at any time. Either side can decide whether or not to use UDP encapsulation for ESP irrespective of the choice made by the other side. However, if a NAT is detected, both devices MUST use UDP encapsulation for ESP.

是必需的。UDP 封装不能在端口 500 上进行。如果支持网络地址转换遍历（NAT-T）（即，如果在 IKE_SA_INIT 期间交换了 NAT_检测*_IP 有效负载），则所有设备必须能够随时接收和处理 UDP 封装的 ESP 和非 UDP 封装的 ESP 数据包。任何一方都可以决定是否对 ESP 使用 UDP 封装，而不考虑另一方的选择。但是，如果检测到 NAT，两个设备都必须对 ESP 使用 UDP 封装。

The specific requirements for supporting NAT traversal [NATREQ] are listed below. Support for NAT traversal is optional. In this section only, requirements listed as MUST apply only to implementations supporting NAT traversal.

下面列出了支持 NAT 穿越[NATREQ]的具体要求。对 NAT 遍历的支持是可选的。仅在本节中，列出的要求必须仅适用于支持 NAT 遍历的实现。

o Both the IKE initiator and responder MUST include in their IKE_SA_INIT packets Notify payloads of type NAT_DETECTION_SOURCE_IP and

NAT_DETECTION_DESTINATION_IP. Those payloads can be used to detect if there is NAT between the hosts, and which end is behind the NAT. The location of the payloads in the IKE_SA_INIT packets is just after the Ni and Nr payloads (before the optional CERTREQ payload).

o IKE 启动器和响应程序都必须在其 IKE_SA_INIT 数据包中包含 NAT_检测_源_IP 和 NAT_检测_目的地_IP 类型的通知有效负载。这些有效负载可用于检测主机之间是否存在 NAT，以及哪一端位于 NAT 后面。IKE_SA_INIT 数据包中的有效负载位置正好位于 Ni 和 Nr 有效负载之后（在可选 CERTREQ 有效负载之前）。

o The data associated with the NAT_DETECTION_SOURCE_IP notification is a SHA-1 digest of the SPIs (in the order they appear in the header), IP address, and port from which this packet was sent. There MAY be multiple NAT_DETECTION_SOURCE_IP payloads in a message if the sender does not know which of several network attachments will be used to send the packet.

o 与 NAT_检测_源_IP 通知相关联的数据是 SPI 的 SHA-1 摘要（按照它们在报头中出现的顺序）、IP 地址和发送此数据包的端口。如果发送方不知道将使用多个网络附件中的哪一个发送数据包，则消息中可能存在多个 NAT_检测_源_IP 有效负载。

o The data associated with the NAT_DETECTION_DESTINATION_IP notification is a SHA-1 digest of the SPIs (in the order they appear in the header), IP address, and port to which this packet was sent.

o 与 NAT_检测_目的地_IP 通知相关联的数据是 SPI 的 SHA-1 摘要（按照它们在报头中出现的顺序）、IP 地址和该数据包发送到的端口。

o The recipient of either the NAT_DETECTION_SOURCE_IP or NAT_DETECTION_DESTINATION_IP notification MAY compare the supplied value to a SHA-1 hash of the SPIs, source or recipient IP address (respectively), address, and port, and if they don't match, it SHOULD enable NAT traversal. In the case there is a mismatch of the NAT_DETECTION_SOURCE_IP hash with all of the NAT_DETECTION_SOURCE_IP payloads received, the recipient MAY reject the connection attempt if NAT traversal is not supported. In the case of a mismatching NAT_DETECTION_DESTINATION_IP hash, it means that the system receiving the NAT_DETECTION_DESTINATION_IP payload is behind a NAT and that system SHOULD start sending keepalive packets as defined in [UDPENCAPS]; alternately, it MAY reject the connection attempt if NAT traversal is not supported.

o NAT_检测\u 源\u IP 或 NAT_检测\u 目的地\u IP 通知的收件人可以将提供的值与 SPI、源或收

件人 IP 地址（分别）、地址和端口的 SHA-1 哈希进行比较，如果它们不匹配，则应启用 NAT 遍历。如果 NAT_检测_源_IP 哈希与接收到的所有 NAT_检测_源_IP 有效负载不匹配，则如果不支持 NAT 遍历，则接收方可以拒绝连接尝试。在 NAT_检测_目的地_IP 哈希不匹配的情况下，这意味着接收 NAT_检测_目的地_IP 有效负载的系统位于 NAT 后面，并且该系统应开始发送[UDPENCAPS]中定义的 keepalive 数据包；或者，如果不支持 NAT 遍历，它可能会拒绝连接尝试。

o If none of the NAT_DETECTION_SOURCE_IP payload(s) received matches the expected value of the source IP and port found from the IP header of the packet containing the payload, it means that the system sending those payloads is behind a NAT (i.e., someone along the route changed the source address of the original packet to match the address of the NAT box). In this case, the system receiving the payloads should allow dynamic updates of the other systems' IP address, as described later.

o 如果接收到的 NAT_检测_源_IP 有效载荷与从包含有效载荷的数据包的 IP 报头中找到的源 IP 和端口的预期值均不匹配，则表示发送这些有效载荷的系统位于 NAT 后面（即，沿途有人更改了原始数据包的源地址以匹配 NAT 盒的地址）。在这种情况下，接收有效负载的系统应允许动态更新其他系统的 IP 地址，如下文所述。

o The IKE initiator MUST check the NAT_DETECTION_SOURCE_IP or NAT_DETECTION_DESTINATION_IP payloads if present, and if they do not match the addresses in the outer packet, MUST tunnel all future IKE and ESP packets associated with this IKE SA over UDP port 4500.

o IKE 启动器必须检查 NAT_检测_源_IP 或 NAT_检测_目的地_IP 有效负载（如果存在），如果它们与外部数据包中的地址不匹配，则必须通过 UDP 端口 4500 对与此 IKE SA 相关联的所有未来 IKE 和 ESP 数据包进行隧道传输。

o To tunnel IKE packets over UDP port 4500, the IKE header has four octets of zero prepended and the result immediately follows the UDP header. To tunnel ESP packets over UDP port 4500, the ESP header immediately follows the UDP header. Since the first four octets of the ESP header contain the SPI, and the SPI cannot validly be zero, it is always possible to distinguish ESP and IKE messages.

o 要通过 UDP 端口 4500 对 IKE 数据包进行隧道传输，IKE 报头有四个八位字节，前缀为零，结果紧跟在 UDP 报头之后。要通过 UDP 端口 4500 对 ESP 数据包进行隧道传输，ESP 报头将紧跟在 UDP 报头之后。由于 ESP 头的前四个八位字节包含 SPI，并且 SPI 不能有效地为零，因此始终可以区分 ESP 和 IKE 消息。

o Implementations MUST process received UDP-encapsulated ESP packets even when no NAT was detected.

o 即使未检测到 NAT，实现也必须处理接收到的 UDP 封装 ESP 数据包。

o The original source and destination IP address required for the transport mode TCP and UDP packet checksum fixup (see [UDPENCAPS]) are obtained from the Traffic Selectors associated with the exchange. In the case of transport mode NAT traversal, the Traffic Selectors MUST contain exactly one IP address, which is then used as the original IP address. This is covered in greater detail in Section 2.23.1.

o 传输模式 TCP 和 UDP 数据包校验和修复所需的原始源和目标 IP 地址（请参见[UDPENCAPS]）可从与交换机关联的流量选择器中获取。在传输模式 NAT 穿越的情况下，流量选择器必须恰好包含一个 IP 地址，然后将其用作原始 IP 地址。第 2.23.1 节详细介绍了这一点。

o There are cases where a NAT box decides to remove mappings that are still alive (for example, the keepalive interval is too long, or the NAT box is rebooted). This will be apparent to a host if it receives a packet whose integrity protection validates, but has a different port, address, or both from the one that was associated with the SA in the validated packet. When such a validated packet is found, a host that does not support other methods of recovery such as IKEv2 Mobility and Multihoming (MOBIKE) [MOBIKE], and that is not behind a NAT, SHOULD send all packets (including retransmission packets) to the IP address and port in the validated packet, and SHOULD store this as the new address and port combination for the SA (that is, they SHOULD dynamically update the address). A host behind a NAT SHOULD NOT do this type of dynamic address update if a validated packet has

o 有些情况下，NAT 盒决定删除仍处于活动状态的映射（例如，keepalive 间隔太长，或者 NAT 盒重新启动）。如果主机接收到完整性保护验证的数据包，但与验证数据包中与 SA 相关联的数据包具有不同的端口、地址或两者，则这对主机来说是显而易见的。当发现这样一个经验证的数据包时，不支持诸如 IKEv2 移动和多址（MOBIKE）[MOBIKE]等其他恢复方法且不在 NAT 后面的主机应将所有数据包（包括重传数据包）发送到经验证数据包中的 IP 地址和端口，并应将其存储为 SA 的新地址和端口组合（即，他们应动态更新地址）。如果已验证的数据包已更新，则 NAT 后面的主机不应执行这种类型的动态地址更新

different port and/or address values because it opens a possible DoS attack (such as allowing an attacker to break the connection with a single packet). Also, dynamic address update should only be done in response to a new packet; otherwise, an attacker can revert the addresses with old replayed packets. Because of this, dynamic updates can only be done safely if replay protection is enabled. When IKEv2 is used with MOBIKE, dynamically updating the addresses described above interferes with MOBIKE's way of recovering from the same situation. See Section 3.8 of [MOBIKE] for more information.

不同的端口和/或地址值，因为它会打开可能的 DoS 攻击（例如允许攻击者使用单个数据包中断连接）。此外，动态地址更新应该只在响应新数据包时进行；否则，攻击者可以使用旧的重播数据包还原地址。因此，只有启用重播保护，才能安全地进行动态更新。当 IKEv2 与 MOBIKE 一起使用时，动态更新上述地址会干扰 MOBIKE 从相同情况中恢复的方式。更多信息参见[MOBIKE]第 3.8 节。

2.23.1. Transport Mode NAT Traversal

2.23.1. 传输模式 NAT 穿越

Transport mode used with NAT Traversal requires special handling of the Traffic Selectors used in the IKEv2. The complete scenario looks like:

NAT 穿越使用的传输模式需要对 IKEv2 中使用的流量选择器进行特殊处理。完整的场景如下所示：

```
   +------+        +------+            +------+         +------+
   |Client| IP1    | NAT  | IPN1  IPN2 | NAT  |     IP2 |Server|
   |node  |<------>|  A   |<---------->|  B   |<------->|      |
   +------+        +------+            +------+         +------+
```

```
   +------+        +------+            +------+         +------+
   |Client| IP1    | NAT  | IPN1  IPN2 | NAT  |     IP2 |Server|
   |node  |<------>|  A   |<---------->|  B   |<------->|      |
   +------+        +------+            +------+         +------+
```

(Other scenarios are simplifications of this complex case, so this discussion uses the complete scenario.)

（其他场景是此复杂案例的简化，因此本讨论使用完整场景。）

In this scenario, there are two address translating NATs: NAT A and NAT B. NAT A is a dynamic NAT that maps the client's source address IP1 to IPN1. NAT B is a static NAT configured so that connections coming to IPN2 address are mapped to the gateway's address IP2, that is, IPN2 destination address is mapped to IP2. This allows the client to connect to a server by connecting to the IPN2. NAT B does not necessarily need to be a static NAT, but the client needs to know how to connect to the server, and it can only do that if it somehow knows the outer address of the NAT B, that is, the IPN2 address. If NAT B is a static NAT, then its address can be configured to the client's configuration. Another option would be to find it using some other protocol (like DNS), but that is outside of scope of IKEv2.

在这个场景中，有两个地址转换 NAT：NAT A 和 NAT B。NAT A 是一个动态 NAT，它将客户端的源地址 IP1 映射到 IPN1。NAT B 是一种静态 NAT，配置为使到 IPN2 地址的连接映射到网关的地址 IP2，即 IPN2 目标地址映射到 IP2。这允许客户端通过连接到 IPN2 连接到服务器。NAT B 不一定必须是静态 NAT，但客户机需要知道如何连接到服务器，并且只有在它知道 NAT B 的外部地址（即 IPN2 地址）时才能这样做。如果 NAT B 是静态 NAT，则其地址可以配置为客户端的配置。另一种选择是使用其他协议（如 DNS）查找它，但这超出了 IKEv2 的范围。

In this scenario, both the client and server are configured to use transport mode for the traffic originating from the client node and destined to the server.

在这种情况下，客户机和服务器都被配置为对源自客户机节点并发送到服务器的流量使用传输模式。

When the client starts creating the IKEv2 SA and Child SA for sending traffic to the server, it may have a triggering packet with source IP address of IP1, and a destination IP address of IPN2. Its Peer Authorization Database (PAD) and SPD needs to have a configuration matching those addresses (or wildcard entries covering them).

当客户端开始创建 IKEv2 SA 和子 SA 以向服务器发送流量时，它可能有一个源 IP 地址为 IP1、目标 IP 地址为 IPN2 的触发包。它的对等授权数据库（PAD）和 SPD 需要具有与这些地址匹配的配置（或包含这些地址的通配符条目）。

Because this is transport mode, it uses exactly same addresses as the Traffic Selectors and outer IP address of the IKE packets. For transport mode, it MUST use exactly one IP address in the TSi and TSr payloads. It can have multiple Traffic Selectors if it has, for example, multiple port ranges that it wants to negotiate, but all TSi entries must use the IP1-IP1 range as the IP addresses, and all TSr entries must have the IPN2-IPN2 range as IP addresses. The first Traffic Selector of TSi and TSr SHOULD have very specific Traffic Selectors including protocol and port numbers, such as from the packet triggering the request.

因为这是传输模式，所以它使用与 IKE 数据包的流量选择器和外部 IP 地址完全相同的地址。对于

传输模式，它必须在 TSi 和 TSr 有效负载中使用一个 IP 地址。例如，如果它有多个要协商的端口范围，它可以有多个流量选择器，但所有 TSi 条目必须使用 IP1-IP1 范围作为 IP 地址，并且所有 TSr 条目必须将 IPN2-IPN2 范围作为 IP 地址。TSi 和 TSr 的第一个流量选择器应该具有非常特定的流量选择器，包括协议和端口号，例如来自触发请求的数据包的流量选择器。

NAT A will then replace the source address of the IKE packet from IP1 to IPN1, and NAT B will replace the destination address of the IKE packet from IPN2 to IP2, so when the packet arrives to the server it will still have the exactly same Traffic Selectors that were sent by the client, but the IP address of the IKE packet has been replaced by IPN1 and IP2.

然后，NAT A 将 IKE 数据包的源地址从 IP1 替换为 IPN1，NAT B 将 IKE 数据包的目标地址从 IPN2 替换为 IP2，因此当数据包到达服务器时，它仍然具有与客户端发送的完全相同的流量选择器，但 IKE 数据包的 IP 地址已被 IPN1 和 IP2 替换。

When the server receives this packet, it normally looks in the Peer Authorization Database (PAD) described in RFC 4301 [IPSECARCH] based on the ID and then searches the SPD based on the Traffic Selectors. Because IP1 does not really mean anything to the server (it is the address client has behind the NAT), it is useless to do a lookup based on that if transport mode is used. On the other hand, the server cannot know whether transport mode is allowed by its policy before it finds the matching SPD entry.

当服务器收到此数据包时，它通常会根据 ID 查找 RFC 4301[IPSECARCH]中描述的对等授权数据库（PAD），然后根据流量选择器搜索 SPD。因为 IP1 对服务器来说并不意味着什么（它是客户端在 NAT 后面的地址），所以如果使用传输模式，那么基于它进行查找是没有用的。另一方面，服务器在找到匹配的 SPD 条目之前，无法知道其策略是否允许传输模式。

In this case, the server should first check that the initiator requested transport mode, and then do address substitution on the Traffic Selectors. It needs to first store the old Traffic Selector IP addresses to be used later for the incremental checksum fixup (the IP address in the TSi can be stored as the original source address and the IP address in the TSr can be stored as the original destination address). After that, if the other end was detected as being behind a NAT, the server replaces the IP address in TSi payloads with the IP address obtained from the source address of the IKE packet received (that is, it replaces IP1 in TSi with IPN1). If the server's end was detected to be behind NAT, it replaces the IP address in the TSr payloads with the IP address obtained from the destination address of the IKE packet received (that is, it replaces IPN2 in TSr with IP2).

在这种情况下，服务器应首先检查启动器是否请求传输模式，然后在流量选择器上进行地址替换。它需要首先存储旧的流量选择器 IP 地址，以便稍后用于增量校验和修复（TSi 中的 IP 地址可以存储为原始源地址，TSr 中的 IP 地址可以存储为原始目标地址）。之后，如果检测到另一端在 NAT 后面，则服务器将 TSi 有效负载中的 IP 地址替换为从接收到的 IKE 数据包的源地址获得的 IP 地址（即，它将 TSi 中的 IP1 替换为 IPN1）。如果检测到服务器端在 NAT 后面，它将用从接收到的 IKE 数据包的目标地址获得的 IP 地址替换 TSr 有效负载中的 IP 地址（即，它将 TSr 中的 IPN2 替换为 IP2）。

After this address substitution, both the Traffic Selectors and the IKE UDP source/destination addresses look the same, and the server does SPD lookup based on those new Traffic Selectors. If an entry is found and it allows transport mode, then that entry is used. If an entry is found but it does not allow transport mode, then the server MAY undo the address substitution and redo the SPD lookup using the

在这个地址替换之后，流量选择器和 IKE UDP 源/目标地址看起来都相同，服务器根据这些新的流量选择器进行 SPD 查找。如果找到一个条目并允许传输模式，则使用该条目。如果找到条目但不允许传输模式，则服务器可以撤消地址替换，并使用

original Traffic Selectors. If the second lookup succeeds, the server will create an SA in tunnel mode using real Traffic Selectors sent by the other end.

原始流量选择器。如果第二次查找成功，服务器将使用另一端发送的真实流量选择器在隧道模式下创建 SA。

This address substitution in transport mode is needed because the SPD is looked up using the addresses that will be seen by the local host. This also will make sure the Security Association Database (SAD) entries for the tunnel exit checks and return packets is added using the addresses as seen by the local operating system stack.

在传输模式下需要这种地址替换，因为使用本地主机将看到的地址查找 SPD。这还将确保使用本地操作系统堆栈看到的地址添加用于隧道出口检查和返回数据包的安全关联数据库（SAD）条目。

The most common case is that the server's SPD will contain wildcard entries matching any addresses, but this also allows making different SPD entries, for example, for different known NATs' outer addresses.

最常见的情况是，服务器的 SPD 将包含与任何地址匹配的通配符条目，但这也允许创建不同的 SPD 条目，例如，针对不同已知 NAT 的外部地址。

After the SPD lookup, the server will do Traffic Selector narrowing based on the SPD entry it found. It will again use the already substituted Traffic Selectors, and it will thus send back Traffic Selectors having IPN1 and IP2 as their IP addresses; it can still narrow down the protocol number or port ranges used by the Traffic Selectors. The SAD entry created for the Child SA will have the addresses as seen by the server, namely IPN1 and IP2.

SPD 查找后，服务器将根据找到的 SPD 条目进行流量选择器缩小。它将再次使用已经替换的流量选择器，并因此将发送回具有 IPN1 和 IP2 作为其 IP 地址的流量选择器；它仍然可以缩小流量选择器使用的协议号或端口范围。为子 SA 创建的 SAD 条目将具有服务器看到的地址，即 IPN1 和 IP2。

When the client receives the server's response to the Child SA, it will do similar processing. If the transport mode SA was created, the client can store the original returned Traffic Selectors as original source and destination addresses. It will replace the IP addresses in the Traffic Selectors with the ones from the IP header of the IKE packet: it will replace IPN1 with IP1 and IP2 with IPN2. Then, it will use those Traffic Selectors when verifying the SA against sent Traffic Selectors, and when installing the SAD entry.

当客户端接收到服务器对子 SA 的响应时，它将执行类似的处理。如果创建了传输模式 SA，客户端可以将原始返回的流量选择器存储为原始源地址和目标地址。它将用 IKE 数据包的 IP 头中的 IP 地址替换流量选择器中的 IP 地址：它将用 IP1 替换 IPN1，用 IPN2 替换 IP2。然后，在根据发送的流量选择器验证 SA 时，以及在安装 SAD 条目时，它将使用这些流量选择器。

A summary of the rules for NAT traversal in transport mode is:

传输模式下 NAT 穿越的规则总结如下：

For the client proposing transport mode:

对于提议运输模式的客户：

- The TSi entries MUST have exactly one IP address, and that MUST match the source address of the IKE SA.

- TSi 条目必须只有一个 IP 地址，并且必须与 IKE SA 的源地址匹配。

- The TSr entries MUST have exactly one IP address, and that MUST match the destination address of the IKE SA.

- TSr 条目必须只有一个 IP 地址，并且必须与 IKE SA 的目标地址匹配。

- The first TSi and TSr Traffic Selectors SHOULD have very specific Traffic Selectors including protocol and port numbers, such as from the packet triggering the request.

- 第一个 TSi 和 TSr 流量选择器应该具有非常特定的流量选择器，包括协议和端口号，例如来自触发请求的数据包的流量选择器。

- There MAY be multiple TSi and TSr entries.

- 可能有多个 TSi 和 TSr 条目。

- If transport mode for the SA was selected (that is, if the server included USE_TRANSPORT_MODE notification in its response):

- 如果选择了 SA 的传输模式（即，如果服务器在其响应中包含使用传输模式通知）：

- Store the original Traffic Selectors as the received source and destination address.

- 将原始流量选择器存储为接收的源地址和目标地址。

- If the server is behind a NAT, substitute the IP address in the TSr entries with the remote address of the IKE SA.

- 如果服务器位于 NAT 后面，则用 IKE SA 的远程地址替换 TSr 条目中的 IP 地址。

- If the client is behind a NAT, substitute the IP address in the TSi entries with the local address of the IKE SA.

- 如果客户端位于 NAT 后面，则用 IKE SA 的本地地址替换 TSi 条目中的 IP 地址。

- Do address substitution before using those Traffic Selectors for anything other than storing original content of them. This includes verification that Traffic Selectors were narrowed correctly by the other end, creation of the SAD entry, and so on.

- 在使用这些流量选择器进行存储原始内容以外的任何操作之前，请先进行地址替换。这包括验证另一端是否正确缩小了流量选择器，创建 SAD 条目，等等。

For the responder, when transport mode is proposed by client:

对于响应者，当客户提议传输模式时：

- Store the original Traffic Selector IP addresses as received source and destination address, in case undo address substitution is needed, to use as the "real source and destination address" specified by [UDPENCAPS], and for TCP/UDP checksum fixup.

- 将原始流量选择器 IP 地址存储为接收的源地址和目标地址，以防需要撤销地址替换，用作[UDPENCAPS]指定的“真实源地址和目标地址”，并用于 TCP/UDP 校验和修复。

- If the client is behind a NAT, substitute the IP address in the TSi entries with the remote address of the IKE SA.

- 如果客户端位于 NAT 后面，则用 IKE SA 的远程地址替换 TSi 条目中的 IP 地址。

- If the server is behind a NAT, substitute the IP address in the TSr entries with the local address of the IKE SA.

- 如果服务器位于 NAT 后面，则用 IKE SA 的本地地址替换 TSr 条目中的 IP 地址。

- Do PAD and SPD lookup using the ID and substituted Traffic Selectors.

- 使用 ID 和替换的流量选择器查找 PAD 和 SPD。

- If no SPD entry was found, or if found SPD entry does not allow transport mode, undo the Traffic Selector substitutions. Do PAD and SPD lookup again using the ID and original Traffic Selectors, but also searching for tunnel mode SPD entry (that is, fall back to tunnel mode).

- 如果未找到 SPD 条目，或者如果找到的 SPD 条目不允许传输模式，请撤消交通选择器替换。使用 ID 和原始流量选择器再次执行 PAD 和 SPD 查找，但也要搜索隧道模式 SPD 条目（即，返回到隧道模式）。

- However, if a transport mode SPD entry was found, do normal traffic selection narrowing based on the substituted Traffic Selectors and SPD entry. Use the resulting Traffic Selectors when creating SAD entries, and when sending Traffic Selectors back to the client.

- 但是，如果找到传输模式 SPD 条目，则根据替换的流量选择器和 SPD 条目进行正常流量选择缩小。创建 SAD 条目以及将流量选择器发送回客户端时，请使用生成的流量选择器。

2.24. Explicit Congestion Notification (ECN)

2.24. 显式拥塞通知（ECN）

When IPsec tunnels behave as originally specified in [IPSECARCH-OLD], ECN usage is not appropriate for the outer IP headers because tunnel decapsulation processing discards ECN congestion indications to the detriment of the network. ECN support for IPsec tunnels for IKEv1- based IPsec requires multiple operating modes and negotiation (see [ECN]). IKEv2 simplifies this situation by requiring that ECN be usable in the outer IP headers of all tunnel mode Child SAs created by IKEv2. Specifically, tunnel encapsulators and decapsulators for all tunnel mode SAs created by IKEv2 MUST support the ECN full-functionality option for tunnels specified in [ECN] and MUST implement the tunnel encapsulation and decapsulation processing specified in [IPSECARCH] to prevent discarding of ECN congestion indications.

当 IPsec 隧道的行为与[IPSECARCH-OLD]中最初指定的相同时，ECN 的使用不适合外部 IP 头，因为隧道解除封装处理会丢弃 ECN 拥塞指示，从而损害网络。ECN 支持基于 IKEv1 的 IPsec 的 IPsec 隧道需要多种操作模式和协商（参见[ECN]）。IKEv2 通过要求 ECN 在 IKEv2 创建的所有隧道模式子 SA 的外部 IP 头中可用，简化了这种情况。具体而言，IKEv2 创建的所有隧道模式 SA 的隧道封装器和去封装器必须支持[ECN]中指定的隧道的 ECN 完整功能选项，并且必须实施[IPSECARCH]中指定的隧道封装和去封装处理，以防止丢弃 ECN 拥塞指示。

2.25. Exchange Collisions

2.25. 交换碰撞

Because IKEv2 exchanges can be initiated by either peer, it is possible that two exchanges affecting the same SA partly overlap. This can lead to a situation where the SA state information is temporarily not synchronized, and a peer can receive a request that it cannot process in a normal fashion.

由于 IKEv2 交换可以由任何一个对等方发起，因此影响同一 SA 的两个交换可能部分重叠。这可能会导致 SA 状态信息暂时不同步的情况，并且对等方可能会收到无法以正常方式处理的请求。

Obviously, using a window size greater than 1 leads to more complex situations, especially if requests are processed out of order. This section concentrates on problems that can arise even with a window size of 1, and recommends solutions.

显然，使用大于 1 的窗口大小会导致更复杂的情况，尤其是在处理请求的顺序不正确的情况下。本节主要介绍窗口大小为 1 时可能出现的问题，并推荐解决方案。

A TEMPORARY_FAILURE notification SHOULD be sent when a peer receives a request that cannot be completed due to a temporary condition such as a rekeying operation. When a peer receives a TEMPORARY_FAILURE notification, it MUST NOT immediately retry the operation; it MUST wait so that the sender may complete whatever operation caused the temporary condition. The recipient MAY retry the request one or more times over a period of several minutes. If a peer continues to receive TEMPORARY_FAILURE on the same IKE SA after several minutes, it SHOULD conclude that the state information is out of sync and close the IKE SA.

当对等方收到由于临时条件（如密钥更新操作）而无法完成的请求时，应发送临时_失败通知。对等方收到临时_失败通知时，不得立即重试该操作；它必须等待，以便发送方可以完成导致临时条件的任何操作。收件人可以在几分钟内重试请求一次或多次。如果对等方在几分钟后继续在同一IKE SA 上接收到临时_故障，则应断定状态信息不同步并关闭 IKE SA。

A CHILD_SA_NOT_FOUND notification SHOULD be sent when a peer receives a request to rekey a Child SA that does not exist. The SA that the initiator attempted to rekey is indicated by the SPI field in the Notify payload, which is copied from the SPI field in the REKEY_SA notification. A peer that receives a CHILD_SA_NOT_FOUND notification SHOULD silently delete the Child SA (if it still exists) and send a request to create a new Child SA from scratch (if the Child SA does not yet exist).

当对等方收到重新设置不存在的子 SA 密钥的请求时，应发送子 SA 未找到通知。启动器尝试重新设置密钥的 SA 由 Notify 有效负载中的 SPI 字段指示，该字段从重新设置密钥 SA 通知中的 SPI 字段复制而来。接收到子 SA_NOT_FOUND 通知的对等方应以静默方式删除子 SA（如果它仍然存在），并发送请求从头开始创建新的子 SA（如果子 SA 还不存在）。

2.25.1. Collisions while Rekeying or Closing Child SAs

2.25.1. 重新设置或关闭子 SAs 时发生冲突

If a peer receives a request to rekey a Child SA that it is currently trying to close, it SHOULD reply with TEMPORARY_FAILURE. If a peer receives a request to rekey a Child SA that it is currently rekeying, it SHOULD reply as usual, and SHOULD prepare to close redundant SAs later based on the nonces (see Section 2.8.1). If a peer receives a request to rekey a Child SA that does not exist, it SHOULD reply with CHILD_SA_NOT_FOUND.

如果一个对等方收到一个请求，要求对其当前正试图关闭的子 SA 重新设置密钥，则该对等方应以临时_失败进行回复。如果对等方收到一个请求，要求对其当前正在重新设置密钥的子 SA 重新设置密钥，则该对等方应照常回复，并应准备稍后根据当前值关闭冗余 SA（参见第 2.8.1 节）。如果一个对等方收到一个请求，要求重新设置一个不存在的子 SA 的密钥，它应该用 Child_SA_not_FOUND 来回复。

If a peer receives a request to close a Child SA that it is currently trying to close, it SHOULD reply without a Delete payload (see Section 1.4.1). If a peer receives a request to close a Child SA that it is currently rekeying, it SHOULD reply as usual, with a Delete payload. If a peer receives a request to close a Child SA that does not exist, it SHOULD reply without a Delete payload.

如果对等方收到关闭其当前试图关闭的子 SA 的请求，则其应在没有删除有效负载的情况下进行回复（参见第 1.4.1 节）。如果对等方收到关闭其当前正在重新设置密钥的子 SA 的请求，它应该像往常一样使用删除负载进行回复。如果对等方收到关闭不存在的子 SA 的请求，它应该在没有删除负载的情况下进行回复。

If a peer receives a request to rekey the IKE SA, and it is currently creating, rekeying, or closing a Child SA of that IKE SA, it SHOULD reply with TEMPORARY_FAILURE.

如果对等方接收到对 IKE SA 重新设置密钥的请求，并且它当前正在创建、重新设置密钥或关闭该 IKE SA 的子 SA，那么它应该以临时_失败的方式进行响应。

2.25.2. Collisions while Rekeying or Closing IKE SAs

2.25.2. 重新设置或关闭 IKE SAs 时发生冲突

If a peer receives a request to rekey an IKE SA that it is currently rekeying, it SHOULD reply as usual, and SHOULD prepare to close redundant SAs and move inherited Child SAs later based on the nonces (see Section 2.8.2). If a peer receives a request to rekey an IKE SA that it is currently trying to close, it SHOULD reply with TEMPORARY_FAILURE.

如果对等方接收到一个请求，要求重新设置其当前正在重新设置密钥的 IKE SA 的密钥，则该对等方应照常回复，并应准备关闭冗余 SA，并在以后根据当前值移动继承的子 SA（参见第 2.8.2 节）。如果对等方收到一个请求，要求对其当前正试图关闭的 IKE SA 重新设置密钥，则该对等方应以临时_失败进行回复。

If a peer receives a request to close an IKE SA that it is currently rekeying, it SHOULD reply as usual, and forget about its own rekeying request. If a peer receives a request to close an IKE SA that it is currently trying to close, it SHOULD reply as usual, and forget about its own close request.

如果对等方收到关闭其当前正在密钥更新的 IKE SA 的请求，它应该像往常一样回复，并忘记自己的密钥更新请求。如果对等方收到关闭其当前试图关闭的 IKE SA 的请求，它应该像往常一样回复，并忘记自己的关闭请求。

If a peer receives a request to create or rekey a Child SA when it is currently rekeying the IKE SA, it SHOULD reply with TEMPORARY_FAILURE. If a peer receives a request to delete a Child SA when it is currently rekeying the IKE SA, it SHOULD reply as usual, with a Delete payload.

如果对等方在当前为 IKE SA 重新设置密钥时收到创建或重新设置子 SA 密钥的请求，则其应以临时_失败的方式进行回复。如果对等方在当前为 IKE SA 重新设置密钥时收到删除子 SA 的请求，它应该像往常一样使用删除负载进行回复。

3. Header and Payload Formats

3. 标题和有效负载格式

In the tables in this section, some cryptographic primitives and configuration attributes are marked as "UNSPECIFIED". These are items for which there are no known specifications and therefore interoperability is currently impossible. A future specification may

在本节的表中，一些加密原语和配置属性被标记为“未指定”。这些项目没有已知的规范，因此目前不可能实现互操作性。未来的规范可能

describe their use, but until such specification is made, implementations SHOULD NOT attempt to use items marked as "UNSPECIFIED" in implementations that are meant to be interoperable.

描述它们的用途，但在制定此类规范之前，实现不应尝试在旨在互操作的实现中使用标记为“未指定”的项。

3.1. The IKE Header

3.1. IKE 头

IKE messages use UDP ports 500 and/or 4500, with one IKE message per UDP datagram. Information from the beginning of the packet through the UDP header is largely ignored except that the IP addresses and UDP ports from the headers are reversed and used for return packets. When sent on UDP port 500, IKE messages begin immediately following the UDP header. When sent on UDP port 4500, IKE messages have prepended four octets of zero. These four octets of zeros are not part of the IKE message and are not included in any of the length fields or checksums defined by IKE. Each IKE message begins with the IKE header, denoted HDR in this document. Following the header are one or more IKE payloads each identified by a "Next Payload" field in the preceding payload. Payloads are identified in the order in which they appear in an IKE message by looking in the "Next Payload" field in the IKE header, and subsequently according to the "Next Payload" field in the IKE payload itself until a "Next Payload" field of zero indicates that no payloads follow. If a payload of type "Encrypted" is found, that payload is decrypted and its contents parsed as additional payloads. An Encrypted payload MUST be the last payload in a packet and an Encrypted payload MUST NOT contain another Encrypted payload.

IKE 消息使用 UDP 端口 500 和/或 4500，每个 UDP 数据报有一条 IKE 消息。除了来自报头的 IP 地址和 UDP 端口被反转并用于返回数据包之外，从数据包开始到 UDP 报头的信息基本上被忽略。当在 UDP 端口 500 上发送时，IKE 消息在 UDP 报头之后立即开始。当在 UDP 端口 4500 上发送时，IKE 消息已在四个八位字节前加上零。这四个八位字节的零不是 IKE 消息的一部分，也不包括在 IKE 定义的任何长度字段或校验和中。每个 IKE 消息都以 IKE 头开始，在本文档中表示为 HDR。在报头之后是一个或多个 IKE 有效载荷，每个有效载荷由前一有效载荷中的“下一有效载荷”字段标识。通过查看 IKE 报头中的“下一个有效载荷”字段，然后根据 IKE 有效载荷本身中的“下一个有效载荷”字段，按照它们在 IKE 消息中出现的顺序识别有效载荷，直到“下一个有效载荷”字段为零表示没有有效载荷跟随。如果找到“加密”类型的有效负载，则该有效负载将被解密，其内容将被解析为附加有效负载。加密的有效负载必须是数据包中的最后一个有效负载，并且加密的有效负载不得包含另一个加密的有效负载。

The responder's SPI in the header identifies an instance of an IKE Security Association. It is therefore possible for a single instance of IKE to multiplex distinct sessions with multiple peers, including multiple sessions per peer.

标头中响应者的 SPI 标识 IKE 安全关联的实例。因此，IKE 的单个实例可以与多个对等方复用不同的会话，包括每个对等方的多个会话。

All multi-octet fields representing integers are laid out in big endian order (also known as "most significant byte first", or "network byte order").

所有表示整数的多个八位字节字段均按大端顺序排列（也称为“最高有效字节优先”或“网络字节顺序”）。

The format of the IKE header is shown in Figure 4.

IKE 头的格式如图 4 所示。

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IKE SA Initiator's SPI                     |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IKE SA Responder's SPI                |
|                                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Payload | MjVer | MnVer | Exchange Type |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Message ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Length                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IKE SA Initiator's SPI                |
|                                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    IKE SA Responder's SPI                |
|                                                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Next Payload | MjVer | MnVer | Exchange Type |     Flags     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Message ID                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Length                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4: IKE Header Format

图 4:IKE 头格式

o Initiator's SPI (8 octets) - A value chosen by the initiator to identify a unique IKE Security Association. This value MUST NOT be zero.

o 发起者的 SPI（8 个八位字节）-发起者选择用于标识唯一 IKE 安全关联的值。此值不能为零。

o Responder's SPI (8 octets) - A value chosen by the responder to identify a unique IKE Security Association. This value MUST be zero in the first message of an IKE initial exchange (including repeats of that message including a cookie).

o 响应者的 SPI（8 个八位字节）-响应者选择的一个值，用于标识唯一的 IKE 安全关联。IKE 初始交换的第一条消息（包括包含 cookie 的该消息的重复）中的该值必须为零。

o Next Payload (1 octet) - Indicates the type of payload that immediately follows the header. The format and value of each payload are defined below.

o 下一个有效负载（1个八位字节）-指示紧跟在报头之后的有效负载类型。每个有效载荷的格式和值定义如下。

o Major Version (4 bits) - Indicates the major version of the IKE protocol in use. Implementations based on this version of IKE MUST set the major version to 2. Implementations based on previous versions of IKE and ISAKMP MUST set the major version to 1. Implementations based on this version of IKE MUST reject or ignore messages containing a version number greater than 2 with an INVALID_MAJOR_VERSION notification message as described in Section 2.5.

o 主要版本（4位）-表示正在使用的IKE协议的主要版本。基于此版本的IKE的实现必须将主版本设置为2。基于以前版本的IKE和ISAKMP的实现必须将主版本设置为1。基于此版本的IKE的实现必须拒绝或忽略包含版本号大于2的消息以及第2.5节所述的无效的主版本通知消息。

o Minor Version (4 bits) - Indicates the minor version of the IKE protocol in use. Implementations based on this version of IKE MUST set the minor version to 0. They MUST ignore the minor version number of received messages.

o 次要版本（4位）-表示正在使用的IKE协议的次要版本。基于此版本的IKE的实现必须将次要版本设置为0。他们必须忽略收到的消息的次要版本号。

o Exchange Type (1 octet) - Indicates the type of exchange being used. This constrains the payloads sent in each message in an exchange. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

o 交换类型（1个八位字节）-表示正在使用的交换类型。这将限制在exchange中的每条消息中发送的有效负载。下表中的值仅为截至RFC 4306发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Exchange Type	Value
IKE_SA_INIT	34
IKE_AUTH	35
CREATE_CHILD_SA	36
INFORMATIONAL	37

Exchange Type	Value
IKE_SA_INIT	34
IKE_AUTH	35
CREATE_CHILD_SA	36
INFORMATIONAL	37

o Flags (1 octet) - Indicates specific options that are set for the message. Presence of options is indicated by the appropriate bit in the flags field being set. The bits are as follows:

o 标志（1 个八位字节）-表示为消息设置的特定选项。选项的存在由正在设置的标志字段中的相应位表示。位如下所示：

```
+-+-+-+-+-+-+-+-+
|X|X|R|V|I|X|X|X|
+-+-+-+-+-+-+-+-+
```

```
+-+-+-+-+-+-+-+-+
|X|X|R|V|I|X|X|X|
+-+-+-+-+-+-+-+-+
```

In the description below, a bit being 'set' means its value is '1', while 'cleared' means its value is '0'. 'X' bits MUST be cleared when sending and MUST be ignored on receipt.

在下面的描述中，“设置”表示其值为“1”，而“清除”表示其值为“0”发送时必须清除 X'位，接收时必须忽略 X'位。

* R (Response) - This bit indicates that this message is a response to a message containing the same Message ID. This bit MUST be cleared in all request messages and MUST be set in all responses. An IKE endpoint MUST NOT generate a response to a message that is marked as being a response (with one exception; see Section 2.21.2).

* R（响应）-此位表示此消息是对包含相同消息 ID 的消息的响应。此位必须在所有请求消息中清除，并且必须在所有响应中设置。IKE 端点不得对标记为响应的消息生成响应（有一个例外；请参见第 2.21.2 节）。

* V (Version) - This bit indicates that the transmitter is capable of speaking a higher

major version number of the protocol than the one indicated in the major version number field. Implementations of IKEv2 MUST clear this bit when sending and MUST ignore it in incoming messages.

* V（版本）-此位表示变送器能够说出比主版本号字段中指示的更高的协议主版本号。IKEv2 的实现在发送时必须清除该位，并且在传入消息中必须忽略该位。

* I (Initiator) - This bit MUST be set in messages sent by the original initiator of the IKE SA and MUST be cleared in messages sent by the original responder. It is used by the recipient to determine which eight octets of the SPI were generated by the recipient. This bit changes to reflect who initiated the last rekey of the IKE SA.

* I（发起方）-此位必须在 IKE SA 的原始发起方发送的消息中设置，并且必须在原始响应方发送的消息中清除。接收者使用它来确定接收者生成了哪八个 SPI 八位字节。此位更改以反映谁发起了 IKE SA 的最后一次重新密钥。

o Message ID (4 octets, unsigned integer) - Message identifier used to control retransmission of lost packets and matching of requests and responses. It is essential to the security of the protocol because it is used to prevent message replay attacks. See Sections 2.1 and 2.2.

o 消息 ID（4 个八位字节，无符号整数）–用于控制丢失数据包的重新传输以及请求和响应的匹配的消息标识符。它对协议的安全性至关重要，因为它用于防止消息重放攻击。见第 2.1 节和第 2.2 节。

o Length (4 octets, unsigned integer) - Length of the total message (header + payloads) in octets.

o 长度（4 个八位字节，无符号整数）-以八位字节为单位的总消息长度（头+有效负载）。

3.2. Generic Payload Header

3.2. 通用有效载荷头

Each IKE payload defined in Sections 3.3 through 3.16 begins with a generic payload header, shown in Figure 5. Figures for each payload below will include the generic payload header, but for brevity, the description of each field will be omitted.

第 3.3 节至第 3.16 节中定义的每个 IKE 有效负载都以通用有效负载头开始，如图 5 所示。下面每个有效载荷的图将包括通用有效载荷标题，但为简洁起见，将省略每个字段的描述。

```
                    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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Next Payload  |C|  RESERVED   |         Payload Length        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                      1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Next Payload  |C|  RESERVED   |         Payload Length        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5: Generic Payload Header

图 5：通用有效负载标题

The Generic Payload Header fields are defined as follows:

通用有效负载标头字段定义如下：

o Next Payload (1 octet) - Identifier for the payload type of the next payload in the message. If the current payload is the last in the message, then this field will be 0. This field provides a "chaining" capability whereby additional payloads can be added to a message by appending each one to the end of the message and setting the "Next Payload" field of the preceding payload to indicate the new payload's type. An Encrypted payload, which must always be the last payload of a message, is an exception. It contains data structures in the format of additional payloads. In the header of an Encrypted payload, the Next Payload field is set to the payload type of the first contained payload (instead of 0); conversely, the Next Payload field of the last contained payload is set to zero). The payload type values are listed here. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

o 下一个有效负载（1 个八位字节）-消息中下一个有效负载的有效负载类型的标识符。如果当前有效负载是消息中的最后一个，则此字段将为 0。此字段提供“链接”功能，通过将每个有效负载追加到消息末尾，并设置前一有效负载的“下一有效负载”字段以指示新有效负载的类型，可以向消息添加额外的有效负载。加密的有效负载（必须始终是消息的最后一个有效负载）是一个例外。它包含附加有效载荷格式的数据结构。在加密有效载荷的报头中，下一有效载荷字段被设置为第一个包含的有效载荷的有效载荷类型（而不是 0）；相反，最后包含的有效负载的下一个有效负

载字段设置为零）。此处列出了有效负载类型值。下表中的值仅为截至RFC 4306发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Next Payload Type	Notation	Value
No Next Payload		0
Security Association	SA	33
Key Exchange	KE	34
Identification - Initiator	IDi	35
Identification - Responder	IDr	36
Certificate	CERT	37
Certificate Request	CERTREQ	38
Authentication	AUTH	39
Nonce	Ni, Nr	40
Notify	N	41
Delete	D	42
Vendor ID	V	43
Traffic Selector - Initiator	TSi	44
Traffic Selector - Responder	TSr	45
Encrypted and Authenticated	SK	46
Configuration	CP	47
Extensible Authentication	EAP	48

Next Payload Type	Notation	Value
No Next Payload		0
Security Association	SA	33
Key Exchange	KE	34
Identification - Initiator	IDi	35
Identification - Responder	IDr	36
Certificate	CERT	37
Certificate Request	CERTREQ	38
Authentication	AUTH	39
Nonce	Ni, Nr	40
Notify	N	41
Delete	D	42
Vendor ID	V	43
Traffic Selector - Initiator	TSi	44
Traffic Selector - Responder	TSr	45
Encrypted and Authenticated	SK	46
Configuration	CP	47
Extensible Authentication	EAP	48

(Payload type values 1-32 should not be assigned in the future so that there is no overlap with the code assignments for IKEv1.)

（未来不应分配有效负载类型值 1-32，以便不会与 IKEv1 的代码分配重叠。）

o Critical (1 bit) - MUST be set to zero if the sender wants the recipient to skip this payload if it does not understand the payload type code in the Next Payload field of the previous payload. MUST be set to one if the sender wants the recipient to reject this entire message if it does not understand the payload type. MUST be ignored by the recipient if the recipient understands the payload type code. MUST be set to zero for payload types defined in this document. Note that the critical bit applies to the current payload rather than the "next" payload whose type code appears in the first octet. The reasoning behind not setting the critical bit for payloads defined in this document is that all implementations MUST understand all payload types defined in this document and therefore must ignore the critical bit's value. Skipped payloads are expected to have valid Next Payload and Payload Length fields. See Section 2.5 for more information on this bit.

o 临界（1 位）-如果发送方不理解上一个有效负载的下一个有效负载字段中的有效负载类型代码，则发送方希望接收方跳过此有效负载，则必须将临界（1 位）设置为零。如果发件人希望收件人在不了解有效负载类型的情况下拒绝整个邮件，则必须将设置为 1。如果收件人理解有效负载类型代码，则收件人必须忽略此项。对于本文档中定义的有效负载类型，必须将设置为零。请注意，关键位适用于当前有效负载，而不是类型代码出现在第一个八位字节中的“下一个”有效负载。没有为本文档中定义的有效负载设置关键位的原因是，所有实现必须理解本文档中定义的所有有效负载类型，因此必须忽略关键位的值。跳过的有效负载应具有有效的下一个有效负载和有效负载长度字段。有关此位的更多信息，请参见第 2.5 节。

o RESERVED (7 bits) - MUST be sent as zero; MUST be ignored on receipt.

o 保留（7 位）-必须作为零发送；必须在收到时忽略。

o Payload Length (2 octets, unsigned integer) - Length in octets of the current payload, including the generic payload header.

o 有效负载长度（2 个八位字节，无符号整数）–当前有效负载的长度（以八位字节为单位），包括通用有效负载标头。

Many payloads contain fields marked as "RESERVED". Some payloads in IKEv2 (and historically in IKEv1) are not aligned to 4-octet boundaries.

许多有效载荷包含标记为“保留”的字段。IKEv2（以及历史上的 IKEv1）中的一些有效载荷未与 4-八位组边界对齐。

3.3. Security Association Payload

3.3. 安全关联有效负载

The Security Association payload, denoted SA in this document, is used to negotiate attributes of a Security Association. Assembly of Security Association payloads requires great peace of mind. An SA payload MAY contain multiple proposals. If there is more than one, they MUST be ordered from most preferred to least preferred. Each proposal contains a single IPsec protocol (where a protocol is IKE, ESP, or AH), each protocol MAY contain multiple transforms, and each transform MAY contain multiple attributes. When parsing an SA, an implementation MUST check that the total Payload Length is consistent with the payload's internal lengths and counts. Proposals, Transforms, and Attributes each have their own variable-length encodings. They are nested such that the Payload Length of an SA includes the combined contents of the SA, Proposal, Transform, and Attribute information. The length of a Proposal includes the lengths of all Transforms and Attributes it contains. The length of a Transform includes the lengths of all Attributes it contains.

安全关联有效负载（在本文档中表示为 SA）用于协商安全关联的属性。安全协会有效载荷的组装需要极大的安心。SA 有效负载可能包含多个方案。如果有多个，则必须从最优先到最不优先顺序排列。每个方案包含一个 IPsec 协议（其中一个协议是 IKE、ESP 或 AH），每个协议可能包含多个转换，每个转换可能包含多个属性。解析 SA 时，实现必须检查总负载长度是否与负载的内部长度和计数一致。建议、转换和属性都有自己的可变长度编码。它们是嵌套的，因此 SA 的有效负载长度包括 SA、建议、转换和属性信息的组合内容。提案的长度包括其包含的所有变换和属性的长度。变换的长度包括其包含的所有属性的长度。

The syntax of Security Associations, Proposals, Transforms, and Attributes is based on ISAKMP; however, the semantics are somewhat different. The reason for the complexity and the hierarchy is to allow for multiple possible combinations of algorithms to be encoded in a single SA. Sometimes there is a choice of multiple algorithms, whereas other times there is a combination of algorithms. For example, an initiator might want to propose using ESP with either (3DES and HMAC_MD5) or (AES and HMAC_SHA1).

安全关联、建议、转换和属性的语法基于 ISAKMP；但是，语义有些不同。复杂性和层次结构的原因是允许在单个 SA 中编码多个可能的算法组合。有时有多种算法可供选择，而有时有多种算法的组合。例如，发起人可能会建议将 ESP 与（3DES 和 HMAC_MD5）或（AES 和

HMAC_SHA1）一起使用。

One of the reasons the semantics of the SA payload have changed from ISAKMP and IKEv1 is to make the encodings more compact in common cases.

SA 有效负载的语义从 ISAKMP 和 IKEv1 更改的原因之一是为了在常见情况下使编码更加紧凑。

The Proposal structure contains within it a Proposal Num and an IPsec protocol ID. Each structure MUST have a proposal number one (1) greater than the previous structure. The first Proposal in the initiator's SA payload MUST have a Proposal Num of one (1). One reason to use multiple proposals is to propose both standard crypto ciphers and combined-mode ciphers. Combined-mode ciphers include both integrity and encryption in a single encryption algorithm, and MUST either offer no integrity algorithm or a single integrity algorithm of "none", with no integrity algorithm being the RECOMMENDED method. If an initiator wants to propose both combined-mode ciphers and normal ciphers, it must include two proposals: one will have all the combined-mode ciphers, and the other will have all

提案结构中包含提案编号和 IPsec 协议 ID。每个结构的提案编号必须比以前的结构大一（1）。发起方 SA 有效负载中的第一个建议必须具有一（1）个建议编号。使用多种方案的一个原因是同时提出标准密码和组合模式密码。组合模式密码包括单一加密算法中的完整性和加密，必须提供无完整性算法或“无”的单一完整性算法，建议使用无完整性算法。如果发起者想要同时提出组合模式密码和普通密码，那么它必须包括两个方案：一个方案拥有所有组合模式密码，另一个方案拥有所有组合模式密码

the normal ciphers with the integrity algorithms. For example, one such proposal would have two proposal structures. Proposal 1 is ESP with AES-128, AES-192, and AES-256 bits in Cipher Block Chaining (CBC) mode, with either HMAC-SHA1-96 or XCBC-96 as the integrity algorithm; Proposal 2 is AES-128 or AES-256 in GCM mode with an 8-octet Integrity Check Value (ICV). Both proposals allow but do not require the use of ESNs (Extended Sequence Numbers). This can be illustrated as:

具有完整性算法的正规密码。例如，一个这样的提案将有两个提案结构。方案 1 为加密分组链（CBC）模式下 AES-128、AES-192 和 AES-256 位的 ESP，完整性算法为 HMAC-SHA1-96 或 XCBC-96；方案 2 为 GCM 模式下的 AES-128 或 AES-256，具有 8 个八位字节的完整性检查值（ICV）。两个方案都允许但不要求使用 ESN（扩展序列号）。这可以说明为：

```
SA Payload
   |
   +--- Proposal #1 ( Proto ID = ESP(3), SPI size = 4,
   |    |          7 transforms,     SPI = 0x052357bb )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 128 )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 192 )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 256 )
   |    |
   |    +-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
   |    +-- Transform INTEG ( Name = AUTH_AES_XCBC_96 )
   |    +-- Transform ESN ( Name = ESNs )
   |    +-- Transform ESN ( Name = No ESNs )
   |
   +--- Proposal #2 ( Proto ID = ESP(3), SPI size = 4,
        |          4 transforms,     SPI = 0x35a1d6f2 )
        |
        +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
        |    +-- Attribute ( Key Length = 128 )
        |
        +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
        |    +-- Attribute ( Key Length = 256 )
        |
        +-- Transform ESN ( Name = ESNs )
        +-- Transform ESN ( Name = No ESNs )
```

```
SA Payload
   |
   +--- Proposal #1 ( Proto ID = ESP(3), SPI size = 4,
   |    |          7 transforms,     SPI = 0x052357bb )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 128 )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 192 )
   |    |
   |    +-- Transform ENCR ( Name = ENCR_AES_CBC )
   |    |    +-- Attribute ( Key Length = 256 )
   |    |
   |    +-- Transform INTEG ( Name = AUTH_HMAC_SHA1_96 )
   |    +-- Transform INTEG ( Name = AUTH_AES_XCBC_96 )
|     +-- Transform ESN ( Name = ESNs )
|     +-- Transform ESN ( Name = No ESNs )
|
+--- Proposal #2 ( Proto ID = ESP(3), SPI size = 4,
      |            4 transforms,      SPI = 0x35a1d6f2 )
      |
      +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
      |     +-- Attribute ( Key Length = 128 )
      |
      +-- Transform ENCR ( Name = AES-GCM with a 8 octet ICV )
      |     +-- Attribute ( Key Length = 256 )
      |
      +-- Transform ESN ( Name = ESNs )
      +-- Transform ESN ( Name = No ESNs )
```

Each Proposal/Protocol structure is followed by one or more transform structures. The number of different transforms is generally determined by the Protocol. AH generally has two transforms: Extended Sequence Numbers (ESNs) and an integrity check algorithm. ESP generally has three: ESN, an encryption algorithm, and an integrity check algorithm. IKE generally has four transforms: a Diffie-Hellman group, an integrity check algorithm, a PRF algorithm,

每个提案/协议结构后面都有一个或多个转换结构。不同转换的数量通常由协议决定。AH 通常有两种转换：扩展序列号（ESN）和完整性检查算法。ESP 通常有三种：ESN、加密算法和完整性检查算法。IKE 通常有四种变换：Diffie-Hellman 组、完整性检查算法、PRF 算法、，

and an encryption algorithm. For each Protocol, the set of permissible transforms is assigned Transform ID numbers, which appear in the header of each transform.

和一个加密算法。对于每个协议，为允许的转换集分配转换 ID 号，该编号显示在每个转换的标题中。

If there are multiple transforms with the same Transform Type, the proposal is an OR of those transforms. If there are multiple transforms with different Transform Types, the proposal is an AND of the different groups. For example, to propose ESP with (3DES or AES-CBC) and (HMAC_MD5 or HMAC_SHA), the ESP proposal would contain two Transform Type 1 candidates (one for 3DES and one for AEC-CBC) and two Transform Type 3 candidates (one for HMAC_MD5 and one for HMAC_SHA). This effectively proposes four combinations of algorithms. If the initiator wanted to propose only a subset of those, for example (3DES and HMAC_MD5) or (IDEA and

HMAC_SHA), there is no way to encode that as multiple transforms within a single Proposal. Instead, the initiator would have to construct two different Proposals, each with two transforms.

如果有多个变换具有相同的变换类型，则建议是这些变换的 OR。如果存在具有不同变换类型的多个变换，则建议是不同组的 AND。例如，若要建议使用（3DES 或 AES-CBC）和（HMAC_MD5 或 HMAC_SHA）的 ESP，ESP 建议将包含两个变换类型 1 候选者（一个用于 3DES，一个用于 AEC-CBC）和两个变换类型 3 候选者（一个用于 HMAC_MD5，一个用于 HMAC_SHA）。这有效地提出了四种算法组合。如果发起人只想提出其中的一个子集，例如（3DES 和 HMAC_MD5）或（IDEA 和 HMAC_SHA），则无法将其编码为单个提案中的多个变换。相反，发起者必须构建两个不同的方案，每个方案都有两个转换。

A given transform MAY have one or more Attributes. Attributes are necessary when the transform can be used in more than one way, as when an encryption algorithm has a variable key size. The transform would specify the algorithm and the attribute would specify the key size. Most transforms do not have attributes. A transform MUST NOT have multiple attributes of the same type. To propose alternate values for an attribute (for example, multiple key sizes for the AES encryption algorithm), an implementation MUST include multiple transforms with the same Transform Type each with a single Attribute.

给定的变换可能有一个或多个属性。当转换可以以多种方式使用时，属性是必需的，例如当加密算法具有可变密钥大小时。转换将指定算法，属性将指定密钥大小。大多数变换没有属性。转换不能具有相同类型的多个属性。要为属性提出备选值（例如，AES 加密算法的多个密钥大小），实现必须包括具有相同变换类型的多个变换，每个变换具有单个属性。

Note that the semantics of Transforms and Attributes are quite different from those in IKEv1. In IKEv1, a single Transform carried multiple algorithms for a protocol with one carried in the Transform and the others carried in the Attributes.

请注意，转换和属性的语义与 IKEv1 中的语义大不相同。在 IKEv1 中，单个转换为协议携带多个算法，其中一个在转换中携带，其他在属性中携带。

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                          <Proposals>                          ~
```

```
|                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                |
~                          <Proposals>                          ~
|                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Security Association Payload

图 6：安全关联负载

o Proposals (variable) - One or more proposal substructures.

o 方案（可变）-一个或多个方案子结构。

The payload type for the Security Association payload is thirty-three (33).

安全关联有效负载的有效负载类型为三十三（33）。

3.3.1. Proposal Substructure

3.3.1. 建议子结构

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 2 |   RESERVED    |         Proposal Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Proposal Num  |  Protocol ID  |    SPI Size   |Num  Transforms|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                |
~                        <Transforms>                           ~
|                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 2 |   RESERVED    |         Proposal Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Proposal Num  |  Protocol ID  |    SPI Size   |Num  Transforms|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                        SPI (variable)                         ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        <Transforms>                           ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: Proposal Substructure

图 7：提案子结构

o 0 (last) or 2 (more) (1 octet) - Specifies whether this is the last Proposal Substructure in the SA. This syntax is inherited from ISAKMP, but is unnecessary because the last Proposal could be identified from the length of the SA. The value (2) corresponds to a payload type of Proposal in IKEv1, and the first four octets of the Proposal structure are designed to look somewhat like the header of a payload.

o 0（最后一个）或 2（更多）（1 个八位组）-指定这是否是 SA 中的最后一个提案子结构。此语法继承自 ISAKMP，但没有必要，因为可以根据 SA 的长度识别最后一个提案。值（2）对应于 IKEv1 中提议的有效负载类型，并且提议结构的前四个八位字节被设计成看起来有点像有效负载的头部。

o RESERVED (1 octet) - MUST be sent as zero; MUST be ignored on receipt.

o 保留（1 个八位字节）-必须作为零发送；必须在收到时忽略。

o Proposal Length (2 octets, unsigned integer) - Length of this proposal, including all transforms and attributes that follow.

o 建议长度（2 个八位字节，无符号整数）-此建议的长度，包括所有后续转换和属性。

o Proposal Num (1 octet) - When a proposal is made, the first proposal in an SA payload MUST be 1, and subsequent proposals MUST be one more than the previous proposal (indicating an OR of the two proposals). When a proposal is accepted, the proposal number in the SA payload MUST match the number on the proposal sent that was accepted.

o 提案编号（1 个八位字节）-提出提案时，SA 有效负载中的第一个提案必须为 1，后续提案必须比前一个提案多一个（表示两个提案中的 OR）。接受提案时，SA 有效负载中的提案编号必须与已接受的已发送提案上的编号相匹配。

o Protocol ID (1 octet) - Specifies the IPsec protocol identifier for the current negotiation. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

o 协议 ID（1 个八位字节）-指定当前协商的 IPsec 协议标识符。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Protocol	Protocol ID
IKE	1
AH	2
ESP	3

Protocol	Protocol ID
IKE	1
AH	2
ESP	3

o SPI Size (1 octet) - For an initial IKE SA negotiation, this field MUST be zero; the SPI is obtained from the outer header. During subsequent negotiations, it is equal to the size, in octets, of the SPI of the corresponding protocol (8 for IKE, 4 for ESP and AH).

o SPI 大小（1 个八位字节）-对于初始 IKE SA 协商，此字段必须为零；SPI 从外部收割台获取。在随后的协商过程中，它等于相应协议的 SPI 的大小（八位字节）（8 表示 IKE，4 表示 ESP 和 AH）。

o Num Transforms (1 octet) - Specifies the number of transforms in this proposal.

o Num Transforms（1 个八位字节）-指定此方案中的转换数。

o SPI (variable) - The sending entity's SPI. Even if the SPI Size is not a multiple of 4 octets, there is no padding applied to the payload. When the SPI Size field is zero, this field is not present in the Security Association payload.

o SPI（变量）-发送实体的 SPI。即使 SPI 大小不是 4 个八位字节的倍数，也不会对有效负载应用填充。当 SPI 大小字段为零时，安全关联有效负载中不存在此字段。

o Transforms (variable) - One or more transform substructures.

o 变换（变量）-一个或多个变换子结构。

3.3.2. Transform Substructure

3.3.2. 变换子结构

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 3 |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Transform Type |   RESERVED    |          Transform ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Transform Attributes                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| 0 (last) or 3 |   RESERVED    |        Transform Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Transform Type |   RESERVED    |          Transform ID         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Transform Attributes                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: Transform Substructure

图 8：变换子结构

o 0 (last) or 3 (more) (1 octet) - Specifies whether this is the last Transform

Substructure in the Proposal. This syntax is inherited from ISAKMP, but is unnecessary because the last transform could be identified from the length of the proposal. The value (3) corresponds to a payload type of Transform in IKEv1, and the first four octets of the Transform structure are designed to look somewhat like the header of a payload.

o 0（最后一个）或 3（更多）（1 个八位组）-指定这是否是提案中的最后一个变换子结构。此语法继承自 ISAKMP，但没有必要，因为可以根据提案的长度识别最后一次转换。值（3）对应于 IKEv1 中转换的有效负载类型，并且转换结构的前四个八位字节被设计成看起来有点像有效负载的报头。

o RESERVED - MUST be sent as zero; MUST be ignored on receipt.

o 保留-必须作为零发送；必须在收到时忽略。

o Transform Length - The length (in octets) of the Transform Substructure including Header and Attributes.

o 变换长度-变换子结构的长度（以八位字节为单位），包括标题和属性。

o Transform Type (1 octet) - The type of transform being specified in this transform. Different protocols support different Transform Types. For some protocols, some of the transforms may be optional. If a transform is optional and the initiator wishes to propose that the transform be omitted, no transform of the given type is included in the proposal. If the initiator wishes to make use of the transform optional to the responder, it includes a transform substructure with Transform ID = 0 as one of the options.

o 变换类型（1 个八位字节）-在此变换中指定的变换类型。不同的协议支持不同的转换类型。对于某些协议，某些转换可能是可选的。如果转换是可选的，并且发起人希望建议省略该转换，则建议中不包括给定类型的转换。如果发起者希望使用响应者可选的转换，它将包含一个转换子结构，其中转换 ID=0 作为选项之一。

o Transform ID (2 octets) - The specific instance of the Transform Type being proposed.

o Transform ID（2 个八位字节）-提出的转换类型的特定实例。

The Transform Type values are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下面列出了变换类型值。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Description	Trans.	Used In
	Type	
Encryption Algorithm (ENCR)	1	IKE and ESP
Pseudorandom Function (PRF)	2	IKE
Integrity Algorithm (INTEG)	3	IKE*, AH, optional in ESP
Diffie-Hellman group (D-H)	4	IKE, optional in AH & ESP
Extended Sequence Numbers (ESN)	5	AH and ESP

Description	Trans.	Used In
	Type	
Encryption Algorithm (ENCR)	1	IKE and ESP
Pseudorandom Function (PRF)	2	IKE
Integrity Algorithm (INTEG)	3	IKE*, AH, optional in ESP
Diffie-Hellman group (D-H)	4	IKE, optional in AH & ESP
Extended Sequence Numbers (ESN)	5	AH and ESP

(*) Negotiating an integrity algorithm is mandatory for the Encrypted payload format specified in this document. For example, [AEAD] specifies additional formats based on authenticated encryption, in which a separate integrity algorithm is not negotiated.

（*）协商完整性算法对于本文档中指定的加密有效负载格式是强制性的。例如，[AEAD]指定了基于身份验证加密的其他格式，其中不协商单独的完整性算法。

For Transform Type 1 (Encryption Algorithm), the Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

对于转换类型 1（加密算法），下面列出了转换 ID。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Name	Number	Defined In
ENCR_DES_IV64	1	(UNSPECIFIED)
ENCR_DES	2	(RFC2405), [DES]
ENCR_3DES	3	(RFC2451)
ENCR_RC5	4	(RFC2451)
ENCR_IDEA	5	(RFC2451), [IDEA]
ENCR_CAST	6	(RFC2451)
ENCR_BLOWFISH	7	(RFC2451)
ENCR_3IDEA	8	(UNSPECIFIED)
ENCR_DES_IV32	9	(UNSPECIFIED)
ENCR_NULL	11	(RFC2410)
ENCR_AES_CBC	12	(RFC3602)
ENCR_AES_CTR	13	(RFC3686)

Name	Number	Defined In
ENCR_DES_IV64	1	(UNSPECIFIED)
ENCR_DES	2	(RFC2405), [DES]
ENCR_3DES	3	(RFC2451)
ENCR_RC5	4	(RFC2451)
ENCR_IDEA	5	(RFC2451), [IDEA]
ENCR_CAST	6	(RFC2451)
ENCR_BLOWFISH	7	(RFC2451)
ENCR_3IDEA	8	(UNSPECIFIED)
ENCR_DES_IV32	9	(UNSPECIFIED)
ENCR_NULL	11	(RFC2410)
ENCR_AES_CBC	12	(RFC3602)
ENCR_AES_CTR	13	(RFC3686)

For Transform Type 2 (Pseudorandom Function), the Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

对于变换类型 2（伪随机函数），下面列出了变换 ID。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Name	Number	Defined In

PRF_HMAC_MD5	1	(RFC2104), [MD5]
PRF_HMAC_SHA1	2	(RFC2104), [SHA]
PRF_HMAC_TIGER	3	(UNSPECIFIED)

Name	Number	Defined In
PRF_HMAC_MD5	1	(RFC2104), [MD5]
PRF_HMAC_SHA1	2	(RFC2104), [SHA]
PRF_HMAC_TIGER	3	(UNSPECIFIED)

For Transform Type 3 (Integrity Algorithm), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

对于转换类型 3（完整性算法），定义的转换 ID 如下所示。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Name	Number	Defined In
NONE	0	
AUTH_HMAC_MD5_96	1	(RFC2403)
AUTH_HMAC_SHA1_96	2	(RFC2404)
AUTH_DES_MAC	3	(UNSPECIFIED)
AUTH_KPDK_MD5	4	(UNSPECIFIED)
AUTH_AES_XCBC_96	5	(RFC3566)

Name	Number	Defined In
NONE	0	
AUTH_HMAC_MD5_96	1	(RFC2403)
AUTH_HMAC_SHA1_96	2	(RFC2404)
AUTH_DES_MAC	3	(UNSPECIFIED)
AUTH_KPDK_MD5	4	(UNSPECIFIED)
AUTH_AES_XCBC_96	5	(RFC3566)

For Transform Type 4 (Diffie-Hellman group), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

对于转换类型 4（Diffie-Hellman 组），定义的转换 ID 如下所示。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Name	Number	Defined In
NONE	0	
768-bit MODP	1	Appendix B
1024-bit MODP	2	Appendix B
1536-bit MODP	5	[ADDGROUP]
2048-bit MODP	14	[ADDGROUP]
3072-bit MODP	15	[ADDGROUP]
4096-bit MODP	16	[ADDGROUP]
6144-bit MODP	17	[ADDGROUP]
8192-bit MODP	18	[ADDGROUP]

Name	Number	Defined In
NONE	0	
768-bit MODP	1	Appendix B
1024-bit MODP	2	Appendix B
1536-bit MODP	5	[ADDGROUP]
2048-bit MODP	14	[ADDGROUP]
3072-bit MODP	15	[ADDGROUP]
4096-bit MODP	16	[ADDGROUP]
6144-bit MODP	17	[ADDGROUP]
8192-bit MODP	18	[ADDGROUP]

Although ESP and AH do not directly include a Diffie-Hellman exchange, a Diffie-Hellman group MAY be negotiated for the Child SA. This allows the peers to employ Diffie-Hellman in the CREATE_CHILD_SA exchange, providing perfect forward secrecy for the generated Child SA keys.

尽管 ESP 和 AH 不直接包括 Diffie-Hellman 交换，但可以为子 SA 协商 Diffie-Hellman 组。这允许对等方在 CREATE_CHILD_SA 交换中使用 Diffie-Hellman，为生成的 CHILD SA 密钥提供完美的前向保密性。

For Transform Type 5 (Extended Sequence Numbers), defined Transform IDs are listed below. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

对于变换类型 5（扩展序列号），定义的变换 ID 如下所示。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Name	Number
No Extended Sequence Numbers	0
Extended Sequence Numbers	1

Name	Number
No Extended Sequence Numbers	0
Extended Sequence Numbers	1

Note that an initiator who supports ESNs will usually include two ESN transforms, with values "0" and "1", in its proposals. A proposal containing a single ESN transform with value "1" means that using normal (non-extended) sequence numbers is not acceptable.

请注意，支持 ESN 的发起人通常会在其提案中包含两个 ESN 转换，其值为“0”和“1”。包含值为“1”的单个 ESN 转换的提案意味着使用正常（非扩展）序列号是不可接受的。

Numerous additional Transform Types have been defined since the publication of RFC 4306. Please refer to the IANA IKEv2 registry for details.

自 RFC 4306 发布以来，已经定义了许多其他转换类型。有关详细信息，请参阅 IANA IKEv2 注册表。

3.3.3. Valid Transform Types by Protocol

3.3.3. 按协议列出的有效转换类型

The number and type of transforms that accompany an SA payload are dependent on the protocol in the SA itself. An SA payload proposing the establishment of an SA has the following mandatory and optional Transform Types. A compliant implementation MUST understand all mandatory and optional types for each protocol it supports (though it

SA 有效负载附带的转换的数量和类型取决于 SA 本身中的协议。建议建立 SA 的 SA 有效负载具有以下强制和可选转换类型。兼容实现必须了解其支持的每个协议的所有强制和可选类型（尽管

need not accept proposals with unacceptable suites). A proposal MAY omit the optional types if the only value for them it will accept is NONE.

不需要接受带有不可接受套件的提案）。如果建议书接受的唯一值为“无”，则建议书可能会忽略可选类型。

Protocol	Mandatory Types	Optional Types
IKE	ENCR, PRF, INTEG*, D-H	
ESP	ENCR, ESN	INTEG, D-H
AH	INTEG, ESN	D-H

Protocol	Mandatory Types	Optional Types
IKE	ENCR, PRF, INTEG*, D-H	
ESP	ENCR, ESN	INTEG, D-H
AH	INTEG, ESN	D-H

(*) Negotiating an integrity algorithm is mandatory for the Encrypted payload format specified in this document. For example, [AEAD] specifies additional formats based on authenticated encryption, in which a separate integrity algorithm is not negotiated.

（*）协商完整性算法对于本文档中指定的加密有效负载格式是强制性的。例如，[AEAD]指定了基于身份验证加密的其他格式，其中不协商单独的完整性算法。

3.3.4. Mandatory Transform IDs

3.3.4. 强制转换 ID

The specification of suites that MUST and SHOULD be supported for interoperability has been removed from this document because they are likely to change more rapidly than this document evolves. At the time of publication of this document, [RFC4307] specifies these suites, but note that it might be updated in the future, and other RFCs might specify different sets of suites.

本文档中删除了互操作性必须和应该支持的套件规范，因为它们的变化可能比本文档的发展更快。在本文档发布时，[RFC4307]指定了这些套件，但请注意，它可能会在将来更新，其他 RFC 可能会指定不同的套件集。

An important lesson learned from IKEv1 is that no system should only implement the mandatory algorithms and expect them to be the best choice for all customers.

从 IKEv1 中学到的一个重要教训是，任何系统都不应该只实现强制算法，并期望它们成为所有客户的最佳选择。

It is likely that IANA will add additional transforms in the future, and some users may want to use private suites, especially for IKE where implementations should be capable of supporting different parameters, up to certain size limits. In support of this goal, all implementations of IKEv2 SHOULD include a management facility that allows specification (by a user or system administrator) of Diffie-Hellman parameters (the generator, modulus, and exponent lengths and values) for new Diffie-Hellman groups. Implementations SHOULD provide a management interface through which these parameters and the associated Transform IDs may be entered (by a user or system administrator), to enable negotiating such groups.

IANA 很可能会在将来添加额外的转换，一些用户可能希望使用私有套件，特别是对于 IKE，在 IKE 中，实现应该能够支持不同的参数，达到一定的大小限制。为了支持这一目标，IKEv2 的所有实现都应该包括一个管理工具，允许（由用户或系统管理员）为新的 Diffie-Hellman 组指定 Diffie-Hellman 参数（生成器、模数和指数长度和值）。实现应提供一个管理界面，通过该界面（由用户或系统管理员）可以输入这些参数和相关的转换 ID，以便能够协商这些组。

All implementations of IKEv2 MUST include a management facility that enables a user or system administrator to specify the suites that are acceptable for use with IKE. Upon receipt of a payload with a set of Transform IDs, the implementation MUST compare the transmitted Transform IDs against those locally configured via the management controls, to verify that the proposed suite is acceptable based on local policy. The implementation MUST reject SA proposals that are

IKEv2 的所有实现必须包括一个管理工具，该工具允许用户或系统管理员指定可与 IKE 一起使用的套件。在收到带有一组转换 ID 的有效负载后，实现必须将传输的转换 ID 与通过管理控制在本地配置的转换 ID 进行比较，以验证基于本地策略提出的套件是可接受的。实施必须拒绝不符合要求的 SA 提案

not authorized by these IKE suite controls. Note that cryptographic suites that MUST be implemented need not be configured as acceptable to local policy.

未经这些 IKE 套件控件授权。请注意，必须实现的加密套件不需要配置为本地策略可以接受。

3.3.5. Transform Attributes

3.3.5. 变换属性

Each transform in a Security Association payload may include attributes that modify or complete the specification of the transform. The set of valid attributes depends on the transform. Currently, only a single attribute type is defined: the Key Length attribute is used by certain encryption transforms with variable-length keys (see below for details).

安全关联有效负载中的每个转换可以包括修改或完成转换规范的属性。有效属性集取决于变换。目前，仅定义了一种属性类型：密钥长度属性由具有可变长度密钥的某些加密转换使用（有关详细信息，请参见下文）。

The attributes are type/value pairs and are defined below. Attributes can have a value with a fixed two-octet length or a variable-length value. For the latter, the attribute is encoded as type/length/value.

属性是类型/值对，定义如下。属性可以具有固定的两个八位字节长度值或可变长度值。对于后者，属性被编码为 type/length/value。

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|       Attribute Type        |    AF=0  Attribute Length     |
|F|                             |    AF=1  Attribute Value      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   AF=0  Attribute Value                       |
|                   AF=1  Not Transmitted                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|A|       Attribute Type        |    AF=0  Attribute Length     |
|F|                             |    AF=1  Attribute Value      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   AF=0  Attribute Value                       |
|                   AF=1  Not Transmitted                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: Data Attributes

图 9：数据属性

o Attribute Format (AF) (1 bit) - Indicates whether the data attribute follows the Type/Length/Value (TLV) format or a shortened Type/Value (TV) format. If the AF bit is zero (0), then the attribute uses TLV format; if the AF bit is one (1), the TV format (with two-byte value) is used.

o 属性格式（AF）（1 位）-指示数据属性是遵循类型/长度/值（TLV）格式还是缩短类型/值（TV）格式。如果 AF 位为零（0），则该属性使用 TLV 格式；如果 AF 位为一（1），则使用 TV 格式（具有两个字节的值）。

o Attribute Type (15 bits) - Unique identifier for each type of attribute (see below).

o 属性类型（15 位）-每种类型属性的唯一标识符（见下文）。

o Attribute Value (variable length) - Value of the attribute associated with the attribute type. If the AF bit is a zero (0), this field has a variable length defined by the Attribute Length field. If the AF bit is a one (1), the Attribute Value has a length of 2 octets.

o 属性值（可变长度）-与属性类型关联的属性的值。如果 AF 位为零（0），则该字段具有由属性长度字段定义的可变长度。如果 AF 位为 1，则属性值的长度为 2 个八位字节。

The only currently defined attribute type (Key Length) is fixed length; the variable-length encoding specification is included only for future extensions. Attributes described as fixed length MUST NOT

当前唯一定义的属性类型（键长度）是固定长度；可变长度编码规范仅用于将来的扩展。不能使用描述为固定长度的属性

be encoded using the variable-length encoding unless that length exceeds two bytes. Variable-length attributes MUST NOT be encoded as fixed-length even if their value can fit into two octets. Note: This is a change from IKEv1, where increased flexibility may have simplified the composer of messages but certainly complicated the parser.

必须使用可变长度编码进行编码，除非该长度超过两个字节。可变长度属性不得编码为固定长度，即使其值可以放入两个八位字节。注意：这是 IKEv1 的一个变化，在 IKEv1 中，增加的灵活性可能简化了消息的编写器，但肯定会使解析器复杂化。

The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Attribute Type	Value	Attribute Format
Key Length (in bits)	14	TV

Attribute Type	Value	Attribute Format
Key Length (in bits)	14	TV

Values 0-13 and 15-17 were used in a similar context in IKEv1, and should not be assigned except to matching values.

值 0-13 和 15-17 在 IKEv1 中的类似上下文中使用，除了匹配值外，不应将其赋值。

The Key Length attribute specifies the key length in bits (MUST use network byte order) for certain transforms as follows:

Key Length 属性指定某些转换的密钥长度（以位为单位）（必须使用网络字节顺序），如下所示：

o The Key Length attribute MUST NOT be used with transforms that use a fixed-length key. For example, this includes ENCR_DES, ENCR_IDEA, and all the Type 2 (Pseudorandom function) and Type 3 (Integrity Algorithm) transforms specified in this document. It is recommended that future Type 2 or 3 transforms do not use this attribute.

o “关键点长度”属性不得与使用固定长度关键点的变换一起使用。例如，这包括 ENCR_DES、ENCR_IDEA 以及本文档中指定的所有类型 2（伪随机函数）和类型 3（完整性算法）转换。建议将来的类型 2 或 3 转换不使用此属性。

o Some transforms specify that the Key Length attribute MUST be always included (omitting the attribute is not allowed, and proposals not containing it MUST be rejected). For example, this includes ENCR_AES_CBC and ENCR_AES_CTR.

o 某些转换指定必须始终包含键长度属性（不允许忽略该属性，并且必须拒绝不包含该属性的建议）。例如，这包括 ENCR_AES_CBC 和 ENCR_AES_CTR。

o Some transforms allow variable-length keys, but also specify a default key length if the attribute is not included. For example, these transforms include ENCR_RC5 and ENCR_BLOWFISH.

o 某些变换允许可变长度的关键帧，但如果不包括属性，则还指定默认的关键帧长度。例如，这些变换包括 ENCR_RC5 和 ENCR_河豚。

Implementation note: To further interoperability and to support upgrading endpoints independently, implementers of this protocol SHOULD accept values that they deem to supply greater security. For instance, if a peer is configured to accept a variable-length cipher with a key length of X bits and is offered that cipher with a larger key length, the implementation SHOULD accept the offer if it supports use of the longer key.

实施说明：为了进一步提高互操作性并支持独立升级端点，该协议的实施者应该接受他们认为可以提供更高安全性的值。例如，如果一个对等方被配置为接受密钥长度为 X 位的可变长度密码，并且该密码具有更大的密钥长度，那么如果该实现支持使用更长的密钥，则该实现应该接受该密码。

Support for this capability allows a responder to express a concept of "at least" a certain level of security -- "a key length of _at least_ X bits for cipher Y". However, as the attribute is always returned unchanged (see the next section), an initiator willing to accept multiple key lengths has to include multiple transforms with the same Transform Type, each with a different Key Length attribute.

对该功能的支持允许响应者表达“至少”某种安全级别的概念——“密码 Y 的密钥长度至少为 X 位”。但是，由于属性总是返回不变（请参见下一节），因此愿意接受多个密钥长度的启动器必须包含具有相同转换类型的多个转换，每个转换都具有不同的密钥长度属性。

3.3.6. Attribute Negotiation

3.3.6. 属性协商

During Security Association negotiation initiators present offers to responders. Responders MUST select a single complete set of parameters from the offers (or reject all offers if none are acceptable). If there are multiple proposals, the responder MUST choose a single proposal. If the selected proposal has multiple transforms with the same type, the responder MUST choose a single one. Any attributes of a selected transform MUST be returned unmodified. The initiator of an exchange MUST check that the accepted offer is consistent with one of its proposals, and if not MUST terminate the exchange.

在安全关联协商期间，发起人向响应者提供报价。响应者必须从报价中选择一组完整的参数（如果没有一个报价是可接受的，则拒绝所有报价）。如果有多个提案，响应者必须选择一个提案。如果所选方案具有多个相同类型的转换，则响应者必须选择一个。所选转换的任何属性都必须未经修改地返回。交易所的发起人必须检查所接受的报价是否与其提议一致，如果不一致，则必须终止交易所。

If the responder receives a proposal that contains a Transform Type it does not understand, or a proposal that is missing a mandatory Transform Type, it MUST consider this proposal unacceptable; however, other proposals in the same SA payload are processed as usual. Similarly, if the responder receives a transform that it does not understand, or one that contains a Transform Attribute it does not understand, it MUST consider this transform unacceptable; other transforms with the same Transform Type are processed as usual. This allows new Transform Types and Transform Attributes to be defined in the future.

如果响应者收到包含不理解的转换类型的提案，或者缺少强制转换类型的提案，则必须认为该提议不可接受；但是，同一 SA 有效载荷中的其他提案将照常处理。类似地，如果响应者接收到它不理解的变换，或者包含其不理解的变换属性的变换，则它必须认为这种变换是不可接受的；具有相同变换类型的其他变换将照常处理。这允许将来定义新的变换类型和变换属性。

Negotiating Diffie-Hellman groups presents some special challenges. SA offers include proposed attributes and a Diffie-Hellman public number (KE) in the same message. If in the initial exchange the initiator offers to use one of several Diffie-Hellman groups, it SHOULD pick the one the responder is most likely to accept and include a KE corresponding to that group. If the responder selects a proposal using a different Diffie-Hellman group (other than NONE), the responder will indicate the correct group in the response and the initiator SHOULD pick an element of that group for its KE value when retrying the first message. It SHOULD, however, continue to propose its full supported set of groups in order to prevent a man-in-the-middle downgrade attack. If one of the proposals offered is for the Diffie-Hellman group of NONE, and the responder selects that Diffie-Hellman group, then it MUST ignore the initiator's KE payload and omit the KE payload from the response.

谈判 Diffie-Hellman 集团面临一些特殊挑战。SA 提供的服务包括在同一消息中建议的属性和

Diffie Hellman 公用号码（KE）。如果在初始交换中，发起方提出使用几个 Diffie-Hellman 组中的一个，那么它应该选择响应方最有可能接受的组，并包括与该组对应的 KE。如果响应者使用不同的 Diffie-Hellman 组（无组除外）选择提案，响应者将在响应中指示正确的组，并且发起者应在重试第一条消息时选择该组的元素作为其 KE 值。然而，为了防止中间人降级攻击，它应该继续提出其完全受支持的组集。如果提供的方案之一是针对 Diffie-Hellman 组的 NONE，并且响应者选择了该 Diffie-Hellman 组，那么它必须忽略启动器的 KE 有效负载，并从响应中忽略 KE 有效负载。

3.4. Key Exchange Payload

3.4. 密钥交换有效载荷

The Key Exchange payload, denoted KE in this document, is used to exchange Diffie-Hellman public numbers as part of a Diffie-Hellman key exchange. The Key Exchange payload consists of the IKE generic payload header followed by the Diffie-Hellman public value itself.

密钥交换有效载荷（在本文档中表示为 KE）用于交换 Diffie-Hellman 公钥，作为 Diffie-Hellman 密钥交换的一部分。密钥交换有效负载由 IKE 通用有效负载头和 Diffie-Hellman 公共值本身组成。

```
                     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 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Diffie-Hellman Group Num    |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Key Exchange Data                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Diffie-Hellman Group Num    |           RESERVED            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Key Exchange Data                       ~
|                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: Key Exchange Payload Format

图 10：密钥交换有效负载格式

A Key Exchange payload is constructed by copying one's Diffie-Hellman public value into the "Key Exchange Data" portion of the payload. The length of the Diffie-Hellman public value for modular exponentiation group (MODP) groups MUST be equal to the length of the prime modulus over which the exponentiation was performed, prepending zero bits to the value if necessary.

密钥交换有效负载是通过将 Diffie-Hellman 公共值复制到有效负载的“密钥交换数据”部分来构建的。模幂运算组（MODP）的 Diffie-Hellman 公共值的长度必须等于在其上执行幂运算的素数模的长度，如有必要，将零位前置到该值。

The Diffie-Hellman Group Num identifies the Diffie-Hellman group in which the Key Exchange Data was computed (see Section 3.3.2). This Diffie-Hellman Group Num MUST match a Diffie-Hellman group specified in a proposal in the SA payload that is sent in the same message, and SHOULD match the Diffie-Hellman group in the first group in the first proposal, if such exists. If none of the proposals in that SA payload specifies a Diffie-Hellman group, the KE payload MUST NOT be present. If the selected proposal uses a different Diffie-Hellman group (other than NONE), the message MUST be rejected with a Notify payload of type INVALID_KE_PAYLOAD. See also Sections 1.2 and 2.7.

Diffie-Hellman 组编号标识计算密钥交换数据的 Diffie-Hellman 组（见第 3.3.2 节）。此 Diffie-Hellman 组编号必须与 SA 有效负载中在同一消息中发送的方案中指定的 Diffie-Hellman 组匹配，并且应与第一个方案中第一个组中的 Diffie-Hellman 组匹配（如果存在）。如果 SA 有效载荷中没有任何建议指定 Diffie-Hellman 组，则 KE 有效载荷不得存在。如果所选方案使用不同的 Diffie-Hellman 组（无组除外），则必须使用类型为 INVALID_KE_payload 的 Notify payload 拒绝该消息。另见第 1.2 节和第 2.7 节。

The payload type for the Key Exchange payload is thirty-four (34).

密钥交换有效负载的有效负载类型为三十四（34）。

3.5. Identification Payloads

3.5. 识别有效载荷

The Identification payloads, denoted IDi and IDr in this document, allow peers to assert an identity to one another. This identity may be used for policy lookup, but does not necessarily have to match anything in the CERT payload; both fields may be used by an implementation to perform access control decisions. When using the

标识有效载荷（在本文档中表示为 IDi 和 IDr）允许对等方彼此声明身份。此标识可用于策略查找，但不一定必须匹配证书有效负载中的任何内容；实现可以使用这两个字段来执行访问控制决策。当使用

ID_IPV4_ADDR/ID_IPV6_ADDR identity types in IDi/IDr payloads, IKEv2 does not require this address to match the address in the IP header of IKEv2 packets, or anything in the TSi/TSr payloads. The contents of IDi/IDr are used purely to fetch the policy and authentication data related to the other party.

IDi/IDr 有效载荷中的 ID_IPV4_ADDR/ID_IPV6_ADDR 标识类型，IKEv2 不要求此地址与 IKEv2 数据包的 IP 头中的地址或 TSi/TSr 有效载荷中的任何内容匹配。IDi/IDr 的内容仅用于获取与另一方相关的策略和身份验证数据。

NOTE: In IKEv1, two ID payloads were used in each direction to hold Traffic Selector (TS) information for data passing over the SA. In IKEv2, this information is carried in TS payloads (see Section 3.13).

注意：在 IKEv1 中，在每个方向上使用两个 ID 有效载荷来保存通过 SA 的数据的流量选择器（TS）信息。在 IKEv2 中，该信息在 TS 有效载荷中携带（见第 3.13 节）。

The Peer Authorization Database (PAD) as described in RFC 4301 [IPSECARCH] describes the use of the ID payload in IKEv2 and provides a formal model for the binding of identity to policy in addition to providing services that deal more specifically with the details of policy enforcement. The PAD is intended to provide a link between the SPD and the IKE Security Association management. See Section 4.4.3 of RFC 4301 for more details.

RFC 4301[IPSECARCH]中描述的对等授权数据库（PAD）描述了 IKEv2 中 ID 有效负载的使用，并提供了身份与策略绑定的正式模型，此外还提供了更具体地处理策略实施细节的服务。PAD 旨在提供 SPD 和 IKE 安全关联管理之间的链接。详见 RFC 4301 第 4.4.3 节。

The Identification payload consists of the IKE generic payload header followed by identification fields as follows:

标识有效负载由 IKE 通用有效负载头和标识字段组成，如下所示：

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Payload  |C|  RESERVED   |         Payload Length        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   ID Type     |                 RESERVED                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                   Identification Data                         ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Payload  |C|  RESERVED   |         Payload Length        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   ID Type     |                 RESERVED                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                   Identification Data                         ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: Identification Payload Format

图 11：识别有效载荷格式

o ID Type (1 octet) - Specifies the type of Identification being used.

o ID 类型（1 个八位字节）-指定正在使用的标识类型。

o RESERVED - MUST be sent as zero; MUST be ignored on receipt.

o 保留-必须作为零发送；必须在收到时忽略。

o Identification Data (variable length) - Value, as indicated by the Identification Type. The length of the Identification Data is computed from the size in the ID payload header.

o 标识数据（可变长度）-由标识类型指示的值。识别数据的长度是根据 ID 有效负载报头中的大

小计算的。

The payload types for the Identification payload are thirty-five (35) for IDi and thirty-six (36) for IDr.

识别有效载荷的有效载荷类型为 IDi 的三十五（35）种，IDr 的三十六（36）种。

The following table lists the assigned semantics for the Identification Type field. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下表列出了标识类型字段的指定语义。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

ID Type	Value
ID_IPV4_ADDR	1

A single four (4) octet IPv4 address.

ID Type	Value
ID_IPV4_ADDR	1

A single four (4) octet IPv4 address.

ID_FQDN 2 A fully-qualified domain name string. An example of an ID_FQDN is "example.com". The string MUST NOT contain any terminators (e.g., NULL, CR, etc.). All characters in the ID_FQDN are ASCII; for an "internationalized domain name", the syntax is as defined in [IDNA], for example "xn--tmonesimerkki-bfbb.example.net".

ID_FQDN 2 是完全限定的域名字符串。ID_FQDN 的一个示例是“example.com”。字符串不得包含任何终止符（例如 NULL、CR 等）。ID_FQDN 中的所有字符均为 ASCII；对于“国际化域名”，语法如[IDNA]中所定义，例如“xn--tmonesimerkki bfbb.example.net”。

ID_RFC822_ADDR 3 A fully-qualified RFC 822 email address string. An example of a ID_RFC822_ADDR is "jsmith@example.com". The string MUST NOT contain any terminators. Because of [EAI], implementations would be wise to treat this field as UTF-8 encoded text, not as pure ASCII.

ID \u RFC822 \u ADDR 3 完全限定的 RFC 822 电子邮件地址字符串。ID_RFC822_ADDR 的

一个示例是“jsmith@example.com". 字符串不能包含任何终止符。由于[EAI]，实现将明智地将此字段视为 UTF-8 编码文本，而不是纯 ASCII。

ID_IPV6_ADDR 5 A single sixteen (16) octet IPv6 address.

ID_IPV6_ADDR 5 单个十六（16）个八位字节的 IPV6 地址。

ID_DER_ASN1_DN 9 The binary Distinguished Encoding Rules (DER) encoding of an ASN.1 X.500 Distinguished Name [PKIX].

ID_DER_ASN1_DN 9 ASN.1 X.500 可分辨名称[PKIX]的二进制可分辨编码规则（DER）编码。

ID_DER_ASN1_GN 10 The binary DER encoding of an ASN.1 X.509 GeneralName [PKIX].

ID_DER_ASN1_GN 10 ASN.1 X.509 通用名称[PKIX]的二进制 DER 编码。

ID_KEY_ID 11 An opaque octet stream that may be used to pass vendor-specific information necessary to do certain proprietary types of identification.

ID_KEY_ID 11 一种不透明的八位字节流，可用于传递进行某些专有类型识别所需的供应商特定信息。

Two implementations will interoperate only if each can generate a type of ID acceptable to the other. To assure maximum interoperability, implementations MUST be configurable to send at least one of ID_IPV4_ADDR, ID_FQDN, ID_RFC822_ADDR, or ID_KEY_ID, and MUST be configurable to accept all of these four types. Implementations SHOULD be capable of generating and accepting all of these types. IPv6-capable implementations MUST additionally be

只有当两个实现都能生成另一个可以接受的 ID 类型时，这两个实现才能互操作。为了确保最大的互操作性，实现必须可配置为发送至少一个 ID_IPV4_ADDR、ID_FQDN、ID_RFC822_ADDR 或 ID_KEY_ID，并且必须可配置为接受这四种类型。实现应该能够生成和接受所有这些类型。支持 IPv6 的实现还必须

configurable to accept ID_IPV6_ADDR. IPv6-only implementations MAY be configurable to send only ID_IPV6_ADDR instead of ID_IPV4_ADDR for IP addresses.

可配置为接受 ID\u IPV6\u 地址。仅 IPv6 实现可以配置为仅发送 IP 地址的 ID\U IPv6\U ADDR，而不是 ID\U IPV4\U ADDR。

EAP [EAP] does not mandate the use of any particular type of identifier, but often EAP is used with Network Access Identifiers (NAIs) defined in [NAI]. Although NAIs look a bit like email addresses (e.g., "joe@example.com"), the syntax is not exactly the same as the syntax of email address in [MAILFORMAT]. For those NAIs that include the realm component, the ID_RFC822_ADDR identification type SHOULD be used. Responder implementations should not attempt to verify that the contents actually conform to the exact syntax given in [MAILFORMAT], but instead should accept any reasonable-looking NAI. For NAIs that do not include the realm component, the ID_KEY_ID identification type SHOULD be used.

EAP[EAP]不强制使用任何特定类型的标识符，但EAP通常与[NAI]中定义的网络访问标识符（NAI）一起使用。虽然NAI看起来有点像电子邮件地址（例如“joe@example.com），语法与[MAILFORMAT]中电子邮件地址的语法不完全相同。对于那些包含领域组件的NAI，应该使用ID_RFC822_ADDR标识类型。响应程序实现不应尝试验证内容是否确实符合[MAILFORMAT]中给出的确切语法，而应接受任何外观合理的NAI。对于不包括领域组件的NAI，应使用ID_KEY_ID标识类型。

3.6. Certificate Payload

3.6. 证书有效负载

The Certificate payload, denoted CERT in this document, provides a means to transport certificates or other authentication-related information via IKE. Certificate payloads SHOULD be included in an exchange if certificates are available to the sender. The Hash and URL formats of the Certificate payloads should be used in case the peer has indicated an ability to retrieve this information from elsewhere using an HTTP_CERT_LOOKUP_SUPPORTED Notify payload. Note that the term "Certificate payload" is somewhat misleading, because not all authentication mechanisms use certificates and data other than certificates may be passed in this payload.

证书有效负载（在本文档中表示为CERT）提供了通过IKE传输证书或其他身份验证相关信息的方法。如果发送方可以使用证书，则应将证书有效负载包括在exchange中。如果对等方表示能够使用HTTP_CERT_LOOKUP_支持的Notify负载从其他地方检索此信息，则应使用证书有效负载的哈希和URL格式。请注意，术语“证书有效负载”有些误导，因为并非所有身份验证机制都使用证书，并且除了证书之外的数据可能在此有效负载中传递。

The Certificate payload is defined as follows:

证书有效负载的定义如下：

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cert Encoding |                                               |
+-+-+-+-+-+-+-+-+                                               |
~                       Certificate Data                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cert Encoding |                                               |
+-+-+-+-+-+-+-+-+                                               |
~                       Certificate Data                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: Certificate Payload Format

图 12：证书有效负载格式

o Certificate Encoding (1 octet) - This field indicates the type of certificate or certificate-related information contained in the Certificate Data field. The values in the following table are only current as of the publication date of RFC 4306. Other values

o 证书编码（1 个八位字节）-此字段指示证书数据字段中包含的证书类型或证书相关信息。下表中的值仅为截至 RFC 4306 发布日期的当前值。其他价值观

may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

此后可能已添加或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

```
Certificate Encoding                 Value
----------------------------------------------------
PKCS #7 wrapped X.509 certificate    1   UNSPECIFIED
```

PGP Certificate	2	UNSPECIFIED
DNS Signed Key	3	UNSPECIFIED
X.509 Certificate - Signature	4	
Kerberos Token	6	UNSPECIFIED
Certificate Revocation List (CRL)	7	
Authority Revocation List (ARL)	8	UNSPECIFIED
SPKI Certificate	9	UNSPECIFIED
X.509 Certificate - Attribute	10	UNSPECIFIED
Raw RSA Key	11	
Hash and URL of X.509 certificate	12	
Hash and URL of X.509 bundle	13	

Certificate Encoding	Value	
PKCS #7 wrapped X.509 certificate	1	UNSPECIFIED
PGP Certificate	2	UNSPECIFIED
DNS Signed Key	3	UNSPECIFIED
X.509 Certificate - Signature	4	
Kerberos Token	6	UNSPECIFIED
Certificate Revocation List (CRL)	7	
Authority Revocation List (ARL)	8	UNSPECIFIED
SPKI Certificate	9	UNSPECIFIED
X.509 Certificate - Attribute	10	UNSPECIFIED
Raw RSA Key	11	
Hash and URL of X.509 certificate	12	
Hash and URL of X.509 bundle	13	

o Certificate Data (variable length) - Actual encoding of certificate data. The type of certificate is indicated by the Certificate Encoding field.

o 证书数据（可变长度）-证书数据的实际编码。证书的类型由证书编码字段指示。

The payload type for the Certificate payload is thirty-seven (37).

证书有效负载的有效负载类型为三十七（37）。

Specific syntax for some of the certificate type codes above is not defined in this document. The types whose syntax is defined in this document are:

本文档中未定义上述某些证书类型代码的特定语法。本文档中定义了其语法的类型有：

o "X.509 Certificate - Signature" contains a DER-encoded X.509 certificate whose public key is used to validate the sender's AUTH payload. Note that with this encoding, if a chain of certificates needs to be sent, multiple CERT payloads are used, only the first of which holds the public key used to validate the sender's AUTH payload.

o “X.509 证书-签名”包含 DER 编码的 X.509 证书，其公钥用于验证发送方的身份验证有效负载。请注意，使用这种编码，如果需要发送证书链，则会使用多个证书有效载荷，其中只有第一个证书有效载荷保存用于验证发送者的身份验证有效载荷的公钥。

o "Certificate Revocation List" contains a DER-encoded X.509 certificate revocation list.

o “证书吊销列表”包含 DER 编码的 X.509 证书吊销列表。

o "Raw RSA Key" contains a PKCS #1 encoded RSA key, that is, a DER-encoded RSAPublicKey structure (see [RSA] and [PKCS1]).

o “原始 RSA 密钥”包含 PKCS#1 编码的 RSA 密钥，即 DER 编码的 RSACPublicKey 结构（请参见[RSA]和[PKCS1]）。

o Hash and URL encodings allow IKE messages to remain short by replacing long data structures with a 20-octet SHA-1 hash (see [SHA]) of the replaced value followed by a variable-length URL that resolves to the DER-encoded data structure itself. This improves efficiency when the endpoints have certificate data

o 哈希和 URL 编码允许 IKE 消息保持简短，方法是将长数据结构替换为替换值的 20 个八位组 SHA-1 哈希（参见[SHA]），后跟解析为 DER 编码数据结构本身的可变长度 URL。这提高了端点具有证书数据时的效率

cached and makes IKE less subject to DoS attacks that become easier to mount when IKE messages are large enough to require IP fragmentation [DOSUDPPROT].

缓存并减少 IKE 受到 DoS 攻击的可能性，当 IKE 消息大到需要 IP 碎片时，DoS 攻击更容易装载[DOSUDPPROT]。

The "Hash and URL of a bundle" type uses the following ASN.1 definition for the X.509 bundle:

“捆绑包的哈希和 URL”类型对 X.509 捆绑包使用以下 ASN.1 定义：

```
CertBundle
```

```
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-cert-bundle(34) }
```

```
CertBundle
  { iso(1) identified-organization(3) dod(6) internet(1)
    security(5) mechanisms(5) pkix(7) id-mod(0)
    id-mod-cert-bundle(34) }
```

```
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
```

```
DEFINITIONS EXPLICIT TAGS ::=
BEGIN
```

```
IMPORTS
  Certificate, CertificateList
  FROM PKIX1Explicit88
     { iso(1) identified-organization(3) dod(6)
       internet(1) security(5) mechanisms(5) pkix(7)
       id-mod(0) id-pkix1-explicit(18) } ;
```

```
IMPORTS
  Certificate, CertificateList
  FROM PKIX1Explicit88
     { iso(1) identified-organization(3) dod(6)
       internet(1) security(5) mechanisms(5) pkix(7)
       id-mod(0) id-pkix1-explicit(18) } ;
```

```
CertificateOrCRL ::= CHOICE {
  cert [0] Certificate,
  crl  [1] CertificateList }
```

```
CertificateOrCRL ::= CHOICE {
  cert [0] Certificate,
  crl  [1] CertificateList }
```

```
CertificateBundle ::= SEQUENCE OF CertificateOrCRL
```

```
   CertificateBundle ::= SEQUENCE OF CertificateOrCRL
```

END

终止

Implementations MUST be capable of being configured to send and accept up to four X.509 certificates in support of authentication, and also MUST be capable of being configured to send and accept the Hash and URL format (with HTTP URLs). Implementations SHOULD be capable of being configured to send and accept Raw RSA keys. If multiple certificates are sent, the first certificate MUST contain the public key used to sign the AUTH payload. The other certificates may be sent in any order.

实现必须能够配置为发送和接受最多四个 X.509 证书以支持身份验证，还必须能够配置为发送和接受哈希和 URL 格式（使用 HTTP URL）。实现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，则第一个证书必须包含用于签名身份验证有效负载的公钥。其他证书可按任何顺序发送。

Implementations MUST support the HTTP [HTTP] method for hash-and-URL lookup. The behavior of other URL methods [URLS] is not currently specified, and such methods SHOULD NOT be used in the absence of a document specifying them.

实现必须支持用于哈希和 URL 查找的 HTTP[HTTP]方法。当前未指定其他 URL 方法[URL]的行为，在没有指定这些方法的文档的情况下，不应使用这些方法。

3.7. Certificate Request Payload

3.7. 证书请求有效负载

The Certificate Request payload, denoted CERTREQ in this document, provides a means to request preferred certificates via IKE and can appear in the IKE_INIT_SA response and/or the IKE_AUTH request. Certificate Request payloads MAY be included in an exchange when the sender needs to get the certificate of the receiver.

本文档中表示为 CERTREQ 的证书请求有效负载提供了通过 IKE 请求首选证书的方法，并且可以出现在 IKE_INIT_SA 响应和/或 IKE_AUTH 请求中。当发送方需要获得接收方的证书时，证书请求有效负载可以包括在交换中。

The Certificate Request payload is defined as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cert Encoding |                                               |
+-+-+-+-+-+-+-+-+                                               |
~                    Certification Authority                    ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Certificate Request payload is defined as follows:

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Cert Encoding |                                               |
+-+-+-+-+-+-+-+-+                                               |
~                    Certification Authority                    ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: Certificate Request Payload Format

图 13：证书请求有效负载格式

o Certificate Encoding (1 octet) - Contains an encoding of the type or format of certificate requested. Values are listed in Section 3.6.

o 证书编码（1 个八位字节）-包含请求的证书类型或格式的编码。数值见第 3.6 节。

o Certification Authority (variable length) - Contains an encoding of an acceptable certification authority for the type of certificate requested.

o 证书颁发机构（可变长度）-包含所请求证书类型的可接受证书颁发机构的编码。

The payload type for the Certificate Request payload is thirty-eight (38).

证书请求有效负载的有效负载类型为三十八（38）。

The Certificate Encoding field has the same values as those defined in Section 3.6.

The Certification Authority field contains an indicator of trusted authorities for this certificate type. The Certification Authority value is a concatenated list of SHA-1 hashes of the public keys of trusted Certification Authorities (CAs). Each is encoded as the SHA-1 hash of the Subject Public Key Info element (see section 4.1.2.7 of [PKIX]) from each Trust Anchor certificate. The 20-octet hashes are concatenated and included with no other formatting.

证书编码字段的值与第 3.6 节中定义的值相同。证书颁发机构字段包含此证书类型的受信任机构的指示器。证书颁发机构值是可信证书颁发机构（CA）公钥的 SHA-1 哈希的串联列表。每个都被编码为来自每个信任锚证书的主题公钥信息元素（见[PKIX]第 4.1.2.7 节）的 SHA-1 散列。20 个八位字节的散列是串联的，不包含其他格式。

The contents of the "Certification Authority" field are defined only for X.509 certificates, which are types 4, 12, and 13. Other values SHOULD NOT be used until Standards-Track specifications that specify their use are published.

“Certificate Authority”（证书颁发机构）字段的内容仅针对 X.509 证书定义，这些证书是类型 4、12 和 13。在发布指定其用途的标准跟踪规范之前，不应使用其他值。

Note that the term "Certificate Request" is somewhat misleading, in that values other than certificates are defined in a "Certificate" payload and requests for those values can be present in a Certificate Request payload. The syntax of the Certificate Request payload in such cases is not defined in this document.

请注意，术语“证书请求”有点误导，因为在“证书”有效负载中定义了证书以外的值，并且对这些值的请求可以出现在证书请求有效负载中。在这种情况下，证书请求有效负载的语法未在本文档中定义。

The Certificate Request payload is processed by inspecting the "Cert Encoding" field to determine whether the processor has any certificates of this type. If so, the "Certification Authority" field is inspected to determine if the processor has any certificates that can be validated up to one of the specified certification authorities. This can be a chain of certificates.

通过检查“Cert Encoding”（证书编码）字段来处理证书请求有效负载，以确定处理器是否具有任何此类证书。如果是，则检查“证书颁发机构”字段，以确定处理者是否有任何证书可通过指定证书颁发机构之一的验证。这可以是一个证书链。

If an end-entity certificate exists that satisfies the criteria specified in the CERTREQ, a certificate or certificate chain SHOULD be sent back to the certificate requestor if the recipient of the CERTREQ:

如果存在满足 CERTREQ 中指定标准的终端实体证书，则如果 CERTREQ 的接收者：

o is configured to use certificate authentication,

o 配置为使用证书身份验证，

o is allowed to send a CERT payload,

o 允许发送证书有效负载，

o has matching CA trust policy governing the current negotiation, and

o 具有管理当前协商的匹配 CA 信任策略，以及

o has at least one time-wise and usage-appropriate end-entity certificate chaining to a CA provided in the CERTREQ.

o 至少具有一个与 CERTREQ 中提供的 CA 链接的时间和使用适当的最终实体证书。

Certificate revocation checking must be considered during the chaining process used to select a certificate. Note that even if two peers are configured to use two different CAs, cross-certification relationships should be supported by appropriate selection logic.

在用于选择证书的链接过程中，必须考虑证书吊销检查。请注意，即使将两个对等方配置为使用两个不同的 CA，也应通过适当的选择逻辑支持交叉认证关系。

The intent is not to prevent communication through the strict adherence of selection of a certificate based on CERTREQ, when an alternate certificate could be selected by the sender that would still enable the recipient to successfully validate and trust it through trust conveyed by cross-certification, CRLs, or other out-of-band configured means. Thus, the processing of a CERTREQ should be seen as a suggestion for a certificate to select, not a mandated one. If no certificates exist, then the CERTREQ is ignored. This is not an error condition of the protocol. There may be cases where there is a preferred CA sent in the CERTREQ, but an alternate might be acceptable (perhaps after prompting a human operator).

这样做的目的不是为了通过严格遵守基于 CERTREQ 的证书选择来阻止通信，当发送方可以选择替代证书时，该替代证书仍然能够使接收方通过交叉认证 CRLs 传递的信任来成功验证和信任该

证书，或其他带外配置方式。因此，CERTREQ 的处理应该被视为选择证书的建议，而不是强制证书。如果不存在证书，则忽略 CERTREQ。这不是协议的错误情况。在某些情况下，CERTREQ 中可能发送了首选 CA，但可以接受备用 CA（可能是在提示人工操作员之后）。

The HTTP_CERT_LOOKUP_SUPPORTED notification MAY be included in any message that can include a CERTREQ payload and indicates that the sender is capable of looking up certificates based on an HTTP-based URL (and hence presumably would prefer to receive certificate specifications in that format).

HTTP_CERT_LOOKUP_支持的通知可以包括在任何消息中，该消息可以包括 CERTREQ 有效负载，并指示发送方能够基于基于 HTTP 的 URL 查找证书（因此可能更愿意接收该格式的证书规范）。

3.8. Authentication Payload

3.8. 身份验证有效负载

The Authentication payload, denoted AUTH in this document, contains data used for authentication purposes. The syntax of the Authentication data varies according to the Auth Method as specified below.

身份验证有效负载（在本文档中表示为 AUTH）包含用于身份验证目的的数据。身份验证数据的语法根据下面指定的身份验证方法而变化。

The Authentication payload is defined as follows:

身份验证有效负载的定义如下：

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Auth Method   |                RESERVED                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Authentication Data                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Auth Method   |                RESERVED                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                      Authentication Data                      ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: Authentication Payload Format

图 14：验证有效负载格式

o Auth Method (1 octet) - Specifies the method of authentication used. The types of signatures are listed here. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

o Auth Method（1 个八位字节）-指定使用的身份验证方法。这里列出了签名的类型。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

```
Mechanism                              Value
-----------------------------------------------------------------
RSA Digital Signature                    1
   Computed as specified in Section 2.15 using an RSA private key
   with RSASSA-PKCS1-v1_5 signature scheme specified in [PKCS1]
   (implementers should note that IKEv1 used a different method for
   RSA signatures).  To promote interoperability, implementations
   that support this type SHOULD support signatures that use SHA-1
   as the hash function and SHOULD use SHA-1 as the default hash
   function when generating signatures.  Implementations can use the
   certificates received from a given peer as a hint for selecting a
   mutually understood hash function for the AUTH payload signature.
```

```
Mechanism                              Value
-----------------------------------------------------------------
RSA Digital Signature                    1
   Computed as specified in Section 2.15 using an RSA private key
   with RSASSA-PKCS1-v1_5 signature scheme specified in [PKCS1]
   (implementers should note that IKEv1 used a different method for
```

RSA signatures). To promote interoperability, implementations
that support this type SHOULD support signatures that use SHA-1
as the hash function and SHOULD use SHA-1 as the default hash
function when generating signatures. Implementations can use the
certificates received from a given peer as a hint for selecting a
mutually understood hash function for the AUTH payload signature.

Note, however, that the hash algorithm used in the AUTH payload signature doesn't have to be the same as any hash algorithm(s) used in the certificate(s).

但是，请注意，身份验证有效负载签名中使用的哈希算法不必与证书中使用的任何哈希算法相同。

Shared Key Message Integrity Code 2 Computed as specified in Section 2.15 using the shared key associated with the identity in the ID payload and the negotiated PRF.

根据第 2.15 节的规定，使用与 ID 有效载荷和协商 PRF 中的身份相关联的共享密钥计算共享密钥消息完整性代码 2。

DSS Digital Signature 3 Computed as specified in Section 2.15 using a DSS private key (see [DSS]) over a SHA-1 hash.

DSS 数字签名 3，按照第 2.15 节的规定，在 SHA-1 哈希上使用 DSS 私钥（见[DSS]）计算。

o Authentication Data (variable length) - see Section 2.15.

o 认证数据（可变长度）-见第 2.15 节。

The payload type for the Authentication payload is thirty-nine (39).

验证有效负载的有效负载类型为三十九（39）。

3.9. Nonce Payload

3.9. 临时有效载荷

The Nonce payload, denoted as Ni and Nr in this document for the initiator's and responder's nonce, respectively, contains random data used to guarantee liveness during an exchange and protect against replay attacks.

在本文档中，发起者和响应者的 Nonce 的 Nonce 负载分别表示为 Ni 和 Nr，其中包含用于保证交换期间的活跃性和防止重播攻击的随机数据。

The Nonce payload is defined as follows:

当前有效载荷定义如下：

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                            Nonce Data                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                            Nonce Data                         ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 15: Nonce Payload Format

图 15：Nonce 有效负载格式

o Nonce Data (variable length) - Contains the random data generated by the transmitting entity.

o Nonce 数据（可变长度）-包含传输实体生成的随机数据。

The payload type for the Nonce payload is forty (40).

当前有效载荷的有效载荷类型为四十（40）。

The size of the Nonce Data MUST be between 16 and 256 octets, inclusive. Nonce values MUST NOT be reused.

Nonce 数据的大小必须介于 16 到 256 个八位字节之间（包括 16 到 256 个八位字节）。不得重复使用 Nonce 值。

3.10. Notify Payload

3.10. 通知有效载荷

The Notify payload, denoted N in this document, is used to transmit informational data, such as error conditions and state transitions, to an IKE peer. A Notify payload may appear in a response message (usually specifying why a request was rejected), in an INFORMATIONAL Exchange (to report an error not in an IKE request), or in any other message to indicate sender capabilities or to modify the meaning of the request.

通知有效负载（在本文档中表示为 N）用于向 IKE 对等方传输信息数据，如错误条件和状态转换。通知有效负载可能出现在响应消息（通常指定拒绝请求的原因）、信息交换（报告错误而不是 IKE 请求）或任何其他消息中，以指示发送方的能力或修改请求的含义。

```
  The Notify payload is defined as follows:
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Payload  |C|  RESERVED   |         Payload Length        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Protocol ID  |   SPI Size    |      Notify Message Type      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                Security Parameter Index (SPI)                 ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                       Notification Data                       ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  The Notify payload is defined as follows:
                        1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Next Payload  |C|  RESERVED   |         Payload Length        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Protocol ID  |   SPI Size    |      Notify Message Type      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                Security Parameter Index (SPI)                 ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       Notification Data                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 16: Notify Payload Format

图16：通知有效负载格式

o Protocol ID (1 octet) - If this notification concerns an existing SA whose SPI is given in the SPI field, this field indicates the type of that SA. For notifications concerning Child SAs, this field MUST contain either (2) to indicate AH or (3) to indicate ESP. Of the notifications defined in this document, the SPI is included only with INVALID_SELECTORS and REKEY_SA. If the SPI field is empty, this field MUST be sent as zero and MUST be ignored on receipt.

o 协议ID（1个八位字节）-如果此通知涉及SPI字段中给出SPI的现有SA，则此字段指示该SA的类型。对于有关子SA的通知，此字段必须包含（2）以表示AH或（3）以表示特别是本文档中定义的通知，SPI仅包含无效的选择器和重新设置密钥的SA。如果SPI字段为空，则此字段必须作为零发送，并且在收到时必须忽略。

o SPI Size (1 octet) - Length in octets of the SPI as defined by the IPsec protocol ID or zero if no SPI is applicable. For a notification concerning the IKE SA, the SPI Size MUST be zero and the field must be empty.

o SPI大小（1个八位字节）-由IPsec协议ID定义的SPI的八位字节长度，如果没有适用的SPI，则为零。对于有关IKE SA的通知，SPI大小必须为零，字段必须为空。

o Notify Message Type (2 octets) - Specifies the type of notification message.

o 通知消息类型（2个八位字节）-指定通知消息的类型。

o SPI (variable length) - Security Parameter Index.

o SPI（可变长度）-安全参数索引。

o Notification Data (variable length) - Status or error data transmitted in addition to the Notify Message Type. Values for this field are type specific (see below).

o 通知数据（可变长度）-除通知消息类型外发送的状态或错误数据。此字段的值是特定于类型的（见下文）。

The payload type for the Notify payload is forty-one (41).

通知有效负载的有效负载类型为四十一（41）。

3.10.1. Notify Message Types

3.10.1. 通知消息类型

Notification information can be error messages specifying why an SA could not be established. It can also be status data that a process managing an SA database wishes to communicate with a peer process.

通知信息可以是错误消息，指定无法建立SA的原因。它也可以是管理SA数据库的进程希望与对等进程通信的状态数据。

The table below lists the Notification messages and their corresponding values. The number of different error statuses was greatly reduced from IKEv1 both for simplification and to avoid giving configuration information to probers.

下表列出了通知消息及其相应的值。从IKEv1中大大减少了不同错误状态的数量，以简化并避免向探测器提供配置信息。

Types in the range 0 - 16383 are intended for reporting errors. An implementation receiving a Notify payload with one of these types that it does not recognize in a response MUST assume that the corresponding request has failed entirely. Unrecognized error types in a request and status types in a request or response MUST be ignored, and they should be logged.

范围为0-16383的类型用于报告错误。接收到Notify有效负载的实现在响应中无法识别其中一种类型的有效负载时，必须假定相应的请求已完全失败。必须忽略请求中无法识别的错误类型以及请求或响应中的状态类型，并记录它们。

Notify payloads with status types MAY be added to any message and MUST be ignored if not recognized. They are intended to indicate capabilities, and as part of SA negotiation, are used to negotiate non-cryptographic parameters.

具有状态类型的Notify payloads可添加到任何消息中，如果无法识别，则必须忽略。它们用于指示功能，并作为SA协商的一部分，用于协商非加密参数。

More information on error handling can be found in Section 2.21.

有关错误处理的更多信息，请参见第2.21节。

The values in the following table are only current as of the publication date of RFC 4306, plus two error types added in this document. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下表中的值仅为截至 RFC 4306 发布日期的当前值，加上本文档中添加的两种错误类型。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

NOTIFY messages: error types	Value
UNSUPPORTED_CRITICAL_PAYLOAD See Section 2.5.	1

NOTIFY messages: error types	Value
UNSUPPORTED_CRITICAL_PAYLOAD See Section 2.5.	1

INVALID_IKE_SPI 4 See Section 2.21.

无效的 SPI 4 见第 2.21 节。

INVALID_MAJOR_VERSION 5 See Section 2.5.

无效的主要版本 5 见第 2.5 节。

INVALID_SYNTAX 7 Indicates the IKE message that was received was invalid because some type, length, or value was out of range or because the request was rejected for policy reasons. To avoid a DoS attack using forged messages, this status may only be returned for and in an encrypted packet if the Message ID and cryptographic checksum were valid. To avoid leaking information to someone probing a node, this status MUST be sent in response to any error not covered by one of the other status types. To aid debugging, more detailed error information should be written to a console or log.

INVALID_SYNTAX 7 表示接收到的 IKE 消息无效，因为某些类型、长度或值超出范围，或者由于策略原因请求被拒绝。为了避免使用伪造消息的 DoS 攻击，只有在消息 ID 和加密校验和有效的情况下，才能在加密数据包中返回此状态。为了避免向探测节点的人泄漏信息，必须发送此状态以响应其他状态类型之一未包含的任何错误。为了帮助调试，应将更详细的错误信息写入控制台或日志。

INVALID_MESSAGE_ID 9 See Section 2.3.

无效消息\u ID 9 请参见第 2.3 节。

INVALID_SPI 11 See Section 1.5.

无效的 SPI 11 见第 1.5 节。

NO_PROPOSAL_CHOSEN 14 None of the proposed crypto suites was acceptable. This can be sent in any case where the offered proposals (including but not limited to SA payload values, USE_TRANSPORT_MODE notify, IPCOMP_SUPPORTED notify) are not acceptable for the responder. This can also be used as "generic" Child SA error when Child SA cannot be created for some other reason. See also Section 2.7.

没有选择任何方案 14 提议的加密套件都不可接受。如果响应者不接受所提供的建议（包括但不限于 SA 有效负载值、使用\传输\模式通知、IPCOMP \支持的通知），则可以发送该建议。当由于其他原因无法创建子 SA 时，这也可以用作“通用”子 SA 错误。另见第 2.7 节。

INVALID_KE_PAYLOAD 17 See Sections 1.2 and 1.3.

无效有效载荷 17 见第 1.2 节和第 1.3 节。

AUTHENTICATION_FAILED 24 Sent in the response to an IKE_AUTH message when, for some reason, the authentication failed. There is no associated data. See also Section 2.21.2.

由于某种原因，身份验证失败时，在对 IKE_身份验证消息的响应中发送的身份验证失败 24。没有关联的数据。另见第 2.21.2 节。

SINGLE_PAIR_REQUIRED 34 See Section 2.9.

所需单对 34 见第 2.9 节。

NO_ADDITIONAL_SAS 35 See Section 1.3.

无额外的 SAS 35，见第 1.3 节。

INTERNAL_ADDRESS_FAILURE 36 See Section 3.15.4.

内部地址故障 36 见第 3.15.4 节。

FAILED_CP_REQUIRED 37 See Section 2.19.

失败的 CP 要求 37 见第 2.19 节。

TS_UNACCEPTABLE 38 See Section 2.9.

参见第 2.9 节。

INVALID_SELECTORS 39 MAY be sent in an IKE INFORMATIONAL exchange when a node receives an ESP or AH packet whose selectors do not match those of the SA on which it was delivered (and that caused the packet to be dropped). The Notification Data contains the start of the offending packet (as in ICMP messages) and the SPI field of the notification is set to match the SPI of the Child SA.

当节点接收到 ESP 或 AH 数据包时，无效的_选择器 39 可能会在 IKE 信息交换中被发送，而 ESP 或 AH 数据包的选择器与发送该数据包的 SA 的选择器不匹配（这导致数据包被丢弃）。通知数据包含违规数据包的开始（如在 ICMP 消息中），通知的 SPI 字段设置为与子 SA 的 SPI 匹配。

TEMPORARY_FAILURE 43 See section 2.25.

临时故障 43 见第 2.25 节。

CHILD_SA_NOT_FOUND 44 See section 2.25.

未找到的儿童 44 见第 2.25 节。

```
NOTIFY messages: status types           Value
-------------------------------------------------------------------
INITIAL_CONTACT                         16384
    See Section 2.4.
```

```
NOTIFY messages: status types           Value
-------------------------------------------------------------------
INITIAL_CONTACT                         16384
    See Section 2.4.
```

SET_WINDOW_SIZE 16385 See Section 2.3.

设置窗口尺寸 16385，见第 2.3 节。

ADDITIONAL_TS_POSSIBLE 16386 See Section 2.9.

其他可能的情况 16386 见第 2.9 节。

IPCOMP_SUPPORTED 16387 See Section 2.22.

IPCOMP_支持 16387 见第 2.22 节。

NAT_DETECTION_SOURCE_IP 16388 See Section 2.23.

NAT 检测源 IP 16388 见第 2.23 节。

NAT_DETECTION_DESTINATION_IP 16389 See Section 2.23.

NAT_检测_目的地_IP 16389 见第 2.23 节。

COOKIE 16390 See Section 2.6.

COOKIE 16390 见第 2.6 节。

USE_TRANSPORT_MODE 16391 See Section 1.3.1.

使用运输模式 16391，见第 1.3.1 节。

HTTP_CERT_LOOKUP_SUPPORTED 16392 See Section 3.6.

支持的 HTTP 证书查找 16392 见第 3.6 节。

REKEY_SA 16393 See Section 1.3.3.

REKEY_SA 16393 见第 1.3.3 节。

ESP_TFC_PADDING_NOT_SUPPORTED 16394 See Section 1.3.1.

ESP 不支持 TFC 填充 16394 参见第 1.3.1 节。

NON_FIRST_FRAGMENTS_ALSO 16395 See Section 1.3.1.

非优先碎片参见第 1.3.1 节。

3.11. Delete Payload

3.11. 删除有效载荷

The Delete payload, denoted D in this document, contains a protocol-specific Security Association identifier that the sender has removed from its Security Association database and is, therefore, no longer valid. Figure 17 shows the format of the Delete payload. It is possible to send multiple SPIs in a Delete payload; however, each SPI MUST be for the same protocol. Mixing of protocol identifiers

MUST NOT be performed in the Delete payload. It is permitted, however, to include multiple Delete payloads in a single INFORMATIONAL exchange where each Delete payload lists SPIs for a different protocol.

本文档中表示的删除有效负载包含特定于协议的安全关联标识符，发送方已将该标识符从其安全关联数据库中删除，因此不再有效。图 17 显示了 Delete 有效负载的格式。可以在删除有效载荷中发送多个 SPI；但是，每个 SPI 必须用于相同的协议。不得在删除有效负载中混合协议标识符。但是，允许在单个信息交换中包含多个删除有效负载，其中每个删除有效负载列出不同协议的 SPI。

Deletion of the IKE SA is indicated by a protocol ID of 1 (IKE) but no SPIs. Deletion of a Child SA, such as ESP or AH, will contain the IPsec protocol ID of that protocol (2 for AH, 3 for ESP), and the SPI is the SPI the sending endpoint would expect in inbound ESP or AH packets.

IKE SA 的删除由协议 ID 1（IKE）指示，但没有 SPI。删除子 SA（如 ESP 或 AH）将包含该协议的 IPsec 协议 ID（2 表示 AH，3 表示 ESP），并且 SPI 是发送端点在入站 ESP 或 AH 数据包中预期的 SPI。

The Delete payload is defined as follows:

删除有效负载的定义如下：

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Protocol ID   |   SPI Size    |          Num of SPIs          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~               Security Parameter Index(es) (SPI)              ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Protocol ID   |   SPI Size    |          Num of SPIs          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~               Security Parameter Index(es) (SPI)              ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 17: Delete Payload Format

图 17：删除有效负载格式

o Protocol ID (1 octet) - Must be 1 for an IKE SA, 2 for AH, or 3 for ESP.

o 协议 ID（1 个八位组）-对于 IKE SA 必须为 1，对于 AH 必须为 2，对于 ESP 必须为 3。

o SPI Size (1 octet) - Length in octets of the SPI as defined by the protocol ID. It MUST be zero for IKE (SPI is in message header) or four for AH and ESP.

o SPI 大小（1 个八位字节）-协议 ID 定义的 SPI 的八位字节长度。IKE 的长度必须为零（SPI 位于消息头中），AH 和 ESP 的长度必须为四。

o Num of SPIs (2 octets, unsigned integer) - The number of SPIs contained in the Delete payload. The size of each SPI is defined by the SPI Size field.

o SPI 数（2 个八位字节，无符号整数）–删除有效负载中包含的 SPI 数。每个 SPI 的大小由 SPI 大小字段定义。

o Security Parameter Index(es) (variable length) - Identifies the specific Security Association(s) to delete. The length of this field is determined by the SPI Size and Num of SPIs fields.

o 安全参数索引（可变长度）-标识要删除的特定安全关联。此字段的长度由 SPI 大小和 SPI 字段数决定。

The payload type for the Delete payload is forty-two (42).

删除有效负载的有效负载类型为四十二（42）。

3.12. Vendor ID Payload

3.12. 供应商 ID 有效负载

The Vendor ID payload, denoted V in this document, contains a vendor-defined constant. The constant is used by vendors to identify and recognize remote instances of their implementations. This mechanism allows a vendor to experiment with new features while maintaining backward compatibility.

供应商 ID 有效负载（在本文档中表示为 V）包含供应商定义的常量。供应商使用该常量来识别和识别其实现的远程实例。此机制允许供应商在保持向后兼容性的同时试验新功能。

A Vendor ID payload MAY announce that the sender is capable of accepting certain extensions to the protocol, or it MAY simply identify the implementation as an aid in debugging. A Vendor ID payload MUST NOT change the interpretation of any information defined in this specification (i.e., the critical bit MUST be set to 0). Multiple Vendor ID payloads MAY be sent. An implementation is not required to send any Vendor ID payload at all.

供应商 ID 有效负载可以宣布发送方能够接受协议的某些扩展，或者它可以简单地将实现标识为调试的辅助。供应商 ID 有效载荷不得改变本规范中定义的任何信息的解释（即，关键位必须设置为 0）。可以发送多个供应商 ID 有效载荷。根本不需要实现来发送任何供应商 ID 有效负载。

A Vendor ID payload may be sent as part of any message. Reception of a familiar Vendor ID payload allows an implementation to make use of private use numbers described throughout this document, such as private payloads, private exchanges, private notifications, etc. Unfamiliar Vendor IDs MUST be ignored.

供应商 ID 有效负载可以作为任何消息的一部分发送。接收熟悉的供应商 ID 有效载荷允许实施使用本文档中描述的专用编号，如专用有效载荷、专用交换、专用通知等。必须忽略不熟悉的供应商 ID。

Writers of documents who wish to extend this protocol MUST define a Vendor ID payload to announce the ability to implement the extension in the document. It is expected that documents that gain acceptance and are standardized will be given "magic numbers" out of the Future Use range by IANA, and the requirement to use a Vendor ID will go away.

希望扩展此协议的文档作者必须定义供应商 ID 有效负载，以宣布在文档中实现扩展的能力。预计 IANA 将为获得认可和标准化的文件提供超出未来使用范围的“幻数”，使用供应商 ID 的要求也将消失。

The Vendor ID payload fields are defined as follows:

供应商 ID 有效负载字段定义如下：

```
              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
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        Vendor ID (VID)                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                        Vendor ID (VID)                        ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 18: Vendor ID Payload Format

图 18：供应商 ID 有效负载格式

o Vendor ID (variable length) - It is the responsibility of the person choosing the Vendor ID to assure its uniqueness in spite of the absence of any central registry for IDs. Good practice is to include a company name, a person name, or some such information. If you want to show off, you might include the latitude and longitude and time where you were when you chose the ID and some random input. A message digest of a long unique string is preferable to the long unique string itself.

o 供应商 ID（可变长度）-选择供应商 ID 的人员有责任确保其唯一性，尽管没有 ID 的中央注册表。良好的做法是包括公司名称、人名或一些此类信息。如果你想炫耀一下，你可以包括纬度和经度，以及你选择 ID 时所在的时间和一些随机输入。长唯一字符串的消息摘要比长唯一字符串本身更可取。

The payload type for the Vendor ID payload is forty-three (43).

供应商 ID 有效负载的有效负载类型为四十三（43）。

3.13. Traffic Selector Payload

3.13. 流量选择器有效载荷

The Traffic Selector payload, denoted TS in this document, allows peers to identify packet flows for processing by IPsec security services. The Traffic Selector payload consists of the IKE generic payload header followed by individual Traffic Selectors as follows:

流量选择器有效负载（在本文档中表示为 TS）允许对等方识别数据包流以供 IPsec 安全服务处理。流量选择器有效负载由 IKE 通用有效负载头和单独的流量选择器组成，如下所示：

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number of TSs |                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       <Traffic Selectors>                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Number of TSs |                 RESERVED                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       <Traffic Selectors>                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 19: Traffic Selectors Payload Format

图 19：流量选择器有效负载格式

o Number of TSs (1 octet) - Number of Traffic Selectors being provided.

o TSs 数量（1 个八位字节）-提供的流量选择器数量。

o RESERVED - This field MUST be sent as zero and MUST be ignored on receipt.

o 保留-此字段必须作为零发送，并且在收到时必须忽略。

o Traffic Selectors (variable length) - One or more individual Traffic Selectors.

o 流量选择器（可变长度）-一个或多个单独的流量选择器。

The length of the Traffic Selector payload includes the TS header and all the Traffic Selectors.

流量选择器有效负载的长度包括TS报头和所有流量选择器。

The payload type for the Traffic Selector payload is forty-four (44) for addresses at the initiator's end of the SA and forty-five (45) for addresses at the responder's end.

流量选择器有效负载的有效负载类型为四十四（44）个，用于SA发起方端的地址，四十五（45）个用于响应方端的地址。

There is no requirement that TSi and TSr contain the same number of individual Traffic Selectors. Thus, they are interpreted as follows: a packet matches a given TSi/TSr if it matches at least one of the individual selectors in TSi, and at least one of the individual selectors in TSr.

没有要求TSi和TSr包含相同数量的单个流量选择器。因此，它们被解释为：如果数据包与TSi中的至少一个单独选择器以及TSr中的至少一个单独选择器相匹配，则它与给定TSi/TSr相匹配。

For instance, the following Traffic Selectors:

例如，以下流量选择器：

```
TSi = ((17, 100, 198.51.100.66-198.51.100.66),
       (17, 200, 198.51.100.66-198.51.100.66))
TSr = ((17, 300, 0.0.0.0-255.255.255.255),
       (17, 400, 0.0.0.0-255.255.255.255))
```

```
TSi = ((17, 100, 198.51.100.66-198.51.100.66),
       (17, 200, 198.51.100.66-198.51.100.66))
TSr = ((17, 300, 0.0.0.0-255.255.255.255),
       (17, 400, 0.0.0.0-255.255.255.255))
```

would match UDP packets from 198.51.100.66 to anywhere, with any of the four combinations of source/destination ports (100,300), (100,400), (200,300), and (200, 400).

将从198.51.100.66到任意位置的UDP数据包与源/目标端口（100300）、（100400）、（200300）和（200400）的四种组合中的任意一种匹配。

Thus, some types of policies may require several Child SA pairs. For instance, a policy matching only source/destination ports (100,300) and (200,400), but not the other two combinations, cannot be negotiated as a single Child SA pair.

因此，某些类型的策略可能需要多个子SA对。例如，仅匹配源/目标端口（100300）和（200400），但不匹配其他两个组合的策略不能作为单个子SA对进行协商。

3.13.1. Traffic Selector

3.13.1. 交通选择器

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TS Type     |IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         Starting Address*                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         Ending Address*                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   TS Type     |IP Protocol ID*|       Selector Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Start Port*         |           End Port*           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         Starting Address*                     ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                         Ending Address*                       ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 20: Traffic Selector

图 20：流量选择器

*Note: All fields other than TS Type and Selector Length depend on the TS Type. The fields shown are for TS Types 7 and 8, the only two values currently defined.

*注意：除 TS 类型和选择器长度以外的所有字段都取决于 TS 类型。显示的字段用于 TS 类型 7 和 8，这是当前定义的仅有的两个值。

o TS Type (one octet) - Specifies the type of Traffic Selector.

o TS 类型（一个八位字节）-指定流量选择器的类型。

o IP protocol ID (1 octet) - Value specifying an associated IP protocol ID (such as UDP, TCP, and ICMP). A value of zero means that the protocol ID is not relevant to this Traffic Selector -- the SA can carry all protocols.

o IP 协议 ID（1 个八位字节）-指定关联 IP 协议 ID（如 UDP、TCP 和 ICMP）的值。值为零表示协议 ID 与此流量选择器无关——SA 可以承载所有协议。

o Selector Length - Specifies the length of this Traffic Selector substructure including the header.

o 选择器长度-指定此流量选择器子结构（包括标题）的长度。

o Start Port (2 octets, unsigned integer) - Value specifying the smallest port number allowed by this Traffic Selector. For protocols for which port is undefined (including protocol 0), or if all ports are allowed, this field MUST be zero. ICMP and ICMPv6 Type and Code values, as well as Mobile IP version 6 (MIPv6) mobility header (MH) Type values, are represented in this field as specified in Section 4.4.1.1 of [IPSECARCH]. ICMP Type and Code values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and Code in the least significant eight bits. MIPv6 MH Type values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and the least significant eight bits set to zero.

o 起始端口（2 个八位字节，无符号整数）–指定此流量选择器允许的最小端口号的值。对于端口未定义的协议（包括协议 0），或者如果允许所有端口，则此字段必须为零。ICMP 和 ICMPv6 类型和代码值以及移动 IP 版本 6（MIPv6）移动报头（MH）类型值在该字段中表示，如[IPSECARCH]第 4.4.1.1 节所述。ICMP 类型和代码值被视为单个 16 位整数端口号，类型在最高有效位，代码在最低有效位。MIPv6 MH 类型值被视为单个 16 位整数端口号，最高有效位的类型为 8 位，最低有效位的类型为 0。

o End Port (2 octets, unsigned integer) - Value specifying the largest port number allowed by this Traffic Selector. For protocols for which port is undefined (including protocol 0), or if all ports are allowed, this field MUST be 65535. ICMP and ICMPv6 Type and Code values, as well as MIPv6 MH Type values, are represented in this field as specified in Section 4.4.1.1 of [IPSECARCH]. ICMP Type and Code values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and Code in the least significant eight bits. MIPv6 MH Type values are treated as a single 16-bit integer port number, with Type in the most significant eight bits and the least significant eight bits set to zero.

o End Port（2 个八位字节，无符号整数）–指定此流量选择器允许的最大端口号的值。对于端口未定义的协议（包括协议 0），或者如果允许所有端口，则此字段必须为 65535。ICMP 和 ICMPv6 类型和代码值以及 MIPv6 MH 类型值在此字段中表示，如[IPSECARCH]第 4.4.1.1 节所述。ICMP 类型和代码值被视为单个 16 位整数端口号，类型在最高有效位，代码在最低有效位。MIPv6 MH 类型值被视为单个 16 位整数端口号，最高有效位的类型为 8 位，最低有效位的类型为 0。

o Starting Address - The smallest address included in this Traffic Selector (length determined by TS Type).

o 起始地址-此流量选择器中包含的最小地址（长度由 TS 类型决定）。

o Ending Address - The largest address included in this Traffic Selector (length determined by TS Type).

o 结束地址-此流量选择器中包含的最大地址（长度由 TS 类型决定）。

Systems that are complying with [IPSECARCH] that wish to indicate "ANY" ports MUST set the start port to 0 and the end port to 65535; note that according to [IPSECARCH], "ANY" includes "OPAQUE". Systems working with [IPSECARCH] that wish to indicate "OPAQUE" ports, but not "ANY" ports, MUST set the start port to

65535 and the end port to 0.

符合[IPSECARCH]要求且希望指示“任意”端口的系统必须将起始端口设置为 0，将结束端口设置为 65535；请注意，根据[IPSECARCH]，“任何”包括“不透明”。使用[IPSECARCH]的系统如果希望指示“不透明”端口，而不是“任何”端口，则必须将起始端口设置为 65535，将结束端口设置为 0。

The Traffic Selector types 7 and 8 can also refer to ICMP or ICMPv6 type and code fields, as well as MH Type fields for the IPv6 mobility header [MIPV6]. Note, however, that neither ICMP nor MIPv6 packets have separate source and destination fields. The method for specifying the Traffic Selectors for ICMP and MIPv6 is shown by example in Section 4.4.1.3 of [IPSECARCH].

流量选择器类型 7 和 8 还可以引用 IPv6 移动头[MIPV6]的 ICMP 或 ICMPv6 类型和代码字段，以及 MH 类型字段。但是，请注意，ICMP 和 MIPv6 数据包都没有单独的源字段和目标字段。[IPSECARCH]第 4.4.1.3 节举例说明了为 ICMP 和 MIPv6 指定流量选择器的方法。

The following table lists values for the Traffic Selector Type field and the corresponding Address Selector Data. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下表列出了“交通选择器类型”字段的值和相应的地址选择器数据。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

TS Type	Value
TS_IPV4_ADDR_RANGE	7

TS Type	Value
TS_IPV4_ADDR_RANGE	7

A range of IPv4 addresses, represented by two four-octet values. The first value is the beginning IPv4 address (inclusive) and the second value is the ending IPv4 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.

IPv4 地址的范围，由两个四个八位组值表示。第一个值是起始 IPv4 地址（包括），第二个值是结束 IPv4 地址（包括）。两个指定地址之间的所有地址都被视为在列表中。

TS_IPV6_ADDR_RANGE 8

TS_IPV6_地址范围 8

A range of IPv6 addresses, represented by two sixteen-octet values. The first value is the beginning IPv6 address (inclusive) and the second value is the ending IPv6 address (inclusive). All addresses falling between the two specified addresses are considered to be within the list.

IPv6 地址的范围，由两个十六位八位组值表示。第一个值是起始 IPv6 地址（含），第二个值是结束 IPv6 地址（含）。两个指定地址之间的所有地址都被视为在列表中。

3.14. Encrypted Payload

3.14. 加密有效载荷

The Encrypted payload, denoted SK{...} in this document, contains other payloads in encrypted form. The Encrypted payload, if present in a message, MUST be the last payload in the message. Often, it is the only payload in the message. This payload is also called the "Encrypted and Authenticated" payload.

本文档中表示为 SK{...}的加密有效负载包含其他加密形式的有效负载。加密的有效负载（如果存在于消息中）必须是消息中的最后一个有效负载。通常，它是消息中唯一的有效负载。该有效载荷也称为“加密和认证”有效载荷。

The algorithms for encryption and integrity protection are negotiated during IKE SA setup, and the keys are computed as specified in Sections 2.14 and 2.18.

加密和完整性保护的算法在 IKE SA 设置期间协商，密钥按照第 2.14 节和第 2.18 节的规定计算。

This document specifies the cryptographic processing of Encrypted payloads using a block cipher in CBC mode and an integrity check algorithm that computes a fixed-length checksum over a variable size message. The design is modeled after the ESP algorithms described in RFCs 2104 [HMAC], 4303 [ESP], and 2451 [ESPCBC]. This document completely specifies the cryptographic processing of IKE data, but those documents should be consulted for design rationale. Future documents may specify the processing of Encrypted payloads for other types of transforms, such as counter mode encryption and authenticated encryption algorithms. Peers MUST NOT negotiate transforms for which no such specification exists.

本文档规定了在 CBC 模式下使用分组密码和完整性检查算法对加密有效载荷进行加密处理，该算法计算可变大小消息上的固定长度校验和。该设计根据 RFCs 2104[HMAC]、4303[ESP]和 2451[ESPCBC]中描述的 ESP 算法建模。本文件完全规定了 IKE 数据的加密处理，但设计原理应参考这些文件。未来的文档可能会为其他类型的转换指定加密有效载荷的处理，例如计数器模式加密和认证加密算法。对等方不得协商不存在此类规范的转换。

When an authenticated encryption algorithm is used to protect the IKE SA, the construction of the Encrypted payload is different than what is described here. See [AEAD] for more information on authenticated encryption algorithms and their use in ESP.

当使用经过身份验证的加密算法来保护 IKE SA 时，加密有效负载的构造与这里描述的不同。有关经过身份验证的加密算法及其在 ESP 中的使用的更多信息，请参见[AEAD]。

The payload type for an Encrypted payload is forty-six (46). The Encrypted payload consists of the IKE generic payload header followed by individual fields as follows:

加密有效负载的有效负载类型为四十六（46）。加密的有效负载由 IKE 通用有效负载头和各个字段组成，如下所示：

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Initialization Vector                     |
|         (length is block size for encryption algorithm)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |             Padding (0-255 octets)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+
|                                               |  Pad Length   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Initialization Vector                     |
|         (length is block size for encryption algorithm)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Encrypted IKE Payloads                     ~
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               |             Padding (0-255 octets)            |
+-+-+-+-+-+-+-+-+                               +-+-+-+-+-+-+-+-+
|                                               |  Pad Length   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                    Integrity Checksum Data                    ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 21: Encrypted Payload Format

图 21：加密有效负载格式

o Next Payload - The payload type of the first embedded payload. Note that this is an exception in the standard header format, since the Encrypted payload is the last payload in the message and therefore the Next Payload field would normally be zero. But because the content of this payload is embedded payloads and there was no natural place to put the type of the first one, that type is placed here.

o 下一个有效负载-第一个嵌入有效负载的有效负载类型。请注意，这在标准标头格式中是一个例外，因为加密的有效负载是消息中的最后一个有效负载，因此下一个有效负载字段通常为零。但是因为这个有效载荷的内容是嵌入的有效载荷，并且没有放置第一个有效载荷类型的自然位置，所以这个类型被放置在这里。

o Payload Length - Includes the lengths of the header, initialization vector (IV), Encrypted IKE payloads, Padding, Pad Length, and Integrity Checksum Data.

o 有效负载长度-包括标头的长度、初始化向量（IV）、加密的 IKE 有效负载、填充、焊盘长度和完整性校验和数据。

o Initialization Vector - For CBC mode ciphers, the length of the initialization vector (IV) is equal to the block length of the underlying encryption algorithm. Senders MUST select a new unpredictable IV for every message; recipients MUST accept any value. The reader is encouraged to consult [MODES] for advice on IV generation. In particular, using the final ciphertext block of the previous message is not considered unpredictable. For modes other than CBC, the IV format and processing is specified in the document specifying the encryption algorithm and mode.

o 初始化向量-对于CBC模式密码，初始化向量（IV）的长度等于基础加密算法的块长度。发件人必须为每封邮件选择一个新邮件；收件人必须接受任何值。鼓励读者参考[MODES]以获得有关IV生成的建议。特别是，使用前一条消息的最后一个密文块并不被认为是不可预测的。对于CBC以外的模式，IV格式和处理在指定加密算法和模式的文档中指定。

o IKE payloads are as specified earlier in this section. This field is encrypted with the negotiated cipher.

o IKE有效载荷如本节前面所述。此字段使用协商密码加密。

o Padding MAY contain any value chosen by the sender, and MUST have a length that makes the combination of the payloads, the Padding, and the Pad Length to be a multiple of the encryption block size. This field is encrypted with the negotiated cipher.

o 填充可以包含发送方选择的任何值，并且必须具有使有效负载、填充和填充长度的组合为加密块大小的倍数的长度。此字段使用协商密码加密。

o Pad Length is the length of the Padding field. The sender SHOULD set the Pad Length to the minimum value that makes the combination of the payloads, the Padding, and the Pad Length a multiple of the block size, but the recipient MUST accept any length that results in proper alignment. This field is encrypted with the negotiated cipher.

o Pad Length是填充字段的长度。发送方应将焊盘长度设置为使有效载荷、焊盘和焊盘长度的组合为块大小的倍数的最小值，但接收方必须接受导致正确对齐的任何长度。此字段使用协商密码加密。

o Integrity Checksum Data is the cryptographic checksum of the entire message starting with the Fixed IKE header through the Pad Length. The checksum MUST be computed over the encrypted message. Its length is determined by the integrity algorithm negotiated.

o 完整性校验和数据是整个消息的加密校验和，从固定IKE头开始，一直到Pad长度。必须对加密消息计算校验和。其长度由协商的完整性算法决定。

3.15. Configuration Payload

3.15. 配置有效载荷

The Configuration payload, denoted CP in this document, is used to exchange configuration information between IKE peers. The exchange is for an IRAC to request an internal IP address from an IRAS and to exchange other information of the sort that one would acquire with Dynamic Host Configuration Protocol (DHCP) if the IRAC were directly connected to a LAN.

配置有效负载在本文档中表示为CP，用于在IKE对等方之间交换配置信息。交换是指IRAC从IRAS请求内部IP地址，并交换其他信息，如果IRAC直接连接到LAN，则可使用动态主机配置协议（DHCP）获取此类信息。

The Configuration payload is defined as follows:

配置有效负载定义如下：

```
                     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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C| RESERVED    |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CFG Type    |                    RESERVED                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                   Configuration Attributes                    ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C| RESERVED    |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   CFG Type    |                    RESERVED                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                   Configuration Attributes                    ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 22: Configuration Payload Format

图 22：配置有效负载格式

The payload type for the Configuration payload is forty-seven (47).

配置有效载荷的有效载荷类型为四十七（47）。

o CFG Type (1 octet) - The type of exchange represented by the Configuration Attributes. The values in the following table are only current as of the publication date of RFC 4306. Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

o CFG 类型（1 个八位字节）-由配置属性表示的交换类型。下表中的值仅为截至 RFC 4306 发布日期的当前值。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

CFG Type	Value
CFG_REQUEST	1
CFG_REPLY	2
CFG_SET	3
CFG_ACK	4

CFG Type	Value
CFG_REQUEST	1
CFG_REPLY	2
CFG_SET	3
CFG_ACK	4

o RESERVED (3 octets) - MUST be sent as zero; MUST be ignored on receipt.

o 保留（3 个八位字节）-必须作为零发送；必须在收到时忽略。

o Configuration Attributes (variable length) - These are type length value (TLV) structures specific to the Configuration payload and are defined below. There may be zero or more Configuration Attributes in this payload.

o 配置属性（可变长度）-这些是特定于配置有效负载的类型长度值（TLV）结构，定义如下。此有效负载中可能有零个或多个配置属性。

3.15.1. Configuration Attributes

3.15.1. 配置属性

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|         Attribute Type      |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                    |
~                      Value                               ~
|                                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|R|         Attribute Type      |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                    |
~                      Value                               ~
|                                                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 23: Configuration Attribute Format

图 23：配置属性格式

o Reserved (1 bit) - This bit MUST be set to zero and MUST be ignored on receipt.

o 保留（1 位）-此位必须设置为零，并且在收到时必须忽略。

o Attribute Type (15 bits) - A unique identifier for each of the Configuration Attribute Types.

o 属性类型（15 位）-每个配置属性类型的唯一标识符。

o Length (2 octets, unsigned integer) - Length in octets of value.

o 长度（2 个八位字节，无符号整数）-值的八位字节长度。

o Value (0 or more octets) - The variable-length value of this Configuration Attribute. The following lists the attribute types.

o 值（0 或更多八位字节）-此配置属性的可变长度值。下面列出了属性类型。

The values in the following table are only current as of the publication date of RFC 4306 (except INTERNAL_ADDRESS_EXPIRY and INTERNAL_IP6_NBNS which were removed by this document). Other values may have been added since then or will be added after the publication of this document. Readers should refer to [IKEV2IANA] for the latest values.

下表中的值仅为截至 RFC 4306 发布日期的最新值（本文件删除的内部地址和内部 IP 地址除外）。此后可能已经添加了其他值，或将在本文件发布后添加。读者应参考[IKEV2IANA]了解最新值。

Attribute Type	Value	Multi-Valued	Length
INTERNAL_IP4_ADDRESS	1	YES*	0 or 4 octets
INTERNAL_IP4_NETMASK	2	NO	0 or 4 octets
INTERNAL_IP4_DNS	3	YES	0 or 4 octets
INTERNAL_IP4_NBNS	4	YES	0 or 4 octets
INTERNAL_IP4_DHCP	6	YES	0 or 4 octets
APPLICATION_VERSION	7	NO	0 or more
INTERNAL_IP6_ADDRESS	8	YES*	0 or 17 octets
INTERNAL_IP6_DNS	10	YES	0 or 16 octets
INTERNAL_IP6_DHCP	12	YES	0 or 16 octets
INTERNAL_IP4_SUBNET	13	YES	0 or 8 octets
SUPPORTED_ATTRIBUTES	14	NO	Multiple of 2
INTERNAL_IP6_SUBNET	15	YES	17 octets

Attribute Type	Value	Multi-Valued	Length
INTERNAL_IP4_ADDRESS	1	YES*	0 or 4 octets
INTERNAL_IP4_NETMASK	2	NO	0 or 4 octets
INTERNAL_IP4_DNS	3	YES	0 or 4 octets
INTERNAL_IP4_NBNS	4	YES	0 or 4 octets
INTERNAL_IP4_DHCP	6	YES	0 or 4 octets
APPLICATION_VERSION	7	NO	0 or more
INTERNAL_IP6_ADDRESS	8	YES*	0 or 17 octets
INTERNAL_IP6_DNS	10	YES	0 or 16 octets
INTERNAL_IP6_DHCP	12	YES	0 or 16 octets
INTERNAL_IP4_SUBNET	13	YES	0 or 8 octets
SUPPORTED_ATTRIBUTES	14	NO	Multiple of 2
INTERNAL_IP6_SUBNET	15	YES	17 octets

* These attributes may be multi-valued on return only if multiple values were requested.

* 仅当请求了多个值时，这些属性在返回时才可以是多值的。

o INTERNAL_IP4_ADDRESS, INTERNAL_IP6_ADDRESS - An address on the internal network, sometimes called a red node address or private address, and it MAY be a private address on the Internet. In a request message, the address specified is a requested address (or a zero-length address if no specific address is requested). If a specific address is requested, it likely indicates that a previous connection existed with this address and the requestor would like to reuse that address. With IPv6, a requestor MAY supply the low-order address octets it wants to use. Multiple internal addresses MAY be requested by requesting multiple internal address attributes. The responder MAY only send up to the number of addresses requested. The INTERNAL_IP6_ADDRESS is made up of two fields: the first is a 16-octet IPv6 address, and the second is a one-octet prefix-length as defined in [ADDRIPV6]. The requested address is valid as long as this IKE SA (or its rekeyed successors) requesting the address is valid. This is described in more detail in Section 3.15.3.

o INTERNAL_IP4_ADDRESS，INTERNAL_IP6_ADDRESS-内部网络上的地址，有时称为红色节点地址或专用地址，它可能是 Internet 上的专用地址。在请求消息中，指定的地址是请求的地址（如果未请求特定地址，则为零长度地址）。如果请求了一个特定的地址，它很可能表示以前存在与该地址的连接，并且请求者希望重用该地址。使用 IPv6，请求者可以提供它想要使用的低阶地址八位字节。可以通过请求多个内部地址属性来请求多个内部地址。响应者最多只能发送请求的地址数。内部_IP6_地址由两个字段组成：第一个是 16 个八位字节的 IPv6 地址，第二个是[ADDRIPV6]中定义的一个八位字节前缀长度。只要请求地址的 IKE SA（或其重新键入的后继者）有效，请求的地址就有效。第 3.15.3 节对此进行了更详细的描述。

o INTERNAL_IP4_NETMASK - The internal network's netmask. Only one netmask is allowed in the request and response messages (e.g., 255.255.255.0), and it MUST be used only with an INTERNAL_IP4_ADDRESS attribute. INTERNAL_IP4_NETMASK in a CFG_REPLY means roughly the same thing as INTERNAL_IP4_SUBNET containing the same information ("send traffic to these addresses through me"), but also implies a link boundary. For instance, the client could use its own address and the netmask to calculate the broadcast address of the link. An empty INTERNAL_IP4_NETMASK attribute can be included in a CFG_REQUEST to request this

o INTERNAL_IP4_网络掩码-内部网络的网络掩码。请求和响应消息中只允许有一个网络掩码

（例如 255.255.255.0），并且只能与内部_IP4_ADDRESS 属性一起使用。CFG_回复中的 INTERNAL_IP4_NETMASK 与包含相同信息的 INTERNAL_IP4_子网大致相同（“通过我向这些地址发送流量”），但也意味着链路边界。例如，客户端可以使用自己的地址和网络掩码来计算链路的广播地址。CFG_请求中可以包含一个空的内部_IP4_NETMASK 属性来请求该属性

information (although the gateway can send the information even when not requested). Non-empty values for this attribute in a CFG_REQUEST do not make sense and thus MUST NOT be included.

信息（尽管网关即使在未请求时也可以发送信息）。CFG_请求中此属性的非空值没有意义，因此不能包含在内。

o INTERNAL_IP4_DNS, INTERNAL_IP6_DNS - Specifies an address of a DNS server within the network. Multiple DNS servers MAY be requested. The responder MAY respond with zero or more DNS server attributes.

o INTERNAL_IP4_DNS，INTERNAL_IP6_DNS-指定网络中 DNS 服务器的地址。可能会请求多个 DNS 服务器。响应者可以使用零个或多个 DNS 服务器属性进行响应。

o INTERNAL_IP4_NBNS - Specifies an address of a NetBios Name Server (WINS) within the network. Multiple NBNS servers MAY be requested. The responder MAY respond with zero or more NBNS server attributes.

o INTERNAL_IP4_NBNS-指定网络中 NetBios 名称服务器（WINS）的地址。可能会请求多个 NBNS 服务器。响应者可以使用零个或多个 NBNS 服务器属性进行响应。

o INTERNAL_IP4_DHCP, INTERNAL_IP6_DHCP - Instructs the host to send any internal DHCP requests to the address contained within the attribute. Multiple DHCP servers MAY be requested. The responder MAY respond with zero or more DHCP server attributes.

o INTERNAL_IP4_DHCP、INTERNAL_IP6_DHCP-指示主机向属性中包含的地址发送任何内部 DHCP 请求。可能会请求多个 DHCP 服务器。响应者可以使用零个或多个 DHCP 服务器属性进行响应。

o APPLICATION_VERSION - The version or application information of the IPsec host. This is a string of printable ASCII characters that is NOT null terminated.

o APPLICATION_VERSION—IPsec 主机的版本或应用程序信息。这是一个非空终止的可打印 ASCII 字符字符串。

o INTERNAL_IP4_SUBNET - The protected sub-networks that this edge-device protects. This attribute is made up of two fields: the first being an IP address and the second being a netmask. Multiple sub-networks MAY be requested. The responder MAY respond with zero or more sub-network attributes. This is discussed in more detail in Section 3.15.2.

o 内部_IP4_子网-此边缘设备保护的受保护子网。此属性由两个字段组成：第一个是 IP 地址，第二个是网络掩码。可以请求多个子网络。响应者可以使用零个或多个子网络属性进行响应。第 3.15.2 节对此进行了更详细的讨论。

o SUPPORTED_ATTRIBUTES - When used within a Request, this attribute MUST be zero-length and specifies a query to the responder to reply back with all of the attributes that it supports. The response contains an attribute that contains a set of attribute identifiers each in 2 octets. The length divided by 2 (octets) would state the number of supported attributes contained in the response.

o 受支持的_属性-当在请求中使用时，该属性的长度必须为零，并指定一个查询给响应程序，以使用其支持的所有属性进行回复。该响应包含一个属性，该属性包含一组属性标识符，每个标识符有两个八位字节。长度除以 2（八位字节）表示响应中包含的受支持属性的数量。

o INTERNAL_IP6_SUBNET - The protected sub-networks that this edge-device protects. This attribute is made up of two fields: the first is a 16-octet IPv6 address, and the second is a one-octet prefix-length as defined in [ADDRIPV6]. Multiple sub-networks MAY be requested. The responder MAY respond with zero or more sub-network attributes. This is discussed in more detail in Section 3.15.2.

o 内部_IP6_子网-此边缘设备保护的受保护子网。此属性由两个字段组成：第一个字段是 16 个八位字节的 IPv6 地址，第二个字段是[ADDRIPV6]中定义的一个八位字节前缀长度。可以请求多个子网络。响应者可以使用零个或多个子网络属性进行响应。第 3.15.2 节对此进行了更详细的讨论。

Note that no recommendations are made in this document as to how an implementation actually figures out what information to send in a response. That is, we do not recommend any specific method of an IRAS determining which DNS server should be returned to a requesting IRAC.

请注意，本文档中没有建议实现如何实际确定在响应中发送哪些信息。也就是说，我们不建议 IRAS 使用任何特定方法来确定应将哪个 DNS 服务器返回给请求的 IRAC。

The CFG_REQUEST and CFG_REPLY pair allows an IKE endpoint to request information from its peer. If an attribute in the CFG_REQUEST Configuration payload is not zero-length, it is taken as a suggestion for that attribute. The CFG_REPLY Configuration payload MAY return that value, or a new one. It MAY also add new attributes and not include some requested ones. Unrecognized or unsupported attributes MUST be ignored in both requests and responses.

CFG_请求和 CFG_应答对允许 IKE 端点从其对等方请求信息。如果 CFG_请求配置有效负载中的属性不是零长度，则将其作为该属性的建议。CFG_REPLY 配置负载可能返回该值，或者返回一个新值。它还可以添加新属性，而不包括一些请求的属性。在请求和响应中都必须忽略无法识别或不受支持的属性。

The CFG_SET and CFG_ACK pair allows an IKE endpoint to push configuration data to its peer. In this case, the CFG_SET Configuration payload contains attributes the initiator wants its peer to alter. The responder MUST return a Configuration payload if it accepted any of the configuration data and it MUST contain the attributes that the responder accepted with zero-length data. Those attributes that it did not accept MUST NOT be in the CFG_ACK Configuration payload. If no attributes were accepted, the responder MUST return either an empty CFG_ACK payload or a response message without a CFG_ACK payload. There are currently no defined uses for the CFG_SET/CFG_ACK exchange, though they may be used in connection with extensions based on Vendor IDs. An implementation of this specification MAY ignore CFG_SET payloads.

CFG_集和 CFG_ACK 对允许 IKE 端点将配置数据推送到其对等方。在这种情况下，CFG_集配置有效负载包含启动器希望其对等方更改的属性。如果响应程序接受任何配置数据，则它必须返回配置有效负载，并且它必须包含响应程序使用零长度数据接受的属性。它不接受的那些属性不能在 CFG_ACK 配置负载中。如果未接受任何属性，响应者必须返回空的 CFG_ACK 有效负载或不带 CFG_ACK 有效负载的响应消息。CFG_SET/CFG_ACK exchange 目前没有定义的用途，尽管它们可以与基于供应商 ID 的扩展一起使用。本规范的实现可能忽略 CFG_集有效载荷。

3.15.2. Meaning of INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET

3.15.2. 内部_IP4_子网和内部_IP6_子网的含义

INTERNAL_IP4/6_SUBNET attributes can indicate additional subnets, ones that need one or more separate SAs, that can be reached through the gateway that announces the attributes. INTERNAL_IP4/6_SUBNET attributes may also express the gateway's policy about what traffic should be sent through the gateway; the client can choose whether other traffic (covered by TSr, but not in INTERNAL_IP4/6_SUBNET) is sent through the gateway or directly to the destination.

Thus, traffic to the addresses listed in the INTERNAL_IP4/6_SUBNET attributes should be sent through the gateway that announces the attributes. If there are no existing Child SAs whose Traffic Selectors cover the address in question, new SAs need to be created.

内部_IP4/6_子网属性可以表示其他子网，即需要一个或多个单独 SA 的子网，这些子网可以通过宣布属性的网关访问。内部_IP4/6_子网属性也可以表示网关关于应通过网关发送哪些流量的策略；客户端可以选择是否通过网关或直接向目的地发送其他流量（TSr 覆盖，但不在内部_IP4/6_子网中）。因此，到内部_IP4/6_子网属性中列出的地址的通信量应通过宣布属性的网关发送。如果没有现有的子 SA，其流量选择器覆盖有问题的地址，则需要创建新的 SA。

For instance, if there are two subnets, 198.51.100.0/26 and 192.0.2.0/24, and the client's request contains the following:

例如，如果有两个子网，198.51.100.0/26 和 192.0.2.0/24，并且客户端的请求包含以下内容：

```
CP(CFG_REQUEST) =
  INTERNAL_IP4_ADDRESS()
TSi = (0, 0-65535, 0.0.0.0-255.255.255.255)
TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)
```

```
CP(CFG_REQUEST) =
  INTERNAL_IP4_ADDRESS()
TSi = (0, 0-65535, 0.0.0.0-255.255.255.255)
TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)
```

then a valid response could be the following (in which TSr and INTERNAL_IP4_SUBNET contain the same information):

然后，有效响应可以是以下内容（其中 TSr 和内部_IP4_子网包含相同的信息）：

```
CP(CFG_REPLY) =
  INTERNAL_IP4_ADDRESS(198.51.100.234)
  INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
  INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
TSr = ((0, 0-65535, 198.51.100.0-198.51.100.63),
       (0, 0-65535, 192.0.2.0-192.0.2.255))
CP(CFG_REPLY) =
  INTERNAL_IP4_ADDRESS(198.51.100.234)
  INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
  INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
TSr = ((0, 0-65535, 198.51.100.0-198.51.100.63),
       (0, 0-65535, 192.0.2.0-192.0.2.255))
```

In these cases, the INTERNAL_IP4_SUBNET does not really carry any useful information.

在这些情况下，内部 IP4 子网实际上并不携带任何有用的信息。

A different possible response would have been this:

另一种可能的反应是：

```
  CP(CFG_REPLY) =
    INTERNAL_IP4_ADDRESS(198.51.100.234)
    INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
    INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
  TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
  TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)
```

```
  CP(CFG_REPLY) =
    INTERNAL_IP4_ADDRESS(198.51.100.234)
    INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
    INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
  TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
  TSr = (0, 0-65535, 0.0.0.0-255.255.255.255)
```

That response would mean that the client can send all its traffic through the gateway, but the gateway does not mind if the client sends traffic not included by INTERNAL_IP4_SUBNET directly to the destination (without going through the gateway).

该响应意味着客户端可以通过网关发送其所有通信量，但网关并不介意客户端是否将内部_IP4_子网未包含的通信量直接发送到目标（而不通过网关）。

A different situation arises if the gateway has a policy that requires the traffic for the two subnets to be carried in separate SAs. Then a response like this would indicate to the client that if it wants access to the second subnet, it needs to create a separate SA:

如果网关的策略要求在单独的 SAs 中承载两个子网的流量，则会出现不同的情况。然后，类似这样的响应将向客户端指示，如果它想要访问第二个子网，则需要创建一个单独的 SA：

```
CP(CFG_REPLY) =
  INTERNAL_IP4_ADDRESS(198.51.100.234)
  INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
  INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
TSr = (0, 0-65535, 198.51.100.0-198.51.100.63)
```

```
CP(CFG_REPLY) =
  INTERNAL_IP4_ADDRESS(198.51.100.234)
  INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
  INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
TSr = (0, 0-65535, 198.51.100.0-198.51.100.63)
```

INTERNAL_IP4_SUBNET can also be useful if the client's TSr included only part of the address space. For instance, if the client requests the following:

如果客户端的 TSr 仅包含部分地址空间，则内部 IP 4 子网也会很有用。例如，如果客户端请求以下内容：

```
CP(CFG_REQUEST) =
  INTERNAL_IP4_ADDRESS()
TSi = (0, 0-65535, 0.0.0.0-255.255.255.255)
TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)
```

```
CP(CFG_REQUEST) =
  INTERNAL_IP4_ADDRESS()
TSi = (0, 0-65535, 0.0.0.0-255.255.255.255)
TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)
```

then the gateway's response might be:

那么网关的响应可能是：

```
CP(CFG_REPLY) =
    INTERNAL_IP4_ADDRESS(198.51.100.234)
    INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
    INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
  TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
  TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)


  CP(CFG_REPLY) =
    INTERNAL_IP4_ADDRESS(198.51.100.234)
    INTERNAL_IP4_SUBNET(198.51.100.0/255.255.255.192)
    INTERNAL_IP4_SUBNET(192.0.2.0/255.255.255.0)
  TSi = (0, 0-65535, 198.51.100.234-198.51.100.234)
  TSr = (0, 0-65535, 192.0.2.155-192.0.2.155)
```

Because the meaning of INTERNAL_IP4_SUBNET/INTERNAL_IP6_SUBNET in CFG_REQUESTs is unclear, they cannot be used reliably in CFG_REQUESTs.

由于 CFG_请求中内部_IP4_子网/内部_IP6_子网的含义不清楚，因此无法在 CFG_请求中可靠地使用它们。

3.15.3. Configuration Payloads for IPv6

3.15.3. IPv6 的配置有效负载

The Configuration payloads for IPv6 are based on the corresponding IPv4 payloads, and do not fully follow the "normal IPv6 way of doing things". In particular, IPv6 stateless autoconfiguration or router advertisement messages are not used, neither is neighbor discovery. Note that there is an additional document that discusses IPv6 configuration in IKEv2, [IPV6CONFIG]. At the present time, it is an experimental document, but there is a hope that with more implementation experience, it will gain the same standards treatment as this document.

IPv6 的配置有效负载基于相应的 IPv4 有效负载，并不完全遵循“正常的 IPv6 做事方式”。特别是，不使用 IPv6 无状态自动配置或路由器广告消息，也不使用邻居发现。请注意，还有一个文档讨论了 IKEv2[IPV6CONFIG]中的 IPv6 配置。目前，这是一份试验性文件，但希望有更多的实施经验，它将获得与本文件相同的标准处理。

A client can be assigned an IPv6 address using the INTERNAL_IP6_ADDRESS Configuration payload. A minimal exchange might look like this:

可以使用内部 IP 地址配置负载为客户端分配 IPv6 地址。最小交换可能如下所示：

```
  CP(CFG_REQUEST) =
    INTERNAL_IP6_ADDRESS()
    INTERNAL_IP6_DNS()
  TSi = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
  TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)


  CP(CFG_REQUEST) =
    INTERNAL_IP6_ADDRESS()
    INTERNAL_IP6_DNS()
  TSi = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
  TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)


  CP(CFG_REPLY) =
    INTERNAL_IP6_ADDRESS(2001:DB8:0:1:2:3:4:5/64)
    INTERNAL_IP6_DNS(2001:DB8:99:88:77:66:55:44)
  TSi = (0, 0-65535, 2001:DB8:0:1:2:3:4:5 - 2001:DB8:0:1:2:3:4:5)
  TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)


  CP(CFG_REPLY) =
    INTERNAL_IP6_ADDRESS(2001:DB8:0:1:2:3:4:5/64)
    INTERNAL_IP6_DNS(2001:DB8:99:88:77:66:55:44)
  TSi = (0, 0-65535, 2001:DB8:0:1:2:3:4:5 - 2001:DB8:0:1:2:3:4:5)
  TSr = (0, 0-65535, :: - FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF)
```

The client MAY send a non-empty INTERNAL_IP6_ADDRESS attribute in the CFG_REQUEST to request a specific address or interface identifier. The gateway first checks if the specified address is acceptable, and if it is, returns that one. If the address was not acceptable, the gateway attempts to use the interface identifier with some other prefix; if even that fails, the gateway selects another interface identifier.

客户端可以在 CFG_请求中发送非空的内部_IP6_地址属性，以请求特定的地址或接口标识符。网关首先检查指定的地址是否可接受，如果可接受，则返回该地址。如果地址不可接受，网关将尝试使用带有其他前缀的接口标识符；即使失败，网关也会选择另一个接口标识符。

The INTERNAL_IP6_ADDRESS attribute also contains a prefix length field. When used in a CFG_REPLY, this corresponds to the INTERNAL_IP4_NETMASK attribute in the IPv4 case.

内部_IP6_ADDRESS 属性还包含前缀长度字段。在 CFG_回复中使用时，它对应于 IPv4 情况下

的内部_IP4_网络掩码属性。

Although this approach to configuring IPv6 addresses is reasonably simple, it has some limitations. IPsec tunnels configured using IKEv2 are not fully featured "interfaces" in the IPv6 addressing architecture sense [ADDRIPV6]. In particular, they do not necessarily have link-local addresses, and this may complicate the use of protocols that assume them, such as [MLDV2].

尽管这种配置 IPv6 地址的方法相当简单，但也有一些局限性。使用 IKEv2 配置的 IPsec 隧道不是 IPv6 寻址体系结构意义上的全功能“接口”[ADDRIPV6]。特别是，它们不一定具有链路本地地址，这可能会使采用它们的协议（如[MLDV2]）的使用复杂化。

3.15.4. Address Assignment Failures

3.15.4. 地址分配失败

If the responder encounters an error while attempting to assign an IP address to the initiator during the processing of a Configuration payload, it responds with an INTERNAL_ADDRESS_FAILURE notification. The IKE SA is still created even if the initial Child SA cannot be created because of this failure. If this error is generated within an IKE_AUTH exchange, no Child SA will be created. However, there are some more complex error cases.

如果响应程序在配置有效负载处理期间尝试将 IP 地址分配给启动器时遇到错误，则响应程序将发出内部地址失败通知。即使由于此故障无法创建初始子 SA，IKE SA 仍会创建。如果此错误是在 IKE_身份验证交换中生成的，则不会创建子 SA。但是，还有一些更复杂的错误情况。

If the responder does not support Configuration payloads at all, it can simply ignore all Configuration payloads. This type of implementation never sends INTERNAL_ADDRESS_FAILURE notifications. If the initiator requires the assignment of an IP address, it will treat a response without CFG_REPLY as an error.

如果响应程序根本不支持配置有效负载，它可以忽略所有配置有效负载。这种类型的实现从不发送内部地址失败通知。如果发起者需要分配 IP 地址，它会将没有 CFG_REPLY 的响应视为错误。

The initiator may request a particular type of address (IPv4 or IPv6) that the responder does not support, even though the responder supports Configuration payloads. In this case, the responder simply ignores the type of address it does not support and processes the rest of the request as usual.

发起方可以请求响应方不支持的特定类型的地址（IPv4 或 IPv6），即使响应方支持配置有效负

载。在这种情况下，响应程序只是忽略它不支持的地址类型，并像往常一样处理其余的请求。

If the initiator requests multiple addresses of a type that the responder supports, and some (but not all) of the requests fail, the responder replies with the successful addresses only. The responder sends INTERNAL_ADDRESS_FAILURE only if no addresses can be assigned.

如果发起方请求响应方支持的多个类型的地址，并且部分（但不是全部）请求失败，则响应方仅使用成功的地址进行响应。只有在无法分配地址时，响应程序才会发送内部地址失败。

If the initiator does not receive the IP address(es) required by its policy, it MAY keep the IKE SA up and retry the Configuration payload as separate INFORMATIONAL exchange after suitable timeout, or it MAY tear down the IKE SA by sending a Delete payload inside a separate INFORMATIONAL exchange and later retry IKE SA from the beginning after some timeout. Such a timeout should not be too short (especially if the IKE SA is started from the beginning) because these error situations may not be able to be fixed quickly; the timeout should likely be several minutes. For example, an address shortage problem on the responder will probably only be fixed when more entries are returned to the address pool when other clients disconnect or when responder is reconfigured with larger address pool.

如果启动器没有收到其策略所需的 IP 地址，它可能会保持 IKE SA 正常运行，并在适当的超时后作为单独的信息交换重试配置有效负载，或者，它可以通过在一个单独的信息交换中发送一个删除有效负载来破坏 IKE SA，然后在超时后从一开始重试 IKE SA。这样的超时不应该太短（特别是如果 IKE SA 从一开始就启动），因为这些错误情况可能无法快速修复；超时时间可能为几分钟。例如，只有当其他客户端断开连接或当响应程序重新配置为更大的地址池时，才能修复响应程序上的地址短缺问题。

3.16. Extensible Authentication Protocol (EAP) Payload

3.16. 可扩展身份验证协议（EAP）有效负载

The Extensible Authentication Protocol payload, denoted EAP in this document, allows IKE SAs to be authenticated using the protocol defined in RFC 3748 [EAP] and subsequent extensions to that protocol. When using EAP, an appropriate EAP method needs to be selected. Many of these methods have been defined, specifying the protocol's use with various authentication mechanisms. EAP method types are listed in [EAP-IANA]. A short summary of the EAP format is included here for clarity.

可扩展身份验证协议有效负载（在本文档中表示为 EAP）允许使用 RFC 3748[EAP]中定义的协议以及该协议的后续扩展对 IKE SA 进行身份验证。使用 EAP 时，需要选择适当的 EAP 方法。其

中许多方法已经定义，指定了协议在各种身份验证机制中的使用。[EAP-IANA]中列出了 EAP 方法类型。为清晰起见，此处包含 EAP 格式的简短摘要。

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       EAP Message                             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Next Payload  |C|  RESERVED   |         Payload Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
~                       EAP Message                             ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 24: EAP Payload Format

图 24:EAP 有效负载格式

The payload type for an EAP payload is forty-eight (48).

EAP 有效载荷的有效载荷类型为四十八（48）。

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      | Identifier    |           Length              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      | Type_Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

```
                     1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      | Identifier    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      | Type_Data...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Figure 25: EAP Message Format

图 25:EAP 消息格式

o Code (1 octet) indicates whether this message is a Request (1), Response (2), Success (3), or Failure (4).

o 代码（1 个八位字节）指示此消息是请求（1）、响应（2）、成功（3）还是失败（4）。

o Identifier (1 octet) is used in PPP to distinguish replayed messages from repeated ones. Since in IKE, EAP runs over a reliable protocol, it serves no function here. In a response message, this octet MUST be set to match the identifier in the corresponding request.

o PPP 中使用标识符（1 个八位组）来区分重播消息和重复消息。因为在 IKE 中，EAP 通过可靠的协议运行，所以它在这里不起任何作用。在响应消息中，必须将此八位字节设置为与相应请求中的标识符匹配。

o Length (2 octets, unsigned integer) is the length of the EAP message and MUST be four less than the Payload Length of the encapsulating payload.

o 长度（2 个八位字节，无符号整数）是 EAP 消息的长度，必须比封装有效负载的有效负载长度小 4。

o Type (1 octet) is present only if the Code field is Request (1) or Response (2). For other codes, the EAP message length MUST be four octets and the Type and Type_Data fields MUST NOT be present. In a Request (1) message, Type indicates the data being requested. In a Response (2) message, Type MUST either be Nak or match the type of the data requested. Note that since IKE passes an indication of initiator identity in the first message in the IKE_AUTH exchange, the responder SHOULD NOT send EAP Identity requests (type 1). The initiator MAY, however, respond to such requests if it receives them.

o 仅当代码字段为请求（1）或响应（2）时，类型（1 个八位组）才存在。对于其他代码，EAP 消息长度必须为四个八位字节，并且类型和类型_数据字段不得存在。在请求（1）消息中，

Type 表示所请求的数据。在响应（2）消息中，类型必须为 Nak 或与请求的数据类型匹配。请注意，由于 IKE 在 IKE_身份验证交换的第一条消息中传递了启动器标识的指示，因此响应者不应发送 EAP 标识请求（类型 1）。但是，如果发起者收到此类请求，则发起者可以对其作出响应。

o Type_Data (Variable Length) varies with the Type of Request and the associated Response. For the documentation of the EAP methods, see [EAP].

o 类型_数据（可变长度）随请求类型和相关响应而变化。有关 EAP 方法的文档，请参阅[EAP]。

Note that since IKE passes an indication of initiator identity in the first message in the IKE_AUTH exchange, the responder should not send EAP Identity requests. The initiator may, however, respond to such requests if it receives them.

请注意，由于 IKE 在 IKE_身份验证交换的第一条消息中传递了启动器标识的指示，因此响应者不应发送 EAP 标识请求。但是，如果发起者收到此类请求，则发起者可以对其作出响应。

4. Conformance Requirements

4. 一致性要求

In order to assure that all implementations of IKEv2 can interoperate, there are "MUST support" requirements in addition to those listed elsewhere. Of course, IKEv2 is a security protocol, and

为了确保 IKEv2 的所有实现都可以互操作，除了其他地方列出的要求外，还有“必须支持”的要求。当然，IKEv2 是一个安全协议

one of its major functions is to allow only authorized parties to successfully complete establishment of SAs. So a particular implementation may be configured with any of a number of restrictions concerning algorithms and trusted authorities that will prevent universal interoperability.

其主要功能之一是仅允许授权方成功完成 SA 的建立。因此，一个特定的实现可能被配置为与算法和可信机构有关的许多限制中的任何一个，这些限制将阻止通用互操作性。

IKEv2 is designed to permit minimal implementations that can interoperate with all compliant implementations. The following are features that can be omitted in a minimal implementation:

IKEv2 被设计为允许与所有兼容实现互操作的最小实现。以下是可以在最小实现中省略的功能：

o Ability to negotiate SAs through a NAT and tunnel the resulting ESP SA over UDP.

o 能够通过 NAT 协商 SAs，并通过 UDP 隧道生成 ESP SA。

o Ability to request (and respond to a request for) a temporary IP address on the remote end of a tunnel.

o 能够在隧道的远程端请求（并响应请求）临时 IP 地址。

o Ability to support EAP-based authentication.

o 能够支持基于 EAP 的身份验证。

o Ability to support window sizes greater than one.

o 能够支持大于 1 的窗口大小。

o Ability to establish multiple ESP or AH SAs within a single IKE SA.

o 能够在单个 IKE SA 内建立多个 ESP 或 AH SA。

o Ability to rekey SAs.

o 重新输入 SAs 的能力。

To assure interoperability, all implementations MUST be capable of parsing all payload types (if only to skip over them) and to ignore payload types that it does not support unless the critical bit is set in the payload header. If the critical bit is set in an unsupported payload header, all implementations MUST reject the messages containing those payloads.

为了确保互操作性，所有实现必须能够解析所有有效负载类型（如果只是跳过它们），并忽略它不支持的有效负载类型，除非在有效负载报头中设置了关键位。如果在不受支持的负载头中设置了关键位，则所有实现都必须拒绝包含这些负载的消息。

Every implementation MUST be capable of doing four-message IKE_SA_INIT and IKE_AUTH exchanges establishing two SAs (one for IKE, one for ESP or AH). Implementations MAY be initiate-only or respond-only if appropriate for their platform. Every implementation MUST be capable of responding to an INFORMATIONAL exchange, but a minimal implementation MAY respond to any request in the INFORMATIONAL exchange with an empty response (note that within the context of an IKE SA, an "empty" message consists of an IKE header followed by an Encrypted payload with no payloads contained in it). A minimal implementation

MAY support the CREATE_CHILD_SA exchange only in so far as to recognize requests and reject them with a Notify payload of type NO_ADDITIONAL_SAS. A minimal implementation need not be able to initiate CREATE_CHILD_SA or INFORMATIONAL exchanges. When an SA expires (based on locally configured values of either lifetime or octets passed), and implementation MAY either try to renew it with a CREATE_CHILD_SA exchange or it MAY delete (close) the old SA and

每个实现必须能够进行四次消息 IKE_SA_INIT 和 IKE_AUTH 交换，建立两个 SA（一个用于 IKE，一个用于 ESP 或 AH）。实现可能仅在适合其平台的情况下启动或响应。每个实现必须能够响应信息交换，但最小实现可能会以空响应响应响应信息交换中的任何请求（请注意，在 IKE SA 的上下文中，“空”消息由 IKE 头和加密的有效负载组成，其中不包含有效负载）。最低限度的实现可能只支持 CREATE_CHILD_SA 交换，以识别请求并使用类型为 NO_ADDITIONAL_SA 的 Notify payload 拒绝它们。最低限度的实现不需要能够启动创建子 SA 或信息交换。当 SA 过期时（基于本地配置的生存期或已传递的八位字节值），实现可以尝试使用 CREATE_CHILD_SA 交换续订 SA，也可以删除（关闭）旧 SA 和

create a new one. If the responder rejects the CREATE_CHILD_SA request with a NO_ADDITIONAL_SAS notification, the implementation MUST be capable of instead deleting the old SA and creating a new one.

创建一个新的。如果响应者拒绝了 CREATE_CHILD_SA 请求，并且没有附加的_SAS 通知，则实现必须能够删除旧 SA 并创建新 SA。

Implementations are not required to support requesting temporary IP addresses or responding to such requests. If an implementation does support issuing such requests and its policy requires using temporary IP addresses, it MUST include a CP payload in the first message in the IKE_AUTH exchange containing at least a field of type INTERNAL_IP4_ADDRESS or INTERNAL_IP6_ADDRESS. All other fields are optional. If an implementation supports responding to such requests, it MUST parse the CP payload of type CFG_REQUEST in the first message in the IKE_AUTH exchange and recognize a field of type INTERNAL_IP4_ADDRESS or INTERNAL_IP6_ADDRESS. If it supports leasing an address of the appropriate type, it MUST return a CP payload of type CFG_REPLY containing an address of the requested type. The responder may include any other related attributes.

实现不需要支持请求临时 IP 地址或响应此类请求。如果实现确实支持发出此类请求，并且其策略要求使用临时 IP 地址，则它必须在 IKE_认证交换的第一条消息中包含 CP 有效负载，其中至少包含一个类型为 INTERNAL_IP4_ADDRESS 或 INTERNAL_IP6_ADDRESS 的字段。所有其他字段都是可选的。如果实现支持响应此类请求，则必须解析 IKE_身份验证交换中第一条消息中

CFG_REQUEST 类型的 CP 有效负载，并识别 INTERNAL_IP4_ADDRESS 或 INTERNAL_IP6_ADDRESS 类型的字段。如果它支持租用适当类型的地址，则必须返回 CFG_REPLY 类型的 CP 有效负载，其中包含请求类型的地址。响应者可以包括任何其他相关属性。

For an implementation to be called conforming to this specification, it MUST be possible to configure it to accept the following:

要调用符合本规范的实现，必须能够将其配置为接受以下内容：

o Public Key Infrastructure using X.509 (PKIX) Certificates containing and signed by RSA keys of size 1024 or 2048 bits, where the ID passed is any of ID_KEY_ID, ID_FQDN, ID_RFC822_ADDR, or ID_DER_ASN1_DN.

o 使用 X.509（PKIX）证书的公钥基础设施，该证书包含大小为 1024 或 2048 位的 RSA 密钥并由其签名，其中传递的 ID 为 ID_Key_ID、ID_FQDN、ID_RFC822_ADDR 或 ID_DER_ASN1_DN 中的任意一个。

o Shared key authentication where the ID passed is any of ID_KEY_ID, ID_FQDN, or ID_RFC822_ADDR.

o 共享密钥身份验证，其中传递的 ID 是 ID_key_ID、ID_FQDN 或 ID_RFC822_ADDR 中的任意一个。

o Authentication where the responder is authenticated using PKIX Certificates and the initiator is authenticated using shared key authentication.

o 身份验证，其中使用 PKIX 证书对响应者进行身份验证，使用共享密钥身份验证对启动器进行身份验证。

5. Security Considerations

5. 安全考虑

While this protocol is designed to minimize disclosure of configuration information to unauthenticated peers, some such disclosure is unavoidable. One peer or the other must identify itself first and prove its identity first. To avoid probing, the initiator of an exchange is required to identify itself first, and usually is required to authenticate itself first. The initiator can, however, learn that the responder supports IKE and what cryptographic protocols it supports. The responder (or someone impersonating the responder) can probe the initiator not only for its identity, but using CERTREQ payloads may be able to determine what certificates the initiator is willing to use.

虽然此协议旨在最大限度地减少向未经验证的对等方披露配置信息，但某些此类披露是不可避免的。一方或另一方必须首先确定自己的身份，并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身，并且通常需要首先对自身进行身份验证。但是，发起者可以了解响应者支持 IKE 以及它支持什么加密协议。响应者（或模拟响应者的人）不仅可以探测启动器的身份，还可以使用 CERTREQ 有效负载确定启动器愿意使用的证书。

Use of EAP authentication changes the probing possibilities somewhat. When EAP authentication is used, the responder proves its identity before the initiator does, so an initiator that knew the name of a valid initiator could probe the responder for both its name and certificates.

EAP 认证的使用在一定程度上改变了探测的可能性。当使用 EAP 身份验证时，响应程序在启动器之前验证其身份，因此知道有效启动器名称的启动器可以探测响应程序的名称和证书。

Repeated rekeying using CREATE_CHILD_SA without additional Diffie-Hellman exchanges leaves all SAs vulnerable to cryptanalysis of a single key. Implementers should take note of this fact and set a limit on CREATE_CHILD_SA exchanges between exponentiations. This document does not prescribe such a limit.

在不进行额外 Diffie-Hellman 交换的情况下，使用 CREATE_CHILD_SA 重复密钥更新会使所有 SA 容易受到单个密钥密码分析的攻击。实现者应该注意这一事实，并对求幂之间的 CREATE_CHILD_SA 交换设置一个限制。本文件未规定此类限制。

The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. Due to these inputs, it is difficult to determine the strength of a key for any of the defined groups. Diffie-Hellman group number two, when used with a strong random number generator and an exponent no less than 200 bits, is common for use with 3DES. Group five provides greater security than group two. Group one is for historic purposes only and does not provide sufficient strength except for use with DES, which is also for historic use only. Implementations should make note of these estimates when establishing policy and negotiating security parameters.

使用此处定义的任何组从 Diffie-Hellman 交换中导出的密钥的强度取决于组的固有强度、所用指数的大小以及所用随机数生成器提供的熵。由于这些输入，很难确定任何已定义组的键的强度。Diffie-Hellman 第二组当与强随机数生成器和不小于 200 位的指数一起使用时，通常用于 3DES。第五组提供了比第二组更高的安全性。第一组仅用于历史目的，除了与 DES 一起使用外，

不提供足够的强度，DES 也仅用于历史用途。在建立策略和协商安全参数时，实现应该注意这些估计。

Note that these limitations are on the Diffie-Hellman groups themselves. There is nothing in IKE that prohibits using stronger groups nor is there anything that will dilute the strength obtained from stronger groups (limited by the strength of the other algorithms negotiated including the PRF). In fact, the extensible framework of IKE encourages the definition of more groups; use of elliptic curve groups may greatly increase strength using much smaller numbers.

请注意，这些限制是针对 Diffie-Hellman 组本身的。IKE 中没有任何内容禁止使用更强的组，也没有任何内容会稀释从更强的组获得的强度（受协商的其他算法（包括 PRF）的强度限制）。事实上，IKE 的可扩展框架鼓励定义更多的组；使用椭圆曲线组可以使用更小的数字大大提高强度。

It is assumed that all Diffie-Hellman exponents are erased from memory after use.

假设所有 Diffie-Hellman 指数在使用后从内存中删除。

The IKE_SA_INIT and IKE_AUTH exchanges happen before the initiator has been authenticated. As a result, an implementation of this protocol needs to be completely robust when deployed on any insecure network. Implementation vulnerabilities, particularly DoS attacks, can be exploited by unauthenticated peers. This issue is particularly worrisome because of the unlimited number of messages in EAP-based authentication.

IKE_SA_INIT 和 IKE_AUTH 交换发生在启动器经过身份验证之前。因此，当部署在任何不安全的网络上时，该协议的实现需要完全健壮。未经身份验证的对等方可以利用实现漏洞，尤其是 DoS 攻击。这个问题尤其令人担忧，因为在基于 EAP 的身份验证中，消息的数量是无限的。

The strength of all keys is limited by the size of the output of the negotiated PRF. For this reason, a PRF whose output is less than 128 bits (e.g., 3DES-CBC) MUST NOT be used with this protocol.

所有密钥的强度都受到协商 PRF 输出大小的限制。因此，输出小于 128 位的 PRF（例如 3DES-CBC）不得与本协议一起使用。

The security of this protocol is critically dependent on the randomness of the randomly chosen parameters. These should be generated by a strong random or properly seeded pseudorandom source (see [RANDOMNESS]). Implementers should take care to ensure that use of random numbers for both keys and nonces is engineered in a fashion that does not undermine the security of the keys.

该协议的安全性主要取决于随机选择的参数的随机性。这些应该由强随机或适当种子的伪随机源生成（参见[随机性]）。实现者应该注意确保对密钥和 nonce 使用随机数的方式不会破坏密钥的安全性。

For information on the rationale of many of the cryptographic design choices in this protocol, see [SIGMA] and [SKEME]. Though the security of negotiated Child SAs does not depend on the strength of the encryption and integrity protection negotiated in the IKE SA, implementations MUST NOT negotiate NONE as the IKE integrity protection algorithm or ENCR_NULL as the IKE encryption algorithm.

有关本协议中许多密码设计选择的基本原理的信息，请参见[SIGMA]和[SKEME]。尽管协商的子 SA 的安全性不取决于 IKE SA 中协商的加密和完整性保护的强度，但实现不能将 NONE 协商为 IKE 完整性保护算法，也不能将 ENCR_NULL 协商为 IKE 加密算法。

When using pre-shared keys, a critical consideration is how to assure the randomness of these secrets. The strongest practice is to ensure that any pre-shared key contain as much randomness as the strongest key being negotiated. Deriving a shared secret from a password, name, or other low-entropy source is not secure. These sources are subject to dictionary and social-engineering attacks, among others.

在使用预共享密钥时，一个重要的考虑因素是如何确保这些秘密的随机性。最强的实践是确保任何预共享密钥包含的随机性与正在协商的最强密钥一样多。从密码、名称或其他低熵源派生共享秘密是不安全的。这些来源受到字典和社会工程攻击等。

The NAT_DETECTION_*_IP notifications contain a hash of the addresses and ports in an attempt to hide internal IP addresses behind a NAT. Since the IPv4 address space is only 32 bits, and it is usually very sparse, it would be possible for an attacker to find out the internal address used behind the NAT box by trying all possible IP addresses and trying to find the matching hash. The port numbers are normally fixed to 500, and the SPIs can be extracted from the packet. This reduces the number of hash calculations to 2^32. With an educated guess of the use of private address space, the number of hash calculations is much smaller. Designers should therefore not assume that use of IKE will not leak internal address information.

NAT_检测_*_IP 通知包含地址和端口的散列，试图将内部 IP 地址隐藏在 NAT 后面。由于 IPv4 地址空间只有 32 位，而且通常非常稀疏，攻击者可以通过尝试所有可能的 IP 地址并尝试查找匹配

的哈希来找出 NAT 框后面使用的内部地址。端口号通常固定为 500，可以从数据包中提取 SPI。这将哈希计算的数量减少到 2^32。通过对私有地址空间使用情况的合理猜测，散列计算的数量要小得多。因此，设计者不应该假设 IKE 的使用不会泄露内部地址信息。

When using an EAP authentication method that does not generate a shared key for protecting a subsequent AUTH payload, certain man-in-the-middle and server-impersonation attacks are possible [EAPMITM]. These vulnerabilities occur when EAP is also used in protocols that are not protected with a secure tunnel. Since EAP is a general-purpose authentication protocol, which is often used to provide single-signon facilities, a deployed IPsec solution that relies on an EAP authentication method that does not generate a shared key (also known as a non-key-generating EAP method) can become compromised due to the deployment of an entirely unrelated application that also happens to use the same non-key-generating EAP method, but in an unprotected fashion. Note that this vulnerability is not limited to just EAP, but can occur in other scenarios where an authentication infrastructure is reused. For example, if the EAP mechanism used by IKEv2 utilizes a token authenticator, a man-in-the-middle attacker

当使用不生成共享密钥的 EAP 身份验证方法来保护后续身份验证有效负载时，可能会发生某些中间人和服务器模拟攻击[EAPMITM]。当 EAP 也用于未受安全隧道保护的协议时，就会出现这些漏洞。由于 EAP 是一种通用身份验证协议，通常用于提供单一登录设施，因此部署的 IPsec 解决方案依赖于不生成共享密钥的 EAP 身份验证方法（也称为非密钥生成 EAP 方法）由于部署一个完全不相关的应用程序，该应用程序也碰巧使用相同的非密钥生成 EAP 方法，但以不受保护的方式，因此可能会受到损害。请注意，此漏洞不仅限于 EAP，而且在重用身份验证基础架构的其他场景中也可能发生。例如，如果 IKEv2 使用的 EAP 机制使用令牌身份验证器，则中间人攻击者

could impersonate the web server, intercept the token authentication exchange, and use it to initiate an IKEv2 connection. For this reason, use of non-key-generating EAP methods SHOULD be avoided where possible. Where they are used, it is extremely important that all usages of these EAP methods SHOULD utilize a protected tunnel, where the initiator validates the responder's certificate before initiating the EAP authentication. Implementers should describe the vulnerabilities of using non-key-generating EAP methods in the documentation of their implementations so that the administrators deploying IPsec solutions are aware of these dangers.

无法模拟 web 服务器，拦截令牌身份验证交换，并使用它启动 IKEv2 连接。因此，应尽可能避免使用非密钥生成 EAP 方法。在使用 EAP 方法的地方，这些 EAP 方法的所有使用都应使用受保护的隧道，在该隧道中，发起方在发起 EAP 身份验证之前验证响应方的证书，这一点非常重要。

实施者应在其实施的文档中描述使用非密钥生成 EAP 方法的漏洞，以便部署 IPsec 解决方案的管理员了解这些危险。

An implementation using EAP MUST also use a public-key-based authentication of the server to the client before the EAP authentication begins, even if the EAP method offers mutual authentication. This avoids having additional IKEv2 protocol variations and protects the EAP data from active attackers.

使用 EAP 的实现还必须在 EAP 身份验证开始之前使用基于公钥的服务器到客户端的身份验证，即使 EAP 方法提供了相互身份验证。这避免了额外的 IKEv2 协议变体，并保护 EAP 数据免受主动攻击者的攻击。

If the messages of IKEv2 are long enough that IP-level fragmentation is necessary, it is possible that attackers could prevent the exchange from completing by exhausting the reassembly buffers. The chances of this can be minimized by using the Hash and URL encodings instead of sending certificates (see Section 3.6). Additional mitigations are discussed in [DOSUDPPROT].

如果 IKEv2 的消息足够长，需要 IP 级别的碎片，则攻击者可能会通过耗尽重组缓冲区来阻止交换完成。通过使用哈希和 URL 编码，而不是发送证书，可以最大限度地减少这种情况的发生（参见第 3.6 节）。[Dosudprot]中讨论了其他缓解措施。

Admission control is critical to the security of the protocol. For example, trust anchors used for identifying IKE peers should probably be different than those used for other forms of trust, such as those used to identify public web servers. Moreover, although IKE provides a great deal of leeway in defining the security policy for a trusted peer's identity, credentials, and the correlation between them, having such security policy defined explicitly is essential to a secure implementation.

接纳控制对协议的安全性至关重要。例如，用于识别 IKE 对等点的信任锚应该与用于其他信任形式的信任锚不同，例如用于识别公共 web 服务器的信任锚。此外，尽管 IKE 在为可信对等方的身份、凭证以及它们之间的相关性定义安全策略方面提供了很大的余地，但明确定义此类安全策略对于安全实现至关重要。

5.1. Traffic Selector Authorization

5.1. 流量选择器授权

IKEv2 relies on information in the Peer Authorization Database (PAD) when determining what kind of Child SAs a peer is allowed to create. This process is described in Section 4.4.3 of [IPSECARCH]. When a peer requests the creation of an Child SA with some Traffic Selectors, the PAD must contain "Child SA Authorization Data" linking the identity authenticated by IKEv2 and the addresses permitted for Traffic Selectors.

IKEv2 在确定允许对等方创建何种子 SA 时，依赖于对等方授权数据库（PAD）中的信息。[IPSECARCH]第 4.4.3 节描述了该过程。当对等方请求使用某些流量选择器创建子 SA 时，PAD 必须包含“子 SA 授权数据”，链接 IKEv2 认证的身份和流量选择器允许的地址。

For example, the PAD might be configured so that authenticated identity "sgw23.example.com" is allowed to create Child SAs for 192.0.2.0/24, meaning this security gateway is a valid "representative" for these addresses. Host-to-host IPsec requires

例如，PAD 可以配置为允许经过身份验证的标识“sgw23.example.com”为 192.0.2.0/24 创建子 SA，这意味着此安全网关是这些地址的有效“代表”。主机到主机 IPsec 需要

similar entries, linking, for example, "fooserver4.example.com" with 198.51.100.66/32, meaning this identity is a valid "owner" or "representative" of the address in question.

类似的条目，例如，将“fooserver4.example.com”链接到 198.51.100.66/32，这意味着此标识是所述地址的有效“所有者”或“代表”。

As noted in [IPSECARCH], "It is necessary to impose these constraints on creation of child SAs to prevent an authenticated peer from spoofing IDs associated with other, legitimate peers". In the example given above, a correct configuration of the PAD prevents sgw23 from creating Child SAs with address 198.51.100.66, and prevents fooserver4 from creating Child SAs with addresses from 192.0.2.0/24.

如[IPSECARCH]中所述，“有必要对子 SA 的创建施加这些约束，以防止经过身份验证的对等方欺骗与其他合法对等方相关的 ID”。在上面给出的示例中，PAD 的正确配置可防止 sgw23 创建地址为 198.51.100.66 的子 SA，并防止 fooserver4 创建地址为 192.0.2.0/24 的子 SA。

It is important to note that simply sending IKEv2 packets using some particular address does not imply a permission to create Child SAs with that address in the Traffic Selectors. For example, even if sgw23 would be able to spoof its IP address as 198.51.100.66, it could not create Child SAs matching fooserver4's traffic.

需要注意的是，仅使用某个特定地址发送 IKEv2 数据包并不意味着允许在流量选择器中使用该地

址创建子 SA。例如，即使 sgw23 能够伪造其 IP 地址为 198.51.100.66，它也无法创建与 fooserver4 流量匹配的子 SAs。

The IKEv2 specification does not specify how exactly IP address assignment using Configuration payloads interacts with the PAD. Our interpretation is that when a security gateway assigns an address using Configuration payloads, it also creates a temporary PAD entry linking the authenticated peer identity and the newly allocated inner address.

IKEv2 规范没有指定使用配置有效负载的 IP 地址分配如何与 PAD 交互。我们的解释是，当安全网关使用配置有效负载分配地址时，它还会创建一个临时 PAD 条目，链接经过身份验证的对等身份和新分配的内部地址。

It has been recognized that configuring the PAD correctly may be difficult in some environments. For instance, if IPsec is used between a pair of hosts whose addresses are allocated dynamically using DHCP, it is extremely difficult to ensure that the PAD specifies the correct "owner" for each IP address. This would require a mechanism to securely convey address assignments from the DHCP server, and link them to identities authenticated using IKEv2.

人们已经认识到，在某些环境中，正确配置 PAD 可能很困难。例如，如果在一对使用 DHCP 动态分配地址的主机之间使用 IPsec，则很难确保 PAD 为每个 IP 地址指定正确的“所有者”。这需要一种机制来安全地传递来自 DHCP 服务器的地址分配，并将它们链接到使用 IKEv2 进行身份验证的身份。

Due to this limitation, some vendors have been known to configure their PADs to allow an authenticated peer to create Child SAs with Traffic Selectors containing the same address that was used for the IKEv2 packets. In environments where IP spoofing is possible (i.e., almost everywhere) this essentially allows any peer to create Child SAs with any Traffic Selectors. This is not an appropriate or secure configuration in most circumstances. See [H2HIPSEC] for an extensive discussion about this issue, and the limitations of host-to-host IPsec in general.

由于这一限制，已知一些供应商将其 PAD 配置为允许经过身份验证的对等方创建具有流量选择器的子 SA，该流量选择器包含用于 IKEv2 数据包的相同地址。在可能进行 IP 欺骗的环境中（即，几乎在任何地方），这基本上允许任何对等方使用任何流量选择器创建子 SA。在大多数情况下，这不是一种适当或安全的配置。请参阅[H2HIPSEC]，了解有关此问题的广泛讨论，以及主机到主机 IPsec 的一般限制。

6. IANA Considerations

6. IANA 考虑

[IKEV2] defined many field types and values. IANA has already registered those types and values in [IKEV2IANA], so they are not listed here again.

[IKEV2]定义了许多字段类型和值。IANA 已经在[IKEV2IANA]中注册了这些类型和值，因此这里不再列出它们。

Two items have been removed from the IKEv2 Configuration Payload Attribute Types table: INTERNAL_IP6_NBNS and INTERNAL_ADDRESS_EXPIRY.

已从 IKEv2 配置有效负载属性类型表中删除两项：内部\u IP6 \u NBNS 和内部\u 地址\u 到期。

Two new additions to the IKEv2 parameters "NOTIFY MESSAGES - ERROR TYPES" registry are defined here that were not defined in [IKEV2]:

此处定义了 IKEv2 参数“NOTIFY MESSAGES-ERROR TYPES”（通知消息-错误类型）注册表中的两个新添加项，它们在[IKEv2]中没有定义：

43 TEMPORARY_FAILURE 44 CHILD_SA_NOT_FOUND

43 暂时性故障 44 未找到儿童

IANA has changed the existing IKEv2 Payload Types table from:

IANA 已将现有的 IKEv2 有效负载类型表从：

46 Encrypted E [IKEV2]

46 加密 E[IKEV2]

to

到

46 Encrypted and Authenticated SK [This document]

46 加密和认证的 SK[本文件]

IANA has updated all references to RFC 4306 to point to this document.

IANA 已更新了 RFC 4306 的所有参考资料，以指向本文件。

7. Acknowledgements

7. 致谢

Many individuals in the IPsecME Working Group were very helpful in contributing ideas and text for this document, as well as in reviewing the clarifications suggested by others.

IPsecME 工作组中的许多个人在为本文件提供想法和文本以及审查其他人提出的澄清方面都非常有帮助。

The acknowledgements from the IKEv2 document were:

来自 IKEv2 文件的确认为：

This document is a collaborative effort of the entire IPsec WG. If there were no limit to the number of authors that could appear on an RFC, the following, in alphabetical order, would have been listed: Bill Aiello, Stephane Beaulieu, Steve Bellovin, Sara Bitan, Matt Blaze, Ran Canetti, Darren Dukes, Dan Harkins, Paul Hoffman, John Ioannidis, Charlie Kaufman, Steve Kent, Angelos Keromytis, Tero Kivinen, Hugo Krawczyk, Andrew Krywaniuk, Radia Perlman, Omer Reingold, and Michael Richardson. Many other people contributed to the design. It is an evolution of IKEv1, ISAKMP, and the IPsec DOI, each of which has its own list of authors. Hugh Daniel suggested the feature of having the initiator, in message 3, specify a name for the responder, and gave the feature the cute name "You Tarzan, Me Jane". David Faucher and Valery Smyslov helped refine the design of the Traffic Selector negotiation.

本文档是整个 IPsec 工作组的协作成果。如果 RFC 上可以出现的作者数量没有限制，那么按字母顺序排列的将列出以下作者：比尔·艾略、斯蒂芬·博列伊、史蒂夫·贝洛文、萨拉·比坦、马特·布雷兹、兰·卡内蒂、达伦·杜克斯、丹·哈金斯、保罗·霍夫曼、约翰·伊奥尼迪斯、查理·考夫曼、史蒂夫·肯特、安吉罗斯·科鲁米蒂斯、泰罗·基维宁、，雨果·克劳奇克、安德鲁·克鲁瓦尼乌克、拉迪亚·帕尔曼、奥马尔·莱因戈尔德和迈克尔·理查森。许多其他人对设计做出了贡献。它是 IKEv1、ISAKMP 和 IPsec DOI 的演变，它们都有自己的作者列表。Hugh Daniel 建议让发起者在消息 3 中为响应者指定一个名称，并给该功能起了一个可爱的名字“You Tarzan，Me Jane”。David Faucher 和 Valery Smyslov 帮助改进了交通选择器协商的设计。

8. References

8. 工具书类

8.1. Normative References

8.1. 规范性引用文件

[ADDGROUP] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, May 2003.

[ADDGROUP]Kivinen，T.和M.Kojo，“互联网密钥交换（IKE）的更多模指数（MODP）Diffie-Hellman组”，RFC 3526，2003年5月。

[ADDRIPV6] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

[ADDRIPV6]Hinden，R.和S.Deering，“IP版本6寻址体系结构”，RFC 42912006年2月。

[AEAD] Black, D. and D. McGrew, "Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol", RFC 5282, August 2008.

[AEAD]Black，D.和D.McGrew，“使用互联网密钥交换版本2（IKEv2）协议的加密有效载荷的认证加密算法”，RFC 5282，2008年8月。

[AESCMACPRF128] Song, J., Poovendran, R., Lee, J., and T. Iwata, "The Advanced Encryption Standard-Cipher-based Message Authentication Code-Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4615, August 2006.

[AESCMACPRF128]Song，J.，Poovendran，R.，Lee，J.，和T.Iwata，“互联网密钥交换协议（IKE）的基于高级加密标准密码的消息认证码伪随机函数-128（AES-CMAC-PRF-128）算法”，RFC 4615，2006年8月。

[AESXCBCPRF128] Hoffman, P., "The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)", RFC 4434, February 2006.

[AESXCBCPRF128]Hoffman，P.，“互联网密钥交换协议（IKE）的AES-XCBC-PRF-128算法”，RFC 4434，2006年2月。

[EAP] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[EAP]Aboba，B.，Blunk，L.，Vollbrecht，J.，Carlson，J.，和 H.Levkowetz，“可扩展认证协议（EAP）”，RFC 37482004 年 6 月。

[ECN] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[ECN]Ramakrishnan，K.，Floyd，S.，和 D.Black，“向 IP 添加显式拥塞通知（ECN）”，RFC 3168，2001 年 9 月。

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

[ESPCBC]Pereira，R.和 R.Adams，“ESP CBC 模式密码算法”，RFC 2451，1998 年 11 月。

[HTTP] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[HTTP]菲尔丁，R.，盖蒂斯，J.，莫卧儿，J.，弗莱斯蒂克，H.，马斯特，L.，利奇，P.，和 T. 伯纳斯李，“超文本传输协议——HTTP/1.1”，RFC2616，1999 年 6 月。

[IKEV2IANA] "Internet Key Exchange Version 2 (IKEv2) Parameters", <http://www.iana.org>.

[IKEV2IANA]“互联网密钥交换版本 2（IKEv2）参数”<http://www.iana.org>.

[IPSECARCH] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[IPSECARCH]Kent，S.和 K.Seo，“互联网协议的安全架构”，RFC 43012005 年 12 月。

[MUSTSHOULD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[MUSTSHOULD]Bradner，S.，“RFC 中用于表示需求水平的关键词”，BCP 14，RFC 2119，1997 年 3 月。

[PKCS1] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[PKCS1]Jonsson，J.和 B.Kaliski，“公钥密码标准（PKCS）#1:RSA 密码规范版本 2.1”，RFC 3447，2003 年 2 月。

[PKIX] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[PKIX]Cooper，D.，Santesson，S.，Farrell，S.，Boeyen，S.，Housley，R.，和W.Polk，“Internet X.509公钥基础设施证书和证书撤销列表（CRL）配置文件”，RFC 52802008年5月。

[RFC4307] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, December 2005.

[RFC4307]Schiller，J.“互联网密钥交换版本2（IKEv2）中使用的加密算法”，RFC 4307，2005年12月。

[UDPENCAPS] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, January 2005.

[UDPENCAPS]Huttunen，A.，Swander，B.，Volpe，V.，DiBurro，L.，和M.Stenberg，“IPsec ESP数据包的UDP封装”，RFC 3948，2005年1月。

[URLS] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[URL]Berners Lee，T.，Fielding，R.，和L.Masinter，“统一资源标识符（URI）：通用语法”，STD 66，RFC 3986，2005年1月。

8.2. Informative References

8.2. 资料性引用

[AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[AH]Kent，S.，“IP认证头”，RFC 4302，2005年12月。

[ARCHGUIDEPHIL] Bush, R. and D. Meyer, "Some Internet Architectural Guidelines and Philosophy", RFC 3439, December 2002.

[ARCHGUIDEPHIL]Bush，R.和D.Meyer，“一些互联网架构指南和哲学”，RFC 34392002年12月。

[ARCHPRINC] Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

[ARCHPRINC]Carpenter，B.，“互联网的建筑原理”，RFC 19581996 年 6 月。

[Clarif] Eronen, P. and P. Hoffman, "IKEv2 Clarifications and Implementation Guidelines", RFC 4718, October 2006.

[Clarif]Eronen，P.和 P.Hoffman，“IKEv2 澄清和实施指南”，RFC 4718，2006 年 10 月。

[DES] American National Standards Institute, "American National Standard for Information Systems-Data Link Encryption", ANSI X3.106, 1983.

[DES]美国国家标准协会，“美国信息系统数据链路加密国家标准”，ANSI X3.1061983。

[DH] Diffie, W. and M. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, V.IT-22 n. 6, June 1977.

[DH]Diffie，W.和 M.Hellman，“密码学的新方向”，IEEE 信息论交易，V.IT-22 n。1977 年 6 月 6 日。

[DIFFSERVARCH] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[DIFFSERVARCH]Blake，S.，Black，D.，Carlson，M.，Davies，E.，Wang，Z.，和 W.Weiss，“区分服务的架构”，RFC 24751998 年 12 月。

[DIFFSERVFIELD] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[DIFFSERVFIELD]Nichols，K.，Blake，S.，Baker，F.，和 D.Black，“IPv4 和 IPv6 报头中区分服务字段（DS 字段）的定义”，RFC 24741998 年 12 月。

[DIFFTUNNEL] Black, D., "Differentiated Services and Tunnels", RFC 2983, October 2000.

[DIFFTUNNEL]Black，D.，“差异化服务和隧道”，RFC 29832000 年 10 月。

[DOI] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[DOI]Piper，D.，“ISAKMP 解释的互联网 IP 安全域”，RFC 2407，1998 年 11 月。

[DOSUDPPROT] C. Kaufman, R. Perlman, and B. Sommerfeld, "DoS protection for

UDP-based protocols", ACM Conference on Computer and Communications Security, October 2003.

[Dosudprot]C.Kaufman，R.Perlman 和 B.Sommerfeld，“基于 UDP 协议的 DoS 保护”，ACM 计算机和通信安全会议，2003 年 10 月。

[DSS] National Institute of Standards and Technology, U.S. Department of Commerce, "Digital Signature Standard", Draft FIPS 186-3, June 2008.

[DSS]美国商务部国家标准与技术研究所，“数字签名标准”，FIPS 186-3 草案，2008 年 6 月。

[EAI] Abel, Y., "Internationalized Email Headers", RFC 5335, September 2008.

[EAI]Abel，Y.，“国际化电子邮件标题”，RFC 53352008 年 9 月。

[EAP-IANA] "Extensible Authentication Protocol (EAP) Registry: Method Types", <http://www.iana.org>.

[EAP-IANA]“可扩展身份验证协议（EAP）注册表：方法类型”<http://www.iana.org>.

[EAPMITM] N. Asokan, V. Nierni, and K. Nyberg, "Man-in-the-Middle in Tunneled Authentication Protocols", November 2002, <http://eprint.iacr.org/2002/163>.

[EAPMITM]N.Asokan，V.Nierni 和 K.Nyberg，“隧道认证协议中的中间人”，2002 年 11 月<http://eprint.iacr.org/2002/163>.

[ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[ESP]Kent，S.，“IP 封装安全有效负载（ESP）”，RFC 4303，2005 年 12 月。

[EXCHANGEANALYSIS] R. Perlman and C. Kaufman, "Analysis of the IPsec key exchange Standard", WET-ICE Security Conference, MIT, 2001, <http://sec.femto.org/wetice-2001/papers/radia-paper.pdf>.

[EXCHANGEANALYSIS]R.Perlman 和 C.Kaufman，“IPsec 密钥交换标准分析”，湿冰安全会议，麻省理工学院，2001 年<http://sec.femto.org/wetice-2001/papers/radia-paper.pdf>.

[H2HIPSEC] Aura, T., Roe, M., and A. Mohammed, "Experiences with Host-to-Host IPsec", 13th International Workshop on Security Protocols, Cambridge, UK, April 2005.

[H2HIPSEC]Aura，T.，Roe，M.和A.Mohammed，“主机对主机IPsec的经验”，第13届安全协议国际研讨会，英国剑桥，2005年4月。

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

[HMAC]Krawczyk，H.，Bellare，M.，和R.Canetti，“HMAC：用于消息身份验证的键控哈希”，RFC 2104，1997年2月。

[IDEA] X. Lai, "On the Design and Security of Block Ciphers", ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlag, 1992.

[IDEA]X.Lai，“关于分组密码的设计和安全”，信息处理ETH系列，v。康斯坦茨：哈东·高尔·韦拉格，1992年。

[IDNA] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[IDNA]Klensin，J.，“应用程序的国际化域名（IDNA）：定义和文档框架”，RFC 58902010年8月。

[IKEV1] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

[IKEV1]Harkins，D.和D.Carrel，“互联网密钥交换（IKE）”，RFC 2409，1998年11月。

[IKEV2] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[IKEV2]Kaufman，C.，“互联网密钥交换（IKEV2）协议”，RFC4306，2005年12月。

[IP] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[IP]Postel，J.，“互联网协议”，STD 5，RFC 7911981年9月。

[IP-COMP] Shacham, A., Monsour, B., Pereira, R., and M. Thomas, "IP Payload Compression Protocol (IPComp)", RFC 3173, September 2001.

[IP-COMP]Shacham，A.，Monsour，B.，Pereira，R.，和M.Thomas，“IP有效载荷压缩协议（IPComp）”，RFC 31732001年9月。

[IPSECARCH-OLD] Kent, S. and R. Atkinson, "Security Architecture for the Internet

Protocol", RFC 2401, November 1998.

[IPSECARCH-OLD]Kent，S.和 R.Atkinson，“互联网协议的安全架构”，RFC 2401，1998 年 11 月。

[IPV6CONFIG] Eronen, P., Laganier, J., and C. Madson, "IPv6 Configuration in Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5739, February 2010.

[IPV6CONFIG]Eronen，P.，Laganier，J.，和 C.Madson，“互联网密钥交换协议版本 2（IKEv2）中的 IPv6 配置”，RFC 5739，2010 年 2 月。

[ISAKMP] Maughan, D., Schneider, M., and M. Schertler, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[ISAKMP]Maughan，D.，Schneider，M.和 M.Schertler，“互联网安全协会和密钥管理协议（ISAKMP）”，RFC 2408，1998 年 11 月。

[MAILFORMAT] Resnick, P., Ed., "Internet Message Format", RFC 5322, October 2008.

[MAILFORMAT]Resnick，P.，Ed.“互联网信息格式”，RFC5322，2008 年 10 月。

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[MD5]Rivest，R.，“MD5 消息摘要算法”，RFC 13211992 年 4 月。

[MIPV6] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[MIPV6]Johnson，D.，Perkins，C.，和 J.Arkko，“IPv6 中的移动支持”，RFC 37752004 年 6 月。

[MLDV2] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004.

[MLDV2]Vida，R.和 L.Costa，“IPv6 多播侦听器发现版本 2（MLDV2）”，RFC 3810，2004 年 6 月。

[MOBIKE] Eronen, P., "IKEv2 Mobility and Multihoming Protocol (MOBIKE)", RFC 4555, June 2006.

[MOBIKE]Eronen，P.，“IKEv2 移动和多址协议（MOBIKE）”，RFC4552006 年 6 月。

[MODES] National Institute of Standards and Technology, U.S. Department of Commerce, "Recommendation for Block Cipher Modes of Operation", SP 800-38A, 2001.

[模式]美国商务部国家标准与技术研究所，“分组密码操作模式建议”，SP 800-38A，2001 年。

[NAI] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[NAI]Aboba，B.，Beadles，M.，Arkko，J.，和 P.Erenen，“网络接入标识符”，RFC 42822005 年 12 月。

[NATREQ] Aboba, B. and W. Dixon, "IPsec-Network Address Translation (NAT) Compatibility Requirements", RFC 3715, March 2004.

[Natraq]Aboba，B.和 W.Dixon，“IPsec 网络地址转换（NAT）兼容性要求”，RFC 3715，2004 年 3 月。

[OAKLEY] Orman, H., "The OAKLEY Key Determination Protocol", RFC 2412, November 1998.

[OAKLEY]Orman，H.，“OAKLEY 密钥确定协议”，RFC 2412，1998 年 11 月。

[PFKEY] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, July 1998.

[PFKEY]McDonald，D.，Metz，C.，和 B.Phan，“PF_密钥管理 API，版本 2”，RFC 2367，1998 年 7 月。

[PHOTURIS] Karn, P. and W. Simpson, "Photuris: Session-Key Management Protocol", RFC 2522, March 1999.

[PHOTURIS]Karn，P.和 W.Simpson，“PHOTURIS:会话密钥管理协议”，RFC2521999 年 3 月。

[RANDOMNESS] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[随机性]Eastlake，D.，Schiller，J.，和 S.Crocker，“安全性的随机性要求”，BCP 106，RFC 40862005 年 6 月。

[REAUTH] Nir, Y., "Repeated Authentication in Internet Key Exchange (IKEv2)

Protocol", RFC 4478, April 2006.

[REAUTH]Nir，Y，“互联网密钥交换（IKEv2）协议中的重复认证”，RFC 4478，2006 年 4 月。

[REUSE] Menezes, A. and B. Ustaoglu, "On Reusing Ephemeral Keys In Diffie-Hellman Key Agreement Protocols", December 2008, <http://www.cacr.math.uwaterloo.ca/techreports/2008/ cacr2008-24.pdf>.

[重用]Menezes，A.和 B.Ustaoglu，“关于在 Diffie-Hellman 密钥协议协议中重用临时密钥”，2008 年 12 月<http://www.cacr.math.uwaterloo.ca/techreports/2008/ cacr2008-24.pdf>。

[ROHCV2] Ertekin, E., Christou, C., Jasani, R., Kivinen, T., and C. Bormann, "IKEv2 Extensions to Support Robust Header Compression over IPsec", RFC 5857, May 2010.

[ROHCV2]Ertekin，E.，Christou，C.，Jasani，R.，Kivinen，T.，和 C.Bormann，“IKEv2 扩展以支持 IPsec 上的健壮报头压缩”，RFC 5857，2010 年 5 月。

[RSA] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", February 1978.

[RSA]R.Rivest、A.Shamir 和 L.Adleman，“获取数字签名和公钥密码系统的方法”，1978 年 2 月。

[SHA] National Institute of Standards and Technology, U.S. Department of Commerce, "Secure Hash Standard", FIPS 180-3, October 2008.

[SHA]美国商务部国家标准与技术研究所，“安全哈希标准”，FIPS 180-32008 年 10 月。

[SIGMA] H. Krawczyk, "SIGMA: the `SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and its Use in the IKE Protocols", Advances in Cryptography - CRYPTO 2003 Proceedings LNCS 2729, 2003, <http:// www.informatik.uni-trier.de/~ley/db/conf/crypto/ crypto2003.html>.

[SIGMA]H.Krawczyk，“SIGMA：认证 Diffie Hellman 的`符号和 MAc'方法及其在 IKE 协议中的使用”，《密码学进展》《密码学 2003 年会议录》，LNCS 27292003，<http://www.informatik.uni-trier.de/~ley/db/conf/CRYPTO/crypto2003.html>。

[SKEME] H. Krawczyk, "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security , 1996.

[SKEME]H.Krawczyk，“SKEME：一种用于互联网的通用安全密钥交换机制”，IEEE 1996年网络和分布式系统安全研讨会论文集，1996年。

[TRANSPARENCY] Carpenter, B., "Internet Transparency", RFC 2775, February 2000.

[透明度]Carpenter，B.，“互联网透明度”，RFC 27752000年2月。

Appendix A. Summary of Changes from IKEv1

附录A.IKEv1的变更汇总

The goals of this revision to IKE are:

IKE本次修订的目标是：

1. To define the entire IKE protocol in a single document, replacing RFCs 2407, 2408, and 2409 and incorporating subsequent changes to support NAT Traversal, Extensible Authentication, and Remote Address acquisition;

1. 在单个文档中定义整个IKE协议，替换RFCs 2407、2408和2409，并合并后续更改以支持NAT遍历、可扩展身份验证和远程地址获取；

2. To simplify IKE by replacing the eight different initial exchanges with a single four-message exchange (with changes in authentication mechanisms affecting only a single AUTH payload rather than restructuring the entire exchange) see [EXCHANGEANALYSIS];

2. 要简化IKE，请将八个不同的初始交换替换为一个四消息交换（身份验证机制的更改仅影响单个身份验证有效负载，而不是重新构造整个交换），请参阅[EXCHANGEANALYSIS]；

3. To remove the Domain of Interpretation (DOI), Situation (SIT), and Labeled Domain Identifier fields, and the Commit and Authentication only bits;

3. 删除解释域（DOI）、情况（SIT）和标记的域标识符字段，以及仅提交和验证位；

4. To decrease IKE's latency in the common case by making the initial exchange be 2 round trips (4 messages), and allowing the ability to piggyback setup of a Child SA on that exchange;

4. 在常见情况下，通过使初始交换为 2 次往返（4 条消息），并允许在该交换上搭载子 SA 的设置，来减少 IKE 的延迟;

5. To replace the cryptographic syntax for protecting the IKE messages themselves with one based closely on ESP to simplify implementation and security analysis;

5. 将用于保护 IKE 消息本身的密码语法替换为紧密基于 ESP 的密码语法，以简化实现和安全分析;

6. To reduce the number of possible error states by making the protocol reliable (all messages are acknowledged) and sequenced. This allows shortening CREATE_CHILD_SA exchanges from 3 messages to 2;

6. 通过使协议可靠（所有消息均已确认）并按顺序排列，减少可能的错误状态数量。这允许将 CREATE_CHILD_SA 交换从 3 条消息缩短到 2 条;

7. To increase robustness by allowing the responder to not do significant processing until it receives a message proving that the initiator can receive messages at its claimed IP address;

7. 通过允许响应方在接收到证明发起方可以在其声明的 IP 地址接收消息之前不进行重要处理来提高健壮性;

8. To fix cryptographic weaknesses such as the problem with symmetries in hashes used for authentication (documented by Tero Kivinen);

8. 修复加密弱点，例如用于身份验证的哈希中的对称性问题（由 Tero Kivinen 记录）;

9. To specify Traffic Selectors in their own payloads type rather than overloading ID payloads, and making more flexible the Traffic Selectors that may be specified;

9. 在自己的有效负载类型中指定流量选择器，而不是重载 ID 有效负载，并使可能指定的流量选择器更加灵活;

10. To specify required behavior under certain error conditions or when data that is not understood is received in order to make it easier to make future revisions in a way that does not break backward compatibility;

10. 指定在某些错误条件下或当接收到不理解的数据时所需的行为，以便更容易以不破坏向后兼容性的方式进行将来的修订;

11. To simplify and clarify how shared state is maintained in the presence of network failures and DoS attacks; and

11. 简化并澄清在出现网络故障和 DoS 攻击时如何保持共享状态；和

12. To maintain existing syntax and magic numbers to the extent possible to make it likely that implementations of IKEv1 can be enhanced to support IKEv2 with minimum effort.

12. 尽可能地维护现有的语法和幻数，使 IKEv1 的实现能够得到增强，以尽可能少的努力支持 IKEv2。

Appendix B. Diffie-Hellman Groups

附录 B.Diffie-Hellman 集团

There are two Diffie-Hellman groups defined here for use in IKE. These groups were generated by Richard Schroeppel at the University of Arizona. Properties of these primes are described in [OAKLEY].

这里定义了两个 Diffie-Hellman 组用于 IKE。这些小组是由亚利桑那大学的 Richard Schroeppel 发明的。[OAKLEY]中描述了这些素数的性质。

The strength supplied by group 1 may not be sufficient for typical uses and is here for historic reasons.

由于历史原因，第 1 组提供的强度可能不足以用于典型用途。

Additional Diffie-Hellman groups have been defined in [ADDGROUP].

[ADDGROUP]中定义了其他 Diffie-Hellman 组。

B.1. Group 1 - 768-bit MODP

B.1. 第 1 组-768 位 MODP

This group is assigned ID 1 (one).

此组被分配 ID 1（一个）。

The prime is: 2^768 - 2 ^704 - 1 + 2^64 * { [2^638 pi] + 149686 } Its hexadecimal value is:

素数是：2^768-2^704-1+2^64*{[2^638 pi]+149686}其十六进制值是：

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

FFFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08 8A67CC74
020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B 302B0A6D
F25F1437 4FE1356D 6D51C245 E485B576 625E6 F44C42E9 A63A3620 FFFFFFFFFF
FFFFFF

The generator is 2.

发电机是 2。

B.2. Group 2 - 1024-bit MODP

B.2. 第 2 组-1024 位 MODP

This group is assigned ID 2 (two).

此组被分配 ID 2（两个）。

The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }. Its hexadecimal value is:

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

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08
8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B
302B0A6D F25F1437 4FE1356D 6D51C245 E485B576 625E7EC6 F44C42E9
A637ED6B 0BFF5CB6 F406B7ED EE386BFB 5A899FA5 AE9F2411 7C4B1FE6
49286651 ECE65381 FFFFFFFF FFFFFFFF

FFFFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1 29024E08 8A67CC74
020BBEA6 3B139B22 514A0879 8E3404DD EF9519B3 CD3A431B 302B0A6D
F25F1437 4FE1356D 6D51C245 E485B576 625E6 F44C42E9 A637ED6B 0BFF5CB6
F406B7 EE386B5A899FA5 AE9F2411 7C4B16 49286651 FFFFFFFFFFFFFFFFFF

The generator is 2.

发电机是 2。

Appendix C. Exchanges and Payloads

附录 C.交换和有效载荷

This appendix contains a short summary of the IKEv2 exchanges, and what payloads can appear in which message. This appendix is purely informative; if it disagrees with the body of this document, the other text is considered correct.

本附录包含 IKEv2 交换的简短摘要，以及在哪个消息中可以显示哪些有效负载。本附录仅供参考；如果与本文件正文不一致，则认为其他文本正确。

Vendor ID (V) payloads may be included in any place in any message. This sequence here shows what are the most logical places for them.

供应商 ID（V）有效载荷可包含在任何消息的任何位置。这里的顺序显示了它们最符合逻辑的位置。

C.1. IKE_SA_INIT Exchange

C.1. IKE_SA_初始交换

```
request             --> [N(COOKIE)],
                        SA, KE, Ni,
                        [N(NAT_DETECTION_SOURCE_IP)+,
                         N(NAT_DETECTION_DESTINATION_IP)],
                        [V+][N+]
```

```
request             --> [N(COOKIE)],
                        SA, KE, Ni,
                        [N(NAT_DETECTION_SOURCE_IP)+,
                         N(NAT_DETECTION_DESTINATION_IP)],
                        [V+][N+]
```

```
normal response     <-- SA, KE, Nr,
(no cookie)             [N(NAT_DETECTION_SOURCE_IP),
                         N(NAT_DETECTION_DESTINATION_IP)],
                        [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                        [V+][N+]
```

```
normal response     <-- SA, KE, Nr,
(no cookie)             [N(NAT_DETECTION_SOURCE_IP),
                         N(NAT_DETECTION_DESTINATION_IP)],
                        [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                        [V+][N+]
```

```
   cookie response      <-- N(COOKIE),
                            [V+][N+]
```

```
   cookie response      <-- N(COOKIE),
                            [V+][N+]
```

different Diffie- <-- N(INVALID_KE_PAYLOAD), Hellman group [V+][N+] wanted

不同的差异-<--N（无效的负载），需要 Hellman 组[V+][N+]

C.2. IKE_AUTH Exchange without EAP

C.2. 无 EAP 的 IKE_身份验证交换

```
   request              --> IDi, [CERT+],
                            [N(INITIAL_CONTACT)],
                            [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                            [IDr],
                            AUTH,
                            [CP(CFG_REQUEST)],
                            [N(IPCOMP_SUPPORTED)+],
                            [N(USE_TRANSPORT_MODE)],
                            [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                            [N(NON_FIRST_FRAGMENTS_ALSO)],
                            SA, TSi, TSr,
                            [V+][N+]
```

```
   request              --> IDi, [CERT+],
                            [N(INITIAL_CONTACT)],
                            [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                            [IDr],
                            AUTH,
                            [CP(CFG_REQUEST)],
                            [N(IPCOMP_SUPPORTED)+],
                            [N(USE_TRANSPORT_MODE)],
                            [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                            [N(NON_FIRST_FRAGMENTS_ALSO)],
                            SA, TSi, TSr,
                            [V+][N+]
```

```
   response             <-- IDr, [CERT+],
                  AUTH,
                  [CP(CFG_REPLY)],
                  [N(IPCOMP_SUPPORTED)],
                  [N(USE_TRANSPORT_MODE)],
                  [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                  [N(NON_FIRST_FRAGMENTS_ALSO)],
                  SA, TSi, TSr,
                  [N(ADDITIONAL_TS_POSSIBLE)],
                  [V+][N+]


  response          <-- IDr, [CERT+],
                  AUTH,
                  [CP(CFG_REPLY)],
                  [N(IPCOMP_SUPPORTED)],
                  [N(USE_TRANSPORT_MODE)],
                  [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                  [N(NON_FIRST_FRAGMENTS_ALSO)],
                  SA, TSi, TSr,
                  [N(ADDITIONAL_TS_POSSIBLE)],
                  [V+][N+]
```

error in Child SA <-- IDr, [CERT+], creation AUTH, N(error), [V+][N+]

子 SA 中的错误<--IDr[CERT+]，创建身份验证，N（错误），[V+][N+]

C.3. IKE_AUTH Exchange with EAP

C.3. IKE_与 EAP 的身份验证交换

```
  first request       --> IDi,
                  [N(INITIAL_CONTACT)],
                  [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                  [IDr],
                  [CP(CFG_REQUEST)],
                  [N(IPCOMP_SUPPORTED)+],
                  [N(USE_TRANSPORT_MODE)],
                  [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                  [N(NON_FIRST_FRAGMENTS_ALSO)],
                  SA, TSi, TSr,
                  [V+][N+]


  first request       --> IDi,
                  [N(INITIAL_CONTACT)],
                  [[N(HTTP_CERT_LOOKUP_SUPPORTED)], CERTREQ+],
                  [IDr],
                  [CP(CFG_REQUEST)],
                  [N(IPCOMP_SUPPORTED)+],
                  [N(USE_TRANSPORT_MODE)],
                  [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                  [N(NON_FIRST_FRAGMENTS_ALSO)],
                  SA, TSi, TSr,
                  [V+][N+]
```

first response <-- IDr, [CERT+], AUTH, EAP, [V+][N+]

第一个响应<--IDr、[CERT+]，AUTH、EAP、[V+][N+]

/ --> EAP repeat 1..N times | \ <-- EAP

/-->EAP 重复 1..N 次| \<--EAP

last request --> AUTH

上次请求-->身份验证

```
  last response       <-- AUTH,
                         [CP(CFG_REPLY)],
                         [N(IPCOMP_SUPPORTED)],
                         [N(USE_TRANSPORT_MODE)],
                         [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                         [N(NON_FIRST_FRAGMENTS_ALSO)],
                         SA, TSi, TSr,
                         [N(ADDITIONAL_TS_POSSIBLE)],
                         [V+][N+]
```

```
  last response       <-- AUTH,
                         [CP(CFG_REPLY)],
                         [N(IPCOMP_SUPPORTED)],
                         [N(USE_TRANSPORT_MODE)],
                         [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                         [N(NON_FIRST_FRAGMENTS_ALSO)],
                         SA, TSi, TSr,
                         [N(ADDITIONAL_TS_POSSIBLE)],
                         [V+][N+]
```

C.4. CREATE_CHILD_SA Exchange for Creating or Rekeying Child SAs

C.4. 创建子 SA 交换以创建或重新键入子 SA

```
request             --> [N(REKEY_SA)],
                   [CP(CFG_REQUEST)],
                   [N(IPCOMP_SUPPORTED)+],
                   [N(USE_TRANSPORT_MODE)],
                   [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                   [N(NON_FIRST_FRAGMENTS_ALSO)],
                   SA, Ni, [KEi], TSi, TSr
                   [V+][N+]
```

```
request             --> [N(REKEY_SA)],
                   [CP(CFG_REQUEST)],
                   [N(IPCOMP_SUPPORTED)+],
                   [N(USE_TRANSPORT_MODE)],
                   [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                   [N(NON_FIRST_FRAGMENTS_ALSO)],
                   SA, Ni, [KEi], TSi, TSr
                   [V+][N+]
```

```
normal              <-- [CP(CFG_REPLY)],
response                 [N(IPCOMP_SUPPORTED)],
                   [N(USE_TRANSPORT_MODE)],
                   [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                   [N(NON_FIRST_FRAGMENTS_ALSO)],
                   SA, Nr, [KEr], TSi, TSr,
                   [N(ADDITIONAL_TS_POSSIBLE)]
                   [V+][N+]
```

```
normal              <-- [CP(CFG_REPLY)],
response                 [N(IPCOMP_SUPPORTED)],
                   [N(USE_TRANSPORT_MODE)],
                   [N(ESP_TFC_PADDING_NOT_SUPPORTED)],
                   [N(NON_FIRST_FRAGMENTS_ALSO)],
                   SA, Nr, [KEr], TSi, TSr,
                   [N(ADDITIONAL_TS_POSSIBLE)]
                   [V+][N+]
```

error case <-- N(error)

错误案例<--N（错误）

different Diffie- <-- N(INVALID_KE_PAYLOAD), Hellman group [V+][N+] wanted

不同的差异-<--N（无效的负载），需要 Hellman 组[V+][N+]

C.5. CREATE_CHILD_SA Exchange for Rekeying the IKE SA

C.5. 创建用于重新设置 IKE SA 密钥的子 SA 交换

```
request            --> SA, Ni, KEi
                   [V+][N+]
```

```
request            --> SA, Ni, KEi
                   [V+][N+]
```

response <-- SA, Nr, KEr [V+][N+]

响应<--SA、Nr、KEr[V+][N+]

C.6. INFORMATIONAL Exchange

C.6. 信息交流

```
request            --> [N+],
                   [D+],
                   [CP(CFG_REQUEST)]
```

```
request            --> [N+],
                   [D+],
                   [CP(CFG_REQUEST)]
```

response <-- [N+], [D+], [CP(CFG_REPLY)]

答复<-[N+]，[D+]，[CP（CFG_答复）]

Authors' Addresses

作者地址

Charlie Kaufman Microsoft 1 Microsoft Way Redmond, WA 98052 US

Charlie Kaufman 微软 1 号微软路雷德蒙德，华盛顿州，美国 98052

Phone: 1-425-707-3335 EMail: charliek@microsoft.com

电话：1-425-707-3335 电子邮件：charliek@microsoft.com

Paul Hoffman VPN Consortium 127 Segre Place Santa Cruz, CA 95060 US

美国加利福尼亚州圣克鲁斯塞格雷广场 127 号保罗·霍夫曼私人有限公司，邮编 95060

Phone: 1-831-426-9827 EMail: paul.hoffman@vpnc.org

电话：1-831-426-9827 电子邮件：保罗。hoffman@vpnc.org

Yoav Nir Check Point Software Technologies Ltd. 5 Hasolelim St. Tel Aviv 67897 Israel

以色列特拉维夫 Hasolelim 街 5 号 Yoav Nir Check Point 软件技术有限公司 67897

EMail: ynir@checkpoint.com

EMail: ynir@checkpoint.com

Pasi Eronen Independent

非依赖性 Pasi

EMail: pe@iki.fi

EMail: pe@iki.fi