3.2.1.4 AH Header

为了讲述的完整性，我们现在补充讲述一下 AH Header 中的各个字段的含义。

我们重新贴一下 AH Header 的图：

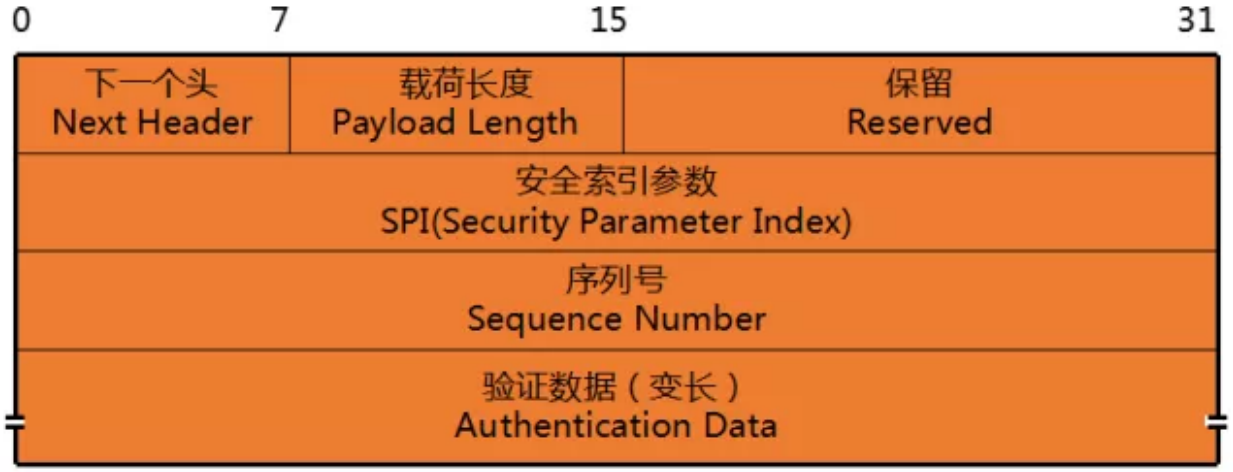

图 1 AH Header

AH Header 每个字段的含义如下：

字段	长度(bit)	含义
下一个头 Next Header	8	AH 后面下一个载荷的协议号，比如：ICMP（1），IGMP（2），TCP（6），UDP（17），ESP（50），IP（4），etc. 需要说明的是：在 IP Header 中，它的协议字段的值是51，表示下一个的协议号是 AH。
载荷长度 Payload Length	8	其值 = Length(AH Header) - 2，整个 AH Header 的长度（包括验证变长字节）减去2。（因为要去除“下一个头”的长度（1字节）和本字段“载荷长度”的长度（1字节））
保留 Reserved	16	暂时不使用，保留为将来适用。现在是全赋值为0
安全索引参数 SPI（Security Parameter Index）	32	后面还会涉及这个字段。现在先了解其取值范围： l 0，保留本地的特定实现使用 l [1，255]，保留为将来使用 l 因此，可用的SPI值为 [256，2^32 - 1]
序列号 Sequence Number	32	作为一个单调递增的计数器，为每个AH包赋予一个序号： l 当通信双方建立SA时，计数器初始化为0。 l SA是单向的，每发送一个包，外出SA的计数器增1；每接收一个包，SA的计数器增1。 前文说过，这个可以起到仿重放攻击的功能
验证数据 Authentication Data	变长	就是前文说的 HAMC 的值，HMAC = Hash(data, key)。此字段又称为 ICV（Integrity Check Value，完整性校验值）。 此字段必须为32位的整数倍，如果 ICV 不是32位的整数倍，必须进行填

表1 AH Header 各字段含义

3.2.2 ESP/Encapsulating Security Payload

在上一小节中，我们讲述了 AH（Authentication Header）。AH 可以保障数据的完整性，但是它并没有对数据进行加密。也就是说，AH 没法防止数据被偷窥！

图 2 AH 并不能防止数据偷窥

ESP/Encapsulating Security Payload，封装安全载荷（协议），其主要目标就是为给数据加密，防止被偷窥！

3.2.2.1 概述

我们仍然是以较简单的 ESP 传输模式为例来进行讲述。

首先我们也是先上传统 IP 报文格式，如下图：

图 3 传统 IP 报文格式

而 ESP 的报文格式，则有点特别，如下图所示：

图 4 ESP 报文格式

ESP 的报文，不像一般的报文，在前面加一个头部，它还在后面加了两个数据结构：ESP 尾部，ESP 验证（这是一个可选数据结构）。ESP Header，ESP 尾部，ESP 验证的数据结构如下图所示：

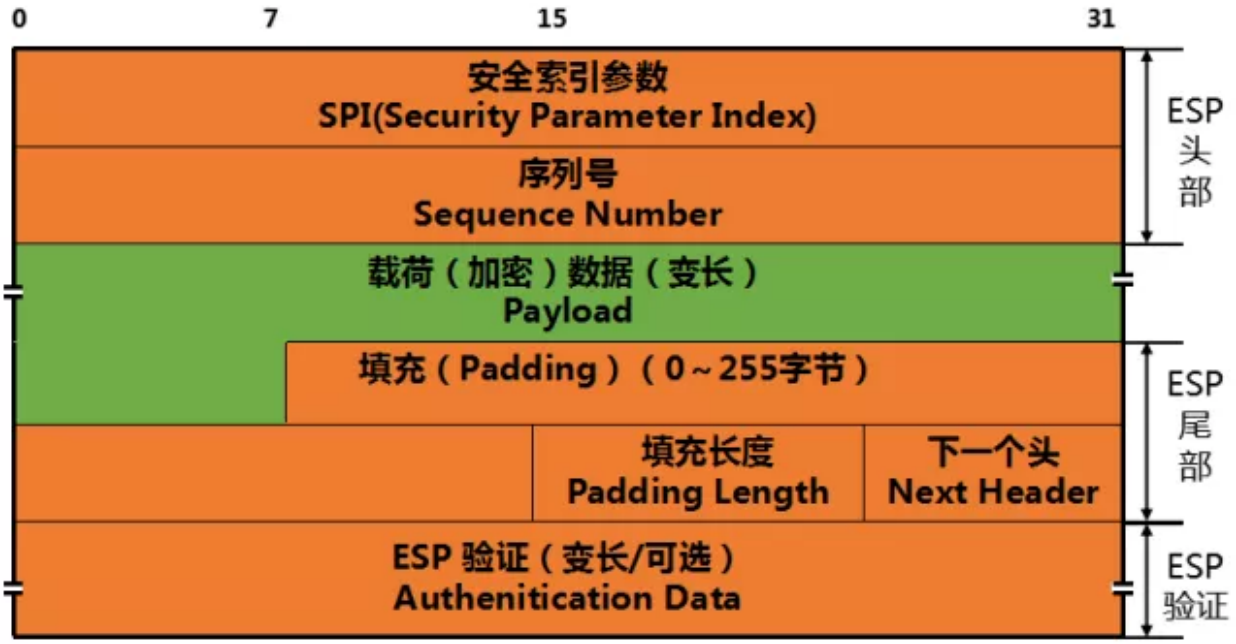

图 5 ESP 数据结构

从图中我们看到，ESP 的数据结构其实比较简单，其含义如下：

分类	字段	长度（bit）	含义
ESP Header	安全索引参数 SPI（Security Parameter Index）	32	后面还会涉及这个字段。现在先了解其取值范围： l 0，保留本地的特定实现使用 l [1，255]，保留为将来使用 l 因此，可用的SPI值为 [256，232- 1] 需要说明的是：在 IP Header 中，它的协议字段的值是 50，表示下一个载荷的协议号是 ESP。
	序列号 Sequence Number	32	作为一个单调递增的计数器，为每个AH包赋予一个序号： l 当通信双方建立SA时，计数器初始化为0。 l SA是单向的，每发送一个包，外出SA的计数器增1；每接收一个包，进入SA的计数器增1。 前文说过，这个可以起到仿重放攻击的功能
ESP 尾部	填充 Padding	0～255 字节	数据长度 + 填充字段长度 + 2个字节（填充长度 1 个字节， 下一个头 1 个字节），需要是8字节（64 bit）的整数倍。如果数据长度不满足这个要求，那么需要用0来填充。
	填充长度 Padding Length	8	参考上一行描述
	下一个头 Next Header	8	这个绝对不是说它下一个载荷的协议号，因为它后面已经没有载荷了（它后面只有“ESP 验证”这一个字段，不能算作载荷）。这个名字取的还是按照习惯用法，取为“下一个头 Next Header”，实际上表达的是它前面封装的载荷的协议号，比如：ICMP（1），IGMP（2），TCP（6），UDP（17），ESP（50），IP（4），etc.
ESP 验证	ESP 验证 （可选） Authentication Data	变长	就是前文说的 HAMC 的值，HMAC = Hash(data, key)。此字段又称为：ICV（Integrity Check Value，完整性校验值）。 此字段必须为32位的整数倍，如果 ICV 不是32位的整数倍，必须进行填充。 在 ESP 协议中，这个是可选字段（如何做到可选，下文会描述）

		会描述）。

表2 ESP 数据结构各字段含义

3.2.2.2 ESP 的基本原理

ESP/Encapsulating Security Payload，封装安全载荷（协议），主要目的是为了给数据进行加密，那么它给那一部分的数据进行加密呢？如下图所示：

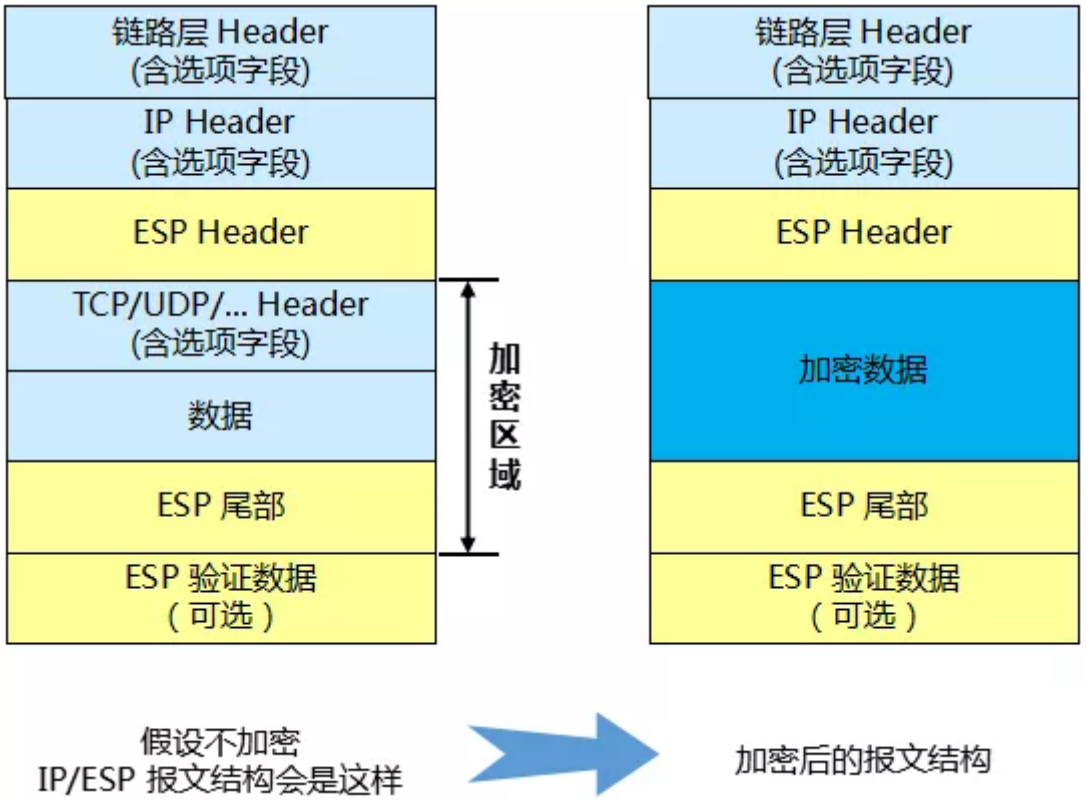

图 6 ESP 加密字段（传输模式）

这里面有一个绕人的地方，ESP 尾部，本来已经被加密了（上图左半部分），但是加密后的报文，又重新添加了 ESP 尾部（上图又半部分）。不过也不用纠结，记住就是了。

ESP 的加密过程，如下图所示：

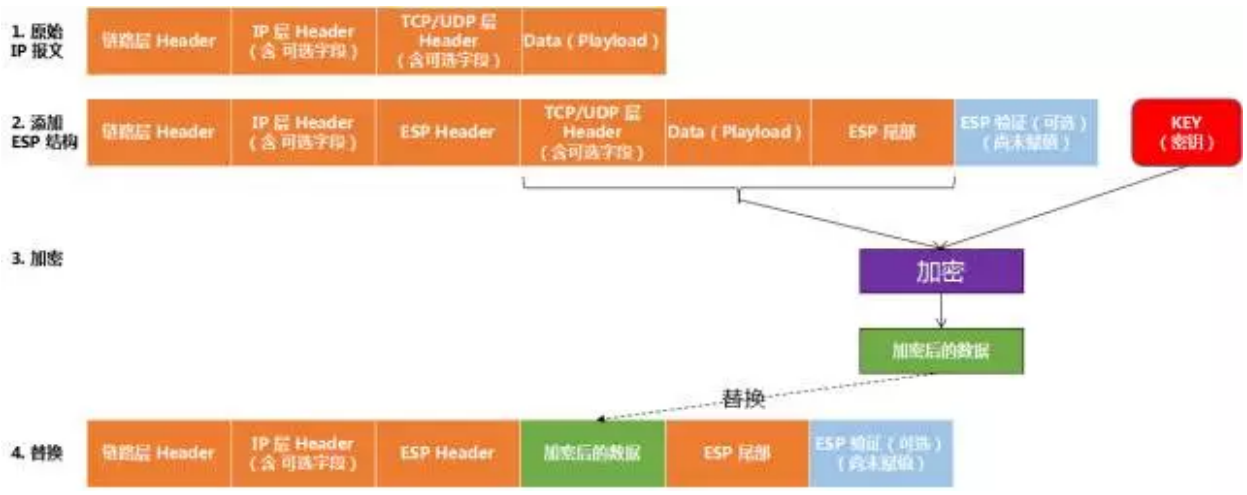

图 7 ESP 加密过程（传输模式）

加密当然需要加密算法和密钥，不过这个问题，我们放在后面的章节再概述一下，这里先忽略。

我们一直再说，ESP 还有一个 “ESP 验证” 字段，是一个可选字段。具体如何可选，我们仍然是放在后面的章节描述。这里先描述一下 ESP 是如何处理这个验证字段的。

ESP 验证字段，与上一节讲的 AH 中的验证数据是同一个原理，都是按照如下公式进行的一个 HASH：

HMAC = Hash（data， key）。

公式中的 data，即 ESP 的验证区域，如下图所示：

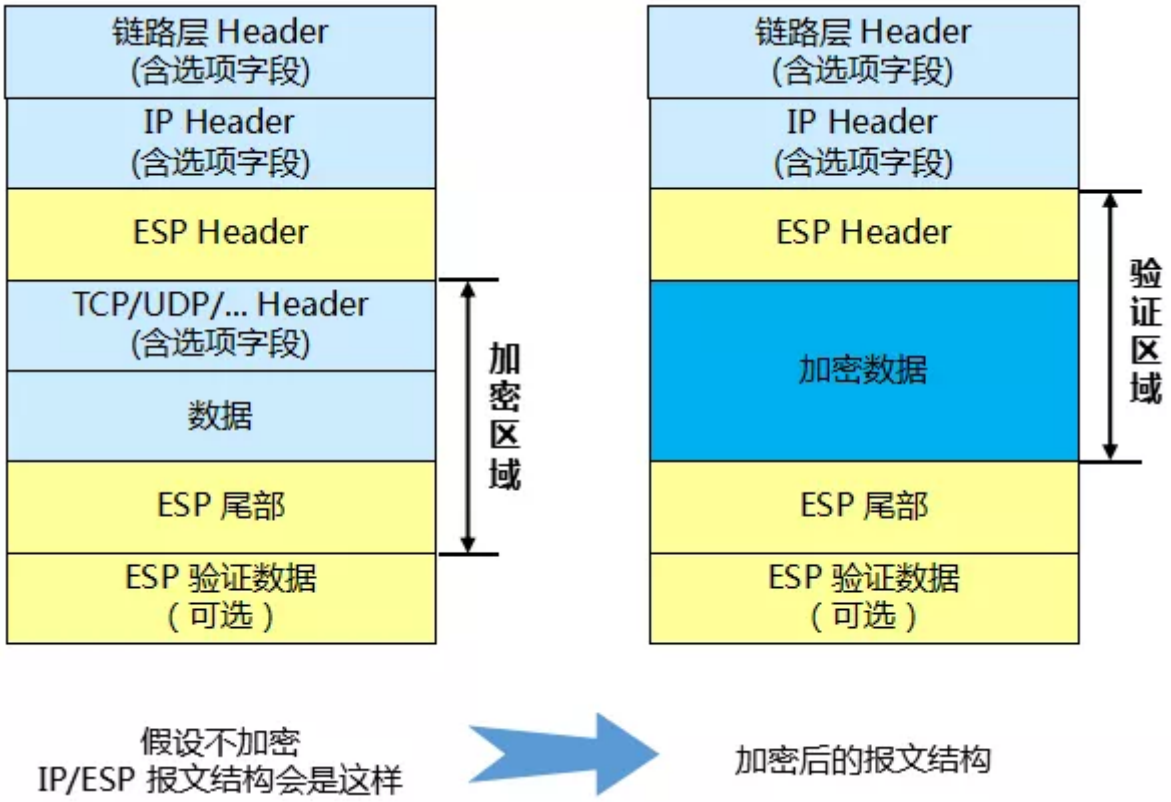

图 8 ESP 验证区域

ESP 验证区域的计算过程，如下图所示：

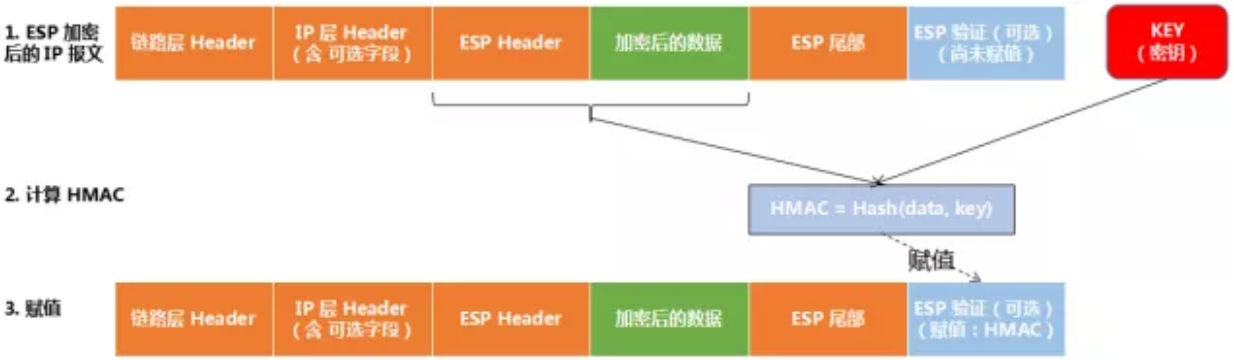

图 9 ESP 验证字段计算过程

ESP 验证字段，与上一节讲的 AH 中的验证数据的作用也是一样的：为了保障数据的完整性。但是它与 AH 不同的是，它 HMAC 所计算的数据中，没有包含 IP Header。也就

是说，如果有人篡改 IP Header 中的字段，那么 ESP 察觉不了。为什么 ESP 这么任性，不包含 IP Header 呢？对不起，我也不知道，^_^。

至此，我们讲述了 ESP 加密和验证的基本过程。

3.2.2.3 ESP 如何保障通信的安全

如果您仔细阅读了上一节“AH/Authentication Header”，那么 ESP 保障通信安全，就比较好理解：

（1）数据的私密性（防止被偷窥），因为 ESP 可以进行数据加密

（2）数据的完整性验证（可选），参见 ESP 数据验证

（3）数据源身份认证（可选），参见 ESP 数据验证

（4）防重放攻击，参见 ESP Header 中的 Sequence Number

在这一章节，我们仍然没有讲述 ESP 的双方（发送方，接收方）如何知道对方的算法（加密算法，Hash 算法），如何知道对方的密钥。这个我们放在后面的章节讲述。