第9章 IPv6身份验证和安全性

很多年来，人们一直在争论IP层是否需要身份验证和安全性及相关的用法问题。本章将讨论如何在IPv6中通过身份验证头(AH)和封装安全性净荷(ESP)头来实现身份验证和安全性，包括安全密码传输、加密和数据包的数字签名。但在探讨 IPv6的安全性头之前，本章将首先介绍IP安全性体系结构以及在IPv6中该体系结构可能实现的部分。该体系结构在 RFC 1825(IP的安全性体系结构)中首次进行了描述。

9.1 为IP增加安全性

IPv4的目的只是作为简单的网络互通协议，因而其中没有包含安全特性。如果 IPv4仅作为研究工具，或者在包括研究、军事、教育和政府网络的相对严格的辖区中作为产品型网络协议而使用，缺乏安全性并不是一个严重的缺陷。但是，随着 IP网络在商用和消费网络中的重要性与日俱增，攻击所导致的潜在危害将具有空前的破坏性。

本节主要内容包括：

- 人们已经为IP定义的安全性目标。
- 这些目标如何满足。
- 这些目标和相关论题如何在IP中定义。

下一节将介绍IP的安全性体系结构(又称为IPsec)本身的细节以及为完成上述目标而安装的一些工具。

应注意，RFC 1825以及后续文档中所定义的IPsec提供的是IP的安全性体系结构，而不是Internet的安全性体系结构。两者的区别很重要： IPsec定义了在IP层使用的安全性服务，对IPv4和IPv6都可用。如果在适当的IPv4选项格式中实现AH和ESP头，IPv4也可以使用这种安全性功能，只是在IPv6中更容易实现。

9.1.1 安全性目标

对于安全性，可以定义如下三个公认的目标：

- 身份验证：能够可靠地确定接收到的数据与发送的数据一致，并且确保发送该数据的实体与其所宣称的身份一致。
- 完整性：能够可靠地确定数据在从源到目的地传送的过程中没有被修改。
- 机密性：确保数据只能为预期的接收者使用或读出，而不能为其他任何实体使用或读出。

完整性和身份验证经常密切相关，而机密性有时使用公共密钥加密来实现，这样也有助于对源端进行身份验证。

AH和ESP头有助于在IP上实现上述目标。很简单， AH为源节点提供了在包上进行数字签名的机制。AH之后的数据都是纯文本格式，可能被攻击者截取。但是，在目的节点接收之后，可以使用AH中包含的数据来进行身份验证。

另一方面，可以使用ESP头对数据内容进行加密。ESP头之后的所有数据都进行了加密，ESP头为接收者提供了足够的数据以对包的其余部分进行解密。

Internet安全性(实际上任何一种安全性)的问题在于很难创建安全性，尤其是在开放的网络中，包可能经过任意数量的未知网络，任一个网络中都可能有包嗅探器在工作，而任何网络都无法察觉。在这样的开放环境中，即使使用了加密和数字签名，安全性也将受到严重的威胁。对IP业务流的攻击也包括诸如侦听之类，致使从一个实体发往另一个实体的数据被未经授权的第三个实体所窃取。此外，IP安全性还应该解决下列安全性威胁：

- 否认服务攻击：即实体使用网络传送数据，致使某个授权用户无法访问网络资源。例如，攻击者可能使某主机淹没于大量请求中，从而致使系统崩溃；或者重复传送很长的 e-mail报文，企图以恶意业务流塞满用户或站点带宽。
- 愚弄攻击：即实体传送虚报来源的包。例如，有一种愚弄攻击是由攻击者发送 e-mail报文，报头的“ From: ”指明该报文的发信人是美国总统。那些在在包头携带错误源地址的攻击则更加阴险。

密钥处理问题则更加复杂。为使身份验证和加密更可靠， IP安全性体系结构要求使用密钥。如何安全地管理和分配密钥，同时又能正确地将密钥与实体结合以避免中间者的攻击，这是Internet业界所面临的最棘手的问题之一。这种中间者的攻击是指，攻击者 (假设为C)将自己置于两个通信实体(假设为A和B)之间，拦截A和B之间传送的所有数据，冒充 A把数据重新发送给B，也冒充B把数据重新发送给A。如果C能够以类似B的公共密钥进行身份验证，从而让A确认它就是B，同样也让B误以为它就是A，那么A和B就会误认为他们之间的传送是安全的。

IPsec本身不能使Internet更加安全。本章只提出与Internet安全性相关的几个最迫切的问题。对Internet安全性的细节感兴趣的读者，请参考本书作者的另一本书《 Personal Encryption Clearly Explained》(AP Professional,1998)，书中讨论了加密、数字签名和 Internet安全性问题。

9.1.2 RFC 1825及建议的更新

RFC 1825于1995年8月发布，共有22页；其第5版修改草案完成于1998年5月，已经达到66页。安全性的正确实现要求认真考虑细节问题，这是对原 RFC进行扩充的主要原因。更新后的文档在最终发布时，在关于如何实现所有的 IP协议(包括ICMP和组播)方面将提供更多的细节，同时将更详细讨论密钥管理相关问题和安全性关联问题。

9.2 IPsec

IPsec的目标是提供既可用于IPv4也可用于IPv6的安全性机制，该服务由IP层提供。一个系统可以使用IPsec来要求与其他系统的交互以安全的方式进行 —— 通过使用特定的安全性算法和协议。IPsec提供了必要的工具，用于一个系统与其他系统之间对彼此可接受的安全性进行协商。这意味着，一个系统可能有多个可接受的加密算法，这些算法允许该系统使用它所倾向的算法和其他系统协商，但如果其他系统不支持它的第一选择，则它也可以接受某些替代算法。

IPsec中可能考虑如下安全性服务：

- 访问控制。如果没有正确的密码就不能访问一个服务或系统。可以调用安全性协议来控制密钥的安全交换，用户身份验证可以用于访问控制。
- 无连接的完整性。使用IPsec，有可能在不参照其他包的情况下，对任一单独的IP包进行完整性校验。此时每个包都是独立的，可以通过自身来确认。此功能可以通过使用安全散列技术来完成，它与使用检查数字类似，但可靠性更高，并且更不容易被未授权实体所篡改。
- 数据源身份验证。IPsec提供的又一项安全性服务是对IP包内包含的数据的来源进行标识。此功能通过使用数字签名算法来完成。
- 对包重放攻击的防御。作为无连接协议， IP很容易受到重放攻击的威胁。重放攻击是指攻击者发送一个目的主机已接收过的包，通过占用接收系统的资源，这种攻击使系统的可用性受到损害。为对付这种花招， IPsec提供了包计数器机制。
- 加密。数据机密性是指只允许身份验证正确者访问数据，对其他任何人一律不准。它是通过使用加密来提供的。
- 有限的业务流机密性。有时候只使用加密数据不足以保护系统。只要知道一次加密交换的末端点、交互的频度或有关数据传送的其他信息，坚决的攻击者就有足够的信息来使系统混乱或毁灭系统。通过使用IP隧道方法，尤其是与安全性网关共同使用， IPsec提供了有限的业务流机密性。

通过正确使用ESP头和AH，上述所有功能都有可能得以实现。目前，人们使用了很多密码功能，在下一节中将对此予以简要描述。后续节将扼要描述密钥管理基础设施。

9.2.1 加密和身份验证算法

由于对安全性的攻击方法多种多样，设计者很难预计到所有的攻击方法，因此设计安全性算法和协议非常困难。普遍为人接受的关于安全性方法的观点是，一个好的加密算法或身份验证算法即使被攻击者了解，该算法也是安全的。这一点对于 Internet安全性尤其重要。在Internet中，使用嗅探器的攻击者通过侦听系统与其连接协商，经常能够确切了解系统使用的是哪一种算法。

与Internet安全性相关的重要的密码功能大致有5类，包括对称加密、公共密钥加密、密钥交换、安全散列和数字签名。

1. 对称加密

大多数人都熟知对称加密这一加密方法。在这种方法中，每一方都使用相同的密钥来加密或解密。只要掌握了密钥，就可以破解使用此法加密的所有数据。这种方法有时也称作秘密密钥加密。通常对称加密效率很高，它是网络传送大量数据中最常用的一类加密方法。

常用的对称加密算法包括：

- 数据加密标准(DES)。DES首先由IBM公司在70年代提出，已成为国际标准。它有 56位密钥。三重DES算法对DES略作变化，它使用DES算法三次加密数据，从而改进了安全性。
- RC2、RC4和RC5。这些密码算法提供了可变长度密钥加密方法，由一家安全性动态公司， RSA数据安全公司授权使用。目前网景公司的 Navigator浏览器及其他很多Internet客户端和服务器端产品使用了这些密码。

- 其他算法。包括在加拿大开发的用于Nortel公司Entrust产品的CAST、国际数据加密算法(IDEA)、传闻由前苏联安全局开发的GOST算法、由Bruce Schneier开发并在公共域发表的Blowfish算法及由美国国家安全局开发并用于 Clipper芯片的契约密钥系统的 Skipjack算法。

安全加密方法要求使用足够长的密钥。短密钥很容易为穷举攻击所破解。在穷举攻击中，攻击者使用计算机来对所有可能的密钥组合进行测试，很容易找到密钥。例如，长度为 40位的密钥就不够安全，因为使用相对而言并不算昂贵的计算机来进行穷举攻击，在很短的时间内就可以破获密钥。同样，单 DES算法已经被破解。一般而言，对于穷举攻击，在可预测的将来，128位还可能是安全的。

对于其他类型的攻击，对称加密算法也比较脆弱。大多数使用对称加密算法的应用往往使用会话密钥，即一个密钥只用于一个会话的数据传送，或在一次会话中使用几个密钥。这样，如果会话密钥丢失，则只有在此会话中传送的数据受损，不会影响到较长时期内交换的大量数据。

2. 公共密钥加密

公共密钥加密算法使用一对密钥。公共密钥与秘密密钥相关联，公共密钥是公开的。以公共密钥加密的数据只能以秘密密钥来解密，同样可以用公共密钥来解密以秘密密钥加密的数据。这样只要实体的秘密密钥不泄露，其他实体就可以确信以公共密钥加密的数据只能由相应秘密密钥的持有者来解密。尽管公共密钥加密算法的效率不高，但它和数字签名 (参见后续讨论)均是最常用的对网络传送的会话密钥进行加密的算法。

最常用的一类公共密钥加密算法是 RSA算法，该算法由 Ron Rivest、Adi Shamir和Len Adleman开发，由RSA数据安全公司授权使用。 RSA定义了用于选择和生成公共 /秘密密钥对的机制，以及目前用于加密的数学函数。

3. 密钥交换

开放信道这种通信媒体上传送的数据可能被第三者窃听。在 Internet这样的开放信道上要实现秘密共享难度很大。但是很有必要实现对共享秘密的处理，因为两个实体之间需要共享用于加密的密钥。关于如何在公共信道上安全地处理共享密钥这一问题，有一些重要的加密算法，是以对除预定接受者之外的任何人都保密的方式来实现的。

Diffie-Hellman密钥交换算法允许实体间交换足够的信息以产生会话加密密钥。按照惯例，假设一个密码协议的两个参与者实体分别是 Alice和Bob，Alice使用Bob的公开值和自己的秘密值来计算出一个值；Bob也计算出自己的值并发给 Alice，然后双方使用自己的秘密值来计算他们的共享密钥。其中的数学计算相对比较简单，而且不属于本书讨论的范围。算法的概要是Bob和Alice能够互相发送足够的信息给对方以计算出他们的共享密钥，但是这些信息却不足以让攻击者计算出密钥。

Diffie-Hellman算法通常称为公共密钥算法，但它并不是一种公共密钥加密算法。该算法可用于计算密钥，但密钥必须和某种其他加密算法一起使用。但是， Diffie-Hellman算法可用于身份验证。Network Associates公司的PGP公共密钥软件中就使用了此算法。

密钥交换是构成任何完整的 Internet安全性体系都必备的。此外， IPsec安全性体系结构还包括Internet密钥交换(IKE)及Internet安全性关联和密钥管理协议 (ISAKMP)。在后续章节中将讨论这些标准和其他相关标准。

4. 安全散列

散列是一定量数据的数据摘要的一种排序。检查数字是简单的散列类型，而安全散列则产生较长的结果，经常是128位。对于良好的安全散列，攻击者很难颠倒设计或以其他方式毁灭。安全散列可以与密钥一起使用，也可以单独使用。其目的是提供报文的数字摘要，用来验证已经收到的数据是否与发送者所发送的相同。发送者计算散列并将其值包含在数据中，接收者对收到的数据进行散列计算，如果结果值与数据中所携带的散列值匹配，接收者就可以确认数据的完整性。

常用的散列方法由RSA数据安全公司提出，包括MD2、MD4和MD5报文摘要函数。安全散列算法(SHA)是由美国国家标准和技术协会(NIST)所开发的标准摘要函数。散列可以单独使用，也可以和数字签名一起使用。

5. 数字签名

前面提到的公共密钥加密依赖于密钥对，而数字签名则依靠公共密钥加密的特性，即允许数据以实体密钥对中的秘密密钥来加密，以公共密钥来解密。发送者首先对于要签名的数据进行安全散列计算，然后对结果使用秘密密钥加密。而接收者首先进行相同的散列计算，然后对发送者所附加的加密值进行解密。如果两次计算的值能够匹配，接收者就可以确信公共密钥的主人就是对报文签名的实体，且报文在传送中并没有被修改。

RSA公共密钥加密算法可以用于数字签名。签名实体为待签名的数据建立散列，然后以自己的密钥对散列加密；证实实体则对接收到的数据进行相同的散列计算，使用签名实体的公共密钥对签名解密，并且比较所得的两个值。如果散列与解密的签名相同，则数据就得到证实。

数字签名有如下几种含义：

- 如果签名得到证实，说明所接收到的报文在从签名到接收的一段时间内未经任何改动。
- 如果不能证实签名，则说明或者是报文在传送过程中受到了破坏或篡改，或者是签名计算错误，又或者是签名在传送过程中被破坏或篡改。在上述任何情况下，未得到证实的签名并不一定是坏事，但是要求对报文重新签名并重传，以便最终能为接收者所接受。
- 如果签名得到证实，意味着与公共密钥相关联的实体是对报文签名的唯一实体。换言之，与公共密钥关联的实体不能否认自己的签名，这是数据签名的重要特性，称为不可抵赖。

还有其他机制可以实现数据签名，而RSA是其中应用最广泛的，并且已在大多数Internet产品中实现。

9.2.2 安全性关联

安全性关联是(SA)IPsec的基本概念。安全性关联包含能够唯一标识一个安全性连接的数据组合。连接是单方向的，每个SA由目的地址和安全性参数索引(SPI)来定义。其中SPI是对RFC 1825修改后的Internet草案中所要求的标识符，它说明使用SA的IP头类型，如AH或ESP。SPI为32位，用于对SA进行标识及区分同一个目的地址所链接的多个SA。进行安全通信的两个系统有两个不同的SA，每个目的地址对应一个。

每个SA还包括与连接协商的安全性类型相关的多个信息。这意味着系统必须了解其SA、与SA目的主机所协商的加密或身份验证算法的类型、密钥长度和密钥生存期。

9.2.3 密钥管理

如何管理密钥是Internet安全性专业人士面临的最复杂的问题之一。密钥管理不仅包括使用密钥协议来分发密钥，还包括在通信系统之间对密钥的长度、生存期和密钥算法进行协商。Internet工作组和研究团体对此已进行了大量工作，但是由于尚未达成一致，目前还没有发表任何RFC。

Internet安全性关联密钥管理协议 (ISAKMP)为密钥的安全交换定义了整个基本构架。ISAKMP实际上是一个应用协议，协议中定义了用于系统之间协商密钥交换的不同类型报文，它在传输层使用UDP。

但是ISAKMP只是特定机制所使用的框架，而没有定义实际完成交换的机制和算法。这些年来在不同的建议中定义了大量的交换机制，通常以 Diffie-Hellman密钥交换为基础。主要的提案包括：

- Photuris。此提案基于Diffie-Hellman 算法，但增加了要求，即要求节点首先发送一个cookie(一个随机数)，然后服务器给予应答，这样减少了否认服务攻击的威胁 (否认服务攻击是由攻击者伪造源地址而导致的)。Photuris也要求通信各方都必须对协商好的密钥签名，以减少中间者攻击的危害 (所谓中间者攻击，是指某个攻击者对系统的 Alice冒充自己是Bob，又对另一个系统的Bob冒充自己是Alice)。
- Sun公司的Internet协议的简单密钥管理 (SKIP)。SKIP也是以Diffie-Hellman密钥交换为基础，但是它并不要求通信各方使用随机数来计算其密钥，而是要求使用静态的密码表。各方查找密码表中的秘密值，然后基于查到的秘密值来计算，并传送所算出的值。
- OAKLEY。此机制与Photuris有某些相似特性，但在不考虑否认服务攻击的情形下，它提供不同的密钥交换模式。

1998年秋，基于OAKLEY和SKEME(Internet的安全密钥交换机制)，Internet密钥交换最终在Internet密钥交换规范中得以定义。

读者应该注意到，人工密钥管理也是一个重要选项，而且在很多情况下是唯一的选项。人工方法要求个人单独交付密钥，并使用密钥来配置网络设备。即使在开放标准已经充分确定并且实现之后，人工密钥管理仍将继续是一个重要选择，对于商业产品尤其如此。

9.2.4 实现IPsec

IP层安全性用于保护 IP数据报。它不一定要涉及用户或应用。这意味着用户可以愉快地使用应用程序，而无需注意所有的数据报在发送到 Internet之前，需要进行加密或身份验证，当然在这种情形下所有的加密数据报都要由另一端的主机正确地解密。

这样就引入了如何实现IPsec的问题，有如下三种可能方法：

- 将IPsec作为IPv4栈或IPv6栈的一部分来实现。这种方法将 IP安全性支持引入IP网络栈，并且作为任何IP实现的一个必备部分。但是，这种方法也要求对整个实体栈进行更新以反映上述改变。
- 将IPsec作为“栈中的一块”(BITS)来实现。这种方法将特殊的 IPsec代码插入到网络栈中，在现有IP网络软件之下、本地链路软件之上。换言之，这种方法通过一段软件来实现安全性，该软件截获从现有 IP栈向本地链路层接口传送的数据报，对这些数据报进行必要的安全性处理，然后再交给链路层。这种方法可用于将现有系统升级为支持 IPsec

的系统，且不要求重写原有的IP栈软件。

- 将IPsec作为“线路的一块”(BITW)来实现。这种方法使用外部加密硬件来执行安全性处理功能。该硬件设备通常是作为一种路由器使用的IP设备，或者更确切一些，是安全性网关，此网关为位于它后面的所有系统发送的IP数据报服务。如果这样的设备只用于一个主机，其工作情况与BITS方法类似，但如果一个BITW设备为多个系统服务，实现相对要复杂得多。

上述各种方法的差别不在于字面上，而在于它们的适用情况不同。要求高级别安全性的应用最好使用硬件方法实现；而如果系统不具备与新的 IPsec兼容的网络栈，应用最好选择BITS方法。

9.2.5 隧道模式与透明模式

本书在后续章节中讨论移植策略时，还将涉及协议隧道概念。而对于 IP安全性，隧道同样重要。见图9-1，两个系统建立了SA，以便在Internet上安全地通信。其中一个系统产生网络业务流，经过加密或者签名，然后发送给目的系统。而在接收方，首先对收到的数据报进行解密或者身份验证，把净荷向上传送给接收系统的网络栈，由使用数据的应用进行最后的处理。两个主机之间的通信如同没有安全性头一样简单，而且数据报实际的 IP头必须要暴露出来以便在Internet选路，因此这种方法称为使用SA的透明模式。

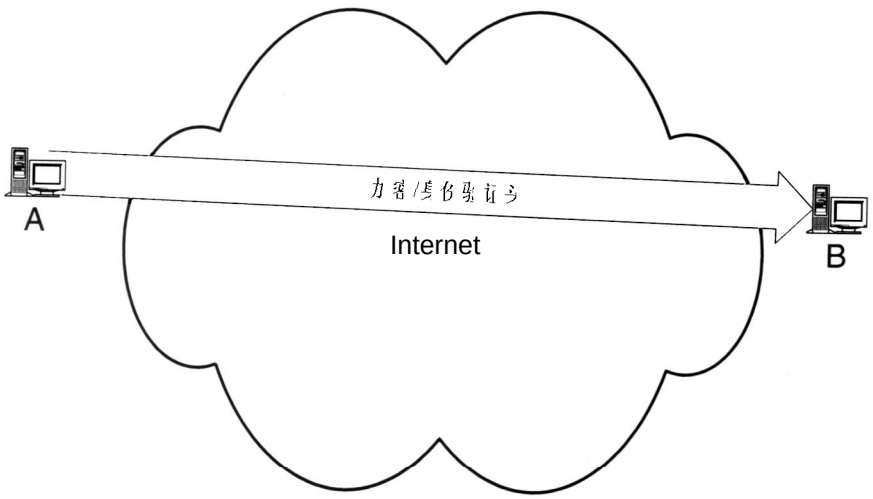

图9-1 一对主机使用IPsec进行透明通信

SA也可以用来将安全IP以隧道方式通过互联网络。见图 9-2，来自系统A的所有IP包首先转发到安全性网关X，由X建立一条跨越Internet、目的地为安全性网关 Y的隧道，由Y对经隧道方式传来的数据拆包并转发。安全性网关 Y可能将包转发给本地互联网络内的任一主机 B、C或D，也可能转发给外部主机，如M。这取决于源主机如何为这些包定向。如果 SA目的节点是安全性网关，则称为隧道关联。即，隧道传送既可以在两个安全性网关之间进行（见图9-2），也可以在正规节点和安全性网关之间进行。因此，图 9-2中的主机M可以与安全性网关X或Y建立隧道连接，M所发送的数据报首先传送给安全性网关，然后经过网关解密或身份验证之后，再进行正确地转发，由此可见这是一种隧道方式。

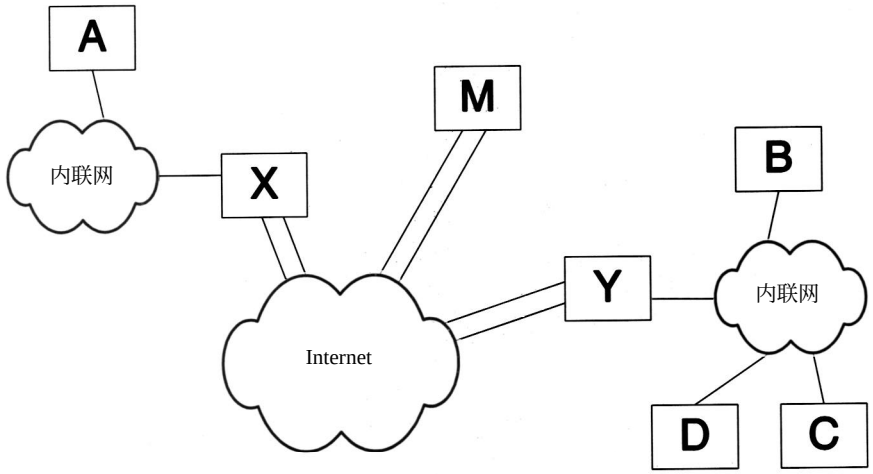

图9-2 IP安全性隧道

9.3 IPv6安全性头

如前所述，IPsec安全性服务完全通过AH和封装安全性净荷(ESP)头相结合的机制来提供，当然还要有正确的相关密钥管理协议。RFC 1826(IP身份验证头)中对AH进行了描述，而ESP头在RFC 1827(IP封装安全性净荷(ESP))中描述。上述RFC及IP安全性体系结构RFC仅仅是解决安全性问题的第一步。IPsec工作组各成员正继续对这些扩展头的规范进行改进，这些文档的当前草案的篇幅几乎是原RFC的两倍。这些草案保留了原RFC的语言和意图，并进行了扩充，对包头及其功能的描述更加完整，综合性更强。

各安全性头可以单独使用，也可以一起使用。如果一起使用多个扩展头，AH应置于ESP头之前，这样，首先进行身份验证，然后再对ESP头净荷解密。使用IPsec隧道时，这些扩展头也可以嵌套。即，源节点对IP包进行加密和数字签名，然后发送给本地安全性网关，该网关则再次进行加密和数字签名，然后发送给另一个安全性网关。

AH和ESP头既可以用于IPv4，也可以用于IPv6，这一点很重要。本节将讨论这些安全性扩展头在IPv6中如何使用，对于IPv4，这些扩展头作为选项加在正常的IPv4头中。

9.3.1 身份验证头

AH的作用如下：

- 为IP数据报提供强大的完整性服务，这意味着AH可用于为IP数据报承载内容验证数据。
- 为IP数据报提供强大的身份验证，这意味着AH可用于将实体与数据报内容相链接。
- 如果在完整性服务中使用了公共密钥数字签名算法，AH可以为IP数据报提供不可抵赖服务。
- 通过使用顺序号字段来防止重放攻击。

AH可以在隧道模式或透明模式下使用，这意味着它既可用于为两个节点间的简单直接的数据报传送提供身份验证和保护，也可用于对发给安全性网关或由安全性网关发出的整个数据报流进行封装。

1. 语义

IPv6中的AH与其他扩展头一起使用时，必须置于那些将由中间路由器处理的扩展头之后，及那些只能由数据报目的地处理的扩展头之前。这意味着 AH应置于逐跳扩展头、选路扩展头或分段扩展头之后。根据不同情况， AH可在目的地选项扩展头之前，也可在其后。

在透明模式中， AH保护初始IP数据报的净荷，也保护在逐跳转发中不变化的部分 IP头，如跳极限字段或选路扩展头。图 9-3中显示了在透明模式中，当计算和增加 AH时，IP数据报的变化情况。图中的目的地选项头也可以置于 AH之前。对于目的IP地址和扩展头，仅在逐跳转发它们不发生变化的情况下，才能得到保护。

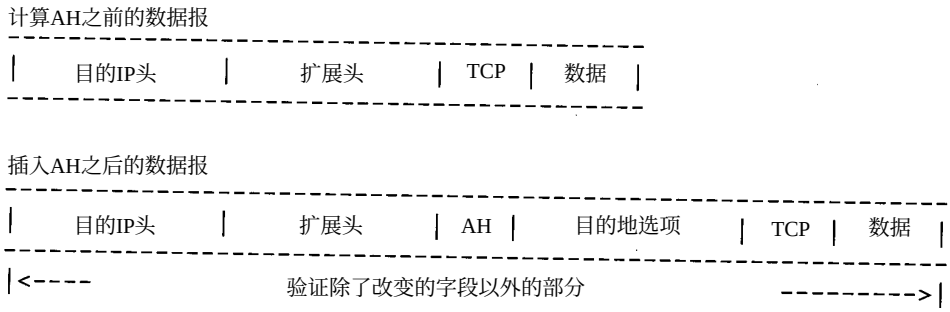

图9-3 在透明模式中为IP数据报增加AH

当AH用于隧道模式中时，使用方法与上不同。图 9-4表明了其中的区别。初始的目的 IP地址与整个初始 IP数据报一起，封装在全新的 IP数据报中，该数据报再发送到安全性网关。因此，整个初始IP数据报以及传送中不变的封装 IP头部分都得以保护。

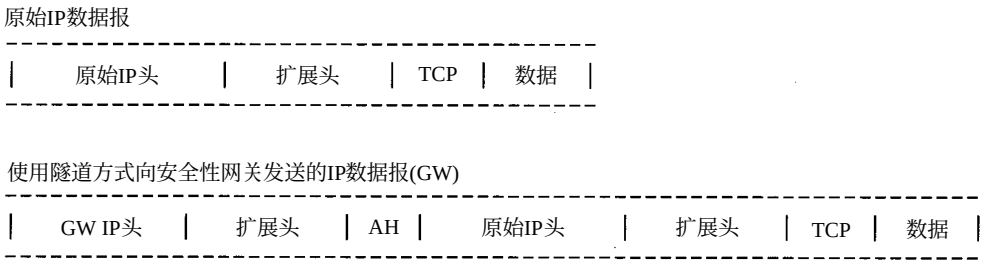

图9-4 在隧道模式中为IP数据报增加AH

2. AH字段

图9-5表示了AH的格式和各字段。与所有的 IPv6扩展头一样，第一个字段是 8位的下一个头字段，它表示后续的扩展头协议。其他字段包括：

(1) 净荷长度。此8位字段指明AH的整个长度，其值以32位字为单位，并减去2 。正如初始的定义， AH包含64位，其余部分为身份验证数据 (参见后续内容)。因此净荷长度字段只指出身份验证数据以32 位字为单位的长度。加入序列号字段 (参见后续内容)后，此值等于身份验证数据加上序列号字段的长度。

(2) 保留。净荷长度字段之后的16位为将来使用而保留。目前，此16位必须全部置为0。

(3) 安全性参数索引(SPI)。此32位字段是一个任意数。与目的IP地址和安全性协议一起使用， SPI是AH使用的SA的唯一标识。若SPI值为0，则表示只用于本地而不予传送；值 1~255被Internet分配号码授权机构(IANA)保留作将来使用。

(4) 序列号。此32位字段是一个必备的计数器，由发送者插入 IP头，但不一定由接收者使

用。从0开始，每发送一个数据报，该计数器增 1，这可用于预防重放攻击。若接收者使用此字段来对抗重放攻击，对于序列号与已收到的数据报相同的数据报，接收者将予以丢弃。这意味着若计数器重新开始循环，即已经接收到 2^{32}个数据报，则必须协商新的 SA。否则，一旦计数器重新置位，接收系统将丢弃所有的数据报。

(5) 身份验证数据。此字段包含完整性检查值 (ICV)，这是AH的核心。其内容的长度必须是32位的整数倍，为满足这个条件，其中可能包含填充字段。下节将讨论该值的计算。

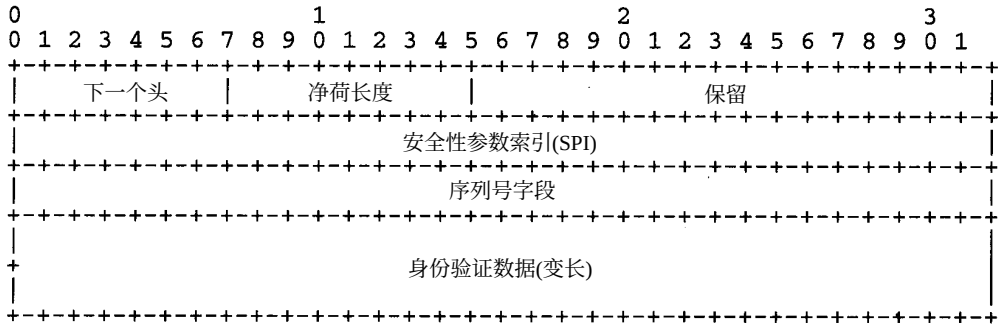

图9-5 AH格式和字段

3. 计算完整性检查值

对于如何计算ICV以及使用什么机制来计算， RFC 1826的描述比较模糊。实际上，术语“完整性检查值”在该文档中并没有出现，而是出现在将要代替 RFC 1826的后续草案中。预期适当的身份验证算法将导致ICV的产生。建议的算法包括：

- 报文身份验证代码(MAC)，然后对其结果用适当的对称加密算法 (如DES)进行加密。
- 安全散列功能，如MD5或SHA的更新版SHA-1。

按照标准的约定，预计 AH的任何实现将必须支持 MD5和SHA-1密钥散列。身份验证数据针对整个IP数据报净荷以及IP头的不变部分或可预测部分来计算。

9.3.2 封装安全性净荷头

ESP头被用于允许IP节点发送和接收净荷经过加密的数据报。更确切一点， ESP头是为了提供几种不同的服务，其中某些服务与 AH有所重叠。ESP头提供的服务包括：

- 通过加密提供数据报的机密性。
- 通过使用公共密钥加密对数据来源进行身份验证。
- 通过由AH提供的序列号机制提供对抗重放服务。
- 通过使用安全性网关来提供有限的业务流机密性。

ESP头可以和 AH结合使用。实际上，如果 ESP头不使用身份验证的机制，建议将 AH和ESP头一起使用。

1. 语义

ESP头必须跟随在去往目的节点所途经的中间节点需要处理的扩展头之后， ESP头之后的数据都可能被加密。实际上，加密的净荷是作为 ESP头的最后一个字段(参见后续内容)。

与AH类似，ESP既可用于隧道模式，也可用于透明模式。在透明模式中，如果有 AH，IP头以及逐跳扩展头、选路扩展头或分段扩展头都在 AH之前，其后跟随ESP头。任何目的地选

项头可以在ESP头之前，也可以在ESP头之后，或者ESP头前后都有，而ESP头之后的扩展头将被加密。

在很多方面，仅仅是常规数据报带着加密净荷从源端传送到目的端。某些情况下，适合在透明模式中使用ESP。但是，这种模式使攻击者有可能研究两个节点之间的业务流，留意正在通信的节点、节点之间交换的数据量、交换的时间等。所有这些信息都可能为攻击者提供有助于对通信双方进行攻击的信息。

类似前面描述的AH的情形，使用安全性网关是一种替代方法。安全性网关可以直接与节点连接，也可以链接到另一个安全性网关。单个节点可以在隧道模式中使用 ESP，即加密所有出境包，并封装到单独的 IP数据报流中，再发送给安全性网关。然后网关解密业务流，并重新将原始IP数据报发往目的地。

使用隧道模式时， ESP头对整个IP数据报进行封装，并作为IP头的扩展将数据报定向到安全性网关。ESP头与AH的结合也有几种不同方式，例如以隧道方法传送的数据报可能有透明模式的AH。

2. 字段

ESP头与其他扩展头不同。其一，下一个头字段的位置接近 ESP头的末端。其二， ESP头之前的扩展头将其下一个头字段值置为 50，以指明随后是ESP头。ESP头的其余部分将可能包括如下字段：

- 安全性参数索引(SPI)。与上节提到的AH中的32位SPI值相同。通信节点使用该值来指出SA，SA用于确定数据应如何加密。
- 序列号。32位，从0开始，每发送一个数据报，该值加 1。如前所述，序列号可用于防御重放攻击，在循环用完所有 2^{32} 个值之前，必须建立新的SA。
- 净荷数据。此字段长度可变，它实际上包含数据报的加密部分以及加密算法需要的补充数据，例如初始化数据。
- 填充。头的加密部分(净荷)必须在正确的边界终止，因此有时需要填充。
- 填充长度。此字段指明净荷数据所需要填充的数据量。
- 下一个头：此字段像其他 IPv6扩展头中的字段一样操作，但是它不位于扩展头的开始，而是靠近扩展头末端。
- 身份验证数据。此字段是一个 ICV，它对除身份验证数据本身之外的整个 ESP头进行计算。这种身份验证计算是可选的。

3. 进行封装

预计一个兼容的ESP实现至少要求支持DES加密和SHA-1身份验证。它也可以支持其他算法，但支持上述两个算法是最低要求。