第 3 章　密码学概述

密码学是网络信息安全的基础，密码学常被认为是数学和计算机科学的分支，和信息论也密切相关。著名的密码学者 Ron Rivest 解释道："密码学是关于如何在敌人存在的环境中通信"。密码学的首要目的是隐藏信息的含义，并不是隐藏信息的存在。密码学也促进了计算机科学的发展，特别是计算机与网络安全中的技术发展。密码学已被应用于人们的日常生活，如自动柜员机的芯片卡、电子商务等。目前主要的网络安全技术也都是以密码学为基础的，使用最广泛的加密机制是对称密码体制和公钥密码体制。本章首先介绍密码学的起源和发展历程，再介绍密码学中的基本概念，最后介绍一些传统加密技术，虽然这些加密技术不再使用了，但这些加密技术中的一些思想依然在现代密码学中得到了延伸。

3.1　密码学起源

密码学一词源自希腊语 krypto 及"理念"两词，意思为"隐藏"及"消息"。它是研究信息系统安全保密的科学，其目的是为两人在不安全的信道上进行通信而不被破译者获取通信的内容。相传最早使用密码捆在木棒上方法是公元前 5 世纪的斯巴达人，公元前 404 年，斯巴达国（今希腊）北路军统帅莱山得在征服雅典之后，本国的信使赶到，献上了一条皮带，上面有文字，通报了敌将断其归路的企图。莱山得当机立断，率师轻装脱离了险境。他们使用的是一根叫 scytale 的棍子，送信人先绕棍子卷一张纸条，然后把要加密的信息写在上面，接着打开纸送给收信人。如果不知道棍子的宽度（这里作为密钥）是不可能解密里面的内容的。后来，罗马的军队用恺撒密码（三个字母表轮换）进行通信。

中国周朝兵书《六韬·龙韬》也记载了密码学的运用，其中的《阴符》和《阴书》便记载了周武王问姜子牙关于征战时与主将通信的方式：

太公曰："主与将，有阴符，凡八等。有大胜克敌之符，长一尺。破军擒将之符，长九寸。降城得邑之符，长八寸。却敌报远之符，长七寸。警众坚守之符，长六寸。请粮益兵之符，长五寸。败军亡将之符，长四寸。失利亡士之符，长三寸。诸奉使行符，稽留，若符事闻，泄告者，皆诛之。八符者，主将秘闻，所以阴通言语，不泄中外相知之术。敌虽圣智，莫之能识。"

武王问太公曰："……符不能明；相去辽远，言语不通。为之奈何？"

太公曰："诸有阴事大虑，当用书，不用符。主以书遗将，将以书问主。书皆一合而再离，三发而一知。再离者，分书为三部。三发而一知者，言三人，人操一分，相参而不相知情也。此谓阴书。敌虽圣智，莫之能识。"

阴符是以八等长度的符来表达不同的消息和指令，可算是密码学中的替代法(substitution)，把信息转变成敌人看不懂的符号。至于阴书则运用了移位法，把书一分为三，分三人传递，要把三份书重新拼合才能获得还原的信息。

在随后的19个世纪中，主要是发明一些更加高明的加密技术，这些技术的安全性通常依赖于用户赋予它们多大的信任程度。然而密码学文献发展有个很奇妙的过程，由于战争和各个国家之间的利益，密码学重要的进展很少在公开的文献中出现。密码学的发展大致可以分为以下三个阶段。

第一阶段是从几千年前到1949年。这一时期密码学还没有成为一门真正的科学，而是一门艺术。密码学专家常常是凭自己的直觉和信念来进行密码设计，而对密码的分析也多基于密码分析者(即破译者)的直觉和经验来进行的。第一次世界大战以后，情况开始变化，完全处于秘密工作状态的美国陆军和海军的机要部门开始在密码学方面取得根本性的进展。加州奥克兰的Edward H. Hebern申请了第一个转轮机专利，这种装置在差不多50年内被指定为美军的主要密码设备。但是由于战争的原因，公开的文件寥寥无几。

第二阶段是从1949年到1975年。1949年，美国数学家、信息论的创始人Shannon Claude Elwood发表了《保密系统的信息理论》一文，它标志着密码学阶段的开始。同时以这篇文章为标志的信息论为对称密钥密码系统建立了理论基础，从此密码学成为一门科学。由于保密的需要，这时人们基本上看不到关于密码学的文献和资料，平常民众是接触不到密码的。1967年David Kahn出版了一本叫做《破译者》(*The Codebreakers*)的小说，它并没有任何新的技术思想，但却对密码学的历史做了相当完整的记述。这部著作的意义不仅在于它涉及相当广泛的领域，而且在于它使成千上万原本不知道密码学的人了解了密码学。新的密码学文章慢慢开始源源不断地被编写出来，使人们知道了密码学。20世纪70年代初期，IBM发表了有关密码学的几篇技术报告，从而使更多的人了解了密码学的存在。但科学理论的产生并没有使密码学失去艺术的一面，如今，密码学仍是一门具有艺术性的科学。

第三阶段为1976年至今。1976年，Diffie和Hellman发表了《密码学的新方向》一文，他们首次证明了在发送端和接收端不需要传输密钥的保密通信的可能性，从而开创了公钥密码学的新纪元。该文章也成了区分古典密码和现代密码的标志。1977年，美国的数据加密标准(DES)公布。这两件事情导致了对密码学的空前研究。从这时候起，开始对密码在民用方面进行研究，密码才开始充分发挥它的商用价值和社会价值，人们才开始接触到密码学。这种转变也促使了密码学的空前发展。密码学发展至今，已有两大类密码系统：第一类为对称密钥密码系统，第二类为非对称密钥(公开密钥)密码系统。

历史车轮滚滚向前，密码学紧跟科学技术前进的步伐，根据密码学所使用的科学技术，还可以将其分为如下发展历程：密码学的初级形式——手工阶段，经过中间形式——机械阶段，发展到今天的高级形式——电子与计算机阶段。现代密码分析依赖数学方面的知识，现代密码学离开数学是不可想象的，密码学涉及数学的各个分支，如代数、数论、

概率论、信息论、几何、组合学等。不仅如此，密码学的研究还需要具有其他学科的专业知识，如物理、电机工程、量子力学、计算机科学、电子学、系统工程、语言学等。反过来，密码学的研究也促进了上述各科学科的发展。

计算机的出现，大大促进了密码学的变革。由于商业应用和大量的计算机网络通信的需要，民间对数据保护、数据传输的安全性、防止工业谍报活动等课题越来越重视，密码学的发展从此进入了一个崭新的阶段，与此同时，密码学的研究开始大规模地扩展到民用。

3.2 密码的基本概念

首先定义一些术语。原始的消息为明文(Plaintext)，而加密后的消息为密文(Cipher)。从明文到密文变换过程称为加密(Encryption)，一般用E表示加密过程，从密文到明文的变换过程称为解密(Decryption)，一般用D表示解密过程。用于加密的各种方案构成的研究领域称为密码编码学。这样的加密方案称为密码体制或密码。不知道任何加密细节的条件下解密消息的技术属于密码分析学的范畴，密码分析学即外行所说的“破译”。密码编码学和密码分析学统称密码学。

3.2.1 密码编码学

密码编码学的主要任务是研究安全、高效的信息加密算法和信息认证算法的设计理论与技术，密码系统设计通常的基本要求是：

(1) 知道密钥 K_{AB} 时，加密 E_{AB} 容易计算。

(2) 知道密钥 K_{AB} 时，解密 D_{AB} 容易计算。

(3) 不知道密钥 K_{AB} 时，由密文 $C=E_{AB}(M)$ 不容易推导出明文 M。

以上三点要求说明密码系统设计的原则是：对合法的通信双方来说，加密和解密变换是容易的；对密码分析员来说，由密文推导出明文是困难的。衡量一个密码系统的好坏，当然应当以它能否被攻破和易于被攻破为基本标准。理论上不可攻破的密码系统是一次一密，密钥只对一个消息进行加解密，之后丢弃不用，每一条新消息都需要一个与其等长的新密钥。这就是著名的一次一密，它是不可攻破的。但是在实际应用中，这种系统却受到很大限制：

- 分发和存储这样大的随机密钥序列(它和明文信息等长)，确保密钥的安全是很困难的。
- 如何生成真正的随机序列也是一个现实问题。因此，人们转而寻求实际上不可攻破的密码系统。

所谓实际上不可攻破的密码系统，是指它们在理论上虽然是可以攻破的，但真正要攻破它们，所需要的计算资源如计算机时间和存储空间超出了实际上的可能性。例如，破解某个密码系统需要耗费计算机机时200年，这个密码系统实际上非常安全。密码编码学

系统具有以下三个独立的特征：

(1) 转换明文为密文的运算类型。所有的加密算法都基于两个原理：代换和置换，代换是将明文中的每个元素(如位、字母、位组或字组等)映射成另一个元素；置换是将明文中的元素重新排列。上述运算的基本要求是不允许有信息丢失(即所有的运算是可逆的)。大多数密码体制，都使用了多层代换和置换。

(2) 所用的密钥数。如果发送方和接收方使用相同的密钥，这种密码就称为对称密码、单密钥密码、秘密钥密码或传统密码。如果收发双方使用不同的密钥，这种密码就称为非对称密码、双钥或公钥密码。

(3) 处理明文的方法。分组密码每次处理输入的一组元素，相应地输出一组元素。流密码则是连续地处理输入元素，每次输出一个元素。

3.2.2 密码分析学

密码分析是指试图找出明文或密钥的工作，通常目标是恢复使用的密钥，而不是仅仅恢复出单个密文对应的明文，这样可以获得更多有价值的信息。攻击对称密码体制一般有两种方法：

(1) 密码分析学：密码分析学攻击通常分析加密算法的性质，利用明文的一般特征或某些明密文对。破译者使用的策略取决于加密方案的固有性质以及破译者掌握的信息。这种形式的攻击企图利用算法的特征来推导出特别的明文或使用的密钥。

(2) 穷举攻击：攻击者对一条密文尝试所有可能的密钥，直到把它转化为可读的有意义的明文。平均而言，获得成功至少要尝试所有可能密钥的一半。

如果上述任意一种攻击能成功地推导出密钥，那么影响将是灾难性的，将会危及所有未来和过去使用该密钥加密的消息的安全。

首先考虑密码分析学，然后讨论穷举攻击。

基于密码分析者知道信息的多少，表 3.1 概括了密码攻击的几种类型，表中唯密文攻击难度最大。有些情况下，攻击者甚至不知道加密算法，但是我们通常假设对手知道。这一条早在 1883 年柯克霍夫斯(A. Kerchoffs)在其名著《军事密码学》中就建立了一个重要原则：密码算法即使为密码分析者所知，也应该无助于用来推导出明文和密钥。这一原则已被人广泛接受，取名为柯克霍夫斯原则，并成为密码系统设计的重要原则之一。原因是依赖加密算法本身的机密性来防范密码分析者的代价较高，一旦对手知道加密算法则所有消息不再安全。重新设计加密算法并进行更换的成本也较大，通过密钥保密而不是加密算法保留来防范密码分析相对容易，可以通过更换密钥加强机密性。和更换算法的成本相比，更换密钥的成本也要小很多。

这种情况下，一种可能的攻击是试遍所有可能密钥的穷举攻击。如果密钥空间非常大，这种方法就不太实际。因此攻击者必须依赖于对密文本身的分析，这一般要运用各种统计方法。使用这种方法，攻击者对隐含的明文类型必须有所了解，例如，说明文是英文文本或法文文本、可执行文件、Java 源代码文件、会计文件等。

表 3.1　密码攻击的类型

攻击类型	密码分析者已知的信息
唯密文攻击 ciphertext only	• 加密算法 • 要解密的密文
已知明文攻击 known plaintext	• 加密算法 • 要解密的密文 • 用(与待解的密文)同一密钥加密的一个或多个密文对
选择明文攻击 chosen plaintext	• 加密算法 • 要解密的密文 • 分析者任意选择的一些明文,以及对应的密文(与待解的密文使用同一密钥加密)
选择密文攻击 chosen ciphertext	• 加密算法 • 要解密的密文 • 分析者有目的地选择一些密文,以及对应的明文(与待解的密文使用同一密钥解密)
选择文本攻击 chosen text	• 加密算法 • 要解密的密文 • 分析者任意选择的明文,以及对应的密文(与待解的密文使用同一密钥加密) • 分析者有目的地选择一些密文,以及对应的明文(与待解的密文使用同一密钥解密)

(1) 唯密文攻击。密码分析者有一些消息的密文,这些消息都用同一算法加密。密码分析者的任务是恢复尽可能多的明文,或者最好能推算出加密消息的密钥来,以便可采用相同的密钥解出其他被加密的消息。

已知:

$$C_1 = E_k(M_1),\ C_2 = E_k(M_2),\cdots,C_i = E_k(M_i)$$

推导出:$M_1,M_2,\cdots,M_i$,或者密钥 K,或者找出一个算法从 $C_{i+1}=E_k(M_{i+1})$ 推导出 M_{i+1}。唯密文攻击是最容易防范的,因为攻击者拥有的信息量最少。不过在很多情况下,分析者可以得到更多的信息。分析者可以捕获到一段或更多的明文信息及相应密文,也可能知道某段明文信息的格式等。例如,按照 Postscript 格式加密的文件总是以相同的格式开头,还有,电子金融消息往往有标准化的文件头或者标志等。这些都是已知明文攻击的例子。拥有这些知识的分析者就可以从分析明文入手来推导出密钥。

(2) 已知明文攻击。密码分析者不仅可以得到一些消息的密文,而且也知道这些消息的明文。分析者的任务是用加密消息推出用来加密的密钥或者推导出一个算法,用此算法可以对用同一密钥加密的任何新的消息进行解密。

已知:

$$M_1,C_1 = E_k(M_1),M_2,C_2 = E_k(M_2),\cdots,M_i,C_i = E_k(M_i)$$

推导出:密钥 K,或者从 $C_{i+1}=E_k(M_{i+1})$ 推导出 M_{i+1} 的算法。

与已知明文攻击紧密相关的是可能词攻击。如果攻击者处理的是一般分散文字信息,他可能对信息的内容一无所知,如果他处理的是一些特定的信息,他就可能知道其中

的部分内容。例如，对于一个完整的数据库文件，攻击者可能知道放在文件最前面的是某些关键词。又如，某公司开发的程序源代码可能含有该公司的版权信息，并且放在某个标准位置。

（3）选择明文攻击。分析者不仅可以得到一些消息的密文和相应的明文，而且他们也可以选择被加密的明文。也就是说分析者能够通过某种方式，让发送方在发送的信息中插入一段由他选择的信息，一个例子是差分密码分析。这个比已知明文攻击更有效，因为密码分析者能选择特定的明文块去加密，那些块可能产生更多关于密钥的信息，分析者的任务是推出用来加密消息的密钥或者导出一个算法，此算法可以对同一密钥加密的任何新的消息进行解密。

已知：M_1，$C_1=E_k(M_1)$，M_2，$C_2=E_k(M_2)$，…，M_i，$C_i=E_k(M_i)$，其中 M_1，M_2，…，M_i 可由密码分析者选择。

推导出：密钥 K，或者从 $C_{i+1}=E_k(M_{i+1})$ 推导出 M_{i+1} 的算法。

一般说来，如果分析者有办法选择明文加密，那么他将特意选取那些最有可能恢复出密钥的数据。

表 3.1 还列举了另外两种类型的攻击方法：选择密文攻击和选择文本攻击。它们在密码分析技术中很少用到，但是仍然是两种可能的攻击方法。

只有相对较弱的算法才抵挡不住唯密文攻击。一般地，加密算法起码要能经受得住已知明文攻击才行。

此外，还有两个概念值得注意。如果一个密码体制满足条件：无论有多少可使用的密文，都不足以唯一的确定密文所对应的明文，则称该加密体制是无条件安全的。也就是说，无论花多少时间，攻击者都无法将密文解密，因为他所需的信息不在密文里。除了一次一密（一次一密的具体内容将会在以后的文中讲到）之外，所有的加密算法都不是无条件安全的。因此，加密算法的使用者应挑选尽量满足以下标准的算法：

- 破译密码的代价超出密文信息的价值。
- 破译密码的时间超出密文信息的有效生命期。

如果满足了上述两条标准，则加密体制是计算上安全的，因为攻击者成功破译密文所需的工作量是非常巨大的。对称密码体制的所有分析方法都利用了这样一个事实，即明文的结构和模式在加密之后仍然保存了下来，并且在密文中能找到一些蛛丝马迹。随着对各种对称密码体制讨论的深入，这一点将会变得很明显。对公钥密码体制的分析是依据一个完全不同的假设，即密钥对的数学性质使得无法从一个密钥推出另一个密钥，在后面章节中会详细介绍公钥密码体制。

穷举攻击试遍所有密钥直到有一个合法的密钥能够把密文还原成明文，这就是穷举攻击。我们可以从这种方法入手，考虑其所需的时间代价。要获得成功一般需要尝试所有可能密钥中的一半，表 3.2 给出了对于不同密钥空间所耗用的时间。DES（数据加密标准）算法使用的是 56 位密钥，3DES 使用的是 168 位密钥，AES（高级加密标准）的最短密钥长度是 128 位。表中最后一行还列出了采用 26 个字母的排列作为密钥的代换密码的一些结果。假设执行一次解密需要 1μs（这是今天普通计算机的速度），表中数据说明了对于不同长度密钥执行穷尽搜索所需的时间。随着大规模并行计算机的应用，处理速度

可能会高出若干个数量级。表 3.2 最后一列列举了每微秒执行 100 万次解密所需的时间。可以看出，DES 算法也不再是安全的算法。

表 3.2　不同密钥空间所耗用的时间

密钥大小(位)	密钥个数	每微秒执行一次解密所需要的时间	每微秒执行 100 万次解密所需要的时间
32	$2^{32}=4.3\times10^{9}$	$2^{31}\mu s=35.8min$	2.15ms
56	$2^{56}=7.2\times10^{16}$	$2^{55}\mu s=1142$ 年	10.01 小时
128	$2^{128}=3.4\times10^{38}$	$2^{127}\mu s=5.4\times10^{24}$ 年	5.4×10^{18} 年
168	$2^{168}=3.7\times10^{50}$	$2^{167}\mu s=5.9\times10^{30}$ 年	5.9×10^{30} 年
26 个字符的排列组合	$26!=4\times10^{26}$	$2\times10^{26}\mu s=6.4\times10^{12}$ 年	6.4×10^{6} 年

3.2.3　密钥管理学

密码算法对密码系统的安全性有决定性的作用，但很多情况下，一个密码应用系统被破解往往不是密码算法本身造成的，而是密码系统的密钥管理方案不当造成的。例如，将密钥与加密算法不加保护地一起存储在计算机中，那么任何侵入到该计算机的攻击者都能得到密码算法和密钥，相应的任何加密消息都可以被破解。再如，使用的密钥过于简单，如使用生日、电话号码等作为密钥，那么就会在本质上降低密钥的安全强度。此时，破译者只需要对那些具有特定意义的可能密钥进行穷举攻击即可。随着互联网的发展，密码学在网络安全中得到了广泛的应用。由于互联网本身是不可靠的，随时可能被攻击者监听、修改、重放数据，因此通信双方为了能达到安全通信的目的，需要在互联网本身不安全这个前提下，对安全通信所使用的密钥进行协商和管理。根据 Kerckhoff 假设，密码分析者知道所使用的密码体制，拥有除了密钥以外的所有关于加密函数的全部知识。因此密码系统的安全性完全取决于所使用的密钥的安全，密钥管理是密码系统不可缺少的重要组成部分，在密码系统中起着根本的作用。密钥管理相当复杂，既有技术问题，也有管理策略问题，从某种程度上密钥管理可以说是密码系统中最重要、最困难的部分，然而密钥管理却往往是人们最容易忽视的地方。

密钥管理包括密钥的生产、装入、存储、备份、分配、更新、吊销、销毁等内容，其中分配和存储是最棘手的问题。密钥管理的内容包括：

(1) 密钥生成。密钥生成是密钥管理的首要环节，密钥生成的主要设备是密钥生成器，密钥生成可分为集中式密钥生成和分布式密钥生成两种模式。对于前者，密钥由可信的密钥管理中心生成；对于后者，密钥由网络中的多个节点通过协商来生成。

(2) 密钥的装入和更换。密钥可通过键盘、密钥注入器、磁卡、智能卡等设备和介质装入。密钥的生命周期结束，必须更换和销毁密钥，同时密钥泄露后也必须对其进行销毁和更新。

(3) 密钥分配。主要有两种模式：集中式分配和分布式分配。集中式分配模式由一个可信的密钥管理中心给用户分发密钥，这种模式具有效率高的优点，但管理中心容易成

为攻击者的攻击目标，存在单点失效问题。分布式密钥分配模式，则由多个服务器通过协商来分配密钥，该模式能极大地提高系统的安全性和密钥的可用性。如果一个服务器被攻击，其他服务器还可以帮助被攻击的服务器恢复密钥，在灾难恢复方面具有优势。

(4) 密钥保护和存储。所有生成和分配的密钥必须具有保护措施，密钥保护装置必须绝对安全，密钥存储要保证密钥的保密性，密钥应以密文形式出现。

(5) 密钥的吊销。如果密钥丢失或因某种原因不能使用，且发生在密钥有效期内，则需要将它从正常使用的密钥集中除去，称为密钥吊销。采用证书机制的公钥可以通过吊销公钥证书实现对公钥的吊销。

(6) 密钥的销毁。不再使用的旧密钥必须销毁，否则敌手可用其解密用该密钥加密的文件，且利用旧密钥进行分析和破译密码体制。

如果对密钥进行分类，可将密钥分为主机主密钥、密钥加密密钥、会话密钥、初始密钥等类型：

(1) 主机主密钥(host master key)。对密钥加密密钥进行加密的密钥称为主机主密钥。它一般保存于网络中心、主节点或主处理器中，受到严格的物理保护。

(2) 密钥加密密钥(key encryption key)。在传输会话密钥时，用来加密会话密钥的密钥称为密钥加密密钥，也称为次主密钥(submaster key)或二级密钥(secondary key)。通信网络中各节点的密钥加密密钥应互不相同，在主机和主机之间以及主机和终端之间传送会话密钥时都需要有相应的密钥加密密钥。

(3) 会话密钥(session key)。用于通信双方交换数据时使用的密钥。根据会话密钥的用途，可分为数据加密密钥，文件密钥等。用于保护传输的数据的会话密钥叫数据加密密钥；用来保护文件的会话密钥称为文件密钥。会话密钥可以由可信的密钥管理中心分配，也可由通信方协商获得。通常会话密钥生存周期很短，一次通信结束后，该密钥就会被销毁。

现有密码系统的设计大都采用了层次化的密钥结构，这种层次化的密钥结构与对系统的密钥控制关系是对应的。一个常见的三级密钥管理层次结构如图 3.1 所示。

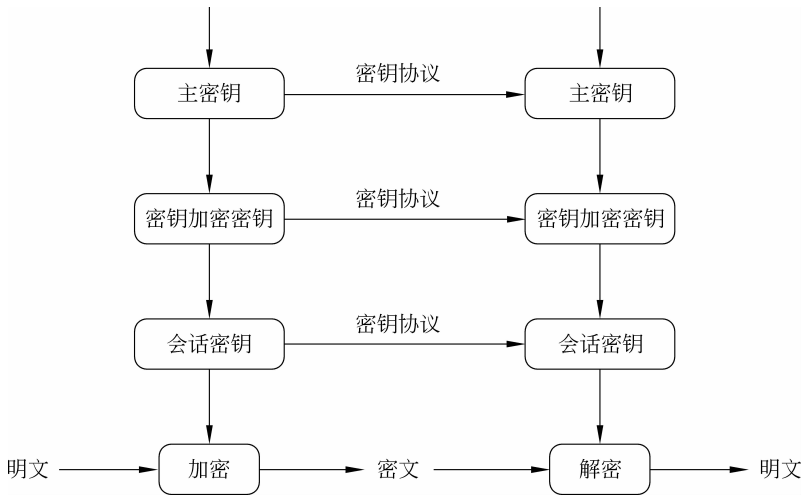


图 3.1 三级密钥管理的层次结构

密钥分级大大提高了密钥的安全性。一般来说，越低级的密钥更换越频繁，最低级的密钥可以做到一次一换。低级密钥具有相对独立性，这样，它的泄露或者破译不会影响到上级密钥的安全，而且它们的生成方式、结构、内容可以根据协议不断变换。于是，对于攻击者而言，密钥分级系统是一个动态系统，对低级密钥的攻击不会影响到高级主密钥。密钥的分级也方便了密钥的管理，使密钥管理自动化成为可能。

密钥还可以分散管理，如采用物理分散管理，将高层密钥保存在不同的地方，如 $K=KR \oplus KI$，K 为高层密钥，KR 保存在密码机内，KI 保存在密钥载体由用户保留，这样即使密码机丢失，或用户密钥载体丢失，密码机内的信息仍然由 K 加密保护。这一思想可以扩展到秘密共享机制，即使用秘密共享来分别保存主密钥。

3.3 传统密码技术

本节讨论的传统密码技术主要是指在计算机出现之前使用的加密方法，本节将对传统的密码学的典型方法进行简要的论述和总结，使读者对密码学的全貌有一个完整的印象。这些传统的密码技术在现在的密码分析技术面前，可以被轻易地破解，已经没有太多的实际意义。但传统加密技术使用的代换和置换技术的基本思想在现代密码算法设计中还有广泛的应用，了解代换和置换技术的基本思想对理解现代密码学产生的背景、为今后研究和改进现代密码系统提供了基础。

3.3.1 置换密码

置换密码根据一定的规则重新安排明文字母，使之成为密文。常用的置换密码有两种：一种是列置换密码，另一种是周期置换密码。下面给出两个例子，分别说明它们的工作情况。

例 3.1 假设有一个密钥是 type 的列置换密码，把明文 we are all together 写成 4 列矩阵，如表 3.3 所示。

表 3.3 置换密码

密钥	type
顺序	3 4 2 1
	w e a r
	e a l l
	t o g e
	t h e r

按照密钥 type 所确定的顺序，按列写出该矩阵中的字母，就得出密文：

R L E R A L G E W E T T E A O H

例 3.2 假设有一个周期是 4 的置换密码，其密钥是 $i=1,2,3,4$ 的一个置换 $f(i)=3,4,2,1$。明文同上例，加密时先将明文分组，每组 4 个字母，然后根据密钥所规定的顺序变换如下：

- 明文：M＝wear eall toge ther
- 密文：C＝A R E W　L L A E　G E O T　E R H T

单纯地置换密码因为有着与原始明文相同的字母频率特征而易被识破。如同列变换所示，密码分析可以直接从将密文排列成矩阵入手，再来处理列的位置。双字母音节和三字母分析方法可以派上用场。

多步置换密码相对来讲要安全得多，这种复杂的置换是不容易重构的。

3.3.2 代换密码

代换法是将明文中每一个字符被替换成密文中的另一个字符，接收者对密文进行逆替换就可以恢复出明文。如果把明文看做是二进制序列，那么代换就是用密文位串来代换明文位串。这里介绍两种代换密码，恺撒密码和单表代换密码。

1. 恺撒密码

已知最早的代换密码是由 Julius Caesar 发明的恺撒(Caesar)密码。它非常简单，就是对字母表中的每个字母用它之后的第三个字母来代换，例如：

- 明文：meet me after the toga party
- 密文：PHHW PH DIWHU WKH WRJD SDUWB

注意到字母表是循环的，即认为紧随 Z 后的是字母 A。通过列出所有的可能来定义如下变换：

- 明文：a b c d e f g h i j k l m n o p q r s t u v w x y z
- 密文：D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

如果让每个字母等价于一个数值：

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

那么加密算法可以这样表达：对每个明文字母 p，代换成密文字母 C

$$C=E(3,p)=(p+3)\bmod 26$$

移位可以是任意整数 k，这样就得到了一般的 Caesar 算法：

$$C=E(k,p)=(p+k)\bmod 26$$

这里 k 的取值范围为 1～25。解密算法是：

$$p=D(k,C)=(C-k)\bmod 26$$

如果已知某给定的密文是恺撒密码，那么穷举攻击是很容易实现的：只要简单地测试所有的 25 种可能的密钥。

恺撒密码的三个重要特征可以采用穷举攻击分析方法：

- 已知加密算法和解密算法。
- 需测试的密钥只有 25 个。
- 明文所用的语言是已知的，且其意义易于识别。

在大多数网络情况下，假设密码算法是已知的。一般来说，密钥空间很大的算法使穷举攻击分析方法不太可能成功。例如，第 6 章介绍的 3DES 算法，它的密钥长度是 168 位，其密钥空间是 2^{168}，或者说有大于 3.7×10^{50} 种可能的密钥。

上述第三个特征也是非常重要的，如果明文所用的语言不为所知，那么明文输出就不

可识别。而且，输入可能按某种方式经过缩写或压缩，也就更不可识别了。例如下面是经过 WINRAR 压缩之后的部分文本文件：

燴/$ 蒜 J;\??B.?Q ~蘞)┼ 4 戢 OWA??.◀ w 悇碇}rw 霪?o 疲｜歜傛 f 顃溺板 r?R 鼫=妈梨柑 I50$ 澹 t?+#?j9W?鲜`砚 O0-+┬ ┌ 痄 ≯QO 皮 {饇庇郻?6)鱞琢?`d3??璩?お 姓肱阜 B 太｜蜇 N 衡?瘓-詖?Z 痟?d 救?d 胜塒 s 攇+r~杷*镌煮'? 媮└ \p 軯┘ e 謓 vA -永 楈有 q W?!u 佯 H 酛?袧镔裉~锻%

如果这个文件用一种简单的代换密码来加密(将字母集合扩充为不止包含 26 个英文字母)，那么即使用穷举攻击进行密码分析，恢复出来的明文也是不可识别的。

2. 单表代换密码

恺撒密码仅有 25 种可能的密钥，远远不够安全。通过允许任意代换，密钥空间将会急剧增大。回忆恺撒密码的对应：如果密文行是 26 个字母的任意置换，那么就有 26！或大于 4×10^{26} 种可能的密钥，这比 DES 的密钥空间要大 10 个数量级，凭直觉这样的密钥空间应该可以抵挡穷举攻击了。这种方法称为单表代换密码，这是因为每条消息用一个字母表(给出从明文字母到密文字母的映射)加密。攻击者尝试所有的 4×10^{26} 种可能密钥的工作量显然太大了，有没有更好的攻击方法呢？

答案是更好的攻击方法仍然存在。如果密码分析者知道明文(例如，未经压缩的英文文本)的属性，他就可以利用语言的一些规律进行攻击。为了说明分析过程，这里给出一段文字。需要解密的密文是：

PBFPVYFBQXZTYFPBFEQJHDXXQVAPTPQJKTOYQWIPBVWLXTOXBTFX
QWAXBVCXQWAXFQJVWLEQNTOZQGGQLFXQWAKVWLXQWAEBIPBFXF
QVXGTVJVWLBTPQWAEBFPBFHCVLXBQUFEVWLXGDPEQVPQGVPPBFTI
XPFHXZHVFAGFOTHFEFBQUFTDHZBQPOTHXTYFTODXQHFTDPTOGHFQ
PBQWAQJJTODXQHFOQPWTBDHHIXQVAPBFZQHCFWPFHPBFIPBQWKF
ABVYYDZBOTHPBQPQJTQOTOGHFQAPBFEQJHDXXQVAVXEBQPEFZBVF
OJIWFFACFCCFHQWAUVWFLQHGFXVAFXQHFUFHILTTAVWAFFAWTE
VOITDHFHFQAITIXPFHXAFQHEFZQWGFLVWPTOFFA

首先把字母使用的相对频率统计出来，与英文字母的使用频率分布进行比较，参见图 3.2和图 3.3。

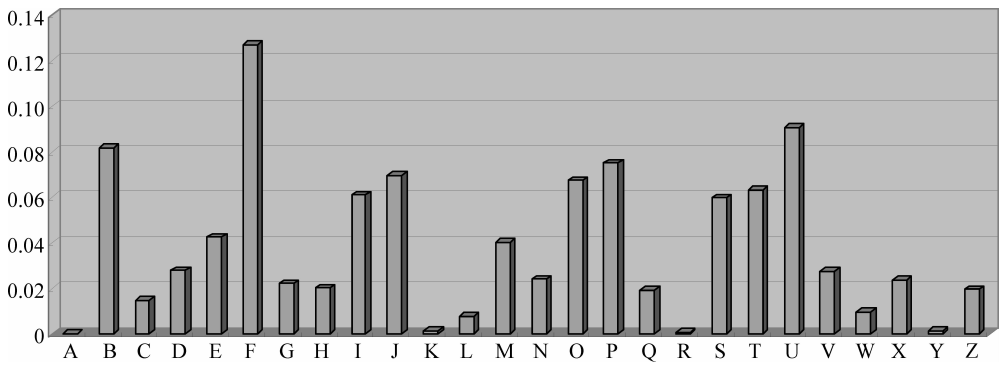


图 3.2 常见的英文字母使用频率

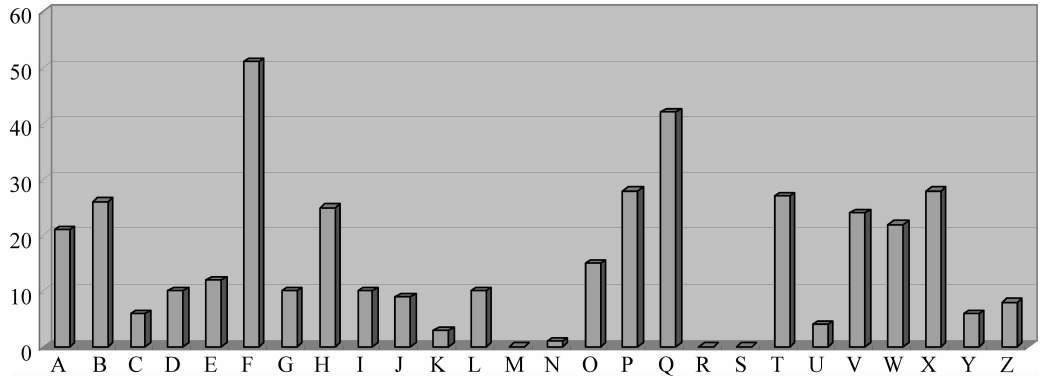


图 3.3 密文中字母出现的次数

如果已知消息足够长，只用这种方法就已经足够了，如果这段消息相对较短，就不能得到准确的字母匹配。将这种统计规律与图 3.2 比较，可以得出结论：密文字母中的 F 或 Q 可能相当于明文中的 e，但是并不能确定 F 对应 e 还是 Q 对应 e。密文中的 P、W、B、H、P、T 和 X 相对频率都比较高，可能与明文中的字母集{t,a,h,i,n,o,r,s}中的某一个相对应。相对频率较低的字母(M,R,S,K,U,Y)可能对应着明文字母集{b,j,k,q,v,x,z}中的某个元素。

据此我们可以从以下几种方法入手。可以尝试着做一些代换，填入明文，看看是否像一个消息的轮廓，更系统一点的方法是寻找其他的规律。例如，明文中有某些词可能是已知的，或者寻找密文字母中的重复序列，推导它们的等价明文。

统计双字母组合的频率是一个很有效的工具。由此可以得到一个类似于图 3.2 的双字母组合的相对频率图。最常用的一个字母组合是 th。而在密文中，用得最多的双字母组合是 PB，它出现了多次。所以可以估计 P 对应明文 t，而 B 对应明文 h。根据先前的假设，可以认为 F 对应 e。现在我们意识到密文中的 PBF 很可能就是 the，这是英语中最常用的三字母组合，这表明我们的思路是正确的。

3.3.3 一次一密

一次一密建议使用与消息一样长且无重复的随机密钥来加密消息，另外，密钥只对一个消息进行加解密，之后丢弃不用。每一条新消息都需要一个与其等长的新密钥。一次一密是不可攻破的，它产生的随机输出与明文没有任何统计关系。因为密文不包含明文的任何信息，所以无法攻破。

下面的例子能够说明我们的观点。假设我们使用的是 27 个字符(第 27 个字符是空格)的密码，但是这里使用的一次性密钥和消息一样长，请看下面的密文：

ANKYODKYUREPFJBYOJTDSPLREYIUNOFDOIUERFPLUYTS

现在我们用两种不同的密钥解密。

（1）密钥1的解密结果：

密文	ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
密钥1	pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
明文	mr mustard with the candlestick in the hall

（2）密钥2的解密结果：

密文	ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
密钥2	mfugpmiydgaxgoufhkllmhsqrdqogtewbqfgyovuhwt
明文	miss scarlet with the knife in the library

假设密码分析者已设法找到了这两个密钥，于是就产生了两个似是而非的明文。分析者如何确定正确的解密呢（即正确的密钥）？如果密钥是在真正随机的方式下产生的，那么分析者就不能说密钥更有可能是哪一种。因此没有办法确定正确的密钥，也就是说，没有办法确定正确的明文。

事实上，给出任何长度与密文一样长的明文，都存在着一个密钥产生这个明文。因此用穷举法搜索所有可能的密钥，就会得到大量可读、清楚的明文，但是没有办法确定哪一个才是真正所需的，因而这种密码是不可破的。

一次一密的安全性完全取决于密钥的随机性。如果构成密钥的字符流是真正随机的，那么构成密文的字符流也是真正随机的。因此分析者没有任何攻击密文的模式和规则可用。

理论上，对一次一密已经讲得很清楚了。但是在实际中，一次一密提供完全的安全性存在两个基本难点：

（1）产生大规模随机密钥有实际困难。任何经常使用的系统都需要建立在某个规则基础上的数百万个随机字符，提供这样规模的真正随机字符是相当艰巨的任务。

（2）更令人担忧的是密钥的分配和保护。对每一条发送的消息，需要提供给发送方和接收方等长度的密钥。因此，存在庞大的密钥分配问题。因为上述这些困难，一次一密实际很少使用，主要用于安全性要求很高的低带宽信道。

3.3.4 转轮机

多步置换得到的算法对密码分析有很大的难度，这对代换密码也适用。20世纪20年代，人们就发明机械加密设备用来自动处理加密，大多数是基于转轮的概念，机械转轮用线连起来完成通常的密码代换。

轮转机有一个键盘和一系列转轮（见图3.4），每个转轮是字母的任意组合，有26个

位置，并且完成一种简单代换。例如，一个转轮可能被用线连起来以完成用 K 代换 A，用 W 代换 D，用 L 代换 T 等，而且转轮的输出栓连接到相邻的输入栓。

例如，有一个密码机，有 4 个转轮，第一个转轮可能用 G 代换 B，第二个转轮可能用 N 代换 G，第三个转轮可能用 S 代换 N，第四个转轮可能用 C 代换 S，C 应该是输出密文。当转轮移动后，下一次代换将不同。为使机器更加安全，可以把几种转轮和移动的齿轮结合起来。因为所有的转轮以不同的速度移动，n 个转轮的机器周期为 $26n$。为进一步阻止密码分析，有些转轮机在每个转轮上还有不同的位置号。

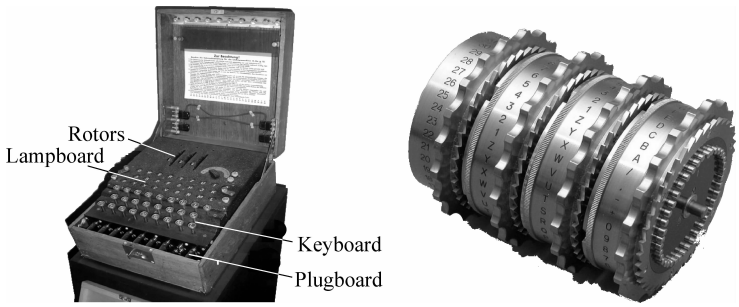


图 3.4　转轮机和其中的机械转轮

今天转轮机的意义在于它曾经给最为广泛使用的密码——数据加密标准 DES 指明了方向。第 4 章对 DES 进行讨论。

3.3.5　电码本

近年热播的电视剧《潜伏》中会出现这样的镜头，当男主角余则成收到密电后，会拿出一本书对密电进行解密，这本书就是加密解密使用的电码本，如果这本书被敌人知道了，密电中的机密则会泄露。从字面中意义而言，传统的电码本密码就是像字典一样的书，包含单词和其相应的码字。表 3.4 给出了德国在一战期间使用的一部著名电码本密码的摘录。

例如，要加密德文单词 Februar，整个单词被替换为 5 位“码字”13605。表 3.4 的电码本用于加密。相应的还有一个将 5 位码字按照数字大小顺序排列的电码本，用于解密。电码本密码属于代替密码，但是与简单代替密码有很大的区别，因为这里是对整个单词甚至短语进行代替。

表 3.4 展示的电码本被用于著名的 Zimmermmann 电报的加密中，在 1917 年，德国外交部长 Arthur Zimmermann 给身在墨西哥城的德国大使发送一个加密的电报（见图 3.5），这份密电文被英国截获，当时英国和法国正在同德国及其同盟国作战，而美国处于中立。

表 3.4　德国电码本摘录

明文	密文
Februar	13605
fest	13732
finanzielle	13850
folgender	13918
Frieden	17142
Friedenschluss	17149
⋮	⋮

俄国破译出了德国电码本的部分内容，并将一部分电码本送给英国。经过艰苦的分析，英国恢复出了足够破译 Zimmermann 电报的电码本。电报陈述了德国政府计划开展“无限制潜艇战”，并且预测到这个计划将导致美国卷入战争。因此 Zimmermann 决定德国应该试图拉拢墨西哥加入同盟并与美国作战，如图 3.6 所示。德国对于墨西哥承诺“夺回其在德克萨斯州、新墨西哥州和亚利桑那州曾经失去的领土”。当解密后的 Zimmermaan 电报被公布给美国后，美国公众开始敌对德国；随后在 Lusitania 航线事件后，美国对德国正式宣战。

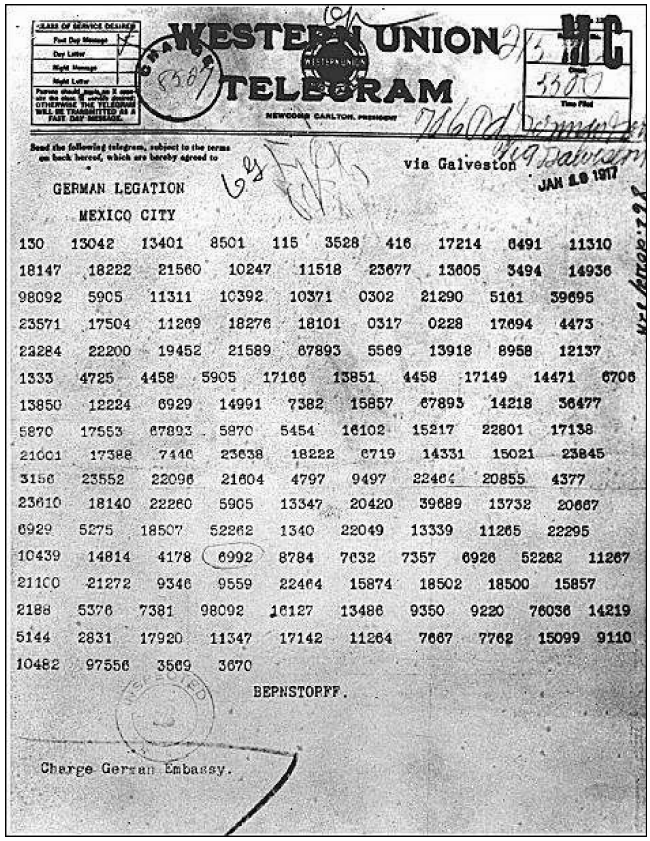
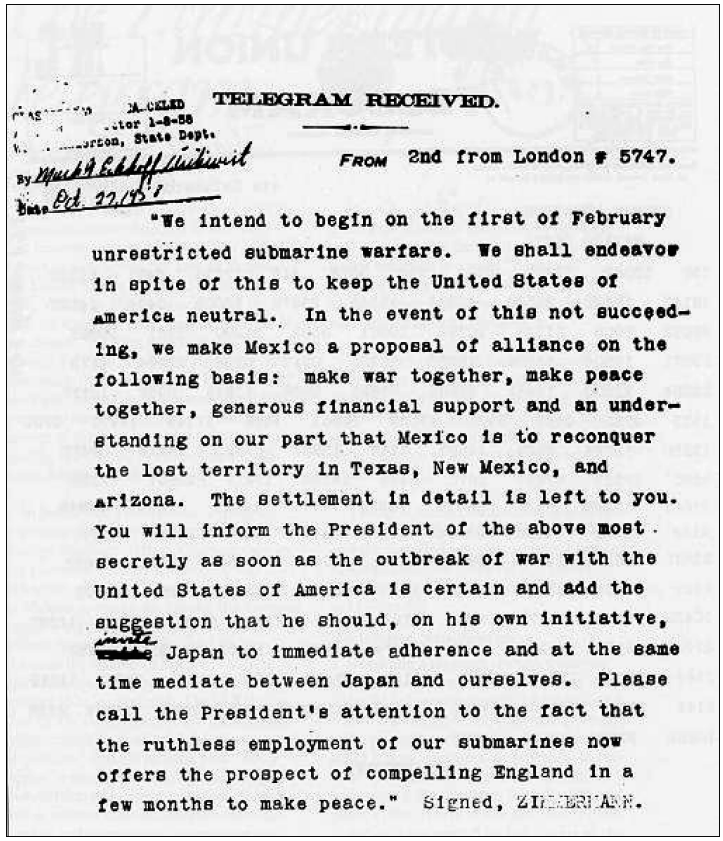

WESTERN UNION
TELEGRAM
via Galveston
JAN 19 1917
GERMAN LEGATION
MEXICO CITY
130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
23284 22200 19452 21589 67893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7110 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3670
BEPNSTORFF.
Charge German Embassy.

图 3.5　德国 Zimmermann 密电

最初，英国曾犹豫是否公布 Zimmermann 电报，因为英国害怕德国发现他们的密码被破译以后就停止使用该密码。然而，通过在 Zimmermann 电报发送的同时期对其他电报进行过滤分析，英国破译者发现还有 Zimmermann 电报的未加密版本被发送。英国随后公布了同未加密版本电报内容相似的 Zimmermann 电报。于是德国就以为他们的电码本密码没有问题，并在整个一战中继续使用其加密敏感信息。

电码本和现代分组密码有着一定的相关性，现代分组密码对明文使用复杂的算法产生密文（反之亦然），但是从宏观上来看分组密码都可以视为一个电码本密码，这里每个密钥都确定一部不同的电码本。



TELEGRAM RECEIVED.

FROM 2nd from London # 5747.

"We intend to begin on the first of February unrestricted submarine warfare. We shall endeavor in spite of this to keep the United States of America neutral. In the event of this not succeeding, we make Mexico a proposal of alliance on the following basis: make war together, make peace together, generous financial support and an understanding on our part that Mexico is to reconquer the lost territory in Texas, New Mexico, and Arizona. The settlement in detail is left to you. You will inform the President of the above most secretly as soon as the outbreak of war with the United States of America is certain and add the suggestion that he should, on his own initiative, invite Japan to immediate adherence and at the same time mediate between Japan and ourselves. Please call the President's attention to the fact that the ruthless employment of our submarines now offers the prospect of compelling England in a few months to make peace." Signed, ZIMMERMANN.

图 3.6 用电码本解密的 Zimmermann 电报

习题

1. 假设已知使用恺撒密码，试从下面的密文恢复出明文：

   VSRQJHEREVTXDUHSDQWU

2. 如果拥有一台每秒可以尝试 2^{40} 次密钥的计算机，那么对于密钥空间为 2^{128} 进行密钥的穷举搜索大概需要多长时间(以年为单位)？

3. 请简述密码学在计算机网络安全的重要作用。

4. 密码管理中使用层次化的密钥结构的作用是什么？

5. 设计一个电码本密码的计算机版本。该密码应包含许多可能的电码本，使用密钥以确定使用哪个电码本对特定消息进行加密或解密。

第4章　对称加密

密码学发展到现在，经历了很多阶段，有很多种算法和应用协议，我们可以将它们分为两类：对称密码系统（或单钥密码系统）和公钥密码系统（或非对称密码系统、双钥密码系统）。

对称加密也称为常规加密、私钥或单钥加密，它的特点是加密和解密时所使用的密钥是相同的或者类似的，即由加密密钥可以很容易得到解密密钥，反之亦然。正因为如此，我们常称其为对称密码系统或单钥密码系统。在一个密码系统中，我们不能假定加密算法和解密算法是保密的，因此密钥必须保密。然而发送消息的通道往往是不安全的，所以在对称密码系统中，通常要求使用不同于发送消息的另外一个安全通道来发送密钥。

与之相反，公钥密码系统却具有不同的特点和优点：加密密钥和解密密钥在算法上是不同的，知道其中一个密钥，也不能有效地推导出另一个密钥，所以公钥密码系统常常称为非对称密码系统或双钥密码系统。因此可以公开加密密钥，不需要额外的安全信道来分发密钥，这样无损于整个系统的保密性，用户只需要保存好解密密钥，这也是公钥密码系统名称的来源。

在20世纪70年代末期公钥加密开发之前，对称加密是唯一被使用的加密类型。现在，它仍然属于使用最广泛的两种加密类型之一，有些读者误认为对称密码已经被公钥密码取代，这是一种错误的认识。对称密码在加解密的速度上比公钥密码要快得多，显然对大量数据进行加解密时，对称密码是首选。

对称密码通常通过流密码和分组密码来实现，本章首先介绍对称流密码，并描述广泛使用的流密码RC4。然后会探讨三种重要的分组加密算法：DES、3DES以及AES。接着讨论随机和伪随机数的生成。最后，介绍分组密码的工作模式这一重要内容。

4.1　流密码

流密码连续处理输入元素，在运行过程中，一次产生一个输出元素。在流密码中，将明文消息按一定长度分组（长度较小，如一个字节），然后对各组用相关但不同的密钥进行加密，产生相应的密文，相同的明文分组会因在明文序列中的位置不同而对应于不同的密文分组。在分组密码中，明文消息也是按一定长度分组（长度较大），每组都使用完全相同的密钥进行加密，产生相应的密文，相同的明文分组不管处在明文序列的什么位置，总是对应相同的密文分组。相对分组密码而言，流密码主要有以下优点：第一，在硬件实施上，流密码的速度一般要比分组密码快，而且不需要有很复杂的硬件电路；第二，在某些情况下（例如，对某些电信上的应用，如GSM移动通信中使用），当缓冲不足或必须对收到的字符进行逐一处理时，流密码就显得更加必要和恰当；第三，流密码有较理想的数学分析工具，如频谱理论和技术、代数方法等；第四，流密码能较好地隐藏明文的统计特征。

尽管分组密码使用得更为普遍，但是对于一些特定的应用，使用流密码更合适，例如在GSM移动通信中用于保护数据机密性的A5/1算法。本节首先概述流密码结构，然后研究流行的对称流密码RC4。

4.1.1 流密码结构

目前关于流密码的理论和技术已取得长足的发展。同时密码学家也提出了大量的流密码算法，有些算法已被广泛地应用于移动通信、军事外交等领域。

流密码的基本原理是在流密码中，明文按一定长度分组后被表示成一个序列，并称为明文流，序列中的一项称为一个明文字。加密时，先由主密钥产生一个密钥流序列，该序列的每一项和明文字具有相同的比特长度，称为一个密钥字。然后依次把明文流和密钥流中的对应项输入加密函数，产生相应的密文字，由密文字构成密文流输出，即

- 设明文流为：$M=m_1\ m_2\cdots m_i\cdots$
- 密钥流为：$K=k_1\ k_2\cdots k_i\cdots$
- 通常密钥流由流密码的函数生成：$\text{StreamCipher}(K)=k_1,k_2,\cdots,k_i,\cdots$
- 加密算法为：$C=c_1\ c_2\cdots c_i\cdots=E_{k1}(m_1)E_{k2}(m_2)\cdots E_{ki}(m_i)\cdots$
- 解密算法为：$M=m_1\ m_2\cdots m_i\cdots=D_{k1}(c_1)D_{k2}(c_2)\cdots D_{ki}(c_i)\cdots$

假设发送者和接受者都拥有相同的流密码算法，并且都知道密钥K，那么这个系统就构成了一个现实的“一次一密”，尽管不像真正的一次一密那样具有可证明的安全性。

流密码与分组密码在对明文的加密方式上是不同的。分组密码对明文进行处理时，明文分组相对较大，所有的明文分组都是用完全相同的函数和密钥来加密的。而流密码对明文消息进行处理时，采用较小的分组长度，对明文流中的每个字用相同的函数和不同的密钥字来加密。

图4.1展示了典型的流密码结构。在这个结构里，密钥输入到一个伪随机字节生成器，产生一个表面随机的8比特数的流。伪随机流是不知道输入密钥就不可预测的，有表面上随机的性质。这个生成器的输出称为密钥流，使用位异或操作与明文流结合，一次一个字节。例如，如果生成器产生的下一字节是01101100，明文的下一个字节是11001100，那么得到的密文字节是：

```
  11001100    明文
⊕ 01101100    密钥流
--------------
  10100000    密文
```

解密需要使用之前加密使用过的同一密钥流：

```
  10100000    密文
⊕ 01101100    密钥流
--------------
  11001100    明文
```

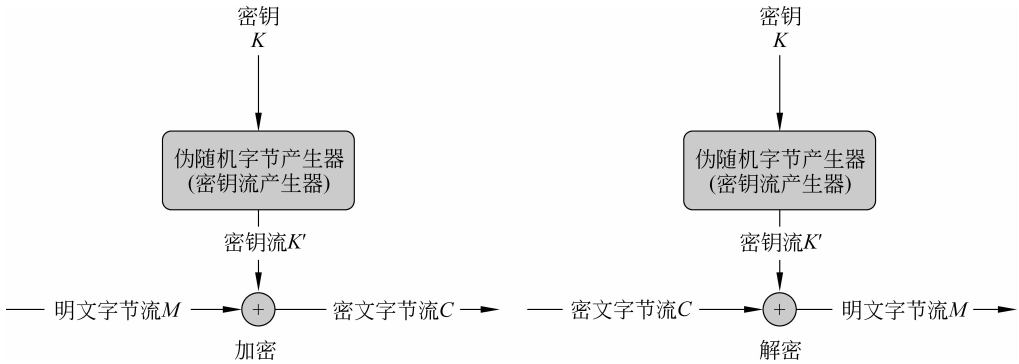


图 4.1　流密码图

设计流密码时主要考虑的因素有：

(1) 加密序列应该有一个长周期。伪随机数生成器使用一个函数产生一个实际上不断重复的确定比特流。这个重复的周期越长，密码破解就越困难。

(2) 密钥流应该尽可能地接近真随机数流的性质。例如，1 和 0 的数目应该近似相等。如果将密钥流视作字节流，那么每个字节的 256 种可能值出现的频率应该近似相等。密钥流表现得越随机，密文就越随机化，密码破译就越困难。

(3) 伪随机数生成器的输出受输入密钥值控制。为了抵抗穷举攻击，这个密钥必须非常长。分组密码中的考虑因素在这里同样适用。因此，就当前的科技水平而言，需要至少 128 比特长度的密钥。

如果伪随机数生成器设计合理，对同样的密钥长度，流密码和分组密码一样安全。流密码的主要优点是流密码与分组密码相比总是更快，而且使用更少的代码，如 RC4 算法能用仅仅几行代码实现。表 4.1 将 RC4 与三个知名的对称分组密码的执行时间进行了比较。分组密码的优点是可以重复使用密钥。如果两个明文使用同一密钥进行流密码加密，密码破译常常会非常容易。如果将这两个密文流进行异或，结果就是原始明文的异或值。如果明文是文本字符串、信用卡号或者其他已知其性质的字节流，密码破解可能会成功。

表 4.1　对称密码速率比较　　使用 CPU：奔腾 Ⅱ

密码	密钥长度	速率(Mbps)	密码	密钥长度	速率(Mbps)
DES	56	9	RC2	可变	0.9
3DES	168	3	RC4	可变	45

对于需要加密/解密数据流的应用，例如在数据通信信道或者浏览器/网络链路上，流密码也许是更好的选择。对于处理数据分组的应用，如文件传递、电子邮件和数据库，分组密码可能更合适。但是，这两种密码都可以在几乎所有的应用中使用。

4.1.2 RC4 算法

RC4 是 RSA 三人组中的头号人物 Ron Rivest 在 1987 年为 RSA Security 公司设计的流密码。它是密钥大小可变的流密码，是由于其核心部分的状态向量 ***S*** 长度可为 1～256 字节，但一般为 256 字节。该算法的速度可以达到 DES 加密的 10 倍左右，且具有很高级别的非线性。RC4 原本被 RSA Security 公司当作商业秘密，但是在 1994 年 9 月，RC4 算法通过 Cypherpunks 匿名邮件发送列表匿名地公布在互联网上，也就不再有什么商业机密。RC4 也称作 ARC4(Alleged RC4——所谓的 RC4)，因为 RSA 从来就没有正式发布过这个算法。SSL/TLS 标准中使用了 RC4 (Secure Sockets Layer/Transport Layer Security，安全套接字层/传输层安全)，该协议可以为网络浏览器和服务器之间提供安全的通信。RC4 也被用于属于 IEEE 802.11 无线局域网标准一部分的 WEP(Wired Equivalent Privacy，有线等效隐私)协议及更新的 WPA(WiFi Protected Access，WiFi 保护访问)协议。

RC4 算法非常简单，从本质上讲，它就是一个 256 字节的置换的查表，在产生密钥流的每一个字节的时候，所查的表就进行一次修改。整个 RC4 都是基于字节的。算法的第一个阶段是对于查表使用的密钥进行初始化，用一个可变长度为 1～256 字节(8～2048 比特)的密钥来初始化 256 字节的状态向量 ***S***，其元素为 $S[0]$，$S[1]$，…，$S[255]$。任何时候，S 都包含所有 0～255 的 8 比特数的排列组合。加密和解密时，一个字节 k 由 ***S*** 产生，通过系统的方式从其 255 个元素中选取一个。每次 k 值产生之后，要再次排列 ***S*** 的元素。

初始化 ***S***。开始时，***S*** 的元素设为等于 0～255 的升序值；即 $S[0]=0$，$S[1]=1$，…，$S[255]=255$。同时创建一个临时向量 ***T***。如果密钥 K 的长度为 256 比特，就把 K 直接赋给 ***T***，否则，对于 keylen 字节长度的密钥，从 K 复制 ***T*** 的前 keylen 个元素，然后一直重复 K 直到填满 ***T***。可以把这些预备操作概括如下：

```
/* 初始化* /
for i=0 to 255 do
S[i]=i;
T[i]=K[i mod keylen];
```

接下来我们使用 ***T*** 来产生 ***S*** 的初始排列。它从 $S[0]$开始一直处理到 $S[255]$，同时对每个 $S[i]$，根据 $T[i]$指定的方案将 $S[i]$与 ***S*** 中的另一个字节交换：

```
/* s 的初始排列* /
j=0;
for i=0 to 255 do
j=(j+S[i]+T[i]) mod 256;
swap (S[i],S[j]);
```

因为对 ***S*** 的唯一操作是交换，其唯一的作用是排列组合。***S*** 仍然包含 0～255 的所