mode)。更为适合 VPN 的隧道模式比运输模式部署得更为广泛。为了进一步讲清 IPsec 和避免许多难题，我们因此专门关注隧道模式。一旦已经牢牢地掌握了隧道模式，应当能够容易地自学运输模式。

IPsec 数据报的分组格式显示在图 8-29 中。你也许认为分组格式是枯燥乏味的，但我们将很快看到 IPsec 数据报实际上尝起来像美式墨西哥风味（Tex-Mex）美食！我们考察图 8-28 的场景中的 IPsec 字段。假设路由器 R1 接收到一个来自主机 172.16.1.17（在总部网络中）的普通 IPv4 数据报，该分组的目的地是主机 172.16.2.48（在分支机构网络中）。路由器 R1 使用下列方法将这个“普通 IPv4 数据报”转换成一个 IPsec 数据报：

- 在初始 IPv4 数据报（它包括初始首部字段!）后面附上一个“ESP 尾部”字段。
- 使用算法和由 SA 规定的密钥加密该结果。
- 在这个加密量的前面附加上一个称为“ESP 首部”的字段，得到的包称为“enchilada”（以辣椒调味的一种墨西哥菜。——译者注）。
- 使用算法和由 SA 规定的密钥生成一个覆盖整个 enchilada 的鉴别 MAC。
- 该 MAC 附加到 enchilada 的后面形成载荷。
- 最后，生成一个具有所有经典 IPv4 首部字段（通常共 20 字节长）的全新 IP 首部，该新首部附加到载荷之前。

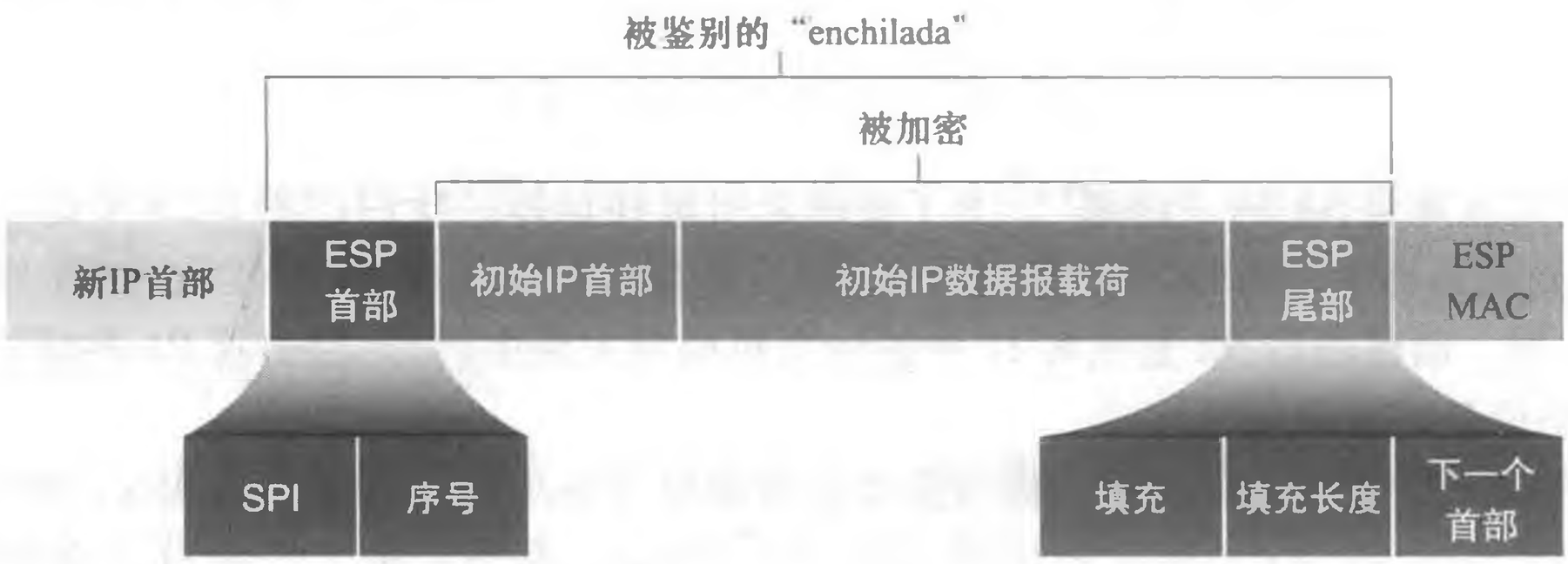

图 8-29　IPsec 数据报格式

注意到得到的 IPsec 数据报是一个货真价实的 IPv4 数据报，它具有传统的 IPv4 首部字段后跟一个载荷。但在这个场合，该载荷包含一个 ESP 首部、初始 IP 数据报、一个 ESP 尾部和一个 ESP 鉴别字段（具有加密的初始数据报和 ESP 尾部）。初始的 IP 数据报具有源 IP 地址 172.16.1.17 和目的地址 172.16.2.48。因为 IPsec 数据报包括了该初始 IP 数据报，这些地址被包含和被加密作为 IPsec 分组负载的组成部分。但是在新 IP 首部中的源和目的地 IP 地址，即在 IPsec 数据报的最左侧首部又该如何处理呢？如你所猜测，它们被设置为位于隧道两个端点的源和目的地路由器接口，也就是 200.168.1.100 和 193.68.2.23。同时，这个新 IPv4 首部字段中的协议号不被设置为 TCP、UDP 或 SMTP，而是设置为 50，指示这是一个使用 ESP 协议的 IPsec 数据报。

在 R1 将 IPsec 数据报发送进公共因特网之后，它在到达 R2 之前将通过许多路由器。这些路由器中的每个将处理该数据报，就像它是一个普通数据报一样，即它们被完全忘记这样的事实：该数据报正在承载 IPsec 加密的数据。对于这些公共因特网路由器，因为在外面首部中的目的 IP 地址是 R2，所以该数据报的最终目的地是 R2。

在考察了如何构造一个 IPsec 数据报的例子后，我们现在更仔细地观察 enchilada 的组成。我们看到在图 8-29 中的 ESP 尾部由三个字段组成：填充、填充长度和下一个首部。

前面讲过块密码要求被加密的报文必须为块长度的整数倍。使用填充（由无意义的字节组成），使得当其加上初始数据报（连同填充长度字段和下一个首部字段）形成的“报文”是块的整数倍。填充长度字段指示接收实体插入的填充是多少（并且需要被删除）。下一个首部字段指示包含在载荷数据字段中数据的类型（例如 UDP）。载荷数据（通常是初始 IP 数据报）和 ESP 尾部级联起来并被加密。

附加到这个加密单元前面的是 ESP 首部，该首部以明文发送，它由两个字段组成：SPI 字段和序号字段。SPI 字段指示接收实体该数据报属于哪个 SA；接收实体则能够用该 SPI 索引其 SAD 以确定适当的鉴别/解密算法和密钥。序号字段用于防御重放攻击。

发送实体也附加一个鉴别 MAC。如前所述，发送实体跨越整个 enchilada（由 ESP 首部、初始 IP 数据报和 ESP 尾部组成，即具有加密的数据报和尾部）计算一个 MAC。前面讲过为了计算一个 MAC，发送方附加一个秘密 MAC 密钥到该 enchilada，进而计算该结果的一个固定长度散列。

当 R2 接收到 IPsec 数据报时，R2 看到该数据报的目的 IP 地址是 R2 自身。R2 因此处理该数据报。因为协议字段（位于 IP 首部最左侧）是 50，R2 明白应当对该数据报施加 IPsec ESP 处理。第一，针对 enchilada，R2 使用 SPI 以确定该数据报属于哪个 SA。第二，它计算该 enchilada 的 MAC 并且验证该 MAC 与在 ESP MAC 字段中的值一致。如果两者一致，它知道该 enchilada 来自 R1 并且未被篡改。第三，它检查序号字段以验证该数据报是新的（并且不是重放的数据报）。第四，它使用与 SA 关联的解密算法和密钥解密该加密单元。第五，它删除填充并抽取初始的普通 IP 报文。最后，它朝着其最终目的地将该初始数据报转发进分支机构网络。这个一种多么复杂的秘诀呀！还未曾有人声称准备并破解 enchilada 是一件容易的事！

实际上还有另一个重要的细微差别需要处理。它以下列问题为中心：当 R1 从位于总部网络中的一台主机收到一个（未加密的）数据报时，并且该数据报目的地为总部以外的某个目的 IP 地址，R2 怎样才能知道它应当将其转换为一个 IPsec 数据报呢？并且如果它由 IPsec 处理，R1 如何知道它应当使用（在其 SAD 中的许多 SA 中）哪个 SA 来构造这个 IPsec 数据报呢？该问题以如下方式解决。除了 SAD 外，IPsec 实体也维护另一个数据结构，它称为**安全策略库**（Security Policy Database，SPD）。该 SPD 指示哪些类型的数据报（作为源 IP 地址、目的 IP 地址和协议类型的函数）将被 IPsec 处理；并且对这些将被 IPsec 处理的数据报应当使用哪个 SA。从某种意义上讲，在 SPD 中的信息指示对于一个到达的数据报做“什么”；在 SAD 中的信息指示“怎样”去做。

IPsec 服务的小结

IPsec 究竟提供什么样的服务呢？我们从某攻击者 Trudy 的角度来考察这些服务，Trudy 是一个中间人，位于图 8-28 中 R1 和 R2 之间路径上的某处。假设通过这些讨论，Trudy 不知道 SA 所使用的鉴别和加密密钥。Trudy 能够做些什么和不能够做些什么呢？第一，Trudy 不能看到初始数据报。如果事实如此，不仅 Trudy 看不到在初始数据报中的数据，而且也看不到协议号、源 IP 地址和目的 IP 地址。对于经该 SA 发送的数据报，Trudy 仅知道该数据报源于 172.16.1.0/24 的某台主机以及目的地为 172.16.2.0/24 的某台主机。她不知道它是否携带 TCP、UDP 或 ICMP 数据；她不知道它是否携带了 HTTP、SMTP 或某些其他类型的应用程序数据。因此这种机密性比 SSL 范围更为宽广。第二，Trudy 试图用反转数据报的某些比特来篡改在 SA 中的某个数据报，当该篡改的数据报到达 R2 时，它

将难以通过完整性核查（使用 MAC），再次挫败了 Trudy 的恶意尝试。第三，假设 Trudy 试图假冒 R1，生成一个源为 200.168.1.100 和目的地为 193.68.2.23 的 IPsec 数据报。Trudy 的攻击将是无效的，因为这个数据报将再次通不过在 R2 的完整性核查。最后，因为 IPsec 包含序号，Trudy 将不能够生成一个成功的重放攻击。总而言之，正如本节开始所言，IPsec 在任何通过网络层处理分组的设备对之间，提供了机密性、源鉴别、数据完整性和重放攻击防护。

8.7.5　IKE：IPsec 中的密钥管理

当某 VPN 具有少量的端点时（例如，图 8-28 中只有两台路由器），网络管理员能够在该端点的 SAD 中人工键入 SA 信息（加密/鉴别算法和密钥及 SPI）。这样的“人工密钥法”对于一个大型 VPN 显然是不切实际的，因为大型 VPN 可能由成百甚至上千台 IPsec 路由器和主机组成。大型的、地理上分散的部署要求一个自动的机制来生成 SA。IPsec 使用**因特网密钥交换**（Internet Key Exchange，IKE）协议来从事这项工作，IKE 由 RFC 5996 定义。

IKE 与 SSL（参见 8.6 节）中的握手具有某些类似。每个 IPsec 实体具有一个证书，该证书包括了该实体的公开密钥。如同使用 SSL 一样，IKE 协议让两个实体交换证书，协商鉴别和加密算法，并安全地交换用于在 IPsec SA 中生成会话密钥的密钥材料。与 SSL 不同的是，IKE 应用两个阶段来执行这些任务。

我们来研究图 8-28 中两台路由器 R1 和 R2 场景下的这两个阶段。第一个阶段由 R1 和 R2 之间报文对的两次交换组成：

- 在报文的第一次交换期间，两侧使用 Diffie-Hellman（参见课后习题）在路由器之间生成一个双向的 IKE SA。为了防止混淆，这个双向 IKE SA 完全不同于 8.6.3 节和 8.6.4 节所讨论的 IPsec SA。该 IKE SA 在这两台路由器之间提供了一个鉴别的和加密的信道。在首个报文对交换期间，创建用于 IKE SA 的加密和鉴别的密钥。还创建了将用于计算后期在阶段 2 使用的 IPsec SA 密钥的一个主密钥。观察在第一步骤期间，没有使用 RSA 公钥和私钥。特别是，R1 或 R2 都没有通过用它们的私钥对报文签字而泄露其身份。
- 在报文的第二次交换期间，两侧通过对其报文签名而透漏了它们的身份。然而，这些身份并未透漏给被动的嗅探者，因为这些报文是经过安全的 IKE SA 信道发送的。同时在这个阶段期间，两侧协商由 IPsec SA 应用的 IPsec 加密和鉴别算法。

在 IKE 的第二个阶段，两侧生成在每个方向的一个 SA。在阶段 2 结束时，对这两个 SA 的每一侧都建立了加密和鉴别会话密钥。然后这两侧都能使用 SA 来发送安全的数据报，如同 8.7.3 节和 8.7.4 节描述的那样。在 IKE 中有两个阶段的基本动机是计算成本，即因为第二阶段并不涉及任何公钥密码，IKE 能够以相对低的计算成本在两个 IPsec 实体之间生成大量 SA。

8.8　使无线 LAN 安全

在无线网络中安全性是特别重要的关注因素，因为这时携带数据帧的无线电波可以传播到远离包含无线基站和主机的建筑物以外的地方。在本节中，我们简要介绍了无线安全性。对于更为深入地探讨，参见由 Edney 和 Arbaugh 撰写的可读性很强的书［Edney

2003]。

在 802.11 中的安全性问题受到了技术界和媒体界的极大关注。在进行大量讨论的同时，一个几乎没有争论的事实是，看起来被广泛认同的初始 802.11 规范具有一些严重的安全性缺陷。现在的确能够下载利用这些漏洞的公共域软件，使得那些使用该普通 802.11 安全性机制的用户面对安全性攻击，就像根本没有使用安全性措施的网络用户一样，门户洞开。

在下面一节中，我们讨论最初在 802.11 规范中标准化的安全性机制，该规范统称为**有线等效保密**（Wired Equivalent Privacy，WEP）。顾名思义，WEP 意欲提供类似于在有线网络中的安全性水平。接下来我们将讨论 WEP 中的安全性漏洞并讨论 802.11i 标准，后者是在 2004 年采纳的 802.11 的本质上更为安全的版本。

8.8.1 有线等效保密

IEEE 802.11 的 WEP 协议设计于 1999 年，为在主机和无线接入点（即基站）之间提供鉴别和数据的加密。WEP 并没有指定密钥管理算法，因此它假定主机和无线接入点之间通过带外方式就密钥达成了某种一致。鉴别以下列方式进行：

1）无线主机通过接入点请求鉴别。

2）接入点以一个 128 字节的不重数值响应该鉴别请求。

3）无线主机用它与这个接入点共享的密钥加密这个不重数值。

4）接入点解密主机加密的不重数值。

如果解密所得不重数值与初始发给主机的值相同，则这个接入点鉴别了该主机。

图 8-30 阐述了 WEP 数据加密算法。假定主机和接入点都知道一个秘密的 40 比特对称密钥 K_S。此外，一个 24 比特的初始向量（IV）附加到这个 40 比特的密钥后面，产生用于加密单个帧的一个 64 比特密钥。每一个帧所使用的 IV 都不同，所以每一帧都由不同的 64 比特密钥加密。加密以如下方式进行。首先为每个数据载荷计算一个 4 字节的 CRC 值（见 6.2 节）。然后用 RC4 流密码加密该载荷和该 4 字节 CRC。我们这里不涉及 RC4 的细节（细节参见［Schneier 1995］和［Edney 2003］）。就我们的目的而言，知道下列事实即可：对于密钥值（此时为 64 比特（K_S、IV）密钥），RC4 算法产生一个密钥值的流为 k_1^{IV}，k_2^{IV}，k_3^{IV}，…，这些密码值用于加密一帧中的数据和 CRC 值。出于实用的目的，我们可以认为每次对一个字节执行这些操作。通过把数据的第 i 字节 d_i 和由（K_S、IV）对生成的密钥值流中的第 i 个密钥 k_i^{IV} 执行异或操作进行加密，以产生密文的第 i 字节 c_i：

$$c_i = d_i \oplus k_i^{IV}$$

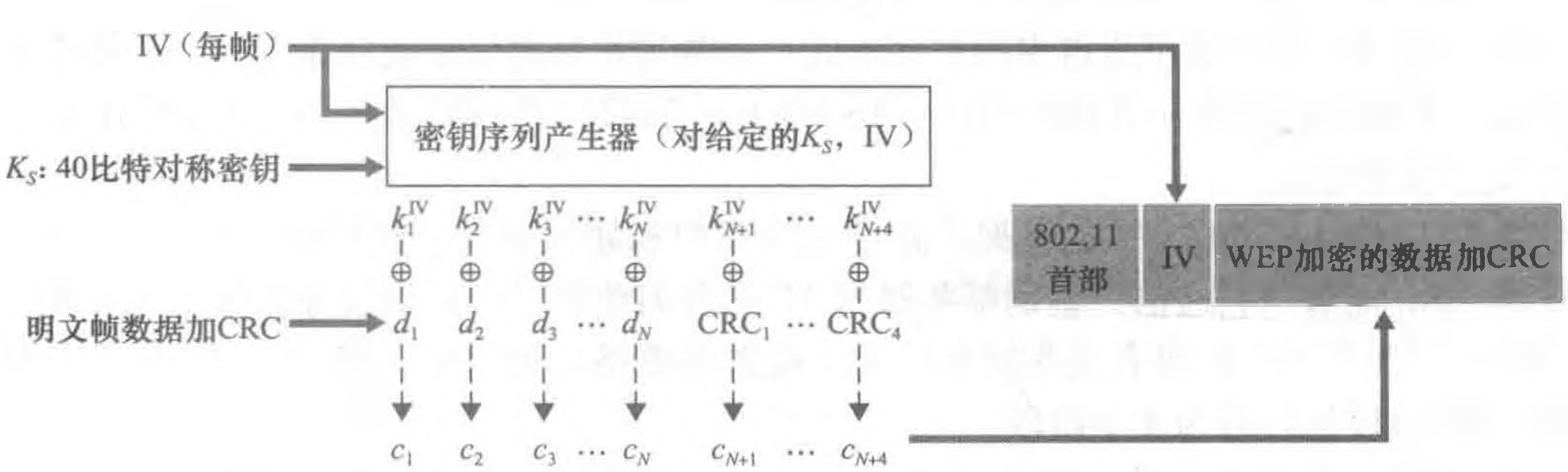

图 8-30 802.11 WEP 协议

如图 8-30 中所示，该 IV 值逐帧而变化，以明文形式出现在每一个 WEP 加密的 802.11 帧的首部中。接收方取它与发送方共享的秘密 40 比特对称密钥，添加上该 IV，并使用形成的 64 比特的密钥（它与发送方执行加密所用的密钥相同）来解密这个帧。

$$d_i = c_i \oplus k_i^{IV}$$

正确使用 RC4 算法要求同一个 64 比特密钥决不能使用超过 1 次。前面讲过 WEP 密钥是每一帧变换一次。对于某给定的 K_S（即使它有变化，也是很少的），这意味着只有 2^{24} 个不同的密钥可用。如果随机选择这些密钥的话，我们能够看到［Edney 2003］，则仅在处理 12 000 帧之后就选中相同的 IV 值（从而使用相同的 64 比特密钥）的概率超过 99%。在 1KB 帧长和 11Mbps 数据传输率的情况下，传输 12 000 帧仅需几秒的时间。此外，由于 IV 值在该帧中以明文形式传输，窃听者就会发现何时使用了一个重复的 IV 值。

为理解重复使用一个密钥可能出现的几个问题之一，考虑下面的选择明文攻击的情况，仍以 Trudy 对 Alice 进行攻击为例。假定 Trudy（可能使用 IP 哄骗）向 Alice 发出一个请求（例如，一个 HTTP 或 FTP 请求），要求 Alice 传输内容已知的文件 $d_1, d_2, d_3, d_4, \cdots$，Trduy 也观察到 Alice 发送的已加密数据 $c_1, c_2, c_3, c_4, \cdots$，由于 $d_i = c_i \oplus k_i^{IV}$，如果在这个等式两边同时异或 c_i，可得到：

$$d_i \oplus c_i = k_i^{IV}$$

根据这个关系，Trudy 就可以使用已知的 d_i 和 c_i 值计算出 k_i^{IV}。下一次 Trudy 看到使用同一 IV 值时，她将知道密钥流为 $k_1^{IV}, k_2^{IV}, k_3^{IV}, \cdots$，并可使用这些密钥解密报文。

对于 WEP 还有其他几个值得关注的安全性问题。［Fluhrer 2001］描述了一种攻击方法，即当选择某些弱密钥时在 RC4 中暴露出的一种已知弱点。［Stubblefield 2002］讨论了实现和开发这种攻击的有效方法。对 WEP 的另一种关注与在图 8-30 中显示并在 802.11 帧中传输的用以检测载荷中改变的比特的 CRC 比特有关。然而，攻击者在改变加密内容（例如用乱七八糟的东西替代初始的加密数据）后，对这些被替换的东西计算出一个 CRC，并将该 CRC 放置在 WEP 帧中产生一个将被接收方接受的 802.11 帧。此时所需要的是诸如我们在 8.3 节中学习的报文完整性技术来检测内容篡改或替换。有关 WEP 安全性更多的细节，参见［Edney 2003；Wright 2015］及其中的参考文献。

8.8.2　IEEE 802.11i

在 IEEE 802.11 于 1999 年发布后不久，就开始研发具有更强安全性机制的 802.11 的新型、改进版本。这个新标准被称为 802.11i，于 2004 年最终得到批准。如我们将看到的那样，虽然 WEP 提供了相对弱的加密、仅有单一方式执行鉴别并且没有密钥分发机制，但 IEEE 802.11i 却提供了强得多的加密形式、一种可扩展的鉴别机制集合以及一种密钥分发机制。下面我们概述一下 802.11i；［TechOnline 2012］是一篇关于 802.11i 的优秀（流式音频）技术概述。

图 8-31 概述了 802.11i 的框架。除了无线客户和接入点外，802.11i 定义了一台鉴别服务器，AP 能够与它通信。鉴别服务器与 AP 的分离使得一台鉴别服务器服务于许多 AP，集中在一台服务器中作出有关鉴别和接入（通常是敏感）的决定，降低了 AP 的成本和复杂性。802.11i 运行分为 4 个阶段：

1）发现。在发现阶段，AP 通告它的存在以及它能够向无线客户节点提供的鉴别和加密形式。客户则请求它希望的特定鉴别和加密形式。尽管客户和 AP 已经交换了报文，但

该客户还没有被鉴别，也还没有加密密钥，因此在该客户能够通过无线信道与任何远程主机通信之前，还需要进行几个其他步骤。

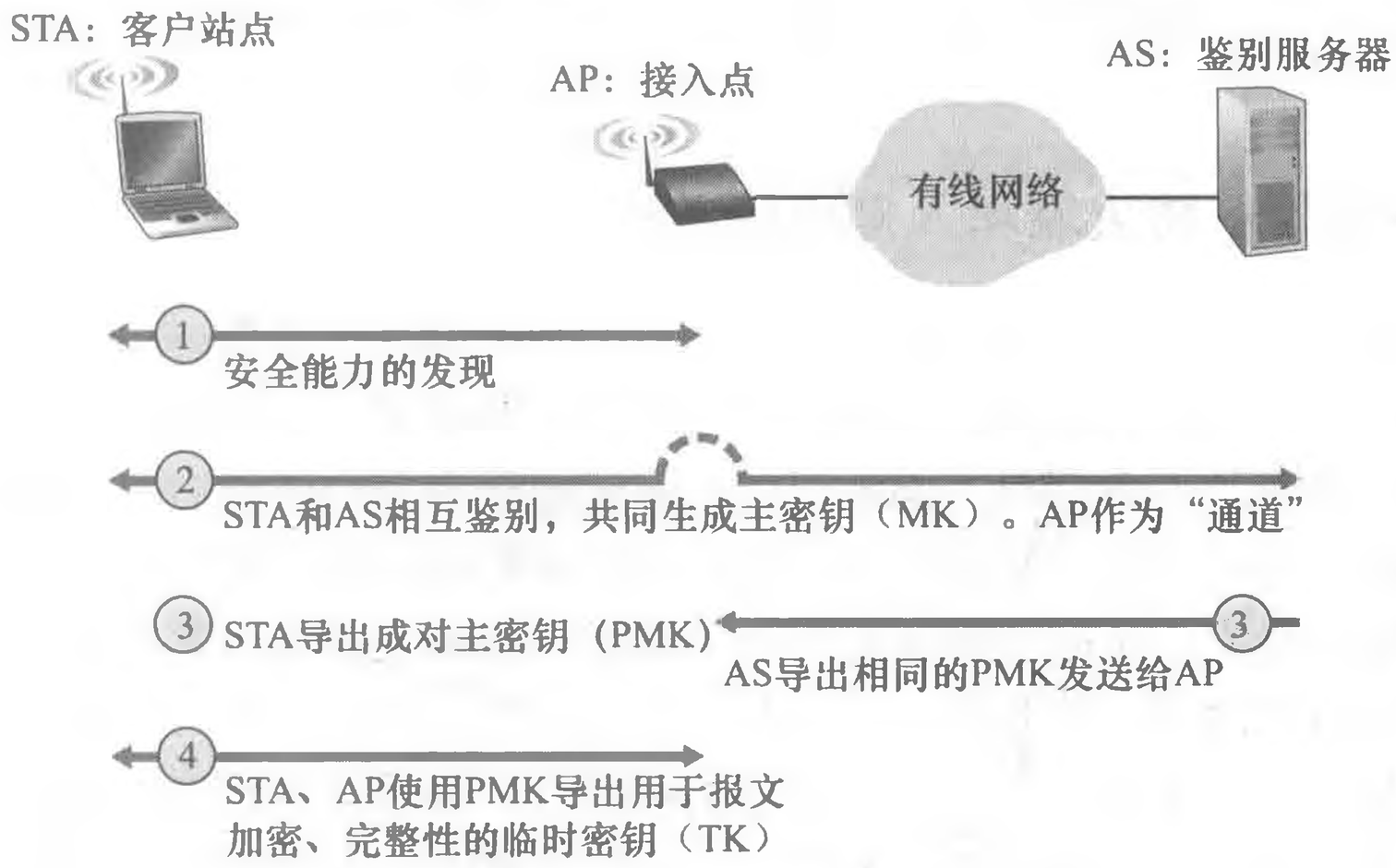

图 8-31　802.11i：运行的 4 个阶段

2）相互鉴别和主密钥（MK）生成。鉴别发生在无线客户和鉴别服务器之间。在这个阶段，接入点基本是起中继的作用，在客户和鉴别服务器之间转发报文。**可扩展鉴别协议**（Extensible Authentication Protocol，EAP）［RFC 3748］定义了客户和鉴别服务器之间交互时简单的请求/响应模式中使用的端到端报文格式。如图 8-32 中所示，EAP 报文使用 EAPoL（EAP over LAN，［IEEE 802.1x］）进行封装，并通过 802.11 无线链路发送。这些 EAP 报文在接入点拆封，然后再使用 RADIUS 协议重新封装，经 UDP/IP 传输到鉴别服务器。尽管 RADIUS 服务器和协议［RFC 2865］并不为 802.11i 所要求，但它们是 802.11i 的事实上的标准组件。最近标准化的 DIAMETER 协议［RFC 3588］很可能在不久的将来替代 RADIUS。

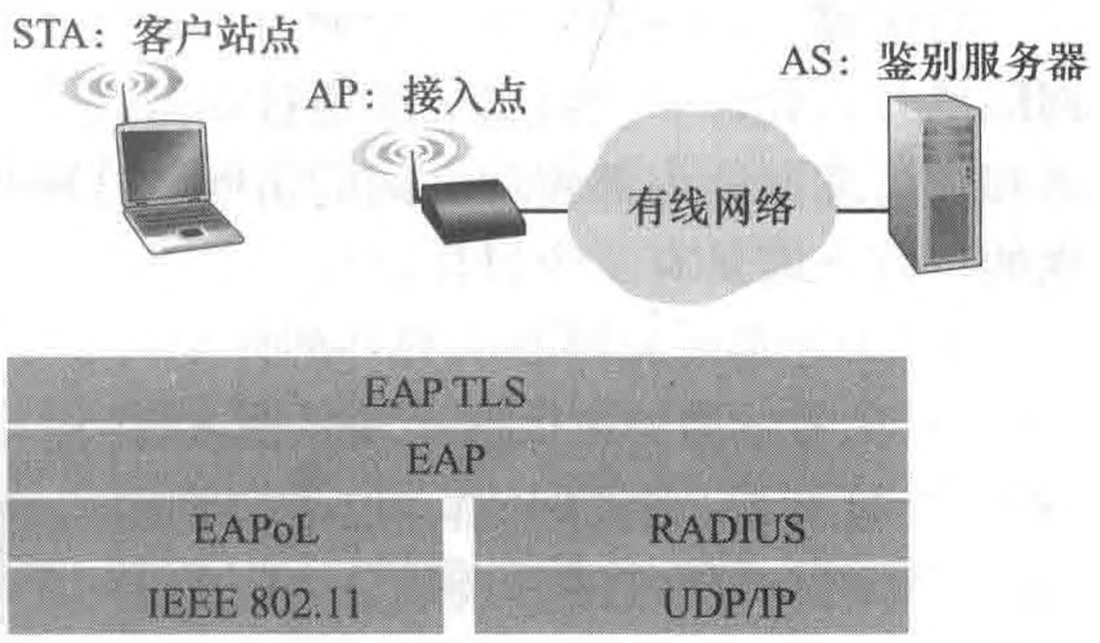

图 8-32　EAP 是一个端到端协议。EAP 报文使用 EAPoL（运行在客户和接入点之间的无线链路上）封装，并使用 RADIUS（运行在接入点和鉴别服务器之间的 UDP/IP 上）

使用 EAP，鉴别服务器能够选择若干方式中的一种来执行鉴别。802.11i 虽未强制一种特殊的鉴别方法，但经常使用 EAP-TLS 鉴别方案［RFC 5216］。EAP-TLS 使用类似于我们在 8.3 节中研究的公钥技术（包括不重数加密和报文摘要），以允许客户和鉴别服务器彼此相互鉴别，并导出为双方所知的一个主密钥。

3）成对主密钥（Pairwise Master Key，PMK）生成。MK 是一个仅为客户和鉴别服务器所知的共享密钥，它们都使用 MK 来生成一个次密钥，即成对主密钥（PMK）。鉴别服务器则向 AP 发送该 PMK。这正是我们所希望达到的目的！客户和 AP 现在具有一个共享的密钥（前面讲过在 WEP 中根本不涉及密钥分发的问题），并彼此相互鉴别。它们此时已经快要能发挥效用了。

4）临时密钥（Temporal Key，TK）生成。使用 PMK，无线客户和 AP 现在能够生成

附加的、将用于通信的密钥。其中的关键是临时密钥，TK 将被用于执行经无线链路向任意远程主机发送数据的链路级的加密。

802.11i 提供了几种加密形式，其中包括基于 AES 的加密方案和 WEP 加密的强化版本。

8.9　运行安全性：防火墙和入侵检测系统

遍及本章我们已经看出，因特网不是一个很安全的地方，即有坏家伙出没，从事着各种各样的破坏活动。给定因特网的不利性质，我们现在考虑一个机构网络和管理它的网络管理员。从网络管理员的角度看，世界可以很明显地分为两个阵营：一部分是好人，他们属于本机构网络，可以用相对不受限制的方式访问该机构网络中的资源；另一部分是坏家伙，他们是其他一些人，访问网络资源时必须经过仔细审查。在许多机构中，从中世纪的城堡到现代公司的建筑物，都有单一的出口/入口，无论好人坏人出入该机构，都需要进行安全检查。在一个城堡中，可以在吊桥的一端的门口执行安全检查；在公司大厦中，这些工作可在安全台完成。在计算机网络中，当通信流量进入/离开网络时要执行安全检查、做记录、丢弃或转发，这些工作都由被称为防火墙、入侵检测系统（IDS）和入侵防止系统（IPS）的运行设备来完成。

8.9.1　防火墙

防火墙（firewall）是一个硬件和软件的结合体，它将一个机构的内部网络与整个因特网隔离开，允许一些数据分组通过而阻止另一些分组通过。防火墙允许网络管理员控制外部和被管理网络内部资源之间的访问，这种控制是通过管理流入和流出这些资源的流量实现的。防火墙具有 3 个目标：

- 从外部到内部和从内部到外部的所有流量都通过防火墙。图 8-33 显示了一个防火墙，位于被管理网络和因特网其余部分之间的边界处。虽然许多大型机构可使用多级防火墙或分布式防火墙[Skoudis 2006]，但在对该网络的单一接入点处设置一个防火墙，如图 8-33 中所示，这使得管理和施加安全访问策略更为容易。

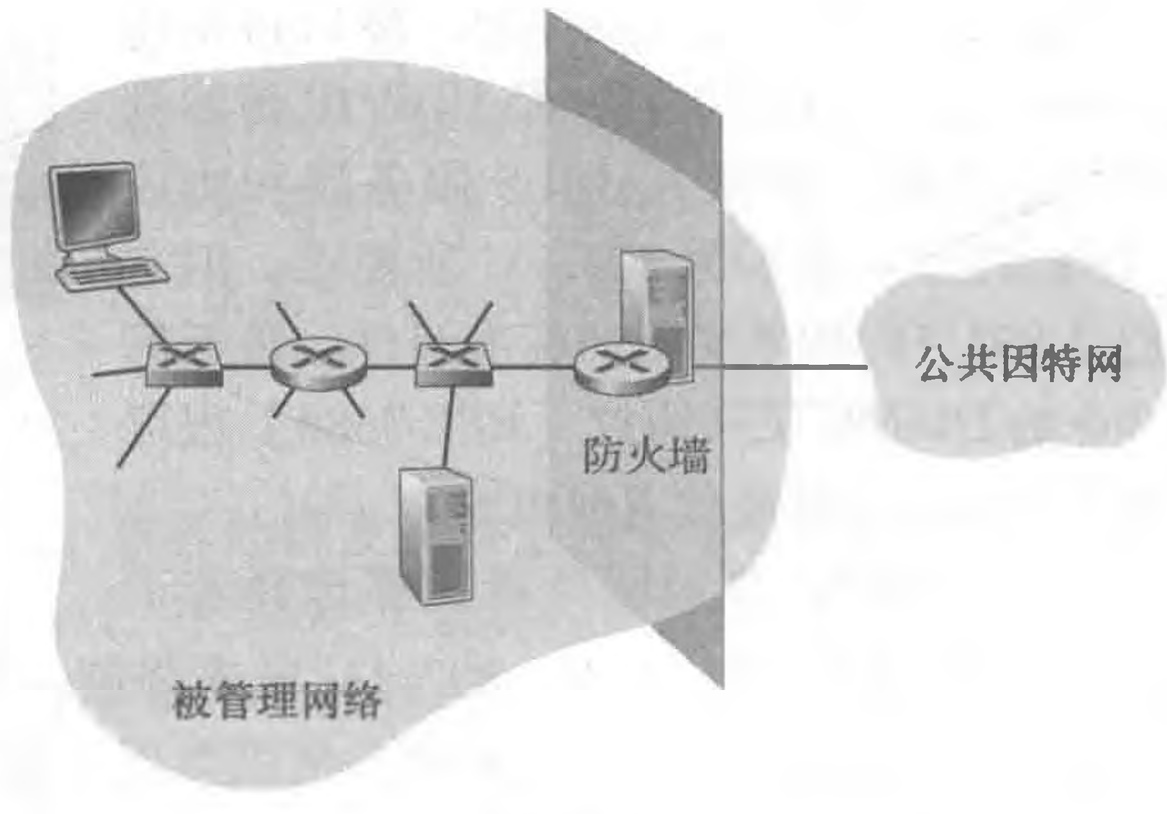

图 8-33　在被管理网络和外部之间放置防火墙

- 仅被授权的流量（由本地安全策略定义）允许通过。随着进入和离开机构网络的所有流量流经防火墙，该防火墙能够限制对授权流量的访问。
- 防火墙自身免于渗透。防火墙自身是一种与网络连接的设备，如果设计或安装不当，将可能危及安全，在这种情况下它仅提供了一种安全的假象（这比根本没有防火墙更糟糕!）。

Cisco 和 Check Point 是当今两个领先的防火墙厂商。你也能够容易地从 Linux 套件使

用 iptables（通常与 Linux 装在一起的公共域软件）产生一个防火墙（分组过滤器）。此外，如第 4 章和第 5 章中所讨论的，防火墙现在经常在路由器中实现并使用 SDN 进行远程控制。

防火墙能够分为 3 类：**传统分组过滤器**（traditional packet filter）、**状态过滤器**（stateful filter）和**应用程序网关**（application gateway）。在下面小节中，我们将依次学习它们。

1. 传统的分组过滤器

如图 8-33 所示，一个机构通常都有一个将其内部网络与其 ISP（并因此与更大的公共因特网相连）相连的网关路由器。所有离开和进入内部网络的流量都要经过这个路由器，而这个路由器正是**分组过滤**（packet filtering）出现的地方。分组过滤器独立地检查每个数据报，然后基于管理员特定的规则决定该数据报应当允许通过还是应当丢弃。过滤决定通常基于下列因素：

- IP 源或目的地址。
- 在 IP 数据报中的协议类型字段：TCP、UDP、ICMP、OSPF 等。
- TCP 或 UDP 的源和目的端口。
- TCP 标志比特：SYN、ACK 等。
- ICMP 报文类型。
- 数据报离开和进入网络的不同规则。
- 对不同路由器接口的不同规则。

网络管理员基于机构的策略配置防火墙。该策略可以考虑用户生产率和带宽使用以及对一个机构的安全性关注。表 8-5 列出了一个机构可能具有的若干可能的策略，以及它们是如何用一个分组过滤器来处理分组的。例如，如果该机构除了允许访问它的公共 Web 服务器外不希望任何入 TCP 连接的话，那么它能够阻挡所有的入 TCP SYN 报文段，但具有目的地端口 80 的 TCP SYN 报文段除外，并且该目的 IP 地址对应于该 Web 服务器。如果该机构不希望它的用户用因特网无线电应用独占访问带宽，那么它能够阻挡所有非关键性 UDP 流量（因为因特网无线电经常是通过 UDP 发送的）。如果该机构不希望它的内部网络被外部绘制结构图（被跟踪路由），那么它能够阻挡所有 ICMP TTL 过期的报文离开该机构的网络。

表 8-5　对于 Web 服务器在 130.207.244.203 的某机构网络 130.207/16，其策略和对应的过滤规则

策略	防火墙设置
无外部 Web 访问	丢弃所有到任何 IP 地址、端口 80 的出分组
无入 TCP 连接，但那些只访问机构公共 Web 服务器的分组除外	丢弃所有到除 130.207.244.203、端口 80 外的任何 IP 地址的入 TCP SYN 分组
防止 Web 无线电占据可用带宽	丢弃所有入 UDP 分组，但 DNS 分组除外
防止你的网络被用于一个 smurf DoS 攻击	丢弃所有去往某“广播”地址（例如 130.207.255.255）的 ICMP ping 分组
防止你的网络被跟踪路由	丢弃所有出 ICMP TTL 过期流量

一条过滤策略能够基于地址和端口号的结合。例如，一台过滤路由器能够转发所有 Telnet 数据报（那些具有端口号 23 的数据报），但那些包括在一个特定的 IP 地址列表中的去往和来自的地址除外。这些策略允许在许可列表上的地址进行 Telnet 连接。不幸的是，

基于外部地址的策略无法对其源地址被哄骗的数据报提供保护。

过滤也可根据 TCP ACK 比特是否设置来进行。如果一个机构要使内部客户连接到外部服务器，却要防止外部客户连接到内部服务器，这个技巧很有效。3.5 节讲过，在每个 TCP 连接中第一个报文段的 ACK 比特都设为 0，而连接中的所有其他报文段的 ACK 比特都设为 1。因此，如果一个机构要阻止外部客户发起到内部服务器的连接，就只需直接过滤进入的所有 ACK 比特设为 0 的报文段。这个策略去除了所有从外部发起的所有 TCP 连接，但是允许内部发起 TCP 连接。

在路由器中使用访问控制列表实现防火墙规则，每个路由器接口有它自己的列表。表 8-6 中显示了对于某机构 222.22/16 的访问控制列表的例子。该访问控制列表适用于将路由器与机构外部 ISP 连接的某个接口。这些规则被应用到通过该接口自上而下传递的每个数据报。前两条规则一起允许内部用户在 Web 上冲浪：第一条规则允许任何具有目的端口 80 的 TCP 分组离开该机构网络；第二条规则允许任何具有源端口 80 且 ACK 比特置位的 TCP 分组进入该机构网络。注意到如果一个外部源试图与一台内部主机建立一条 TCP 连接，该连接将被阻挡，即使该源或目的端口为 80。接下来的两条规则一起允许 DNS 分组进入和离开该机构网络。总而言之，这种限制性相当强的访问控制列表阻挡所有流量，但由该机构内发起的 Web 流量和 DNS 流量除外。[CERT Filtering 2012] 提供了一个推荐的端口/协议分组过滤的列表，以避免在现有网络应用中的一些周知的安全性漏洞。

表 8-6　用于某路由器接口的访问控制列表

动作	源地址	目的地址	协议	源端口	目的端口	标志比特
允许	222.22/16	222.22/16 的外部	TCP	>1023	80	任意
允许	222.22/16 的外部	222.22/16	TCP	80	>1023	ACK
允许	222.22/16	222.22/16 的外部	UDP	>1023	53	—
允许	222.22/16 的外部	222.22/16	UDP	53	>1023	—
拒绝	全部	全部	全部	全部	全部	全部

2. 状态分组过滤器

在传统的分组过滤器中，根据每个分组分离地做出过滤决定。状态过滤器实际地跟踪 TCP 连接，并使用这种知识作出过滤决定。

为了理解状态过滤器，我们来重新审视表 8-6 中的访问控制列表。尽管限制性相当强，表 8-6 中的访问控制列表仍然允许来自外部的 ACK = 1 且源端口为 80 的任何分组到达，通过该过滤器。这样的分组能够被试图用异常分组来崩溃内部系统、执行拒绝服务攻击或绘制内部网络的攻击者使用。幼稚的解决方案是也阻挡 TCP ACK 分组，但是这样的方法将妨碍机构内部的用户在 Web 上冲浪。

状态过滤器通过用一张连接表来跟踪所有进行中的 TCP 连接来解决这个问题。这种方法是可能的：因为防火墙能够通过观察三次握手（SYN、SYNACK 和 ACK）来观察一条新连接的开始；而且当它看到该连接的一个 FIN 分组时，它能够观察该连接的结束。当防火墙经过比如说 60 秒还没有看到该连接的任何活动性，它也能够（保守地）假设该连接结束了。某防火墙的一张连接表例子显示在表 8-7 中。这张连接表指示了当前有 3 条进行中的 TCP 连接，所有的连接都是从该机构内部发起的。此外，该状态过滤器在它的访问控制列表中包括了一个新栏，即“核对连接”，如表 8-8 中所示。注意到表 8-8

与表8-6中的访问控制列表相同，只是此时它指示应当核对其中两条规则所对应的连接。

表8-7　用于状态过滤器的连接表

源地址	目的地址	源端口	目的端口
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

表8-8　用于状态过滤器的访问控制列表

动作	源地址	目的地址	协议	源端口	目的端口	标志比特	核对连接
允许	222.22/16	222.22/16 的外部	TCP	>1023	80	任意	
允许	222.22/16 的外部	222.22/16	TCP	80	>1023	ACK	X
允许	222.22/16	222.22/16 的外部	UDP	>1023	53	—	
允许	222.22/16 的外部	222.22/16	UDP	53	>1023	—	X
拒绝	全部	全部	全部	全部	全部	全部	

我们浏览某些例子来看看连接表和扩展的访问控制列表是如何联手工作的。假设一个攻击者通过发送一个具有TCP源端口80和ACK标志置位的数据报，试图向机构网络中发送一个异常分组。进一步假设该分组具有源端口号12543和源IP地址150.23.23.155。当这个分组到防火墙时，防火墙核对表8-8中的访问控制列表，该表指出在允许该分组进入机构网络之前还必须核对连接表。该防火墙正确地核对了连接表，发现这个分组不是某进行中的TCP连接的一部分，从而拒绝了该分组。举第二个例子，假设一个内部的用户要在外部Web站点冲浪。因为该用户首先发送了一个TCP SYN报文段，所以该用户的TCP连接在连接表中有了记录。当Web服务器发送回分组（ACK比特进行了必要的设置），该防火墙核对了连接表并明白一条对应的连接在进行中。防火墙因此将让这些分组通过，从而不会干扰内部用户的Web冲浪活动。

3. 应用程序网关

在上面的例子中，我们已经看到了分组级过滤使得一个机构可以根据IP的内容和TCP/UDP首部（包括IP地址、端口号和ACK比特）执行粗粒度过滤。但是如果一个机构仅为一个内部用户的受限集合（与IP地址情况正相反）提供Telnet服务该怎样做呢？如果该机构要这些特权用户在允许创建向外部的Telnet会话之前首先鉴别他们自己该怎样做呢？这些任务都超出了传统过滤器和状态过滤器的能力。的确，有关内部用户的身份信息是应用层数据，并不包括在IP/TCP/UDP首部中。

为了得到更高水平的安全性，防火墙必须把分组过滤器和应用程序网关结合起来。应用程序网关除了看IP/TCP/UDP首部外，还基于应用数据来做策略决定。一个**应用程序网关**（application gateway）是一个应用程序特定的服务器，所有应用程序数据（入和出的）都必须通过它。多个应用程序网关可以在同一主机上运行，但是每一个网关都是有自己的进程的单独服务器。

为了更深入地了解应用程序网关，我们来设计一个防火墙，它只允许内部客户的受限集合向外Telnet，不允许任何外部客户向内Telnet。这一策略可通过将分组过滤（在一台

路由器上）和一个 Telnet 应用程序网关结合起来实现，如图 8-34 所示。路由器的过滤器配置为阻塞所有 Telnet 连接，但从该应用程序网关 IP 地址发起的连接除外。这样的过滤器配置迫使所有向外的 Telnet连接都通过应用程序网关。现在考虑一个要向外部 Telnet 的内部用户。这个用户必须首先和应用程序网关建立一个 Telnet 会话。在网关（网关监听进入的 Telnet 会话）上一直运行的应用程序提示用户输入用户 ID 和口令。当这个用户提供这些信息时，应用程序网关检查这个用户是否得到许可向外 Telnet。如果没有，网关则中止这个内部用户向该网关发起的 Telnet 连接。如果该用户得到许可，则这个网关：①提示用户输入它所要连接的外部主机的主机名；②在这个网关和某外部主机之间建立一个 Telnet 会话；③将从这个用户到达的所有数据中继到该外部主机，并且把来自这个外部主机的所有数据都中继给这个用户。所以，该 Telnet 应用程序网关不仅执行用户授权，而且同时充当一个 Telnet 服务器和一个 Telnet 客户，在这个用户和该远程 Telnet 服务器之间中继信息。注意到过滤器因为该网关发起向外部的 Telnet 连接，将允许执行步骤②。

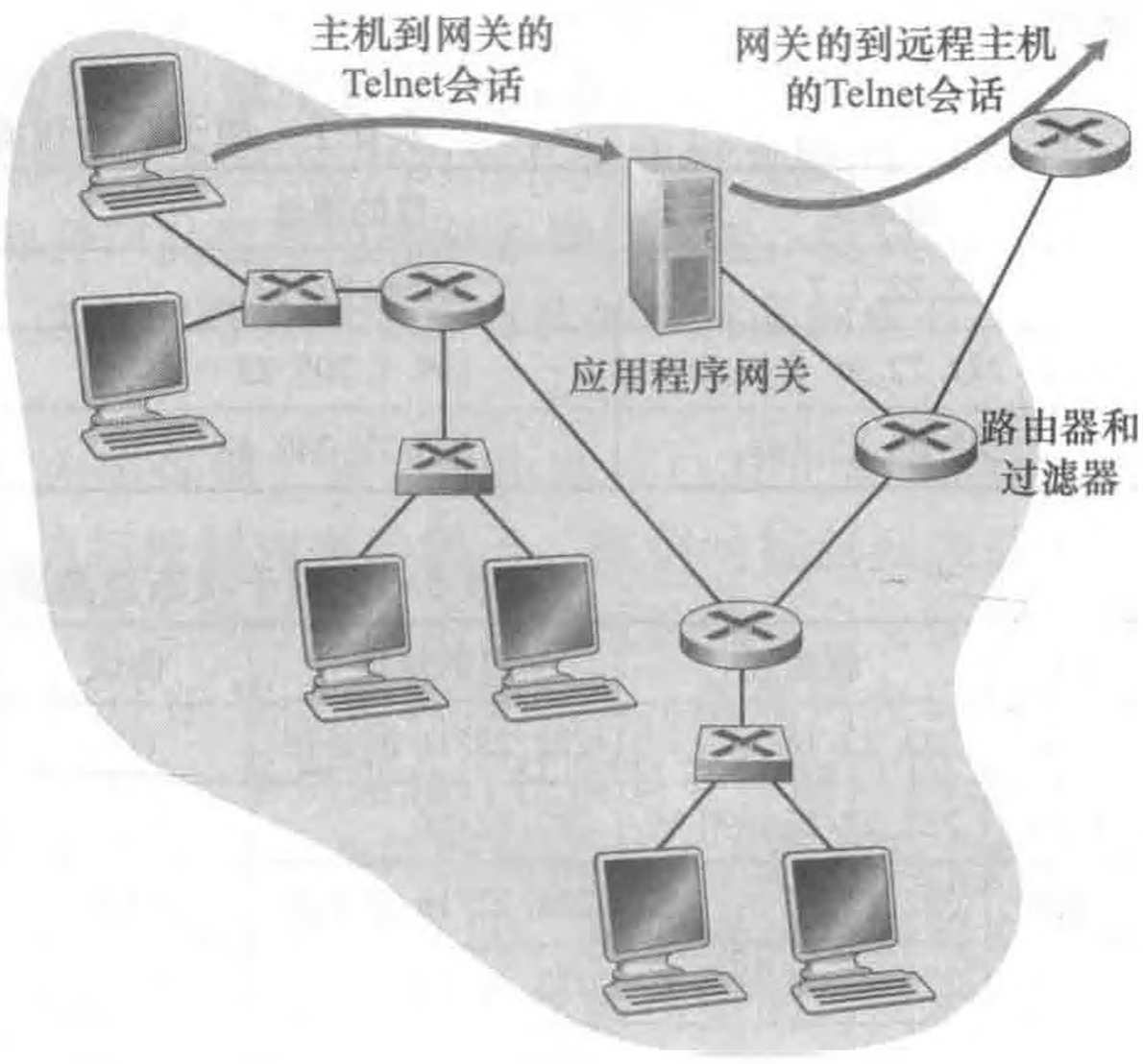

图 8-34　由应用程序网关和过滤器组成的防火墙

历史事件

匿名与隐私

假定你要访问一个有争议的 Web 网站（例如某政治活动家的网站），并且你：①不想向该 Web 网站透漏你的 IP 地址；②不想要你的本地 ISP（它可能是你住家或办公室的 ISP）知道你正在访问该站点；③不想要你的本地 ISP 看到你正在与该站点交换的数据。如果你使用传统的方法直接与该 Web 站点连接而没有任何加密，你无法实现这三个诉求。即使你使用 SSL，你也无法实现前两个诉求：你的源 IP 地址呈现在你发送给 Web 网站的每个数据报中；你发送的每个分组的目的地址能够容易地被你本地 ISP 嗅探到。

为了获得隐私和匿名，你能够使用如图 8-35 所示的一种可信代理服务器和 SSL 的组合。利用这种方法，你首先与可信代理建立一条 SSL 连接。然后你在该 SSL 连接中向所希望站点的网页发送一个 HTTP 请求。当代理接收到该 SSL 加密的 HTTP 请求，它解密请求并向 Web 站点转发该明文 HTTP 请求。接下来 Web 站点响应该代理，该代理经过 SSL 再向你转发该响应。因为该 Web 站点仅看到代理的 IP 地址，并非你的客户 IP 地址，你的确获得了对该 Web 站点的匿名访问。并且因为你和代理之间的所有流量均被加密，你的本地 ISP 无法通过对你访问的站点做日志和记录你交换的数据来侵犯你的隐私。今天许多公司（例如 proxify. com）提供了这种代理服务。

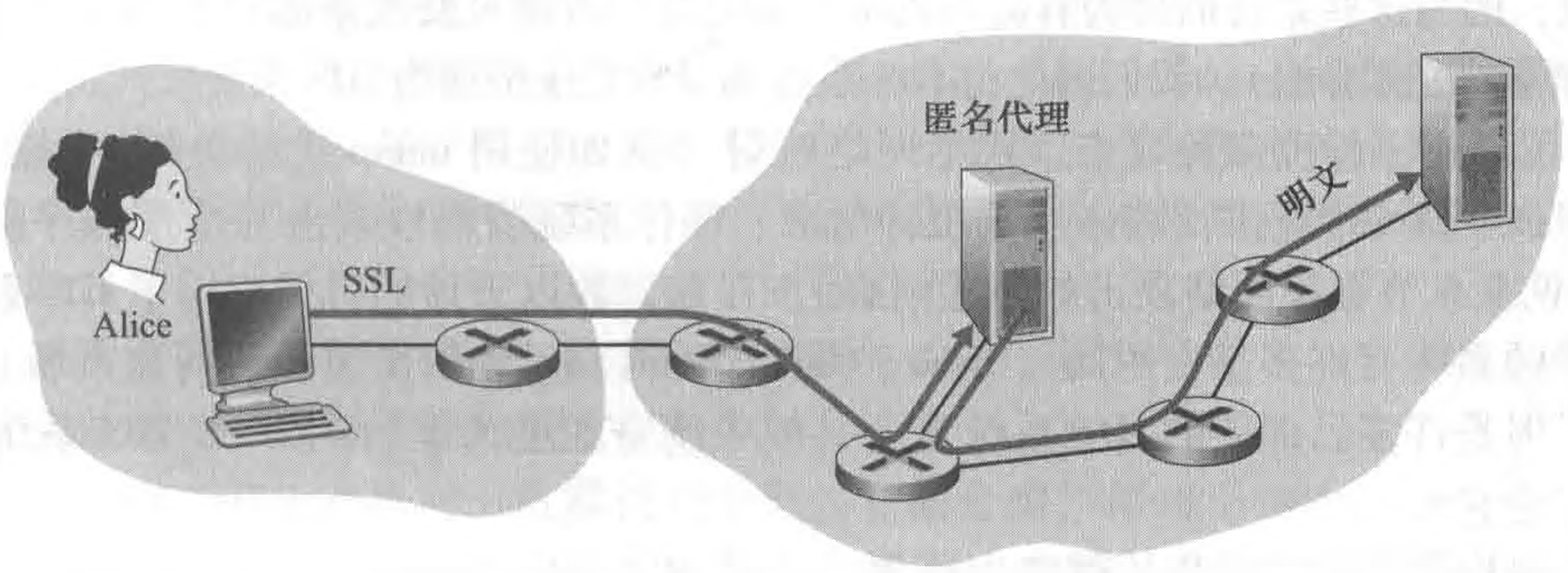

图 8-35 利用代理提供匿名和隐私

当然，在这个解决方案中，你的代理知道一切：它知道你的 IP 地址和你正在冲浪的站点的 IP 地址；并且它能够看到你与该 Web 站点之间以明文形式交换的所有流量。因此，这种解决方案的好坏取决于该代理的可信度。由 TOR 匿名和隐私服务所采用的一种更为健壮的方法是，让你的流量路由通过一系列“不串通”的代理服务器［TOR 2012］。特别是，TOR 允许独立的个体向其代理池贡献代理。当某用户使用 TOR 与一个服务器连接，TOR 随机地（从它的代理池）选择一条三个代理构成的链，并通过该链在客户和服务器之间路由所有流量。以这种方式，假设这些代理并不串通，无人知道在你的 IP 地址和目标 Web 站点之间发生的通信。此外，尽管在最后的代理和服务器之间发送明文，但这个最后代理并不知道哪个 IP 地址正在发送和接收明文。

内部网络通常有多个应用程序网关，例如 Telnet、HTTP、FTP 和电子邮件网关。事实上，一个机构的邮件服务器（见 2.3 节）和 Web 高速缓存都是应用程序网关。

应用程序网关也有其缺陷。首先，每一个应用程序都需要一个不同的应用程序网关。第二，因为所有数据都由网关转发，付出的性能负担较重。当多个用户或应用程序使用同一个网关计算机时，这成为特别值得关注的问题。最后，当用户发起一个请求时，客户软件必须知道如何联系这个网关，并且必须告诉应用程序网关如何连接到哪个外部服务器。

8.9.2 入侵检测系统

我们刚刚看到，当决定让哪个分组通过防火墙时，分组过滤器（传统的和状态的）检查 IP、TCP、UDP 和 ICMP 首部字段。然而，为了检测多种攻击类型，我们需要执行**深度分组检查**（deep packet inspection），即查看首部字段以外部分，深入查看分组携带的实际应用数据。如我们在 8.9.1 节所见，应用程序网关经常做深度分组检查。而一个应用程序网关仅对一种特定的应用程序执行这种检查。

显然，这为另一种设备提供了商机，即一种不仅能够检查所有通过它传递的分组的首部（类似于分组过滤器），而且能执行深度分组检查（与分组过滤器不同）的设备。当这样的设备观察到一个可疑的分组时，或一系列可疑的分组时，它能够防止这些分组进入该机构网络。或者仅仅是因为觉得该活动可疑，该设备虽说能够让这些分组通过，但要向网络管理员发出告警，网络管理员然后密切关注该流量并采取适当的行动。当观察到潜在恶意流量时能产生告警的设备称为**入侵检测系统**（Intrusion Detection System，IDS）。滤除可疑流量的设备称为**入侵防止系统**（Intrusion Prevention System，IPS）。在本节中我们一起学习 IDS 和 IPS 这

两种系统，因为这些系统的最为有趣的技术方面是它们检测可疑流量的原理（而不是它们是否发送告警或丢弃分组）。我们因此将 IDS 系统和 IPS 系统统称为 IDS 系统。

IDS 能够用于检测多种攻击，包括网络映射（例如使用 nmap 进行分析）、端口扫描、TCP 栈扫描、DoS 带宽洪泛攻击、蠕虫和病毒、操作系统脆弱性攻击和应用程序脆弱性攻击。（参见 1.6 节有关网络攻击的概述内容。）目前，数以千计的机构应用了 IDS 系统。这些已部署的系统有许多是专用的，Cisco、Check Point 和其他安全装备厂商在市场上销售这些系统。但是许多已部署的 IDS 系统是公共域系统，如极为流行的 Snort IDS 系统（我们将简要讨论它）。

一个机构可能在它的机构网络中部署一个或多个 IDS 传感器。图 8-36 显示了一个具有 3 个 IDS 传感器的机构。当部署了多个传感器时，它们通常共同工作，向一个中心 IDS 处理器发送有关可疑流量活动的信息，中心处理器收集并综合这些信息，当认为适合时向网络管理员发送告警。在图 8-36 中，该机构将其网络划分为两个区域：一个高度安全区域，由分组过滤器和应用程序网关保护，并且由 IDS 系统监视；一个较低度安全区域（称之为**非军事区**（DeMilitarized Zone，DMZ）），该区域仅由分组过滤器保护，但也由 IDS 系统监视。注意到 DMZ 包括了该机构需要与外部通信的服务器，如它的公共 Web 服务器和它的权威 DNS 服务器。

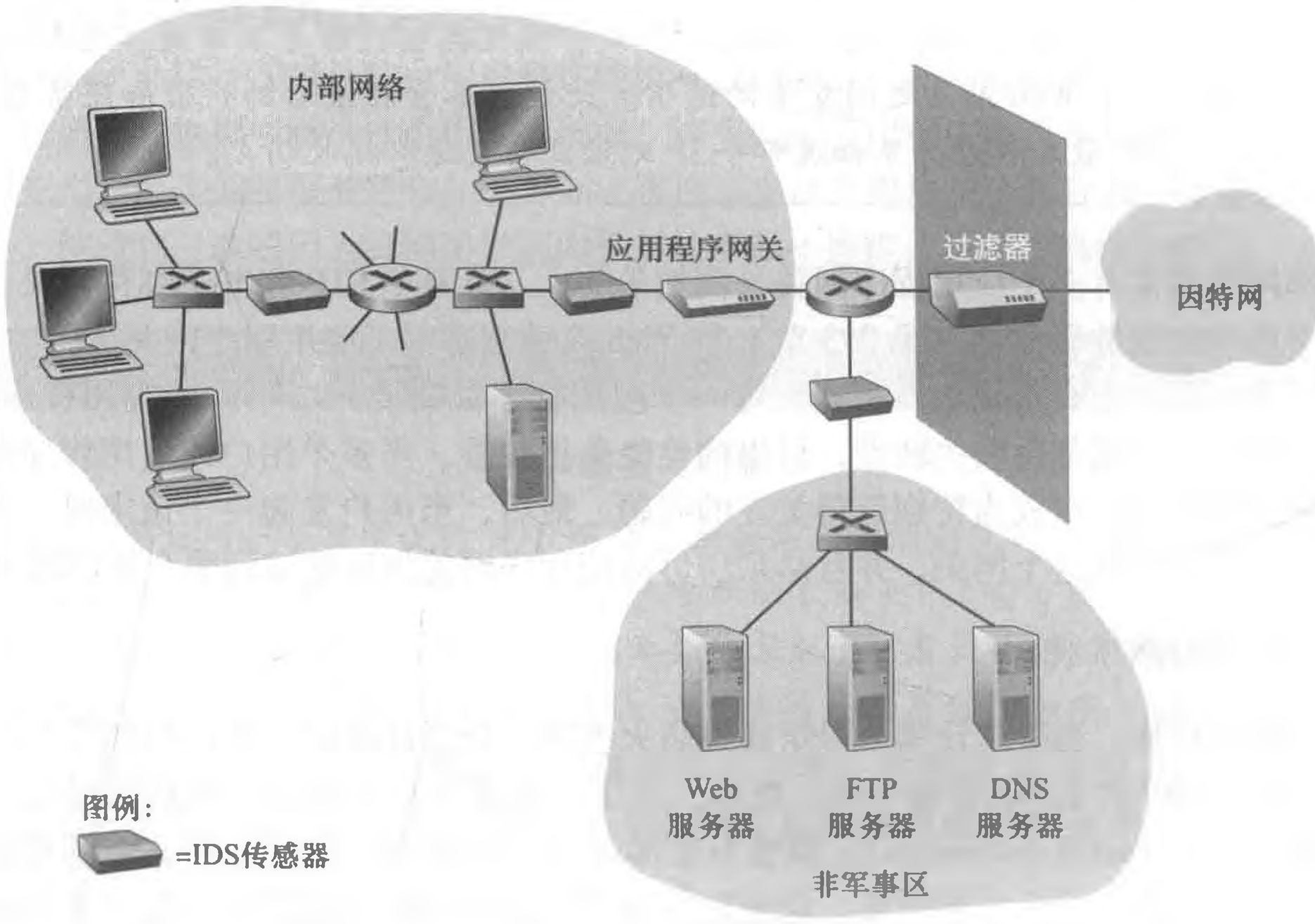

图 8-36　部署一个过滤器、一个应用程序网关和多个 IDS 传感器的机构

此时你也许想知道，为什么使用多个 IDS 传感器？为什么在图 8-36 中不只是在分组过滤器后面放置一个 IDS 传感器（或者甚至与分组过滤器综合）？我们将很快看到，IDS 不仅需要做深度分组检查，而且必须要将每个过往的分组与数以万计的“特征（signature）”进行比较；这可能导致极大的处理量，特别是如果机构从因特网接收每秒数十亿比特的流量时更是如此。将 IDS 传感器进一步向下游放置，每个传感器仅看到该机构流量的一部分，维护能够更容易。无论如何，目前有许多高性能 IDS 和 IPS 系统可供使用，许

多机构实际上能够在靠近其接入路由器附近只使用一个传感器。

IDS 系统大致可分类为**基于特征的系统**（signature-based system）或**基于异常的系统**（anomaly-based system）。基于特征的 IDS 维护了一个范围广泛的攻击特征数据库。每个特征是与一个入侵活动相关联的规则集。一个特征可能只是有关单个分组的特性列表（例如源和目的端口号、协议类型和在分组载荷中的特定比特串），或者可能与一系列分组有关。这些特征通常由研究了已知攻击、技艺熟练的网络安全工程师生成。一个机构的网络管理员能够定制这些特征或者将其加进数据库中。

从运行上讲，基于特征的 IDS 嗅探每个通过它的分组，将每个嗅探的分组与数据库中的特征进行比较。如果某分组（或分组序列）与数据库中的一个特征相匹配，IDS 产生一个告警。该告警能够发送一个电子邮件报文给网络管理员，能够发送给网络管理系统，或只是做日志以供以后检查。

尽管基于特征的 IDS 系统部署广泛，但仍具有一些限制。更重要的是，它们要求根据以前的攻击知识来产生一个准确的特征。换言之，基于特征的 IDS 对不得不记录的新攻击完全缺乏判断力。另一个缺点是，即使与一个特征匹配，它也可能不是一个攻击的结果，因此产生了一个虚假告警。最后，因为每个分组必须与范围广泛的特征集合相比较，IDS 可能处于处理过载状态并因此难以检测出许多恶意分组。

当基于异常的 IDS 观察正常运行的流量时，它会生成一个流量概况文件。然后，它寻找统计上不寻常的分组流，例如，ICMP 分组不寻常的百分比，或端口扫描和 ping 掠过导致指数性突然增长。基于异常的 IDS 系统最大的特点是它们不依赖现有攻击的以前知识。在另一方面，区分正常流量和统计异常流量是一个极具挑战性的问题。迄今为止，大多数部署的 IDS 主要是基于特征的，尽管某些 IDS 包括了某些基于异常的特性。

Snort

Snort 是一种公共域开放源码的 IDS，现有部署达几十万［Snort 2012；Koziol 2003］。它能够运行在 Linux、UNIX 和 Windows 平台上。它使用了通用的嗅探接口 libpcap，Wireshark 和许多其他分组嗅探器也使用了 libpcap。它能够轻松地处理 100Mbps 的流量；安装在千兆比特/秒流量速率下工作，需要多个 Snort 传感器。

为了对 Snort 有一些认识，我们来看一个 Snort 特征的例子：

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any
(msg:"ICMP PING NMAP"; dsize: 0; itype: 8;)
```

这个特征由从外部（$EXTERNAL_NET）进入机构网络（$HOME_NET）的任何 ICMP 分组所匹配，其类型是 8（ICMP ping）并且具有空负载（dsize = 0）。因为 nmap（参见 1.6 节）用这些特定的特征产生这些 ping 分组，所以设计出该特征来检测 nmap 的 ping 扫描。当某分组匹配该特征时，Snort 产生一个包括“ICMP PING NAMP”报文的告警。

也许关于 Snort 印象最为深刻的是巨大的用户社区和维护其特征数据库的安全专家。通常在一个新攻击出现后的几个小时内，Snort 社区就编写并发布一个攻击特征，然后它就能被分布在全世界的数十万 Snort 部署者下载。此外，使用 Snort 特征的语法，网络管理员能够根据他们自己的机构需求，通过修改现有的特征或通过创建全新的特征来裁剪某个特征。

8.10 小结

在本章中，我们考察了秘密情人 Bob 和 Alice 能够用于安全通信的各种机制。我们看

到 Bob 和 Alice 对下列因素感兴趣：机密性（因此只有他们才能理解传输的报文内容）、端点鉴别（因此他们确信正在与对方交谈）和报文完整性（因此他们确信在传输过程中他们的报文未被篡改）。当然，安全通信的需求并不限于秘密情人。的确，我们在 8.5 ~ 8.8 节中看到，可以在网络体系结构中的各个层次使用安全性，使之免受采用各种各样攻击手段的坏家伙们的侵扰。

本章前面部分给出了安全通信所依赖的各种原理。在 8.2 节中，我们涉及了加密和解密数据的密码技术，包括对称密钥密码和公开密钥密码。作为今天网络中两种重要的密码技术的特定的学习案例，我们考察了 DES 和 RSA。

在 8.3 节中，我们研究了提供报文完整性的两种方法：报文鉴别码（MAC）和数字签名。这两种方法有一些共同之处。它们都使用了密码散列函数，这两种技术都使我们能够验证报文的源以及报文自身的完整性。一个重要的差异是 MAC 不依赖于加密，而数字签名要求公钥基础设施。如我们在 8.5 ~ 8.8 节所见，这两种技术广泛在实际中都得到了广泛应用。此外，数字签名用于生成数字证书，数字证书对于证实公钥的合法性是重要的。在 8.4 节中，我们考察了端点鉴别并引入了不重数以防御重放攻击。

在 8.5 ~ 8.8 节中，我们研究了几种在实践中得到广泛使用的安全性网络协议。我们看到了对称密钥密码在 PGP、SSL、IPsec 和无线安全性中的核心地位。我们看到了公开密钥密码对 PGP 和 SSL 是至关重要的。我们看到 PGP 使用数字签名而 SSL 和 IPsec 使用 MAC 来保证报文完整性。在目前理解了密码学的基本原理以及学习了这些原理的实际应用方法之后，你现在已经有能力设计你自己的安全网络协议了！

利用 8.2 ~ 8.4 节所包含的技术，Bob 和 Alice 就能够安全通信了。（只希望他们是学习了这些材料的网络专业学生，因此能够使他们的约会不会被 Trudy 发现！）而机密性仅是整个网络安全的一小部分。如我们在 8.9 节中所学习，现在网络安全的焦点越来越多地关注网络基础设施的安全性，以防止“坏家伙”的潜在猛烈攻击。在本章的后面部分，我们因此学习了防火墙和 IDS 系统，它们检查进入和离开一个机构网络的分组。

本章已经涉及了许多基础性问题，同时关注了现代网络安全性中最为重要的主题。希望深入钻研的读者最好研究本章中引用的文献。特别是，我们推荐以下读物：关于攻击和运行安全性的［Skoudis 2006］，关于密码学及其如何应用于网络安全的［Kaufman 1995］，有深度且可读性强的关于 SSL 处理的［Rescorla 2001］，以及透彻地讨论 802.11 安全性且包括对 WEP 及其缺陷的深入研究的［Edney 2003］。

课后习题和问题

复习题

8.1 节

R1. 报文机密性和报文完整性之间的区别是什么？你能具有机密性而没有完整性吗？你能具有完整性而没有机密性吗？证实你的答案。

R2. 因特网实体（路由器、交换机、DNS 服务器、Web 服务器、用户端系统等）经常需要安全通信。给出三个特定的因特网实体对的例子，它们要安全通信。

8.2 节

R3. 从服务的角度，对称密钥系统和公开密钥系统之间一个重要的差异是什么？

R4. 假定某入侵者拥有一个加密报文以及该报文的解密版本。这个入侵者能够发起已知密文攻击、已知