3.2.3 SA(Security Association) & IKE(Internet Key Exchange)

在前面两个小节中，我们是有意忽略通信双方如何知道加密算法、密钥等信息，只是假设它们就是知道。这一小节，我们就来讲述，通信双方是如何知道这些信息的。

3.2.3.1 SPD(Security Policy Database)

在讲述 SA 与 IKE 之前，我们首先要讲述 SPD（Security Policy Database），安全策略数据库。这需要从一个路由器的路由转发说起，如下图所示，是一个传统 IP 报文转发过程（简化描述）：

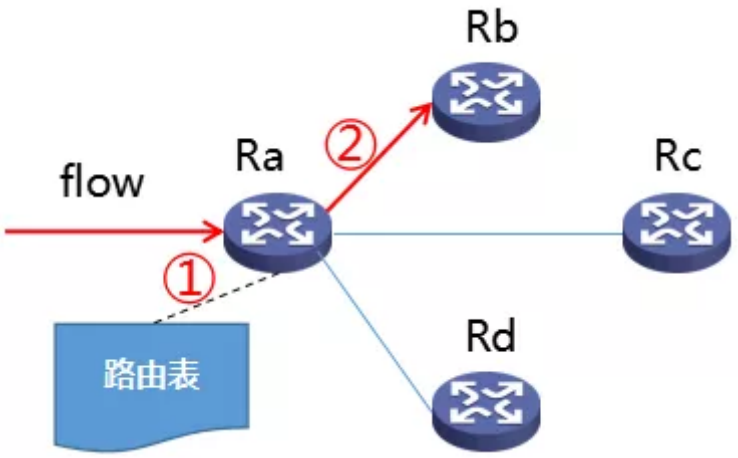

图 1 传统 IP 报文转发过程（简化描述）

一个传统的 IP 报文转发过程，简单地说，就两步：

（1）一个 flow 进来以后，根据路由表，查找出口

（2）做相应的处理，从出口转发出去

但是，如果需要 IPSec，不能就这样直接转发出去了。这样的话，就没 IPSec 啥事了。所以，一个 IPSec 转发的过程，如下图所示（简化描述）：

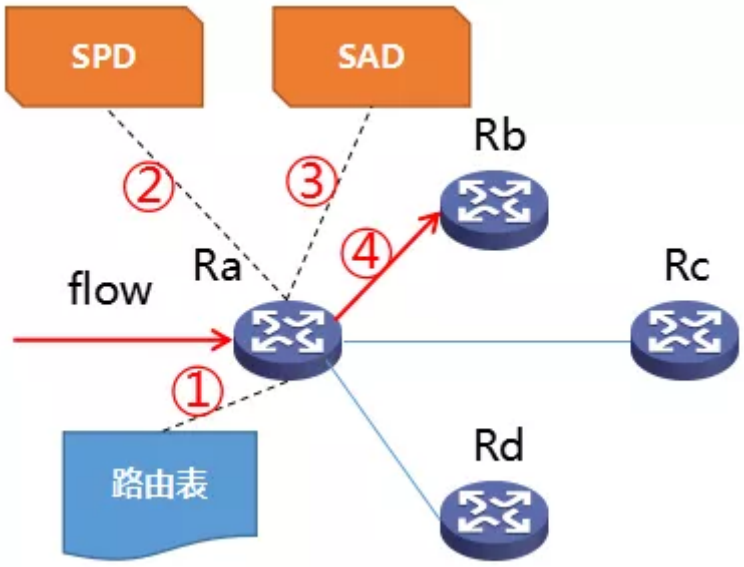

图 2 IPSec 转发过程（简化描述）

一个 IPSec 转发过程，需要四步（简化描述，并且是示意描述）：

（1）一个 flow 进来以后，根据路由表，查找出口

（2）查找 SPD（Security Policy Database），确定安全策略（下文会讲述）

（3）查找 SAD（Security Association Database），确定安全行为和安全参数（下小节会讲述）

（4）做相应的处理，从出口转发出去

这里涉及到了 SPD 和 SAD，SAD 暂时先不用纠结，我们本小节先看看 SPD。当一个流经过一个路由器时，在 IPSec 的视角，它有三种行为，如下图所示：

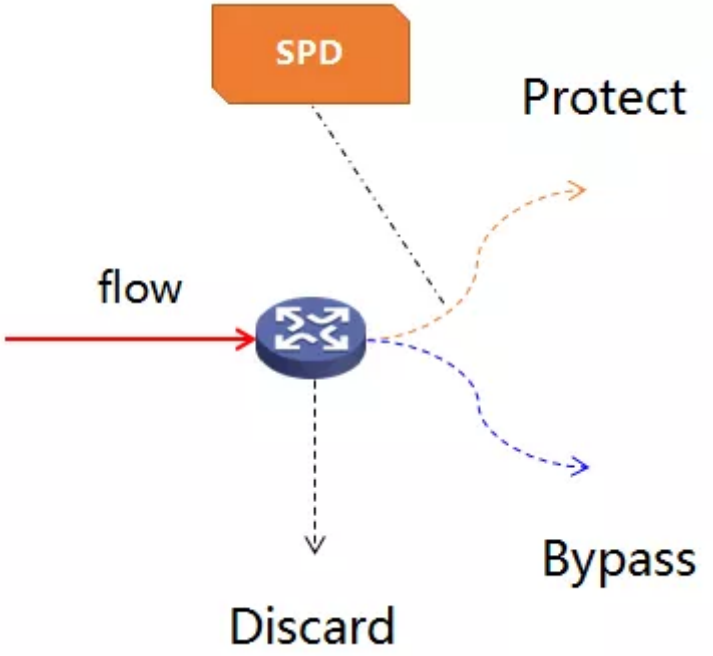

图 3 一个流的三种行为

这三种行为，就是安全策略（Security Policy）：

l Discard：就是直接丢弃。

l Bypass：是相对于 IPSec 而言的，即不使用 IPSec 进行保护，而是当作一个普通的路由转发。

l Protect：就是使用 IPSec，在保护的基础上，再进行路由转发。（IPSec 的行为有两种，一个是传输模式，另一个是隧道模式。前文我们一直以传输模式为例，进行讲述。后文我们会讲述隧道模式）

而 IPSec，针对一个流，是如何知道该采用哪个策略呢。这就涉及到了 SPD。IPSec 就是通过查询 SPD 里面的策略，来决定如何处理一个流。

SPD 并不是传统意义上，真的是一个数据库，这只是一种称谓。实际实现时，可以用任何可行的方法（比如一个文本文件，比如一个 excel 文件，比如真的是使用一个 mysql 数据库，等等，只要你觉得这个方法可行，你就使用，没有人限制你）。

SPD 可以简单地理解为一个表，而这个表结构，也可以简单理解如下：

Flow					Behavior
源 IP	目的 IP	源端口	目的端口	协议	

***	***	***	***	***	Discard
***	***	***	***	***	Bypass
***	***	***	***	***	Protect

表1 简化版 SPD 表结构

SPD 的内容，是人工输入的，说的高大上一点，就是通过一个管理接口配置的。

关于 SPD 更详细的内容，请参阅 RFC4301，笔者在这里就不详述了。

3.2.3.2 SAD（Security Association Database)

在 SPD 这一小节了，我们讲述了，针对一个流，可以有三种策略： Discard，Bypass， Protect。如果选择了 Protect 这一策略，那么更进一步的内容该如何选择呢？比如前文说的，可以是 AH，也可以是 ESP，可以是传输模式，也可以是隧道模式，等等。

当一个流是采取 Protect 策略时，IPSec 下一步就是去 SAD 查询进一步的信息。SAD 跟 SPD 一样，并不一定需要是一个真正的数据库，它只是一个将所有 SA（Security Association）存储的地方而已。所以，重点是 SA。

SA（Security Association），可能是中外语言的差异，没法顾名思义，只能硬理解——不用管它的名字到底是啥意思，只管知道它到底有哪些内容。SA 的内容比较多，我们逐步打开。

3.2.3.2.1 SA 表达了针对一个“流”所使用的 IPSec 协议

我们知道，IPSec 的安全协议有两种：AH 和 ESP。IPSec Peer（比如两个路由器）之间，首先就需要协商它们之间需要采取哪个协议（当然，所谓的协商，归根结底是人的协商。所以，这种“协商”或者就是人工直接配置好，直接将 IPSec Peer 配置好。或者人工通过间接配置相关参数，由 IP Peer 通过 IKE 协议进行协商。这个是后话了，我们会在下一个小节进行描述）。

SA 不是协商这些参数的协议，而是定义这些参数的一个数据结构（最终存储在 SAD 中）。我们先不完全打开 SA 的数据结构，先看一部分内容，三个字段：

l SPI，Security Parameter Index，就是前文介绍的 AH Header 及 ESP Header 中的 SPI。他们中的 SPI 的值，就是来自 SA 这个字段。IPSec 会读取这个值，然后赋给 AH Header 或者 ESP Header 中的 SPI 字段。SA 中 的 SPI，通过人工配置，或者由 IKE 协商。SPI 是一个32位的整数。（无符号？）

l IP目的地址，这个是鄙人迷惑的地方。标识一个流，传统的做法是通过一个五元组（源 IP， 目的 IP，源端口，目的端口，协议号）。而这里仅仅是使用一个目的 IP，显得流太“粗”了一点。不过先不纠结，这个字段实际上就是标识一个流。

l 安全协议标识符，这个字段表示，一个流是采用 AH Protocol 还是 ESP Protocol 进行的数据流保护。

我们先忽略 IPSec 其他参数（比如安全算法，秘钥等等），假设就只需要 AH 或者 ESP 就可以了。我们通过下图来分析上述三个参数：

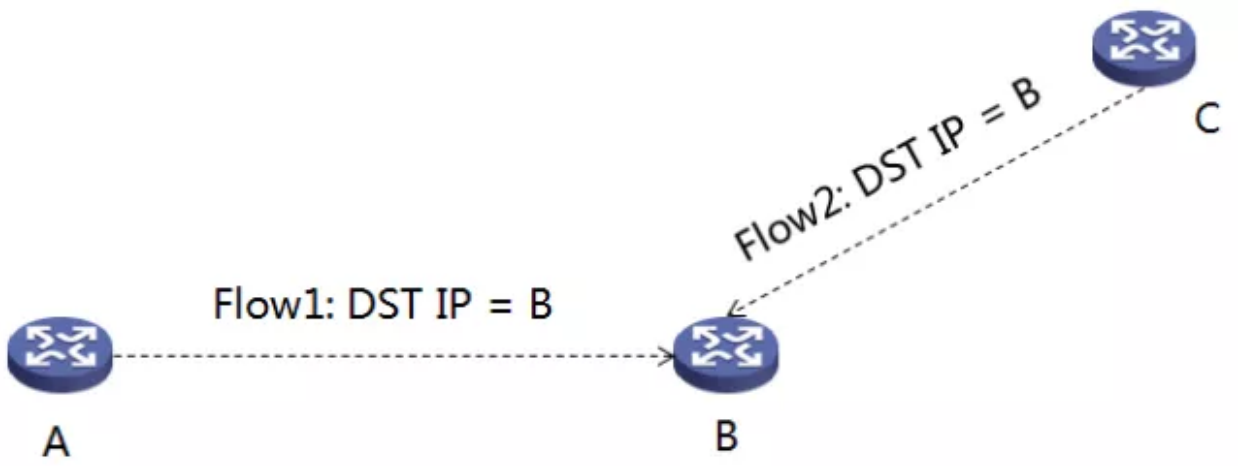

图 4 SA Flow 简单示意图

我们首先看 A 路由器，看 Flow1。这个时候，我们会发现，SPI 是没有意义的，因为对于 A 路由器而言，它所需要的 SA 内容，只需两个字段，如下表所示：

目的 IP	IPSec Protocol
B	AH
***	ESP
***	***

表2 通过目的 IP 查找 IPSec Protocol

它只需要查找这样的一个表，就能确定该给出口的流使用哪种 IPSec 协议。

但是，我们再看路由器 B，同时看 Flow1 和 Flow2，就会发现，这两个流的目的 IP 都是“B”，没法区分。这时，就需要 SPI 这个参数了。由于 A 与 B 协商的 SPI 参数，与 B 与 C 协商的 IP 参数不同，所以 B 路由器能够区分这两个流。

所以，SPI 这个参数，是针对一个“入”流而设计的，而不是针对一个“出”流而设计的。

也正是从这个意义上讲，SPI 是单向的。对于 IPSec Peer 而言，需要两个 SA，如下图所示：

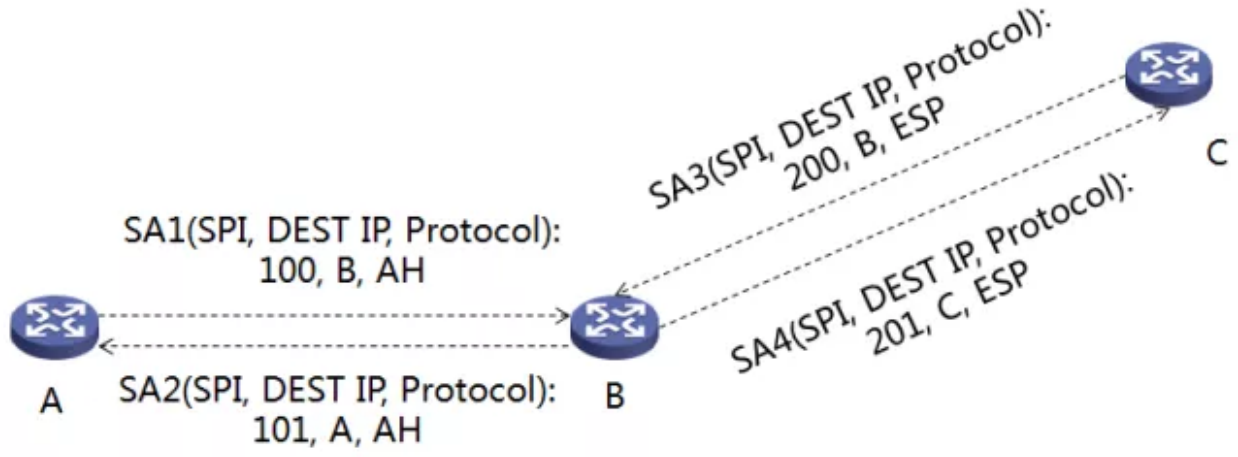

图 5 SA 示意图

3.2.3.2.2 SA 的数据结构

SA，存储在 SAD 中。作为一个数据库/表（虽然更多只是形式上的），当然得有索引。前文描述的三个字段“SPI，DST IP，Protocol”，就是 SAD 中的索引字段。SAD 除了存储索引字段以外，还存储 SA 的内容（SA 数据结构对应的实例化数据）。

SA 数据结构如下（全文摘自《RFC4301(中文)-IP 安全架构(废除了 RFC2401).pdf》）：

l Security Parameter Index(SPI)：前文已经介绍

l Sequence Number Counter：用于生成AH首部或ESP首部中Sequence Number字段的64位计数器。64位序列号是默认值，但是如果协商同意，也支持32位序列号。

l Sequence Counter Overflow：一个标记，指出是否序列号计数器溢出应当产生审计事件并阻止在SA上发送附加分组，或者是否允许序列号计数器翻转。这个事件的审计日志条目应当包括SPI值、当前日期/时间、Local Address、Remote Address以及来自相关SAD条目的选择器(们)。

l Anti-Replay Window：一个64位计数器，和用于决定是否入境AH或ESP分组是重播的位图(或等效物)。

l AH Authentication算法、密钥，等等。这仅是支持AH时要求的。

l ESP Encryption算法、密钥、模式、IV，等等。如果使用组合模式算法，这些字段不能使用。

l ESP完整性算法、密钥，等等。如果不选择完整性业务，这些字段不能使用。如果使用组合模式算法，这些字段不能使用。

l ESP组合模式算法、密钥(或多个密钥)，等等。当与ESP一起使用组合模式(加密和完整性)算法时，使用这个数据。如果不使用组合模式算法，这些字段不能使用。

l SA的生存期：时间间隔，此时间间隔过后，SA必须被新SA(和新SPI)代 替或终结，加上一个指示，说明这些行动中哪一个应当发生。

l IPsec协议模式：隧道或传输。指出将AH或ESP的哪一种模式应用于这个SA上的流量。

l 有状态分段检验标记。指出不管怎样将把有状态分段检验应用到这个SA。

l 旁路DF位(T/F) -- 应用于隧道模式SAs，那里内层首部和外层首部都是IPv4。

l DSCP值 – 可供这个SA上携带的分组使用的DSCP值集合。

l 路径MTU：任何监测到的路径MTU和迟滞变量。

l 隧道首部IP源地址和目的地地址 -- 两个地址必须或者是IPv4地址或者是IPv6地址。此解释暗示使用的IP首部类型。仅适用隧道IPsec协议模式。

3.2.3.2.3 SAD 的管理

前文说过，SAD 就是存储 SA 实例的地方。那么说道 SAD 的管理，它的行为也跟一个数据库是一样的，四个行为：增删改查。修改和查询，比较简单。我们这里主要讲述增加和删除。

增加 SA 有两种模式，一种是人工增加模式，一种是 IKE 协商模式，自动增加一个（具体的 IKE 协商，下文会讲述）。

人工增加模式，就是人工协商好相关参数（比如 IPSec Protocol，算法，密钥等等）以后，在 SAD 中增加一条 SA 记录。

人工增加很难应对复杂网络，因为每一个 IPSec Peer，都需要2条 SA 记录，这个不仅工作量大，还容易出错。

另外，人工增加SA没有生存周期限制，永不过期，除非手工删除，因此有安全隐患。

IKE 协商模式，自动增加，与人工增加相比，除了能大量减少工作量以外，还有生命周期管理。当如下情况之一发生时，SA 会被自动删除：

l 存活时间过期；

l 密钥已遭破解；

l 使用SA加密/解密或验证的字节数已超过策略设定的某一个阈值；

l 另一端要求删除这个SA。

正是这样的生命周期管理，IKE 协商模式具备非常高的安全性。

另外需要说明一下，SA 有两类生存期（存活时间）：

l 软生存期，生存期到期之前，它会预先通知 IKE 协商发起重新建立替代SA的行动（比如，双发正在通信时，生存期到期，这个时候必须提前重新建立一个 SA，否则通信会中断）

l 硬生存期，指生存期到期以后，当前的SA 被直接终止（删除）。

3.2.3.3 IKE（Internet Key Exchange）

SA 实例的增加有两种模式，一种是人工模式，一种是 IKE 协商模式。本小节就讲述 IKE。

首先必须要说明，IKE 自己是一套独立的协议，它并不是专门为 IPSec 而生的，恰恰相反，是 IPSec 采用了 IKE 协议来做 SA 的协商和建立。IPSec 自身能够对 IP 数据提供安全保障，但是谁为 IPSec SA 的协商提供安全保障呢？也正是基于此，IPSec 采用了 IKE 协议。

另外需要说明的是 IKE 是一个“混合”协议，它是Oakley（Sorry，不知道这个名字的出处）和SKEME（Secure Key Exchange Mechanism for Internet，Internet 安全密钥交换机制）协议的一种混合，并在由 ISAKMP（Internet Security Association Key Management Protocol，Internet安全联盟密钥管理协议）规定的一个框架内运作。在 RFC2409 里，是这么描述他们的关系的：

ISAKMP 中对验证和密钥交换提出了结构框架，但没有具体定义。ISAKM被设计用来独立的进行密钥交换，即被设计用于支持多种不同的密钥交换。

Oakley 中描述了一系列被称为“模式”的密钥交换，并详述了每一种提供的服务（如密钥的完全后继保密（perfect forward secrecy），身份保护，以及验证）。

SKEME 中描述了一种提供匿名，否认，和快速密钥更新的通用密钥交换技术。

本文档将描述使用部分Oakley，部分SKEME，并结合ISAKMP的一种协议，它使用ISAKMP来得到已验证的用于生成密钥和其它安全联盟（如AH，ESP）中用于IETE IPsec DOI（Domain of Interpretation，DOI）的材料。

以上描述的也有点复杂，我们需要这样理解：（1）它是一个协议，不是一个协议族（不象 TCP/IP 那样，由很多协议组成，是一个协议族。）；（2）它是一个“混合协

议”，它不是完全自己“创新发明”的一个协议，它是综合“Oakley， SEKME，ISAMAP”三个协议，“混合”而成。

仅仅一个名字的解释，都这么复杂，毫无疑问，IKE 协议本身也非常复杂。本文由于篇幅和主题的原因，就简单介绍一下。（有机会，笔者再详细补充）。

IKE（Internet Key Exchange），Inernet 密钥交换协议，实际上交换的并不仅仅是密钥，还包括加密算法（通信双方采用何种加密算法）。当然，这是必然的，没有加密算法，要密钥有什么用呢？而且，IKE 交换密钥，也不是傻乎乎的直接用明文告诉对方密钥，那样等于毫无秘密可言。

IKE，密钥交换（不仅仅是密钥），分为两个阶段。下文做一个简要描述

3.2.3.3.1 第一阶段交换

前文说了， IKE 为 IPSec SA 的协商提供了安全保障，那么 IKE 的第一阶段，就是构建了一个安全通道。是的，不能无限依赖，如果 IKE 还依赖于其他协议来构建安全通道，那么那个“其他”协议又有谁来保障安全呢——不能这么无限依赖下去，所以，IKE 自己需要能搞定一个安全通道，这就是第一阶段要做的事情。

第一阶段，如果详细描述，涉及到报文格式，消息交互，这里我们就不展开了。只须记住：

（一）交换模式分为主模式或者野蛮模式两种

主模式，通信双方有6次交互，野蛮模式双方只有3次交互。无论是哪种模式，第一阶段都是为了协商 IKE SA（Security Association，安全联盟）。注意，这个 SA 是 IKE SA，不是 IPSec SA（大家都用到了 SA 这个词）。

另外，我也不懂老外的命名方法，为啥叫野蛮模式，感觉他们这个名字取得够野蛮的，^_^。

（二）主模式的交互过程简述

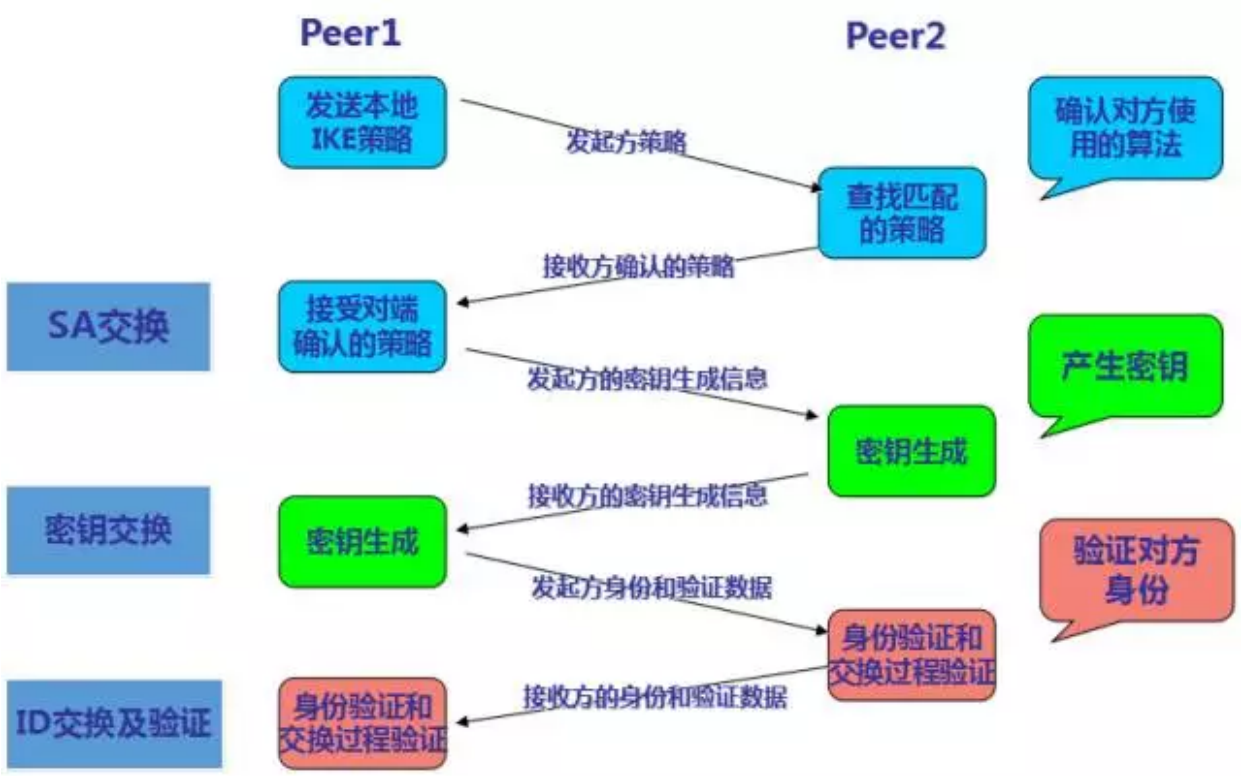

图 6 主模式交换过程

（三）野蛮模式交换过程简述

图 7 野蛮模式交换过程

3.2.3.3.2 第二阶段交换

由于第一阶段已经建立了安全通道，所以，IPSec 应用 IKE 的第二阶段，就比较直接，双方直接交换相关参数即可：

l 加密算法

l Hash算法

l 安全协议

l 封装模式

l 存活时间

篇幅原因，这里就不再进一步描述了。

3.2.3.3.3 IKE 与 IPSec 安全参数比较

最后补充一下两者的安全参数比较，如下图：

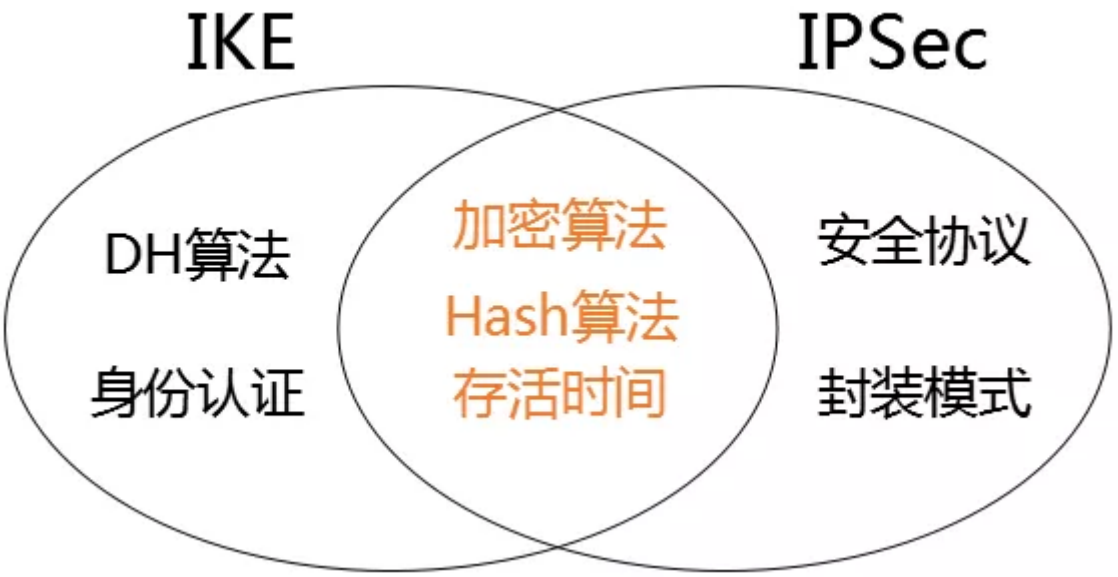

图 8 IKE 与 IPSec 安全参数比较