第15章　电子商务的安全问题

在本章中，我们将介绍电子商务安全的重要性。我们将分析所有可能对信息最感兴趣的人，以及他们为了得到它所有可能采用的手段，还将讨论涉及创建一套能够避免这类问题的策略的原则，此外，还有一些用于保护网站安全的技术，包括加密、身份验证和跟踪。

在本章中，我们将主要介绍以下内容：

- 信息的重要程度
- 安全威胁
- 建立一套安全策略
- 易用性、性能、成本和安全性
- 身份验证原则
- 在站点应用身份验证
- 加密技术基础
- 私有密钥加密
- 公有密钥加密
- 数字签名
- 数字证书
- 安全的Web服务器
- 审计与日志记录
- 防火墙
- 备份数据
- 自然环境的安全性

15.1　信息的重要程度

考虑到安全的时候，首先要评估的是所保护信息的重要性。应该既要考虑这些信息对你的重要性，又考虑它对潜在入侵者的重要性。

人们可能会认为，所有网站时时刻刻都要求有最高级别的安全保护，但是保护措施的实施需要成本。在决定要对安全保护提供多少人力和物力之前，必须判定信息价值。

保存在一个计算机业余爱好者、企业、银行和军事组织的计算机中的信息的价值是不同的。同样，一个入侵者要窃取这些信息可能要经过的途径也是不同的。机器中的内容对恶意访问者有多大的吸引力呢？

计算机业余爱好者很可能只有有限的时间来了解或提高他们系统的安全性。除了对本人的价值之外，保存在他们机器上的信息，对其他任何人的价值可能非常有限，因此被别人攻击的可能性也就非常小。同时，攻击者付出的努力也会是有限的。但是，所有网络计算机用户都应

该采取明智的防范措施。即使计算机上没有什么能让别人感兴趣的东西，也可能被攻击者用作攻击别人的系统的跳板，或者作为复制病毒和蠕虫的载体。

很显然，军事用途的计算机对个人和外国政府来说都是政击的目标。由于攻击者可能拥有丰富的资源，所以在人员和其他资源方面进行充分投资是明智的，以便确保在该领域内采取所有实际防范措施。

对于一个商业网站的负责人来讲，需要考虑介于上述两种极端情况之间的黑客攻击，因此投入的资源和努力也就应该介于二者之间。

15.2 安全威胁

网站上存在什么样的危险？网站之外有什么威胁？我们在第14章中已经讨论了电子商务交易的一些威胁。

根据网站实际情况，安全威胁可能包括：

- 机密数据的泄露
- 数据丢失和数据损坏
- 数据修改
- 拒绝服务
- 软件错误
- 否认

我们将逐一介绍每个安全威胁。

15.2.1 机密数据的泄露

存储在计算机上的数据，或者从计算机发送或接收的数据都可能是机密的。它可能仅仅是一些人要看的信息，例如批发价清单。也可能是一个顾客提供的机密信息，例如密码、联系方法及信用卡号码。

不要将不希望被别人看到的信息存储到Web服务器上。Web服务器不应该是存放机密信息的地方。如果要将你的薪水记录等机密信息放到计算机上，最好使用非服务器计算机。Web服务器本身就是公众访问的机器，应该只包含需要提供给公众的信息，或者最近从公众那里收集到的信息。

要减少数据泄露的危险，必须限制访问信息的方法以及能够访问这些信息的用户。这就要求在设计的时候要考虑安全问题，正确配置服务器与软件，编程时要小心谨慎，进行完全的测试，从Web服务器上删除不必要的服务，并且要求身份验证。

小心谨慎地设计、配置、编码和测试可以减少成功恶意攻击的危险，同样重要的是，可以减少由于软件错误导致的信息意外泄露。

我们还需要从服务器上删除不必要的服务，这样可以减少潜在弱点的数量。正在运行的每个服务都可能存在弱点。每个服务必须经常更新以保证那些众所周知的弱点不再呈现出来。没有使用的服务可能更加危险。如果从来没有使用过命令`rcp`，服务器为什么已经安装了这个服务呢？即使目前使用了`rcp`命令，也应该删除它，而使用`scp`（secure copy，安全复制）命令。

如果告诉安装程序机器是一台网络主机，主Linux分区和Windows NT会安装许多不必要的服务，应该删除它们。

身份验证的意思是请人们证明他们的身份。当系统知道哪个用户正在请求的时候，它可以判断这个人是否有访问权限。身份验证的具体实现可以有许多不同的方法，但是通常只使用其中两种形式——密码和数字签名。我们在后面将更详细地讨论它们。

CD Universe就是一个很好的例子，它因为机密信息的泄露而导致经济和名誉的双重损失。

1999年下半年，据传闻一个自称是Maxus的入侵者联系了CD Universe，声称已经从他们的网站上窃取了300 000个信用卡号码。他以销毁这些号码为条件要求100 000美元。他们拒绝了，然后发现自己处于非常尴尬的境地，他们上了主要报纸的头条，因为Maxus将这些号码发放出去供别人滥用了。

当数据在网络上传输的时候，也存在泄露的危险。尽管TCP/IP网络有许多很好的性能，这些使TCP/IP成为将不同网络连接成互联网的实际上的标准。但是安全性能不是这些很好性能之一。TCP/IP将数据分成信息包，然后将这些信息包从一台机器向另一台机器发送直到终点。这意味着数据在发送的路途中经过了许多不同的机器，如图15-1所示。数据途经这些机器中的任何一台时，这台机器都有可能看到数据。

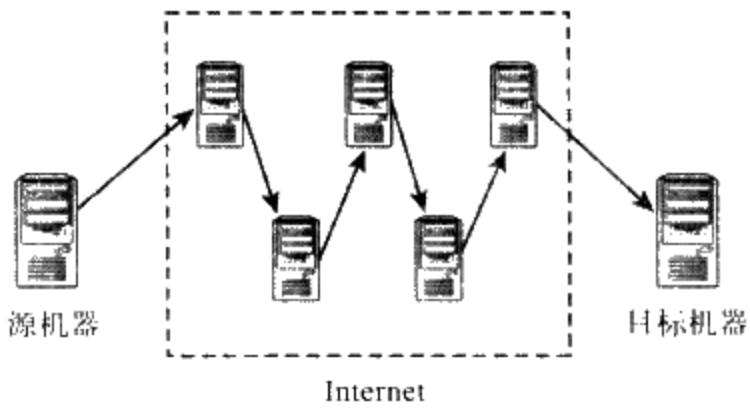

图15-1　通过互联网传输信息，将使信息途经一些并不安全的主机

要查看数据发送到特定机器所途径的路由，可以使用命令traceroute（在UNIX机器上）。该命令给出数据到达目的主机所经历机器的地址。对本国内的目的主机，数据可能经过10台不同的机器。对于一台国际性的目标主机，中间可能经过了多于20台的机器。如果一个公司的网络大而复杂，数据可能甚至在离开办公楼之前就要经过5台机器。

要保护机密信息，可以在将它们通过Internet发送之前进行加密，然后再在另一端解密。通常，Web服务器会使用Secure Sockets Layer（SSL，加密套接字层），它由Netscape开发，用于数据在Web服务器和浏览器之间传输时的加密和解密。这是一个成本低、使用简单的安全传输方法。但是因为服务器需要加密数据而不是简单地发送和接受数据，该机器可以容纳的每秒访问量可能会急剧下降。

15.2.2　数据丢失和数据破坏

对我们来说，数据丢失可能比数据泄露的损失更大。如果已经耗费了数月时间构建了网站，同时又收集了一些用户数据的订单，丢失所有这些信息对时间、声誉和金钱将是多大的损失！如果没有任何数据备份，就必须从头开始匆匆忙忙地重写网站。还可能会遇到顾客或客户抱怨他们还没有收到所订购的商品。

入侵者可能会进入系统，格式化硬盘。粗心的程序员或管理员更有可能不小心删除一些东西，而我们几乎肯定会偶尔损失一个磁盘。硬盘每分钟旋转几千次，偶尔，它们也会出现问题。莫菲法则告诉我们失去的东西是最重要的东西，尤其是很久没有备份以后。可以采取各种措施

以减少数据丢失。加强服务器的安全以防止入侵者。尽量减少可以访问机器的职员人数。只雇佣有能力、细心的人们。购买高质量的硬盘驱动器。使用廉价冗余磁盘阵列（RAID）以便多个驱动器可以像一个更快、更可靠的驱动器一样工作。

无论是什么原因，对数据丢失只有一种真正的保护措施：备份。备份数据不是火箭科技。相反，它是一种枯燥无味的技术，而且我们并不希望使用它，但是它却是至关重要的。请确认是否有规律地备份数据，确认已经测试了备份过程，确保它可以恢复数据。确认备份数据远离计算机。尽管机房被烧毁或者遭遇其他灾害的可能性很小，但是将备份数据与网站分开总是一个相当廉价的保险策略。

15.2.3　数据修改

尽管数据丢失可能具有破坏性，而数据修改则可能更糟。如果一个人得到系统访问权并修改文件将是什么样的情况呢？尽管大规模删除可能会被管理员注意到，也可以从备份恢复，但是要多长时间才能注意到这些数据修改呢？

文件修改可能包括对数据文件的修改和对可执行文件的修改。一个入侵者修改一个数据文件的动机可能是要涂改网站或者获取非法利益。而使用旧版本的可执行文件代替一个新版本的可执行文件，可能给入侵者提供一个网站秘密后门，以便将来访问或获取更高系统权限。

可以通过计算一个签名来防止数据在网络传输过程中被人修改。当然，这不能阻止别人修改数据，但是如果文件到达的时候，接收者检查签名仍然匹配，那么他就知道文件是否被人修改。如果数据经过加密以防止未授权用户浏览，使用签名还使得它很难在传输途中在没有监测的情况下被修改。

保护服务器上的文件以防修改要求我们在操作系统中应用文件权限，以防止未授权的访问。使用文件权限，系统可以授权用户使用系统，但不是给用户修改系统文件和其他用户文件的权限。Windows 95、Windows 98和ME都缺少基本的权限机制，因此这也就是它们不适合作服务器操作系统的原因之一。

检测修改可能是很困难的。如果在某种程度上意识到系统的安全已经遭到破坏，那么如何知道重要的文件是否已经被人修改了呢？有一些文件，例如保存在数据库中的数据文件，它会在一段时间后就被更新。许多其他文件则从安装起就保持原样，如果没有专门对它们进行升级的话。修改程序和数据可能是阴险的，但是尽管怀疑发生了修改，程序可重新安装，但是我们很可能不知道哪个版本的数据是“干净的”。

文件的完整性评估软件（例如，Tripwire）记录了安全状态下的重要文件信息，它很可能是在安装之后立即记录的，而这些信息可以在过一段时间后用来验证文件是否已经被修改。可以从下列网址下载其商业版本或者有条件的免费版本：http: //www.tripwire.com

15.2.4　拒绝服务

要防范的最重要威胁之一是拒绝服务。拒绝服务（DoS）是指某人的操作使得其他用户很难或者不可能访问一个服务，或者延迟对一个时间临界服务的访问。

在2000年年初，出现了一次针对高科技网站的、闻名的分布式拒绝服务攻击（DDoS）。

这些高科技网站目标包括Yahoo!、eBay、Amazon、E-Trade和Buy.com。这些网站的流量级别是我们大多数人很难想象的，但是仍然而受到了攻击，仍然在一个DoS攻击下关闭了几个小时。尽管攻击者通常不会因为关闭一个网站捞到什么好处，但是经营者可能会损失金钱、时间和声誉。

有些站点在特定的时候会有大量的业务。当一个重大的体育事件发生之前，在线书签站点就可能经历这样的情况。2004年就曾经发生过黑客攻击人员通过DDoS攻击获利的事情，他们通过在站点业务高峰时间进行攻击的威胁向在线书签站点勒索现金。

这些攻击之所以难以防范的一个重要原因是有大量的方法来实现这样的攻击。这些方法可能包括在目标机器上安装一个程序，该程序将消耗系统的大多数处理器时间；或者使用一个自动工具发送垃圾邮件。发送垃圾邮件的方式还包括以攻击目标作为邮件的发送者将垃圾邮件发送给人们。这样，攻击目标就会收到成千上万封愤怒的回复信件。

自动工具用于对攻击目标发动分布式拒绝服务的攻击。不需要知道许多信息，一个攻击者就可以扫描大量的机器，通过常见的弱点破解其中一台机器，安装上这个自动工具。因为这些过程都是自动的，攻击者可以在5s之内将自动工具安装到一个被攻击主机。当足够的机器安装了这样的工具之后，这些工具将指示所有这些机器向目标主机发动潮水般的网络流量攻击。

一般来说，我们很难防范DoS攻击。但通过简单的研究，可以发现被常规的DDoS工具占用的默认端口并关闭这些端口。路由器可以提供类似于限制使用特定协议的流量百分比的机制(例如，ICMP)。检测网络中正被用来攻击其他主机的主机比保护主机避免他人攻击更容易。如果每个网络管理员能够警惕地监视自己的网络，DDoS可能就不是那么严重的问题。

因为有如此之多的方法可以用来进行攻击，因此唯一真正有效的防范措施是监视常规流量，以及配备一批专家以在发生异常的时候采取对策。

15.2.5　软件错误

购买的、免费获取的或编写的软件都可能包含严重的错误。由于网络项目的开发周期通常都很短，那么该软件就很有可能会存在一些错误。任何高度依赖于计算机程序的企业都可能遭受到软件错误的攻击。

软件中的错误可能导致许多无法预见的行为，其中包括服务无效、安全缺口、经济损失和服务质量低下。

我们可以发现，导致这些错误的原因一般都包括低质量的设计说明书、开发人员做出的不完善假设和不充分的测试。

1. 低质量的设计说明书

设计文档越简单或者越模糊，最终产品就越有可能出错。尽管对我们而言，在顾客的信用卡被拒绝的时候，特定订单的产品不应该发送给顾客——这个逻辑有些多余，然而对一个大预算的网站不应该出现这样的错误。开发人员对他们所使用系统的经验越少，设计说明书就应该越详细明确。

2. 开发人员做出的假设

系统的设计人员和程序员需要做出许多假设。如果可能的话，他们要将这些假设记录到文

档中，并且应该保证这些假设都是正确的。但是有些时候，人们做出的假设却非常糟糕。这些假设可能包括输入数据都是有效的，不包含不经常用到的字符，或者输入数据会小于一个特定的长度。此外，还可能包括关于计算时间的假设，例如两个互相冲突的操作在同一时刻出现的可能性或者一个复杂的任务经常比一个简单的任务耗时多。

这类假设很可能会被我们所忽略，因为在大多数情况下它们都是正确的。一个入侵者可能利用缓冲区溢出的问题，因为程序员假定输入数据不会超过某个长度；或者合法用户可能获得让人混淆的错误信息而离开，因为系统开发人员没有见过一个人的名字中会包含有省略号。这类错误可以通过充分的测试与仔细的代码检查找出来并进行改正。

从历史的角度看，入侵者找出来的操作系统级别或应用程序级别的弱点通常与缓冲区溢出或者竞争条件有关。

3. 不充分的测试

在所有可能的硬件类型上，运行所有可能的操作系统以及所有可能的用户设置的条件下，测试所有可能的输入条件几乎是不可能的。基于网络的系统更是如此。

我们需要的是经过精密设计的测试计划，在一个有代表性的常见机器类型上测试软件的所有功能。一个规划得很好的系列测试应该把目标制定在对项目中的每行代码，应该保证至少测试一次。理想情况下，这套测试应该自动执行，以便它可以在特定的测试机器上运行而不要花费很多周折。

测试的最大问题是它非常单调并且具有重复性。尽管有一些人喜欢打破规则，但是没有人喜欢一次又一次地打破同样的规则。让原始开发人员之外的人进行测试是非常重要的。测试的一个重要目标就是检查出开发人员所作的不完善假设。一个局外人更可能作出不同的假设。除此之外，专家们也很难非常专注地查找自己工作中的毛病。

15.2.6　否认

我们需要考虑的最后一个风险是否认。否认通常发生在事务参与的一方否认已经参与了事务处理。在电子商务领域中的一个例子就是，一个人在某个网站预订了一件货物，然后否认自己授权该网站从信用卡扣除费用；或者一个人在邮件中答应某件事情，然后声称是别人伪造了该邮件。

理想情况下，涉及金融事务的双方应该为参与事务的对方提供不可否认的证据。任一方都不能否认他们在一件事务处理中的参与行为；或者，更精确地说，双方都向第三方（例如法庭）最终证明对方的参与行为。事实上，这种事情很少发生。

身份验证可以为识别正在参与事务的一方身份提供保证。如果是由一个可信任的组织分配的，证明身份的数字证书可以提供更充分的保证。

每一方发送的信息也需要被证明是准确的，而不是胡乱捏造的。如果不能证明所收到的信息恰恰就是Corp Pty Ltd发送过来的信息，就没有很多证据能够证明他们发送了此信息。

正如前面所介绍的，签名信息或者加密信息可以防止信息被秘密修改。

对于关系正处于持续过程中的双方事务，使用加密的或经过签名的数字证书进行通信是有效的防止否认发生的方法。而对于一次性的事务处理，例如电子商务网站和一个持有信用卡的

陌生人之间的初次接触，这就不那么实际。

一个电子商务公司应该愿意提交它的身份证明，并且愿意花费几百美元在能提供身份证明的权威机构获得身份证明，例如，VeriSign（http://www.verisign.com/）或Thawte (http://www.tha-wte.com/)，这样才能向用户保证公司的坦诚。这样的公司会拒绝每个不愿在其订单中同样证明其身份的顾客吗？对许多小型事务，商人们通常可以接受一定程度的欺骗或否认的风险，而不愿就此错过生意。

15.3　易用性、性能、成本和安全性

就其自身特性来说，网络是危险的。人们将它设计成允许许多匿名用户请求我们机器上的服务。这些请求大多数都是完全合法的网页请求。但是将机器连接到Internet会允许人们尝试其他连接方式。

尽管我们可以试图假定最高级别的安全是合适的，但是现实情况却极少如此。如果希望真正的安全，那么就关掉所有计算机，从所有网络上断开，关在一间紧锁的屋子里。要使计算机可以利用并且便于使用，我们对安全性能必须有一些放宽。

在易用性、性能、成本和安全性之间有一个折中。使服务更安全可能会降低服务的易用性，例如限制人们可以做什么，或者要求他们证明自己的身份。增加安全性能也可能降低机器的性能级别。运行一些软件以使系统更安全，例如加密软件、入侵检测系统、病毒扫描软件和扩展的日志软件，这些软件将消耗一定的系统资源。例如，对于一个常规的Web连接来说，提供一个加密的连接（如对一个网站的SSL连接）将消耗更多的资源。

这些性能的损失可以通过在专门为加密设计的更快的机器或硬件上花费更多的资金而解决。

我们可以将性能、易有性、成本和安全看作相互制约的目标。需要调查其平衡并作出明智的折中决定。根据信息价值、投资预算、预期访问量和我们认为合法用户能够忍受的障碍等这些因素，可以提出一个折中的方法。

15.4　建立一套安全政策

安全政策是一个文档，它描述了：

- 公司的一般安全要求
- 安全保护对象——软件、硬件、数据
- 负责保护这些项目的人
- 安全标准及其度量标准，度量标准衡量这些标准在多大程度上适合编写安全政策的一个可以借鉴的好策略就是编写过程像为软件编写功能需求一样。安全政策不应该谈论具体的应用或者解决办法，只需要标明安全目标与现实环境下的安全要求。它也不应该被频繁地更新。

安全政策应该使用单独的文档，它阐明了在特定环境中所要求的安全政策方针。对于公司的不同部门，可以采取不同的方针。这更类似设计文档或者一个程序手册，它们详细记录要保证某级别的安全性所要求的实际操作。

15.5 身份验证原则

身份验证试图证明某人的确是他本人。有许多可以提供身份验证的方法，但是与大多数安全措施一样，方法越安全，使用起来就越麻烦。

身份验证技术包括密码、数字签名、生物鉴定措施（例如指纹扫描），以及涉及硬件（例如智能卡）的措施。在网络上，只有两种技术是经常使用的：密码和数字签名。

生物鉴定措施和大多数硬件解决办法都包含了特殊的输入设备，因此限定授权用户必须到指定的机器上接触这些设备。这对于要访问某个组织的内部系统来说，它是可以接受的，甚至是令人满意的，但是它会丧失让一个系统在网络上得到广泛应用的许多好处。

密码易于网络应用，也易于用户使用，并且不需要特殊的设备。它们提供一定级别的身份验证，但是也许不能独立地适于安全级别较高的系统。

密码是一个简单的概念。用户和系统都知道密码。如果一个访问者声称是某个用户，并知道密码，系统就没有理由不相信他就是该用户。只要别人不知道或者猜不出密码，那么采用密码就是安全的。但是只使用密码仍然存在许多潜在的弱点，例如它不能提供健壮的身份验证。

许多密码很容易就被别人猜到。如果允许用户选择自己的密码，大概50%的用户会选择容易破解的密码。这种密码中通常包含字典单词或者用户名。以易用性作为代价，可以迫使用户在密码中包含数字或者标点符号，但是这会导致一些用户难以记住密码。

告诉用户选择更好的密码可能会有所帮助，但是即便如此，仍然有约25%的用户会选择容易破解的密码。这可能需要通过加强密码政策来实现，防止用户选择容易破解的字符组合，这可以通过使用非字典单词，或要求密码是数字或标点符号或大小写字母的混合来实现。严格的密码规则的一个危险是它可能会导致许多合法用户不能记住自己的密码。难以记住密码会增加用户不安全操作的可能性，例如，他可能写一个便条“username fred password rover”，把这个便条贴在显示器上。需要教育用户不要将密码写下来或做其他蠢事，例如在电话中将密码告诉自称正在系统中工作的人。

密码也可以通过电子化的方式捕获。通过运行一个程序捕获终端的按键或者使用一个“嗅控器”捕获网络信息，入侵者可以（并且肯定可以）捕获一对可以使用的登录名和密码。我们可以通过加密网络信息来限制捕获密码的机会。

尽管密码有许多潜在的缺点，但仍然不失为一种简单而相对有效的用户身份验证方法。它们提供的保密级别可能不适于国家级安全，但是用于检查一个顾客订单的分发状况则是非常理想的。

身份验证机制内置于大多数流行的Web浏览器和Web服务器之中。Web服务器可能会要求请求服务器上特定目录文件的人们输入用户名和密码。

当需要一个登录名和密码时，浏览器将弹出一个如图15-2所示的对话框。

图15-2　当用户试图访问Web服务器上一个受限目录时，Web浏览器将要求用户进行身份验证

Apache的Web服务器和Microsoft的IIS都采

用了这样的办法，它能够很简单地保护网站的一部分或者全部。使用 PHP或者MySQL，可以通过许多其他方法来实现同样的效果。使用MySQL比内置的身份验证速度更快。使用PHP，可以提供更灵活的身份验证，或者以更具吸引力的方式呈现此要求。

我们将在第17章中详细介绍身份验证的示例。

15.6　加密技术基础

加密算法是将信息转变为一个看起来是任意数据串的数学过程。

通常，要被加密的初始数据称为*普通文本*，但是该信息代表什么并不重要——无论它是真正的文本，还是其他类型的数据。类似地，已加密的信息称为*密文*，它们看起来完全不像文本。图15-3所示的就是加密的简单流程。首先，普通文本被载入到加密引擎，以前，加密引擎可能是一个机械设备，例如World War II Engima机器，但现在，绝大多数引擎都是计算机程序。然后由加密引擎产生密文。

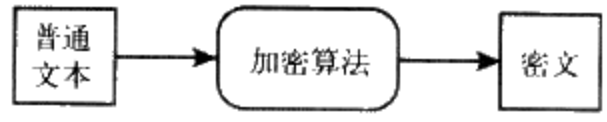

图15-3　加密过程将接收普通文本并将其转变为看起来随机的密文

要创建一个如图15-2所示的、需要身份验证的受保护目录，我们可以使用由Apache的身份验证所提供的大多数基本类型（在下一章中，我们将了解它们的使用方法）。这将在储密码之前对它们进行加密。我们创建了一个用户，其密码是password。该密码经过加密后，将以aWDuA3X3H.mc2的形式保存在数据库中。可以发现，普通文本和密文之间并没有明显的相似之处。

这种加密方法是不可逆的。许多密码使用一种单方向的加密算法进行存储。要检查输入的密码是否正确，不需要对加密的密码进行解密。只需要加密尝试输入的密码然后将它与存储的密码比较即可。

许多加密过程都是可逆的，但并非所有加密都是这样。这些可逆的过程称为*解密*，图15-4所示的是双向加密过程。

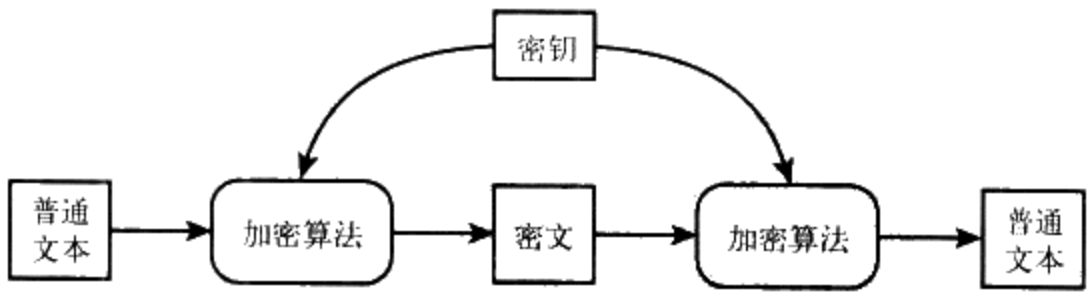

图15-4　加密接收普通文本并将其转换为看起来是随机的密文，解密将密文转换为普通文本

加密技术的诞生已经有将近4000年的历史了，但是真正的广泛应用开始于第二次世界大战。从那时起，它的发展模式与计算机网络的应用非常类似，开始用于军事和金融公司，20世纪70年代开始广泛应用于公司，到20世纪90年代便已经普遍应用。在最近几年中，加密已经从普通人只能在第二次世界大战电影和令人毛骨悚然的小说中看到的概念，发展到在报纸上经常看到，并且每次用Web浏览器购买东西时都将亲身经历的东西。

目前，有许多不同的加密算法可供使用。有些算法，例如DES，使用一个公有密钥或者

个私有密钥；有一些算法，如RSA，使用一个公有密钥和一个单独的私有密钥。

15.6.1　私有密钥加密

私有密钥加密也称作保密密钥加密，它信赖于授权用户知道或者可以访问一个密钥。该密钥必须是保密的。如果密钥落入别人手中，未授权的用户也可以阅读加密消息。如图15-4所示，发送方（加密消息的人）和接收方（解密消息的人）都有同样的密钥。

使用最广泛的密钥算法是数据加密标准（DES）。该方案是IBM公司在20世纪70年代发起并被用作美国商业和未分类的政府通信的标准。现在的计算机速度比20世纪70年代快了几个数量级，因此，1998年以后，DES就已经开始过时了。

其他著名的密钥系统包括RC2、RC4、RC5、triple DES（3DES）和IDEA。其中triple DES非常安全。它使用与DES相同的算法，3次分别应用3个不同的密钥。一个普通文本消息将必须顺序地使用密钥1解密，使用密钥2解密，再用密钥3解密。

提示：有趣的是，triple DES的安全性能只是DES的两倍。如果需要安全性能3倍于DES的加密算法，可以编写一个5倍于DES安全性能的算法。

显然，密钥加密的一个缺点是，要向某人发送一个机密的消息，需要通过秘密的方式把密钥告诉对方。如果可以通过秘密的方式来分发一个密码，为什么不通过这个秘密的方式来分发消息呢？

幸运的是，当Diffie和Hellman在1976年最初公布第一个公有密钥方案时，加密技术有了突破。

15.6.2　公有密钥加密

公有密钥加密依赖于两个不同的密钥，一个公有密钥和一个私有密钥。如图15-5所示，公有密钥用于加密消息，私有密钥用于解密它们。

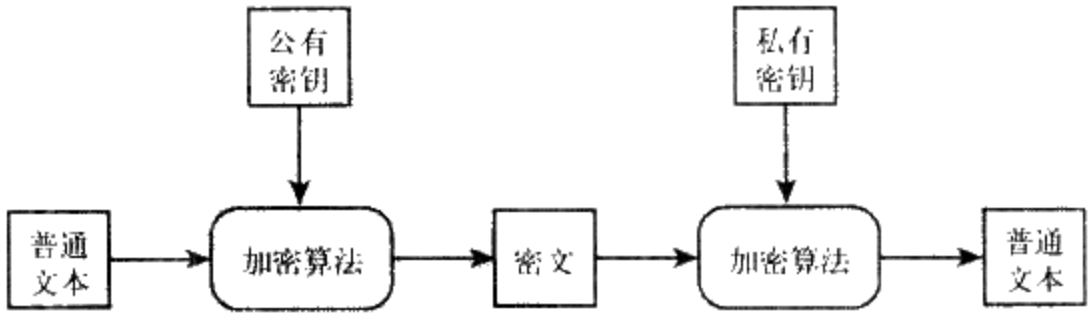

图15-5　公有密钥在加密和解密的过程中都必须使用独立的密码

该系统的好处在于公有密钥的使用，顾名思义，它可以公之于众。任何获得公有密钥的人都可以发送一个秘密消息。只有拥有私有密钥的人才可以解密消息。

最常使用的公有密钥算法是RSA，它是由麻省理工学院的Rivest、Shamir和Adelman在1978年发表的。RSA是一个专利系统，但是其专利保护时间在2000年9月已经到期。

以公开的方式发送一个公有密钥而无须担心被第三者看到的特性是一个巨大的优点，但是密钥系统仍然只适用于常规用途。通常，我们会使用一种混合的系统。公有密钥系统应用于为

密钥系统传输密钥，密钥系统应用于会话通信的其余部分。虽然这样会增加系统的复杂度，但是这却是可以容忍的，因为密钥系统比公有密钥系统快约1000倍。

15.6.3　数字签名

数字签名与公有密钥加密相关，但是与公有密钥和私有密钥的作用相反。一个发送者可以用密钥加密和签名一个消息。当人们接收消息的时候，接收者可以用发送者的公有密钥对它解密。因为发送者是唯一可以访问密钥的人，接收者可以确认消息由谁发送以及它有没有被修改。

数字签名可能确实很有意义。它们允许接收者确认消息没有被篡改，发送者也很难否认消息的内容，或者否认发送过该消息。

值得注意的是，尽管消息被加密了，但拥有公有密钥的任何人都可以阅读它。尽管使用了相同的技术和密钥，但是在这里，加密的目的是为了防止篡改和否认，而不是防止别人阅读消息。

对于大量消息来说，由于公有密钥的速度比较慢，就出现了另一种称为*哈希函数*的算法。通常，它可以用于提高效率。哈希函数为给定的消息计算出一个消息摘要或者哈希值。算法产生什么值并不重要。重要的是其输出是唯一的，也就是说，每次使用一个特定的输入时候，输出是一样的，该输出较小，因此算法较快。

最常见的哈希函数是MD5和SHA。

哈希函数可以为特定消息产生相匹配的消息摘要。如果有一个消息和一个消息摘要，只要确认摘要没有被篡改，就可以确认该消息没有被篡改。最后，创建一个数字签名最常用的方法是用一个快速哈希函数为整个消息创建一个消息摘要，然后使用速度较慢的公有密钥加密算法对简短的消息摘要进行加密。现在，签名可以通过任何常规的、可能不安全的方式发送。

当接收到一个签名消息的时候，我们可检查它。用发送者的公有密钥解密签名。使用与发送者同样方法产生该消息的一个哈希值。如果解密的哈希值匹配生产的哈希值，就可以确认消息来自发送者而且没有被篡改。

15.7　数字证书

能够验证一条消息没有被篡改以及一组消息都来自一个特定的用户或机器是很好的。对于商业往来来说，将用户或者服务器与一个真实的法律实体（如人或公司）结合成一体可能会更好。

一个数字证书可以将一个公有密钥与一个个人或者组织的细节以签名的数字格式结合起来。如果给用户一个证书，他就拥有对方的公有密钥，如果该用户要发送一个加密消息的话，他还拥有对方的详细消息，并且知道这些消息没有改变。

这里有一个问题，就是该消息的可信度只与签发它的人的可信度一样。任何人都可以产生并签发一个证书，声称是某人。对于商业贸易来说，有可信的第三方验证所参与的实体，以及他们证书上记录的细节是非常有意义的。

这些第三方验证称为*认证授权*（CA）。认证授权公司可以为个人或组织颁发数字证书，从而证明其身份。两个最著名的CA公司是VeriSign（http://www.verisign.com/）和Thawte

(http://www.thawte.com/)，但是你还可以选择其他一些公司的授权认证。

VeriSign 和Thawte在实际用途中，两者还是存在一些区别。此外，一些不很著名的认证授权公司，例如Network Solution公司（http://www.networksolutions.com）和GoDaddy公司(http://www.godaddy.com/)，他们的认证收费则便宜很多。

认证授权公司可以颁发一个证书，用来证明他们可以保证该个人和组织的身份或标识。需要注意的是，要将证书作为一个信誉说明书或者信誉陈述是没有什么价值的。它并不能保证与我们打交道的人或组织具有良好的信誉，它真正的意义是，如果对方违约，我们就有机会知道对方真实的地址和个人，以便控告他们。

证书提供了一种网络信任。如果选择信任CA，就可以选择信任他们所信任的个人，以及被证明方信任的人。

数字证书最常见的用途是为电子商务网站提供一个可以信任的氛围。拥有一个由著名CA所颁发的证书，Web浏览器可以通过SSL的方式连接到网站而不会出现警告对话框。支持SSL连接的Web服务器通常又称为安全的Web服务器。

15.8　安全的Web服务器

通过加密套接字协议层，我们可以使用Apache Web服务器、Microsoft IIS或者其他免费或商业Web服务器，与浏览器之间进行安全的通信。使用Apache产品，我们可以使用类似于UNIX的操作系统。这肯定比IIS更可靠一些，但是也更难安装和配置。当然，也可以选择在Windows平台下使用 Apache。

在IIS上使用SSL包括简单地安装IIS，生成一个密钥对，以及安装证书。在 Apache上使用SSL要求安装OpenSSL，以及在服务器软件的安装过程启用mod_ssl模块。

“你可以在获得蛋糕以后再吃掉它”，就像购买商业版的Apache一样。若干年以来，Red Hat一直在销售 Stronghold产品，它如今已经捆绑在Red Hat企业版本的Linux产品上。通过购买这样的解决方案，我们即可以获得 Linux的可靠性，同时它又是一个简易安装的产品并有厂商的技术支持。

在附录A中，我们给出了两个最流行的Web服务器Apache和IIS的安装说明。在生成自己的数字证书以后，就可以立即开始使用SSL了，但是Web浏览器将向网站访问者发出警告，警告将告诉访问者，证书是你自己签发的。要有效地使用SSL，还需要一个认证授权公司颁发的一个证书。

对于不同的CA公司来说，获取证书的确切过程各不相同，但是一般地说，需要向CA证明公司是法律承认的公司，有确切的实际地址并且拥有相关域名。

必须生成一个证书签发请求（CSR）。而证书生成过程在不同的服务器之间各不相同。

其用法说明在CA的网站上可以找到。 Stronghold和IIS提供一个对话框驱动的过程，而Apache要求输入命令。然而，对所有服务器来说，该请求过程是基本相同的。最终结果是一个加密的签发请求证书（certificate signing request，CSR）。CSR看起来应该类似于如下：

```
---BEGIN NEW CERTIFICATE REQUEST---
MIIBuwIBAAKBgQCLn1XX8faMHhtzStp9wY6BVTPuEU9bpMmhrb6vgaNZy4dTe6VS
```

```
84p7wGepq5CQjfOL4Hjda+g12xzto8uxBkCDO98Xg9q86CY45HZk+q6GyGOLZSOD
8cQHwh1oUP65s5Tz018OFBzpI3bHxfO6aYelWYziDiFKp1BrUdua+pK4SQIVAPLH
SV9FSz8Z7IHOg1Zr5H82oQOlAoGAWSPWyfVXPAF8h2GDb+cf97k44VkHZ+Rxpe8G
ghlfBn9L3ESWUZNOJMfDLlny7dStYU98VTVNekidYuaBsvyEkFrny7NCUmiuaSnX
4UjtFDkNhX9j5YbCRGLmsc865AT54KRu31O2/dKHLo6NgFPirijHy99HJ4LRY9Z9
HkXVzswCgYBwBFH2QfK88C6JKW3ah+6cHQ4DeoiltxI627WN5HcQLwkPGn+WtYSZ
jG5tw4tqqogmJ+1P2F/5G6FI2DQP7QDvKNeAU8jXcuijuWo27S2sbhQtXgZRTZvO
jGn89BC0mIHgHQMkI7vz35mx1Skk3VNq3ehwhGCvJlvoeiv2J8X2IQIVAOTRp7zp
En7QlXnXwls7xXbbuKP0
---END NEW CERTIFICATE REQUEST---
```

拥有一个CSR，缴纳相应的会费，提交证明你存在的文档，并验证你正在使用的域名与商业文档中的域名一致，就可以与CA签订一个证书了。

CA颁发证书之后，必须将证书保存到系统中并告诉Web服务器它的地址。最终的证书只是一个文本文件，它看起来如前面显示的CSR。

15.9 审计与日志记录

操作系统允许把各种各样的事件记入到日志文件中。从安全的角度考虑，我们可能关心的事件包括网络错误，对特定数据文件（例如，配置文件或NT注册表）的访问，对一些特定的程序（例如，在UNIX系统中使用su命令，可用来将自己变成另一个用户，通常是root用户）的调用。

日志文件可以帮助我们在出错的时候检测错误或者恶意操作。如果在注意到问题之后检查它们，它们还可以告诉我们一个问题或者非法入侵是如何发生的。日志文件有两个主要问题：大小和准确性。

如果将检测和记录问题的条件设置为最极端的情况，最终将得到庞大的日志文件而难以检查。要帮助整理庞大的日志文件，可能需要使用一个现存的工具或者从安全政策中得到的审计脚本，这样就可以在日志中搜索“感兴趣”的事件。审计过程可以实时发生，也可以定期发生。

特别情况下，日志文件容易受到攻击。如果入侵者拥有系统的root用户权限或管理员权限，他就可以随意修改日志文件以掩饰行踪。UNIX可以将事件记录到一个独立的机器中。

这意味着一个入侵者必须控制至少两台机器才能掩饰行踪。类似的功能在NT中也可以实现，但是没有在UNIX下实现容易。

系统管理员可以进行定期审计。但是我们可能还要一个外部的审计人员定期检查管理员的操作。

15.10 防火墙

在网络中，设计防火墙的目的是将本地网络与外部网络相分离。与一个建筑中或者一个停车场用防火墙以防止火灾蔓延到其他区域一样，网络防火墙也是防止混乱蔓延到我们的网络。

防火墙用于保护内部网络中的机器以防外来攻击。它过滤和拒绝不符合标准的消息，限制防火墙之外的个人和机器的行为。

有时，防火墙也用于限制其内部的个人和机器的行为。一个防火墙可以限制用户使用的网

络协议，限制他们可以连接的主机，或者迫使他们使用代理服务器以降低带宽费用。

防火墙可能是一个硬件设备，例如，具有过滤规则的路由器，或者运行于一台机器上的一个软件程序。任何情况下，防火墙都需要两个网络接口和一组规则。它可以监视所有试图从一个网络流到另一个网络的信息。如果被监视的这些信息符合规则，就将它发送到另一网络；否则，就终止它或者拒绝它。

防火墙可以根据信息包的类型、源地址、目的地址或端口信息对其进行过滤。对于一些信息包，可能只是简单地丢弃它，但是一些事件可能触发日志记录或者警告。

15.11　备份数据

在任何灾难恢复计划中，都不能够低估备份的重要性。硬件和建筑物可以买保险和替换，或者网站主机的位置可以更换，但是如果定制的网络软件遭到毁坏，没有保险公司可以恢复它。

我们必须定期备份网站的所有组件——静态网页、脚本和数据库。备份的频率取决于网站的动态程度。如果它完全是静态的，可以只在修改网站的时候对其进行备份。但是，在本书中，我们所讨论的这类网站可能都要频繁修改，特别是如果接收在线订单的话。

大多数规模适当的网站都需要在服务器上使用RAID（廉价冗余磁盘阵列），它可以支持镜像。RAID考虑了可能有一个硬盘出现故障的情况。但是，如果整个硬盘阵列、机器或者建筑出现问题该怎么办呢？

应该根据更新量的大小以一定的频率进行独立的备份。这些备份应该保存在独立的介质上，这些介质更适宜放置于一个安全的、独立的地方，以防火灾、盗窃或自然灾害。

如今，在Internet上，有许多关于备份和恢复的资料。在这里，我们将集中讨论如何备份由PHP和MySQL数据库建立的网站。

15.11.1　备份常规文件

在大多数系统中，可以使用备份软件来备份HTML、PHP、图像和其他非数据库文件，这些操作是非常简单的。

最常用的免费软件是AMANDA（Advanced Maryland Automated Network Disk Archiver），它是由Maryland大学开发。它适于备份UNIX机器，也可以通过SAMBA备份Windows机器。要了解更多消息，请访问其网站：http: //www.amanda.org/

15.11.2　备份与恢复MySQL数据库

备份一个正在工作的数据库比较复杂。要避免在数据库修改的过程中复制任何表数据。关于如何备份与恢复一个MySQL数据库，请参阅第12章的详细介绍。

15.12　自然环境的安全性

到目前为止，我们考虑的安全威胁都与无形的东西（如软件）有关，但是，我们不应该忽略系统的自然环境安全。网站需要空调，需要防火、防人（笨拙的人和罪犯）、防止电力故障

和网络故障。

系统应该安全地锁起来。根据公司运作的规模，这可能是一个房间、一个机柜或者一个壁橱。不必访问机器房间的人员不要进入。未授权的人们可能有意或无意地拔掉电缆，或者使用一张可引导的磁盘尝试绕过安检机制。

火灾发生时的喷水装置也可能对电子设备造成极大损害，如同火灾一样。在过去，Halon（一种化学物质）灭火系统可用于避免这个问题。现在，在“消耗臭氧层的物质的蒙特利尔协议”的约束下，Halon已经禁止生产，因此新的灭火系统必须采用其他危害更小灭火物质，例如，氩气或者二氧化碳。要了解更详细的消息，请参阅如下网站：http: //www.epa.gov/Ozone/snap/fire/qa.html

在许多地方，偶然的短暂电力故障是经常遇到的事实。在天气恶劣并且采用架空电线的地方，长时间的故障也可能经常发生。如果系统的持续运作很重要的话，应该购买一个不间断电源支持设备（UPS）。一个可以为一台机器供电10分钟的UPS价格低于200美元（美国）。如果考虑更长时间的故障，或者更多设备，花费可能会更昂贵一些。长时间的故障则需要一台发电机，以运行空调同时运行计算机。

与电力故障一样，几分钟或者几小时的网络故障是无法控制的，而且肯定会偶尔发生。如果网络至关重要，那么准备多个互联网服务商的连接就很有意义了。拥有两个连接将使成本更高，也意味着出现故障的时候，处理能力可能会降低，但是至少不会变得不可访问。

这些类型的问题就是将机器集中放置在一个专用场所的一些原因。尽管一个中型的公司可能不必使用能够运行多于几分钟的UPS，采用多个冗余的网络连接，以及配置灭火系统，但是一个容纳100台公司机器的场所绝对有必要采取这些措施。

15.13　下一章

在第16章中，我们将进一步学习Web应用的安全性。我们将了解谁是我们的敌人，以及如何保护我们自己，如何保护我们的服务器，网络和代码。此外，还有如何制定灾难计划。