1.1 IPSec 体系结构

IP安全(IP Security)体系结构，简称IPSec，是IETF IPSec工作组于1998年制定的一组基于密码学的安全的开放网络安全协议。IPSec工作在IP层，为IP层及其上层协议提供保护。

IPSec提供访问控制、无连接的完整性、数据来源验证、防重放保护、保密性、自动密钥管理等安全服务。IPSec独立于算法，并允许用户(或系统管理员)控制所提供的安全服务粒度。比如可以在两台安全网关之间创建一条承载所有流量的加密隧道，也可以在穿越这些安全网关的每对主机之间的每条TCP连接间建立独立的加密隧道。

IPSec在传输层之下，对应用程序和终端用户来说是透明的。当在路由器或防火墙上安装IPSec时，无需更改用户或服务器系统中的软件设置。即使在终端系统中执行IPSec，应用程序之类的上层软件也不会受到影响。

1.1.1 IPSec 的组成

IPSec是因特网工程任务组（IETF）定义的一种协议套件，由一系列协议组成，验证头（AH）、封装安全载荷（ESP）、Internet安全关联和密钥管理协议ISAKMP的Internet IP安全解释域（DOI）、ISAKMP、Internet密钥交换（IKE）、IP安全文档指南、OAKLEY密钥确定协议等，它们分别发布在RFC2401～RFC2412的相关文档中。图2.3显示了IPSec的体系结构、组件及各组件间的相互关系。

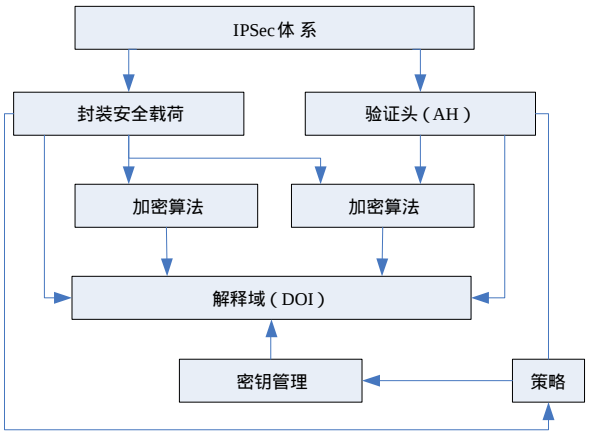

图　2.3 IPSec的体系结构

AH（认证头）和ESP（封装安全载荷）：是IPSec体系中的主体，其中定义了协议的载荷头格式以及它们所能提供的服务，另外还定义了数据报的处理规则，

正是这两个安全协议为数据报提供了网络层的安全服务。两个协议在处理数据报文时都需要根据确定的数据变换算法来对数据进行转换，以确保数据的安全，其中包括算法、密钥大小、算法程序以及算法专用的任何信息。

IKE（Internet 密钥交换）：IKE利用ISAKMP语言来定义密钥交换，是对安全服务进行协商的手段。IKE交换的最终结果是一个通过验证的密钥以及建立在通信双方同意基础上的安全服务——亦即所谓的“IPSec安全关联”。

SA（安全关联）：一套专门将安全服务/密钥和需要保护的通信数据联系起来的方案。它保证了IPSec数据报封装及提取的正确性，同时将远程通信实体和要求交换密钥的IPSec数据传输联系起来。即SA解决的是如何保护通信数据、保护什么样的通信数据以及由谁来实行保护的问题。

策略：策略是一个非常重要的但又尚未成为标准的组件，它决定两个实体之间是否能够通信；如果允许通信，又采用什么样的数据处理算法。如果策略定义不当，可能导致双方不能正常通信。与策略有关的问题分别是表示与实施。“表示”负责策略的定义、存储和获取，“实施”强调的则是策略在实际通信中的应用。

1.1.2 IPSec 的工作原理

设计IPSec是为了给IPv4和IPv6数据提供高质量的、可互操作的、基于密码学的安全性。IPSec通过使用两种通信安全协议来达到这些目标：认证头（AH）和封装安全载荷（ESP），以及像Internet密钥交换（IKE）协议这样的密钥管理过程和协议来达到这些目标。

IP AH协议提供数据源认证，无连接的完整性，以及一个可选的抗重放服务。ESP协议提供数据保密性，有限的数据流保密性，数据源认证，无连接的完整性以及抗重放服务。对于AH和ESP都有两种操作模式：传输模式和隧道模式。IKE协议用于协商AH和ESP所使用的密码算法，并将算法所需要的密钥放在合适的位置。

IPSec所使用的协议被设计成与算法无关的。算法的选择在安全策略数据库（SPD）中指定。IPSec允许系统或网络的用户和管理员控制安全服务提供的粒度。通过使用安全关联（SA），IPSec能够区分对不同数据流提供的安全服务。

IPSec本身是一个开放的体系，随着网络技术的进步和新的加密、验证算法的出现，通过不断加入新的安全服务和特性，IPSec就可以满足未来对于信息安全的需要。随着互联网络技术的不断进步，IPSec作为网络层安全协议，也是在

不断地改进和增加新的功能。其实在IPSec的框架设计时就考虑过系统扩展问题。例如在ESP和AH的文档中定义有协议、报头的格式以及它们提供的服务，还定义有数据报的处理规则，但是没有指定用来实现这些能力的具体数据处理算法。AH默认的、强制实施的加密MAC是HMAC－MD5和HMAC－SHA，在实施方案中其它的加密算法DES－CBC、CAST－CBC以及3DES－CBC等都可以作为加密器使用。

1.1.3 IPSec 的模式

IPSec协议（包括AH和ESP）既可以用来保护一个完整的IP载荷，也可以用来保护某个IP载荷的上层协议。这两个方面的保护分别由IPSec两种不同的“模式”来提供：传输模式和隧道模式。

传输模式：在传输模式中，IP头与上层协议头之间需插入一个特殊的IPSec头。传输模式保护的是IP包的有效载荷或者说保护的是上层协议（如TCP、UDP和ICMP），如图2.4所示。在通常情况下，传输模式只用于两台主机之间的安全通信。

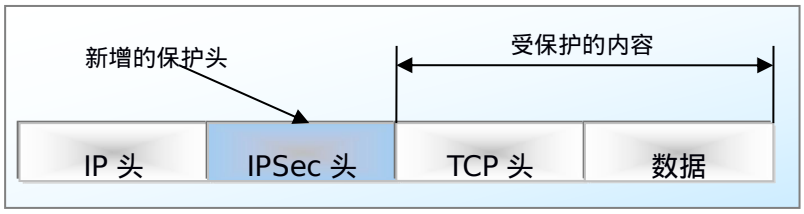

图2.4 IPSec传输模式的IP数据报格式

隧道模式：隧道模式为整个IP包提供保护。如图2.5所示，要保护的整个IP包都需封装到另一个IP数据报中，同时在外部与内部IP头之间插入一个IPSec头。所有原始的或内部包通过这个隧道从IP网的一端传递到另一端，沿途的路由器只检查最外面的IP报头，不检查内部原来的IP报头。由于增加了一个新的IP报头，因此，新IP报文的目的地址可能与原来的不一致。

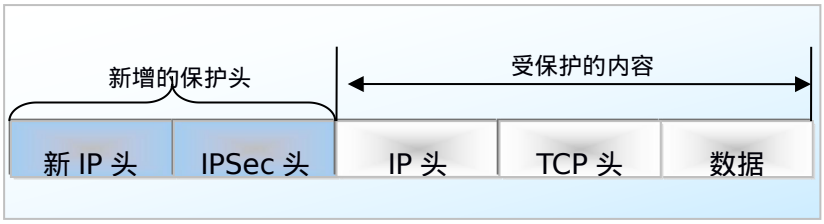

图2.5 IPSec隧道模式的IP数据报格式

在千兆网络加密工程实现上，我们需要的是根据IPSec协议，实现一个安全网关设备，为了保证数据的机密性，考虑采用隧道模式的ESP封装。由于IKE协议

是通过软件实现的，而本文着重讨论IPSec的硬件实现部分，故本文不再介绍IKE协议。

1.1.4 IPSec 的实现方式

IPSec可以在主机、路由器或防火墙（创建一个安全网关）中同时实施和部署。用户可以根据对安全服务的需要决定究竟在什么地方实施，IPSec的实现方式可分为集成方式、BITS方式、BITW方式三种。

集成方式：把IPSec集成到IP协议的原始实现中，这需要处理IP源代码，适用于在主机和安全网关中实现。

“堆栈中的块（BITS）”方式：把IPSec作为一个“锲子”插在原来的IP协议栈和链路层之间。这不需要处理IP源代码，适用于对原有系统的升级改造。这种方法通常用在主机方式中。

“线缆中的块（BITW）”方式：这是本文采用实现IPSec的方式，它将IPSec的实现在一个设备中进行，该设备直接接入路由器或主机设备。当用于支持一台主机时，与BITS实现非常相似，但在支持路由器或防火墙时，它必须起到一台安全网关的作用。

1.1.5 IPSec 协议的处理

IPSec处理分两类：外出处理和进入处理。

1.1.5.1 外出处理

在外出处理的过程中，数据包从传输层流进IP层。IP层首先取出IP头的有关参数，检索SPDB数据库，判断应为这个包提供那些安全服务。输入SPDB的是传送报头中的源地址和目的地址的“选择符”。SPDB输出的是根据“选择符”查询的策略结果，有可能出现以下几种情况：

丢弃这个包。此时包不会得以处理，只是简单地丢掉。

绕过安全服务。在这种情况下，这个IP包不作任何处理，按照一个普通的IP包发送出去。

应用安全服务。在这种情况下，需要继续进行下面的处理。

如果SPDB的策略输出中指明该数据包需要安全保护，那么接着就是查询SADB来验证与该连接相关联的SA是否已经建立，查询的结果可能是下面的两种情况之一：如果相应的SA已存在，对SADB的查询就会返回指向该SA的指针；如果查询不到相应的SA,说明该数据包所属的安全通信连接尚未建立,就会调用IKE进行协商，将所需要的SA建立起来。如果所需要的SA已经存在，那么SPDB结构中包含指向SA或SA集束的一个指针（具体由策略决定）。如果SPDB的查询输出规定必须将IPSec应用于数据包，那么在SA成功创建完成之前，数据包是不被允许传送出去的。

对于从SADB中查询得到的SA还必须进行处理，处理过程如下：

1. 如果SA的软生存期已满，就调用IKE建立一个新的SA。
2. 如果SA的硬生存期已满，就将这个SA删除。
3. 如果序列号溢出，就调用IKE来协商一个新的SA。

SA处理完成后，IPSec的下一步处理是添加适当的AH或ESP报头，开始对数据包进行处理。其中涉及到对负载数据的加密、计算校验等在下面的内容中会给予详细的介绍。SA中包含所有必要的信息，并已排好顺序，使IPSec报头能够按正确的顺序加以构建。在完成IPSec的报头构建后，将生成的数据报传送给原始IP层进行处理，然后进行数据报的发送。

1.1.5.2 进入处理

进入处理中，在收到IP包后，假如包内根本没有包含IPSec报头，那么IPSec就会查阅SPDB，并根据为之提供的安全服务判断该如何对这个包进行处理。因为如果特定通信要求IPSec安全保护，任何不能与IPSec保护的那个通信的SPDB定义相匹配的进入包就应该被丢弃。它会用“选择符”字段来检索SPDB数据库。策略的输出可能是以下三种情况：丢弃、绕过或应用。如果策略的输出是丢弃，那么数据包就会被放弃；如果是应用，但相应的SA没有建立，包同样会被丢弃；否则就将包传递给下一层作进一步的处理。

如果IP包中包含了IPSec报头，就会由IPSec层对这个包进行处理。IPSec从数据包中提取出SPI、源地址和目的地址组织成<SPI,目的地址，协议>三元组对SADB数据库进行检索（另外还可以加上源地址，具体由实施方案决定）。协议值要么是AH，要么是ESP。根据这个协议值，这个包的处理要么由AH协议来处理，要么由ESP来处理。在协议处理前，先对重放攻击和SA的生存期进行检查，把重放的报文或SA生存期已到的包简单丢弃而不作任何处理。协议载荷处理完

成之后，需要查询SPDB对载荷进行校验，“选择符”用来作为获取策略的依据。验证过程包括：检查SA中的源和目的地址是否与策略相对应，以及SA保护的传输层协议是否和要求的相符合。

IPSec完成了对策略的校验后，会将IPSec报头剥离下来，并将包传递到下一层。下一层要么是一个传输层，要么是网络层。假如说数据包是IP【ESP【TCP】】，下一层就是传输层；假如这个包是IP【AH【ESP【TCP】】】，下一层仍然是IPSec层。

1.1.6 认证头（AH）协议

1.1.6.1 AH 的目标

IP协议中，用来提供IP数据包完整性的认证机制是非常简单的。IP头通过头部的校验和域来保证IP数据包的完整性。而校验和只是对IP头的每16位计算累加和的反码。这样并没有提供多少安全性，因为IP头很容易修改，可以对修改过的IP头重新计算校验和并用它代替以前的校验和。这样接受端的主机就无法知道数据包已经被修改。

设计认证头(AH)协议的目的是用于增加IP数据包的安全性。AH协议提供无连接的完整性(connectionless integrity)、数据源认证(data originauthentication)和反重播(anti-replay)攻击服务。然而，AH不提供任何保密性服务，也就是说它不加密所保护的数据包。AH的作用是为IP数据流提供高强度的密码认证，以确保被修改过的数据包可以被检查出来。AH使用消息认证码(MAC)对IP进行认证。MAC不同于杂凑函数，因为它需要密钥来产生消息摘要，而杂凑函数不需要密钥。常用的MAC是 HMAC，它与任何迭代密码杂凑函数(如MD5, SHA-l, Tiger等)结合使用，而不用对杂凑函数进行修改。由于生成IP数据包的消息摘要需要密钥，所以IPSec的通信双方需要共享一个同样的认证密钥。这个密钥就是由双方的SA信息来提供的。

1.1.6.2 AH 协议包格式

AH只用于保证收到的数据包在传输过程中不被修改，保证由要求发送它的当事人将它发送出去，以及保证它是一个新的非重播的数据包。AH用于传送模式时，保护的是端到端的通信。通信的终点必须是IPSec终点，所以在我们所研

究的VPN的隧道方式中不预考虑。AH协议隧道模式的包格式如图2.6所示：

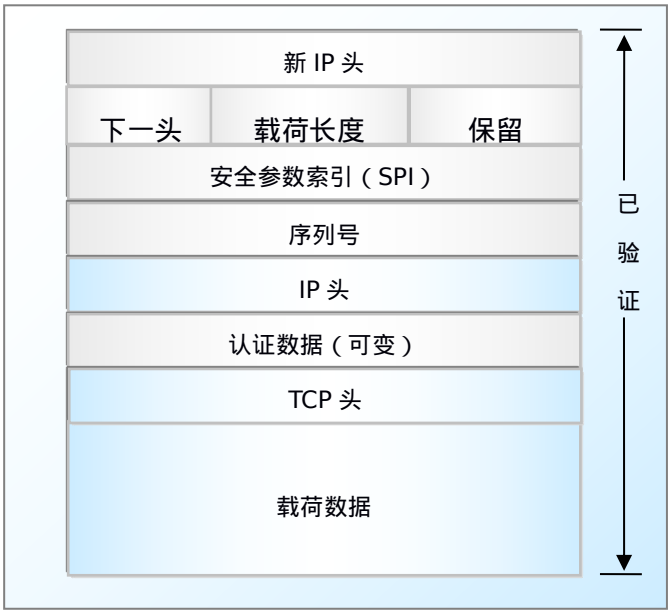

图2.6　隧道模式的AH

下一个头(8bit):指示下一个负载的协议类型。

载荷长度(8bit): AH的负载长度。

保留(8bit)：供将来使用。

安全参数索引SPI (32bit):它是一个32位长的整数。它与源地址或目的地址以及工PSEC协议(AH或ESP)来共同唯一标识一个数据包所属的数据流的安全联合(SA)。SPI的值1 ~ 255被IANA留作将来使用；0被保留，用于本地和具体实现。所以目前有效的SPI值从256 ~ 2^{32}-1。

序列号(32bit)：这里包含了一个作为单调增加计数器的32位无符号整数，用于防止对数据包的重演。所谓重演指的是数据包被攻击者截取并重新发送。如果接收端启动了反重演攻击功能，它将使用滑动接收窗口检测重演数据包。具体的滑动窗口因不同的工PSEC实现而不同，一般具有一下功能。窗口长度最小32比特，窗口的右边界代表一特定SA所接收倒的验证有效的最大序列号，序列号小于窗口左边界的数据包将被丢弃。将序列号值位于串口之内的数据包与位于窗口内的接收到的数据包清单进行比照，如果接收到的数据包的序列号位于窗口内并且是新的，或者序列号大于窗口右边界且有效，那么接收主机继续处理认证数据的计算。

认证数据：这是一个变长域（必须是32bit字的整数倍）。它包含数据包的认证数据，该认证数据被称为这个数据包的完整性校验值(ICV)。用于计算ICV的可用的算法因IPSEC的实现不同而不同；然而，为了保证互操作性，AH强制所有的IPSec必须包含两个MAC: HMAC-MD5和

HMAC-SHA-I。

1.1.6.3 AH 的处理

无论是ESP或者AH协议，数据报的处理都是向数据报中添加IPSec报头或者剥离IPSec报头。为了能正确地完成数据报的封装，需要从SADB中获得各个字段的数据；为了能够正确完成数据报的解封装，需要根据SPDB的查询结果对各个字段进行校验。下面就简单介绍AH协议的如何完成IPSec报头的封装和解封装。

对于外出的数据包，AH协议处理的目标是向数据包合适的位置增加AH报头。具体的处理步骤如下：

1. 外出数据包与一个SPDB条目匹配时，查看SADB是否有合适的SA。如果有，就将AH应用到与这个与之相符的数据包，该数据包在SPDB条目指定的那个模式中。如果没有，可用IKE动态地建立一个，并把序列号计数器初始化为0。在利用这个SA构建一个AH头之前，计算器就开始递增，这样保证了每个AH报头中的序列号都是一个独一无二的、非零的和单向递增的数。
2. 向AH的其余字段填满恰当的值。SPI字段分配的值是取自SA的SPI；下一个头字段分配的是跟在AH之后的数据类型值；而载荷长度分配的则是“32位字减二”；“验证数据”字段设成0。需要注意的是：AH协议将安全保护扩展到外部IP报头的原有的字段，因此将“完整性检查值(ICV)”之前的不定字段清零是必要的。这和ESP协议的处理是不一样的。
3. 根据验证算法的要求，或出于排列方面的原因，需要进行适当的填充。对有些MAC算法来说，比如DES－CBC MAC要求应用MAC的数据必须是算法的块尺寸大小的倍数。在这种情况下就必须进行填充以便正确地使用MAC（注意两种强制算法均无此要求）。填充的数据报必须为零，并且填充数据的长度不包括在载荷长度中。对IPv4来说AH报头必须是32比特的倍数，IPv6则是64比特的倍数。如果MAC算法的输出不符合这项要求就必须添加AH报头。对填充项的值没有什么别的要求，但必须把它包括在ICV的计算中，而载荷长度中必须反映出填充项大小。
4. 计算ICV。从外出SA中取出验证密钥，连同整个IP包（包括AH报头）传到特定的算法（也就是SA中的“身份验证程序”）计算ICV。由于不定字段已清零，它们不会被包括在ICV的计算中。将计算得到的ICV复制到

AH的“验证数据”字段中，IP报头中的不定字段就可根据IP处理的不同得以填充。

5. 输出经过处理的报文。AH处理结束后就形成了AH保护下的IP数据报，根据数据报的大小，在传输到网络上前可将它分段处理，或在两个IPSec同级之间的传送过程中，由路由器分段。

对于进入的数据报，AH协议处理的目标就是从数据报中将AH报头剥离下来，还原出封装在IPSec内的高层数据包：

1. 重组分段。如果一个受AH安全保护的包在接收时被证实是分段数据，那么在AH输入处理之前需要对这些分段数据进行重新组合。因为如果分段的数据报没有重组为原来的完整数据，ICV检查就会失败。只有完整的AH保护的IP包可传送到AH输入处理。
2. 查询SADB，找出保护这个包的SA。用基于IP报头的SPI、目的IP地址和安全协议（AH）组成的三元组来对SA进行查询。如果没有找到合适的SA，这个包会被丢弃。
3. 进行序列号检查。如果检查失败，这个包就会被丢弃。
4. 检查ICV。首先把AH报头中的“验证数据”字段中的ICV值取出来，然后将这个字段清零，同时将IP中所有不定字段也清零。根据验证算法的要求以及载荷长度的要求可能还要进行零数据的填充，使验证数据的长度符合算法的要求。随后对整个数据包应用验证算法，并将获得的摘要同保存下来的ICV值进行比较。如相符，IP包就通过了身份验证；如不符，该数据报丢弃。
5. 接收窗口的序列号可以递增，结束AH处理过程。验证通过的整个数据报传递给下一步的IP来处理。

1.1.7 封装安全载荷（ESP）协议

1.1.7.1 ESP 的目标

ESP为IP报文以无连接的方式（以包为单位）提供完整性校验、认证和加密服务，同时还可能提供防重放攻击保护。在建立SA时可选择所期望得到的安全服务，建议遵守以下约定：

完整性校验和身份认证建议同时使用。

使用防重放攻击时建议同时使用完整性校验和身份认证。

防重放攻击保护的使用建议由接收端选择。

加密独立于其他的安全服务，但建议使用加密时同时使用完整性校验和身份认证。

1.1.7.2 ESP 协议包格式

ESP协议通道模式的包格式如图2.7所示。

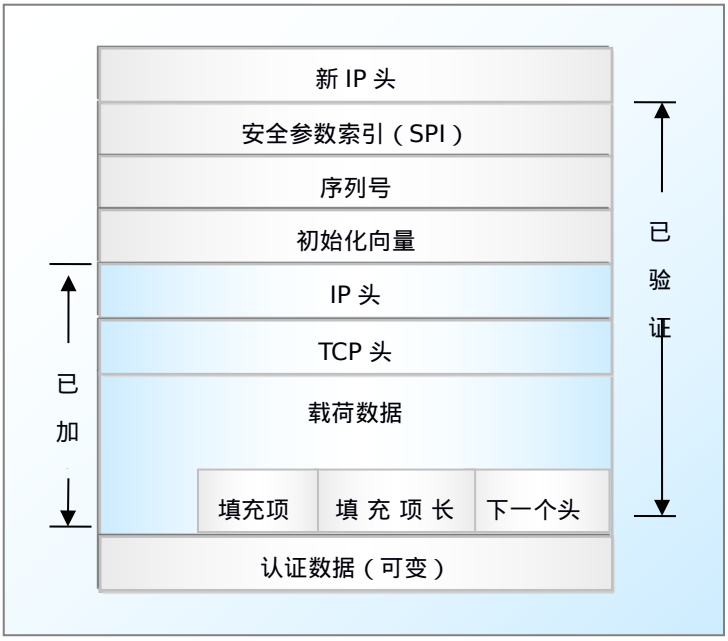

图2.7ESP通道模式的数据报格式

安全参数索引（SPI）：32bit字段，与目的IP地址和安全协议结合在一起，用来标识处理数据报的特定安全关联SA。SPI一般在IKE交换过程中由目标主机选定。SPI值为0时，表示预留给本地的特定实现使用。

序列号(Sequence Number)：32比特字段，是一个单项递增的计数器。无论接收者是否选择使用特定SA的抗重放服务，都必须使用序列号，并由接收者选择是否需要处理序列号字段。当建立一个SA时，发送者和接收者的计数器初始化为0，并在进行IPSec输出处理前，令这个值递增。新的SA必须在序列号回归位零之前创建。由于序列号长度为32位，所有在传送232个包之前，必须重置发送者和接收者的计数器。

载荷数据（Payload Data）：可变长字段，是ESP保护的实际数据报。

在这个域中，包含“下一个头”字段，也可包含一个加密算法可能需要使用到的初始化向量 (Initialization Vector,IV) ,虽然载荷数据是加密的，但IV是没有加密的。

填充项（Padding）：填充项的使用是为了保证ESP的边界适合于加密算法的需要。因为有些加密算法要求输入数据是以一定数量的字节为单位的块的整数倍，即使SA没有机密性要求，仍然需要通过加入Pad数据把ESP报头的“填充长度”和“下一个头”这两个字段靠右排列。

填充项长度（Pad Length）：指出上面的填充项填充了多少字节的数据。通过填充长度，接收端可以恢复出载荷数据的真实长度。

下一个头（NextHeader）：8bit字段，表明包含在载荷数据字段的类型。字段的大小从IP协议数据中选择。在通道模式下使用ESP，这个值是4，表示IP-in-IP。

认证数据（Authentication Data）：字段的长度由选择的认证功能指定。它包含数据完整性检验结果（Integrity Check Value，ICV）。验证数据计算的是ESP包中除验证数据域以外的所有项。如果对ESP数据报进行处理的SA中没有指定身份验证器，就没有这一项。

1.1.7.3 ESP 处理

无论采用哪种模式，对ESP来说，密文是得到验证的，验证的明文则是未加密的。即：对于外出的包，首先进行的是加密处理；而对于进入的包来说，验证是首先进行的。使用这种处理顺序能够简化检测过程，抵抗重放攻击以及减少拒绝服务攻击的影响。

外出包处理:

1. 安全关联查询：得到处理包的策略和SA，其中包括SPI，密钥等。
2. 包加密：在增加了必要的填充项后，使用密钥、加密算法、由SA指定的算法模式以及密码同步对载荷数据、填充项、填充长度、下一个头进行加密。如果选择认证，则在认证前要进行加密，加密不包含认证数据字段。由于认证数据不被加密保护，因此要使用认证算法计算ICV。
3. 序列号产生：当创建一个SA时，发送者的计数器初始化为0。利用这个SA,发送的第一个包的序列号设置为1，计数器的值从此开始递增，并将新值插入到序列号字段，这样就可以保证序列号的唯一性、非零性和单

向递增性。

4. 完整性校验值（ICV）计算：计算ICV的参数包含SPI、序列号、载荷数据（包括初始化向量，原IP头、TCP头和原载荷数据）、填充项、填充长度和下个一头字段的密文数据。
5. 分段：在进行ESP处理后IPSec要进行IP分段。传输模式ESP只适用于整个IP数据报，由路由器对IP包进行分段，在ESP处理之前由接收端进行分段重组。在隧道模式中，应用ESP协议处理IP包，载荷是分段的IP包。
6. 重新计算位于ESP前面的IP头校验和，按IPSec格式重新封装数据报。

进入包处理：

1. 重组：在ESP处理之前执行分段包的重组。
2. SA查询：接收到包含ESP头的包时，接收者根据目的地址、安全协议和SPI查询单向的SA。SA指示出是否检查序列号字段，认证数据字段是否出现，说明解密和ICV计算使用的算法和密钥等。
3. 序列号验证：验证每个接收的包是否包含不重复的序列号。通过使用滑动接收窗口可以拒绝重复序列号。序列号未重复，接收者就进行ICV验证。如果ICV验证失败，接收者丢弃无效的IP数据报。如果ICV验证成功，刷新接收窗口。
4. ICV验证：接收者使用认证算法，根据包的字段计算ICV,验证包的认证数据字段内的ICV是否相同。
5. 包解密：接收者使用密钥、加密算法、算法模式和密码同步数据，对ESP载荷数据、填充项、填充长度和下一个头进行解密。在解密数据传送到上一层之前，接收者应检查填充项字段。原始数据报的重组取决于ESP的工作模式。

如果篡改了SPI、目的地址或IPSec协议类型字段，那么所选择的SA就是不正确的。如果将包映射到另一个这样的SA,造成的错误和坏包将很难区分。通过使用认证算法，可以检测出IPSec头是否已被篡改。如果篡改了IP目的地址或IPSec协议类型字段，就会发生SA不匹配的事情。