网络安全理论与应用

实用网络技术丛书

● 杨 波 编著

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

第1章　引　　言

- 网络安全面临的威胁
- 网络入侵者和病毒
- 网络安全的模型

1.1 网络安全面临的威胁

1.1.1 安全威胁

Internet 为人类交换信息，促进科学、技术、文化、教育、生产的发展，提高现代人的生活质量提供了极大的便利，但同时对国家、单位和个人的信息安全带来了极大的威胁。由于因特网的全球性、开放性、无缝连通性、共享性、动态性发展，使得任何人都可以自由地接入 Internet，其中有善者，也有恶者。恶者会采用各种攻击手段进行破坏活动。网络安全面临的攻击有独立的犯罪者、有组织的犯罪集团和国家情报机构。对网络的攻击具有以下新特点：无边界性、突发性、蔓延性和隐蔽性。因此我们考虑网络安全，就要首先知道网络安全面临有哪些威胁。

网络安全所面临的威胁来自很多方面，并且随着时间的变化而变化。这些威胁可以宏观地分为人为威胁和自然威胁。

自然威胁可能来自于各种自然灾害、恶劣的场地环境、电磁辐射和电磁干扰、网络设备自然老化等。这些无目的的事件，有时会直接威胁网络的安全，影响信息的存储媒体。

我们主要讨论人为威胁，也就是说对网络的人为攻击。这些攻击手段都是通过寻找系统的弱点，以便达到破坏、欺骗、窃取数据等目的，造成经济上和政治上不可估量的损失。

图 1.1（a）表示网络中两个计算机系统之间的正常信息流，图 1.1（b）至（e）说明以下四类基本的攻击类型。

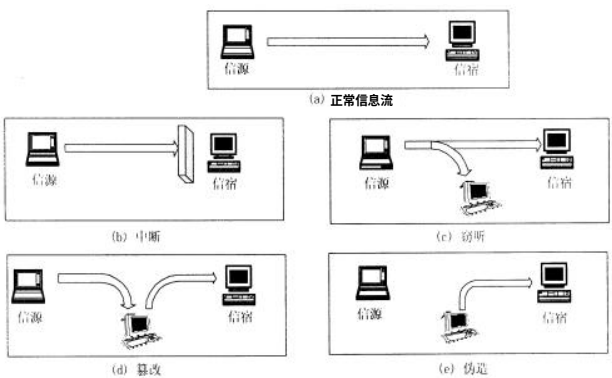


图 1.1 攻击类型示意图

① 中断：是对系统的可用性进行攻击，如破坏计算机硬件、线路或文件管理系统。
② 窃听：是对系统的保密性进行攻击，如搭线窃听、对文件或程序的非法拷贝。

③ 篡改：是对系统的完整性进行攻击，如修改数据文件中的数据、替换某一程序使其执行不同的功能、修改网络中传送的消息内容。

④ 伪造：是对系统的真实性进行攻击。如在网络中插入伪造的消息或在文件中插入伪造的记录。

攻击类型又可分为被动攻击和主动攻击，如图 1.2 所示。

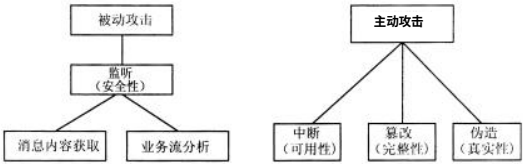


图 1.2 被动攻击和主动攻击

1．被动攻击

被动攻击相应于攻击类型中的窃听，敌手的目标是窃取传输中的信息。被动攻击又分为两类，一类是获取消息的内容，很容易理解；第二类是进行业务流分析，假如我们通过某种手段，比如加密，使得敌手从截获的消息无法得到消息的真实内容，然而敌手却有可能获得消息的格式、确定通信双方的位置和身份以及通信的次数和消息的长度，这些信息可能对通信双方来说是敏感的，例如公司间的合作关系可能是保密的、电子函件用户可能不想让他人知道自己正在和谁通信、电子现金的支付者可能不想让别人知道自己正在消费、Web 浏览器用户也可能不愿意让别人知道自己正在浏览哪一站点。

被动攻击因不对传输的消息做任何修改，因而是难以检测的，所以抗击这种攻击的重点在于预防而非检测。

2．主动攻击

这种攻击包括对数据流的某些篡改或产生某些假的数据流。主动攻击又可分为以下四个子类。

① 假冒：某个实体（人或者系统）假装成另外一个实体，以使某一防线的守卫者相信它是一个合法的实体，此后便可僭取合法用户的权利和特权。这是侵入安全防线最为常用的方法。

② 重放：攻击者对截获的某次合法数据进行拷贝，以后出于非法的目的而重新发送。

③ 消息的篡改：指某一通信数据在传输过程中被改变、删除或替代，如“允许甲读账目文件”改为“允许乙读账目文件”。

④ 业务拒绝：对通信设备的使用和管理被无条件地拒绝。这种攻击可能有一个特定的目标，例如某个实体对到某一特定终端的所有消息都予以阻止。还有一类业务拒绝是对整个网络实施破坏，例如使网络瘫痪或用大量无用信息使其资源耗尽。

绝对防止主动攻击是十分困难的，因为需要随时随地对通信设备和通信线路进行物理保护，因此抗击主动攻击的主要途径是检测，以及对此攻击造成的破坏进行恢复。

1.1.2 安全业务

在网络通信中，主要的安全防护措施称作安全业务，有以下 5 种。

① 保密业务：保护数据以防被动攻击。保护方式可根据保护范围的大小分为若干级，其中最高级保护可在一定时间范围内保护两个用户之间传输的所有数据，低级保护包括对单个消息的保护或对一个消息中某个特定域的保护。保密业务还有对业务流实施保密，防止敌手进行业务流分析以获得通信的信源、信宿、次数、消息长度和其他信息。

② 认证业务：用于保证通信的真实性。在单向通信的情况下，认证业务的功能是使接收者相信消息确实是由它自己所声称的那个信源发出的。在双向通信的情况下，如计算机终端和主机的连接，在连接开始时，认证服务则使通信双方都相信对方是真实的（即的确是它所声称的实体）；其次，认证业务还保证通信双方的通信连接不能被第三方介入，以假冒其中的一方而进行非授权的传输或接收。

③ 完整性业务：完整性业务和保密业务一样，也能应用于消息流、单个消息或一个消息的某一选定域。用于消息流的完整性业务目的在于保证所接收的消息未经复制、插入、篡改、重排或重放，因而是和所发出的消息完全一样的；这种服务还能对已毁坏的数据进行恢复，所以这种业务主要是针对对消息流的篡改和业务拒绝的。应用于单个消息或一个消息某一选定域的完整性业务仅用来防止对消息的篡改。

④ 不可否认业务：用于防止通信双方中的某一方对所传输消息的否认，因此，一个消息发出后，接收者能够证明这一消息的确是由通信的另一方发出的。类似地，当一个消息被接收后，发出者能够证明这一消息的确已被通信的另一方接收了。

⑤ 访问控制：访问控制的目标是防止对网络资源的非授权访问，控制的实现方式是认证，即检查欲访问某一资源的用户是否具有访问权。

1.2 网络入侵者和病毒

网络安全的人为威胁主要来自用户（恶意的或无恶意的）和恶意软件的非法侵入，入侵网络的用户也称为黑客，黑客可能是某个无恶意的人，其目的仅仅是以破译和进入一个计算机系统为满足，或者是某个心怀不满的雇员，其目的是对计算机系统实施破坏，也可能是一个犯罪分子，其目的是非法窃取系统资源（如窃取信用卡号或非法资金传送），对数据进行未授权的修改或破坏计算机系统。

恶意软件是病毒、蠕虫等恶意程序，分为两类，如图 1.3 所示，一类需要主程序，另一类不需要。前者是某个程序中的一段，不能独立于实际的应用程序或系统程序；后者是能被操作系统调度和运行的独立程序。

对恶意软件也可根据其能否自我复制来进行分类。不能自我复制的是程序段，这种程序段在主程序被调用执行时就可激活；能够自我复制的或者是程序段（病毒）或者是独立的程序（蠕虫、细菌等），当这种程序段或独立的程序被执行时，可能复制一个或多个自己的副本，以后这些副本可在这一系统或其他系统中被激活。以上仅是大致分类，因为逻辑炸弹或特洛伊木马可能是病毒或蠕虫的一部分。下面对恶意程序作一简单介绍。

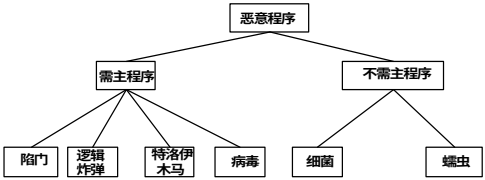


图 1.3 恶意程序分类

1．陷门

陷门是进入程序的秘密入口，掌握陷门的人可不经过通常的安全访问程序而访问该程序。陷门通常是由程序员调试程序时合法使用的，程序员在调试具有认证功能且设置很长的应用程序时，也许希望有特别的权限或避免所有必要的设置和认证，因此希望有一种激活程序的方法。陷门是识别某些特定输入序列的码或通过运行某个特定用户 ID 或一个不可能事件序列能激活的码。

陷门一旦被原来的程序员利用，或者被无意或有意的人发现将会带来严重的安全后果。比如，可能利用陷门在程序中建立隐蔽通道，甚至植入一些隐蔽的恶意程序。非法利用陷门可以使原来相互隔离的网络信息形成某种潜在的关联，进而可以非法访问网络，达到窃取、篡改、伪造和破坏信息等目的，甚至造成网络的大面积瘫痪。

2．逻辑炸弹

逻辑炸弹是早于病毒和蠕虫出现的最早的恶意程序之一，它是镶嵌于合法程序并且设置了“爆炸”条件的代码。一旦满足设置的条件（例如某个特定文件的存在或缺省、某个特定的日期、运行应用程序的某个特定的用户等），逻辑炸弹可能会修改或删除数据甚至整个文件，造成死机甚至网络瘫痪。

3．特洛伊木马

特洛伊木马是包含在有用程序中的隐藏码，当有用程序被调用时，这种隐藏码将执行某些有害功能。

特洛伊木马能用于间接实现非授权用户不能直接实现的功能，例如一用户欲访问共享系统中另一用户的文件，他可建立一个特洛伊木马程序，该程序执行时能修改被访问用户文件的访问许可权限，使得任一用户都能读这一文件。特洛伊木马一般难于被发现，例如编译程序中的特洛伊木马程序，可在编译某个程序（如系统登录程序）时在其中插入附加的码，这个码在登录程序中就建立了一个陷门以便特洛伊木马的作者使用某个特定的口令登录系统。这种特洛伊木马程序无法通过读登录程序的源代码而被发现。

特洛伊木马另一个常见的目的是毁坏数据。含有特洛伊木马程序的有用程序在执行某个有用功能时，其中的特洛伊木马也许正在悄悄地删除用户的文件。

4．病毒

病毒是一个程序，它能通过修改其他程序而将其感染。其中修改过程包括病毒程序本身的复制，复制得到的副本又可继续感染其他程序。

生物病毒是遗传代码 DNA 或 RNA 的一个片段。它能进入并欺骗生物细胞，从而由生

物细胞为其复制成千上万个副本。与生物病毒类似，计算机病毒也可被复制。驻留在机器中的病毒可暂时控制计算机的磁盘操作系统，被感染的计算机一旦与其他软件接触，病毒将给这一软件复制自己的一个新副本。因此病毒的感染可通过用户之间交换磁盘或在网络中发送程序而得以蔓延。

5．蠕虫

网络中的蠕虫程序通过网络连接关系从一个系统蔓延到另一系统。系统中，网络蠕虫一旦被激活，其行为或者与病毒或细菌一样、或者在网络中植入特洛伊木马程序、或者直接进行破坏活动。

为了复制自己，网络蠕虫需利用某种网络载体，例如：

① 电子邮件：网络蠕虫可通过电子邮件将自己复制到其他系统。

② 远程执行能力：蠕虫可将自己复制到另一系统。

③ 远程登录能力：蠕虫可像用户一样登录到某个远程系统，然后又将自己从这一系统复制到另一系统。

和计算机病毒一样，网络蠕虫也有以下四个阶段：潜伏阶段、传播阶段、激活阶段和执行阶段。传播阶段一般有以下方式：

① 检查主表或类似的远程系统地址库，以搜索并感染其他系统。

② 建立和远程系统的连接。

③ 将自己复制到远程系统并引起副本运行。

网络蠕虫在将自己复制到某个系统时，也可能检查这个系统以前是否已被感染。在多道程序系统中，蠕虫可能会将自己伪装成一个系统程序或使用某些不被系统操作员注意的其他名称。

6．细菌

细菌是不明显危害其他文件的程序，它们的惟一目的是复制自己。一个典型的细菌程序在多道程序系统中可能仅同时建立自己的两个副本，或者建立两个新文件，每个新文件都是细菌程序最初源文件的副本，而每个副本又继续建立两个新的副本。如此下去，细菌数目指数地增长，最终占据所有处理器、存储器或磁盘空间，拒绝用户对这些资源的访问。

1.3 网络安全的模型

图 1.4 表示网络安全的基本模型。

通信双方欲传递某个消息，需通过以下方式建立一个逻辑上的信息通道，首先在网络中定义从发方到收方的一个路由，然后在该路由上共同执行通信协议。

如果需要保护所传信息以防敌手对其保密性、认证性等构成的威胁，则需要考虑通信的安全性。安全传输技术有以下两个基本成分。

① 消息的安全传输：包括对消息的加密和认证。加密的目的是将消息搞乱以使敌手无法读懂，认证的目的是检查发送者的身分。

② 通信双方共享的某些秘密信息，如加密密钥。

为获得消息的安全传输，可能还需要一个可信的第三方，其作用可能是负责向通信双方分布秘密信息或者在通信双方有争议时进行仲裁。

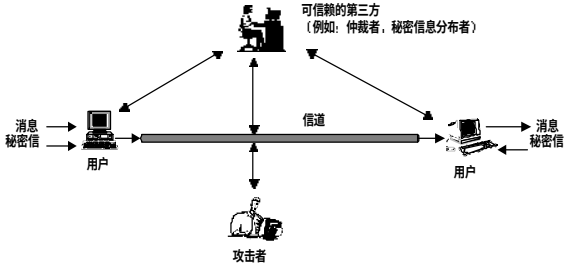


图 1.4　网络安全的基本模型

一个安全的网络通信必须考虑以下四个方面：

① 加密算法；

② 用于加密算法的秘密信息；

③ 秘密信息的分布和共享；

④ 使用加密算法和秘密信息以获得安全服务所需的协议。

以上考虑的是网络安全的一般模型，然而还有其他一些情况。图 1.5 表示保护信息系统以防未授权访问的一个模型。

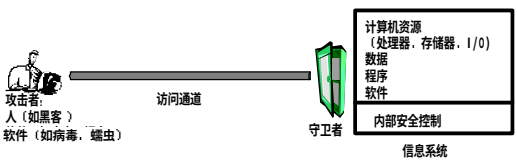


图 1.5　信息系统的保护模型

对付未授权访问的安全机制可分为两道防线，第一道称为守卫者，它包括基于通行字的登录程序和屏蔽逻辑程序，分别用于拒绝非授权用户的访问、检测和拒绝病毒。第二道防线由一些内部控制部件构成，用于管理系统内部的各项操作和分析所存有的信息，以检查是否有未授权的入侵者。

上面我们介绍了网络安全面临的威胁以及网络安全的一般模型。网络安全是一个属于综合、交叉的学科领域。它要利用数学、电子、信息、通信、计算机等诸多学科的长期积累的知识和最新发展成果。网络安全研究的内容很多，它涉及安全体系结构、安全协议、密码理论、信息分析、安全监控、应急处理等，其中密码是网络安全的关键技术。本书首先介绍网络安全所需的密码学原理，包括加解密算法及其设计原理、消息认证及杂凑函数、数字签字及公钥证书。第二部分介绍网络安全实际技术，包括 Kerberos、X.509 证书、PGP、S/MIME、IPSec、Web 的安全性、防火墙、虚拟专用网的安全性、电子商务的安全性等。

第 2 章　单钥密码体制

- 密码学基本概念
- 流密码
- 分组密码概述
- 数据加密标准（DES）
- 差分密码分析与线性密码分析
- 分组密码的运行模式
- IDEA
- AES 简介

2.1 密码学基本概念

2.1.1 保密通信系统

通信双方采用保密通信系统可以隐蔽和保护需要发送的消息，使未授权者不能提取信息。发方将要发送的消息称做明文，明文被变换成看似无意义的随机消息，称为密文，这种变换过程称做加密，其逆过程，即由密文恢复出原明文的过程称为解密。对明文进行加密操作的人员称做加密员或密码员。密码员对明文进行加密时所采用的一组规则称做加密算法。传送消息的预定对象称做接收者，接收者对密文进行解密时所采用的一组规则称做解密算法。加密和解密算法的操作通常都是在一组密钥控制下进行的，分别称做加密密钥和解密密钥。传统密码体制所用的加密密钥和解密密钥相同，或实质上等同，即从一个易于得出另一个，称其为单钥或对称密码体制。若加密密钥和解密密钥不相同，从一个难于推出另一个，则称为双钥或非对称密码体制。密钥是密码体制安全保密的关键，它的产生和管理是密码学中的重要研究课题。

在信息传输和处理系统中，除了已定的接收者外，还有非授权者，他们通过各种办法（如搭线窃听、电磁窃听、声音窃听等）来窃取机密信息，称其为截收者。截收者虽然不知道系统所用的密钥，但通过分析可能从截获的密文推断出原来的明文或密钥，这一过程称做密码分析，从事这一工作的人称做密码分析员，研究如何从密文推演出明文、密钥或解密算法的学问称做密码分析学。对一个保密通信系统采取截获密文进行分析的这类攻击称做被动攻击。现代信息系统还可能遭受的另一类攻击是主动攻击，非法入侵者、攻击者或黑客主动向系统窜扰，采用删除、增添、重放、伪造等篡改手段向系统注入假消息，达到利已害人的目的。这是现代信息系统中更为棘手的问题。

保密通信系统可用图 2.1 表示，它由以下几部分组成：明文消息空间 M，密文消息空间 C，密钥空间 K_1 和 K_2，在单钥体制下 $K_1 = K_2 = K$，此时密钥 K 需经安全的密钥信道由发方传给收方；加密变换 $E_{k_1} : M \to C$，其中 $k_1 \in K_1$，由加密器完成；解密变换

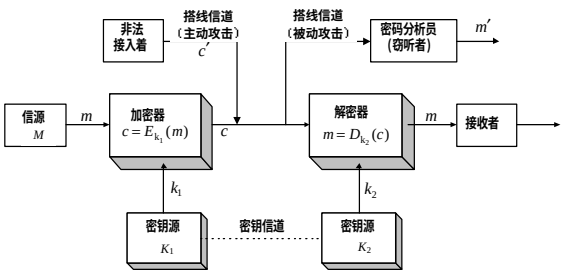


图 2.1 保密通信系统模型

$D_{k_2}: C \to M$，其中 $k_2 \in K_2$，由解密器实现。称总体 $(M, C, K_1, K_2, E_{k_1}, D_{k_2})$ 为保密通信系统。对于给定的明文消息 $m \in M$，密钥 $k_1 \in K_1$，加密变换将明文 m 变换为密文 c，即

$$c = f(m, k_1) = E_{k_1}(m) \qquad m \in M, \quad k_1 \in K_1$$

收方利用通过安全信道送来的密钥 k（单钥体制下）或用本地密钥发生器产生的解密密钥 $k_2 \in K_2$(双钥体制下)控制解密操作 D，对收到的密文进行变换得到恢复的明文消息

$$m = D_{k_2}(c) \qquad m \in M, \quad k_2 \in K_2$$

而密码分析者，则用其选定的变换函数 h 对截获的密文 c 进行变换，得到的明文是明文空间中的某个元素

$$m' = h(c)$$

一般 $m' \neq m$。如果 $m' = m$，则分析成功。

为了保护信息的保密性，抗击密码分析，保密系统应当满足下述要求：

① 系统即使达不到理论上是不可破的，即 $p_r\{m' = m\} = 0$，也应当为实际上是不可破的。就是说，从截获的密文或某些已知明文密文对，要决定密钥或任意明文在计算上是不可行的。

② 系统的保密性不依赖于对加密体制或算法的保密，而依赖于密钥。这是著名的 Kerckhoff 原则。

③ 加密和解密算法适用于所有密钥空间中的元素。

④ 系统便于实现和使用。

2.1.2　密码体制分类

密码体制从原理上可分为两大类，即单钥体制和双钥体制。

单钥体制的加密密钥和解密密钥相同。系统的保密性主要取决于密钥的安全性，与算法的安全性无关，即由密文和加解密算法不可能得到明文。换句话说算法无需保密，需保密的仅是密钥。根据单钥密码体制的这种特性，单钥加解密算法可通过低费用的芯片来实现。密钥可由发方产生然后再经一个安全可靠的途径(如信使递送)送至收方，或由第三方产生后安全可靠地分配给通信双方。如何产生满足保密要求的密钥以及如何将密钥安全可靠地分配给通信双方是这类体制设计和实现的主要课题。密钥产生、分配、存储、销毁等问题，统称为密钥管理。这是影响系统安全的关键因素，即使密码算法再好，若密钥管理问题处理不好，就很难保证系统的安全保密。

单钥体制对明文消息的加密有两种方式：一是明文消息按字符(如二元数字)逐位地加密，称之为流密码；另一种是将明文消息分组(含有多个字符)，逐组地进行加密，称之为分组密码。单钥体制不仅可用于数据加密，也可用于消息的认证。

双钥体制是由 Diffie 和 Hellman 于 1976 年首先引入的。采用双钥体制的每个用户都有一对选定的密钥：一个是可以公开的，可以像电话号码一样进行注册公布；另一个则是秘密的。因此双钥体制又称做公钥体制。

双钥密码体制的主要特点是将加密和解密能力分开，因而可以实现多个用户加密的消息只能由一个用户解读，或由一个用户加密的消息而多个用户可以解读。前者可用于公共

网络中实现保密通信，而后者可用于实现对用户的认证。详细介绍见第 3 章。

2.1.3 密码攻击概述

表 2.1 所示是攻击者对密码系统的四种攻击类型，类型的划分由攻击者可获取的信息量决定。其中，最困难的攻击类型是惟密文攻击，这种攻击的手段一般是穷搜索法，即对截获的密文依次用所有可能的密钥试译，直到得到有意义的明文。只要有足够多的计算时间和存储容量，原则上穷搜索法总是可以成功的。但实际上，任何一种能保障安全要求的实用密码都会设计得使这一方法在实际上是不可行的。敌手因此还需对密文进行统计测试分析，为此需要知道被加密的明文的类型，比如英文文本、法文文本、MD-Dos 执行文件、Java 源列表等。

表 2.1 对密码系统的攻击类型

攻击类型	攻击者掌握的内容
惟密文攻击	加密算法 截获的部分密文
已知明文攻击	加密算法 截获的部分密文 一个或多个明文—密文对
选择明文攻击	加密算法 截获的部分密文 自己选择的明文消息及由密钥产生的相应密文
选择密文攻击	加密算法 截获的部分密文 自己选择的密文消息及相应的被解密的明文

惟密文攻击时，敌手知道的信息量最少，因此最易抵抗。然而，很多情况下，敌手可能有更多的信息，也许能截获一个或多个明文及其对应的密文，也许知道消息中将出现的某种明文格式。例如 ps 格式文件开始位置的格式总是相同的，电子资金传送消息总有一个标准的报头或标题。这时的攻击称为已知明文攻击，敌手也许能够从已知的明文被变换成密文的方式得到密钥。

与已知明文攻击密切相关的一种攻击法称为可能字攻击。例如对一篇散文加密，敌手可能对消息含义知之甚少。然而，如果对非常特别的信息加密，敌手也许能知道消息中的某一部分。例如，发送一个加密的账目文件，敌手可能知道某些关键字在文件报头的位置。又如，一个公司开发的程序的源代码中，可能在某个标准位置上有该公司的版权声明。

如果攻击者能在加密系统中插入自己选择的明文消息，则通过该明文消息对应的密文，有可能确定出密钥的结构，这种攻击称为选择明文攻击。

选择密文攻击是指攻击者利用解密算法，对自己所选的密文解密出相应的明文。

还有两个概念值得注意，一个加密算法是无条件安全的，如果算法产生的密文不能给出惟一决定相应明文的足够信息。此时无论敌手截获多少密文、花费多少时间，都不能解

密密文。Shannon 指出，仅当密钥至少和明文一样长时，才能达到无条件安全。也就是说除了一次一密方案外，再无其他的加密方案是无条件安全的。因此，加密算法只要满足以下两条准则之一就行。

① 破译密文的代价超过被加密信息的价值。

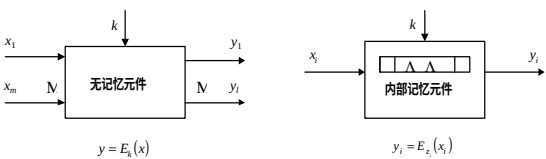
② 破译密文所花的时间超过信息的有用期。

满足以上两个准则的加密算法称为计算上安全的。

2.2 流 密 码

流密码的基本思想是利用密钥 k 产生一个密钥流 $z = z_0 z_1 \Lambda$ ，并使用如下规则对明文串 $x = x_0 x_1 x_2 \Lambda$ 加密： $y = y_0 y_1 y_2 \Lambda = E_{z_0}(x_0) E_{z_1}(x_1) E_{z_2}(x_2) \Lambda$ 。密钥流由密钥流发生器 f 产生： $z_i = f(k, s_i)$，这里 s_i 是加密器中的记忆元件（存储器）在时刻 i 的状态，f 是由密钥 k 和 s_i 产生的函数。

分组密码与流密码的区别就在于记忆性（如图 2.2）。流密码的滚动密钥 $z_0 = f(k, s_0)$ 由函数 f 、密钥 k 和指定的初态 s_0 完全确定。此后，由于输入加密器的明文可能影响加密器中内部记忆元件的存储状态，因而 $s_i (i > 0)$ 可能依赖于 $k, s_0, x_0, x_1, \Lambda, x_{i-1}$ 等参数。



(a) 分组密码　　(b) 流密码

图 2.2　分组密码和流密码的比较

2.2.1 同步流密码

根据加密器中记忆元件的存储状态 s_i 是否依赖于输入的明文字符，流密码可进一步分成同步和自同步两种。 s_i 独立于明文字符的叫做同步流密码，否则叫做自同步流密码。由于自同步流密码的密钥流的产生与明文有关，因而较难从理论上进行分析。目前大多数研究成果都是关于同步流密码的。在同步流密码中，由于 $z_i = f(k, s_i)$ 与明文字符无关，因而此时密文字符 $y_i = E_{z_i}(x_i)$ 也不依赖于此前的明文字符。因此，可将同步流密码的加密器分成密钥流产生器和加密变换器两个部分。如果与上述加密变换对应的解密变换为 $x_i = D_{z_i}(y_i)$，则我们可给出同步流密码的模型如图 2.3 所示。

同步流密码的加密变换 E_{z_i} 可以有多种选择，只要保证变换是可逆的即可。实际使用的数字保密通信系统一般都是二元系统，因而在有限域 GF(2) 上讨论的二元加法流密码（如图 2.4）是目前最受欢迎的流密码体制，其加密变换可表示为 $y_i = z_i + x_i$ 。

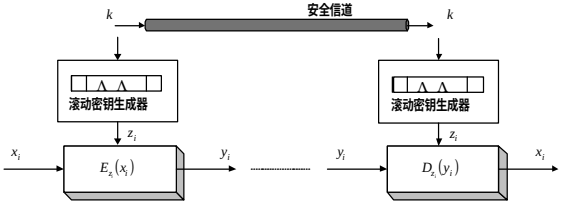


图 2.3 同步流密码体制模型

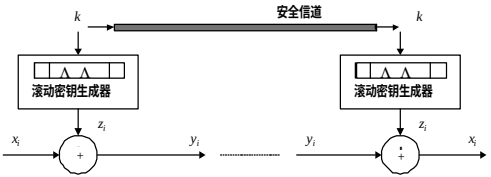


图 2.4 加法流密码体制模型

一次一密密码是加法流密码的原型。事实上，如果 $z_i = k_i$（即密钥用做滚动密钥流），则加法流密码就退化成一次一密密码。实际使用中，密码设计者的最大愿望是设计出一个滚动密钥生成器，使得密钥 k 经其扩展成的密钥流序列 z 具有如下性质：极大的周期、良好的统计特性、抗线性分析、抗统计分析。

2.2.2 密钥流产生器

同步流密码的关键是密钥流产生器。一般可将其看成一个参数为 k 的有限状态自动机，由一个输出符号集 Z、一个状态集 Σ、两个函数 j 和 y 以及一个初始状态 s_0 所组成（如图 2.5 所示）。状态转移函数 $j: s_i \to s_{i+1}$，将当前状态 s_i 变为一个新状态 s_{i+1}，输出函数 $\Psi: s_i \to z_i$，当前状态 s_i 变为输出符号集中的一个元素 z_i。这种密钥流生成器设计的关键在于找出适当的状态转移函数 j 和输出函数 y，使得输出序列 z 满足密钥流序列 z 应满足的几个条件，并且要求在设备上是节省的和容易实现的。为了实现这一目标，必须采用非线性函数。

由于具有非线性的 j 的有限状态自动机理论很不完善，相应的密钥流产生器的分析工作受到极大的限制。相反地，当采用线性的 j 和非线性的 y 时，我们将能够进行深入的分析并可以得到好的生成器。为方便，可将这类生成器分成驱动部分和非线性组合部分（如图 2.6 所示）。驱动部分控制生成器的状态转移，并为非线性组合部分提供统计性能好的序列。而非线性组合部分要利用这些序列组合出满足要求的密钥流序列。

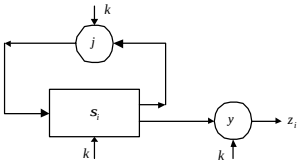


图 2.5　作为有限状态自动机的密钥流生成器

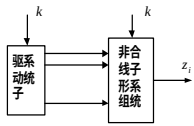


图 2.6　密钥流生成器的分解

目前最为流行和实用的密钥流产生器如图 2.7 所示，其驱动部分是一个或多个线性反馈移位寄存器（LFSR）。

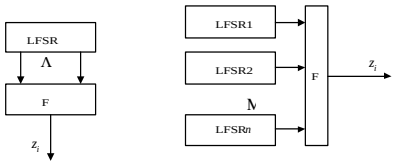


图 2.7　常见的两种密钥流产生器

2.2.3　线性反馈移位寄存器序列

线性反馈移位寄存器（简记为 LFSR）因其实现简单、速度快、有较为成熟的理论等优点而成为构造密钥流生成器的最重要的部件之一。设 GF(q)表示 q 元有限域，GF(q)上一个 n 级 LFSR 由 n 个 q 元存储器与若干个 GF(q)上的乘法器 和加法器连接而成（如图 2.8 所示，当 q=2 时，不需要乘法器），每一存储器称为 LFSR 的一级。初始状态由用户确定，当第 i 个移位时钟脉冲到来之时，LFSR 的状态由 $a_i, a_{i+1}, \Lambda\ , a_{i+n-1}$ 变为 $a_{i+1}, a_{i+2}, \Lambda\ , a_{i+n}$，并输出 a_i 作为序列 a 的一位。补入 LFSR 的最右边一级的 a_{i+n} 的值由下列线性递归关系式(也称反馈函数)决定。

$$a_{j+n} = -\sum_{j=1}^{n} c_j a_{j+n-1}, \quad j \geqslant 0$$

设 D 为 LFSR 延迟算子，则 $Da_i = a_{i-1}$，$i \geqslant 1$，因而 $f(D)a_i = 0$，$i \geqslant n$，这里

$$f(D) = c_0 + c_1 D + \Lambda\ + c_n D^n, \quad c_0 = 1$$

称为 LFSR 的反馈多项式。如果用未定元 x 取代 D 则得

$$f(x) = c_0 + c_1 x + \Lambda\ + c_{n-1} x^{n-1} + x^n$$

称为 LFSR 的连接多项式。

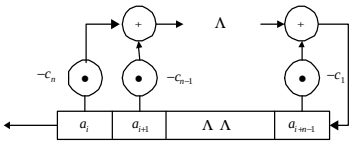


图 2.8 GF(q) **上一个** n **级线性反馈移位寄存器**

2.2.4 周期序列

用 s **表示无限序列** s_0, s_1, Λ **，** s^N **表示有限序列** $s_0, s_1, \Lambda\ s_{N-1}$ **。如果存在正整数** n**，使得** $s_{i+n} = s_i$**，** $i = 0,1,2,\Lambda$ **，则称序列** s **为周期序列，满足上式的正整数** n **称为序列** s **的周期，其中最小的一个称为最小周期。如果** s **满足**

$$s_j + c_1 s_{j-1} + \Lambda + c_L s_{j-L} = 0 \ (j \geqslant L)$$

其中， L **是正整数，** $c_1, c_2, \Lambda\ , c_L$ **是在** GF(p^m) **中，则称** s **是一个** L **阶线性递归序列，满足上式的最小的正整数** L**，称为该递归序列** s **的线性复杂度，记为** $c(s)$**。**

对于序列 s **和** s^N **，它们的生成函数（也称形式幂级数）定义为**

$$s(x) = s_0 + s_1 x + \Lambda + s_n x^n + \Lambda = \sum_{i=0}^{\infty} s_i x^i$$

和

$$s^N(x) = s_0 + s_1 x + \Lambda + s_{N-1} x^{N-1}$$

如果 s **是周期序列，周期为** N**，** s^N **是它的第一个周期，则**

$$s(x) = s^N(x)(1 + x^N + x^{2N} + \Lambda) = \frac{s^N(x)}{1 - x^N}$$

从而 $s(x)$**可以表示成**

$$s(x) = \frac{s^N(x) / \gcd(s^N(x), 1 - x^N)}{(1 - x^N) / \gcd(s^N(x), 1 - x^N)} = \frac{g(x)}{f_s(x)}$$

这里 $f_s(x) = (1 - x^N) / \gcd(s^N(x), 1 - x^N)$**，** $g(x) = s^N(x) / \gcd(s^N(x), 1 - x^N)$ **。**

显然， $g(x) / f_s(x)$ **是既约的，** $\deg g(x) < \deg f_s(x)$**，** $f_s(x)$ **为** s **的极小多项式，** $\deg f_s(x) = c(s)$ **为** s **的线性复杂度。**

2.2.5 B-M 综合算法

设 $a^N = a_0 a_1 \Lambda\ a_{N-1}$ **是一个有限序列，** $f(x) = 1 + c_1 x + \Lambda + c_l x^{l-1}$ **是一个多项式。用** $(f(x), l)$ **表示以** $f(x)$ **为反馈多项式的** l **级线性反馈移位寄存器。如果** a^N **满足线性递归关系式**

$$a_k = -\sum_{i=1}^{l} c_1 a_{k-i}, \quad k \geqslant l$$

则称$(f(x), l)$产生a^N。

线性反馈移位寄存器综合问题，即是对于一个给定 N 长的序列，求产生它的最短线性反馈移位寄存器。B-M 综合算法有效地解决了线性移位反馈寄存器的综合问题，从而使序列的线性复杂度成为同步流密码强度的一个重要的度量指标。该算法递归地求出一系列线性反馈移位寄存器，$(f_n(x), l_n)$，n=1,2,…，N，使得每个$(f_n(x), l_n)$都是产生序列$a^n = a_0 a_1 \Lambda\ a_{n-1}$的最短线性移位寄存器。具体算法如下。

B-M 综合算法　设n_0是一个满足$a_0 = a_1 = \Lambda\ = a_{n_0-1} = 0$，$a_{n_0} \neq 0$的非负整数

初始值：$d_0 = d_1 = \Lambda\ = d_{n_0-1} = 0$，$d_{n_0} = a_{n_0}$，

$$f_0(x) = f_1(x) = \Lambda\ = f_{n_0}(x) = 1,$$

$$l_1 = l_2 = \Lambda\ = l_{n_0} = 0$$

（1）如果$n_0 = N$停止；否则，$(f_{n_0+1}(x), l_{n_0+1}) = (1 - d_{n_0} x^{n_0+1}, n_0 + 1)$，$n = n_0 + 1$转向（2）。

（2）如果$n = N$停止；否则，$d_n = f_n(E)a_n$，转向（3）。

（3）如果$d_n = 0$，$n \leftarrow n+1$，$f_n(x) = f_{n-1}(x)$，$l_n = l_{n-1}$，转向（2）；否则，$n \leftarrow n+1$，找出 m $(1 \leqslant m < n-1)$，使得 $l_m < l_{m+1} = l_{m+2} = \Lambda\ = l_{n-1}$，取 $f_n(x) = f_{n-1}(x) - d_{n-1} d_m^{-1} x^{n-1-m} f_m(x)$，$l_n = \max\{l_{n-1}, n - l_{n-1}\}$，转向（2）。

如此算出的$(f_N(x), l_N)$即是一个产生a^N的最短线性反馈移位寄存器。

产生一个序列的最短线性反馈移位寄存器一般不惟一，惟一的充分和必要条件是$2l_N \leqslant N$。因而，对于周期为 N 的序列，由 B-M 算法算出的$(f_{2N}(x), l_{2N})$即是产生此周期序列的惟一最短线性反馈移位寄存器。

2.3　分组密码概述

在许多密码系统中，单钥分组密码是系统安全的一个重要组成部分，用分组密码易于构造伪随机数生成器、流密码、消息认证码（MAC）和杂凑函数等，还可进而成为消息认证技术、数据完整性机制、实体认证协议以及单钥数字签字体制的核心组成部分。实际应用中对于分组密码可能提出多方面的要求，除了安全性外，还有运行速度、存储量(程序的长度、数据分组长度、高速缓存大小)、实现平台(硬、软件，芯片)、运行模式等限制条件。这些都需要与安全性要求之间进行适当的折衷选择。

分组密码是将明文消息编码表示后的数字序列 $x_0, x_1, \Lambda\ , x_i, \Lambda$ 划分成长为 n 的组 $x = (x_0, x_1, \Lambda\ , x_{n-1})$，各组(长为 n 的矢量)分别在密钥$k = (k_0, k_1, \Lambda\ , k_{i-1})$控制下变换成等长的输出数字序列$y = (y_0, y_1, \Lambda\ , y_{m-1})$（长为 m 的矢量），其加密函数 E：$\boldsymbol{V}_n \times K \to \boldsymbol{V}_m$，$\boldsymbol{V}_n$和 $\boldsymbol{V}_m$ 分别是 n 维和 m 维矢量空间，K为密钥空间，参见图 2.9 所示。它与流密码不同之处在于输出的每一位数字不是只与相应时刻输入的明文数字有关，而是与一组长为 n 的明文数字有关。在相同密钥下，分组密码对长为 n 的输入明文组所实施的变换是等同的，所以只需研