第3章

物联网安全密码学基础

本章要点

密码学概论(密码发展史、密码系统、密码分类)
典型对称密码算法 AES
典型非对称密码算法 RSA
国产密码算法

导入案例

为中国革命胜利发挥重要作用的“豪密”

1929 年末,中国共产党在上海的第一部秘密地下电台建立。同年 12 月,香港电台建立。1930 年 1 月,上海与香港电台之间第一次通报成功,这是党历史上第一次无线电通报成功。当时使用的是一种“明码颠倒”的加密方式,香港电台很快被英国殖民政府破坏。周恩来得到电报被破译的消息后,非常震惊,立刻决定重新编制密码。周恩来提出编码思想后,汇总集体智慧,亲自动手,终于编制出了一套密码,也是中共第一部高级密码,因周恩来在党内代号是“伍豪”,这套密码便被称作“豪密”。

周恩来亲自编制的这套密码,从未被敌人破译。因为在密码破译上,全世界都有一个共同的破译规律,就是寻找重复,当无法破译的时候,就把各封电报搜集在一起,然后通过重复的码来寻找规律。而一次一密的“豪密”是不会重复的。“豪密”只是一种简单的密码,是一种“底本”加“乱数”的密码。所谓的“底本”就是类似明码一样的单表代替式密码本,而“乱数”是编成的表,若干随机编排的数字。然后再加上“加减法”一样的“算法”进行加密。例如,“我在等你”的电码是“1234-2345-3456-4567”,如果后面加上一组乱码“9876-8765-7654-6543”,用上加法之后,则发送的时候就是“0000-0000-0000-0000”。换言之,这就好比现在的支付工具,除了要有支付密码外,还要有短信发来的“验证码”,尽管支付密码是固定的,但是验证码是随机的,两个密码同时发挥作用才能成功。

周恩来创造的这套密码体制为中国革命的胜利发挥了特别的作用,一直到解放战争结束,国民党方面的专家也未能破译“豪密”。正是在它的帮助下,“龙潭三杰”才能一次次将宝

贵的情报安全传递到党中央，使得我党在对敌无线电斗争中始终掌握主动权，为党的战略布局赢得先机。

基于此背景材料，你是否对密码学产生了浓厚兴趣，那让我们开启本章的学习之旅吧！

3.1　密码学概述

密码学(Cryptography)是一种为信息安全提供解决方案的学科，是数学的一个分支，在物联网诸多应用场景中发挥巨大作用。物联网中广泛采用的加密、消息认证、数字签名等都是基于密码学的。例如，大多数物联网终端设备都处于开放的网络环境下，为确保安全性，需要利用加密算法，对物联网设备之间传输的数据进行加密，防止攻击者窃取数据，从而保证信息的保密性；通过消息认证技术，可以判断物联网设备之间传输的消息是否被篡改，从而保证信息的完整性；验证收到的消息是否真的来自发送设备，保证消息的可认证性等。Cryptography 一词来源于古希腊语的 crypto 和 graphen，意思是密写，它以认识密码变换为本质，以加密与解密基本规律为研究对象，包括密码编码学和密码分析学。其中，密码编码学是研究各种加密方案的科学；密码分析学是研究密码破译的科学。

3.1.1　密码发展史

密码技术最早源于战争需求，从某种意义上说，战争是科学技术进步的催化剂。人类自从有了战争，就面临着通信安全的需求，许多古代文明，包括埃及人、希伯来人、亚述人都在实践中逐步发明了密码系统。存于石刻或史书中的记载表明，密码技术源远流长。密码学的发展历史大致经历了四个阶段：古代加密方法、古典密码、现代密码和量子加密。

1. 古代加密方法(手工阶段)

古代加密方法大约起源于公元前 440 年，出现在古希腊战争中的隐写术，当时为了安全传送军事情报，奴隶主剃光奴隶的头发，将情报写在奴隶的光头上，待头发长长后将奴隶送到另一个部落，再次剃光头发，原有的信息复现出来，从而实现这两个部落之间的秘密通信。我国古代也早有以藏头诗、藏尾诗、漏格诗及绘画等形式，将要表达的真正意思或“密语”隐藏在诗文或画卷中特定位置，一般人只注意诗或画的表面意境，而不会去注意或很难发现隐藏其中的“话外之音”。比如，我画蓝江水悠悠，爱晚亭上枫叶愁。秋月溶溶照佛寺，香烟袅袅绕轻楼。(藏头诗)。

最早的密码技术来源于公元前 2000 年。希伯来人的一种加密方法是把字母表调换顺序，这样的字母表的每一个字母就被映射成调换顺序后的字母表中的另一个字母，这种加密方法被称为 Atbash。例如，单词 security 就被加密成 hvxfirgb，这是一种代换密码，因为一个字母被另一个字母所代替。这种代换密码被称为单一字母替换法，因为它只使用一个字母表，而其他加密方法一次用多个字母表，则称为多字母替换法。

公元前 400 年，斯巴达人发明了“塞塔式密码”，即把长条纸螺旋形地斜绕在一个多棱棒上，将文字沿棒的水平方向从左到右书写，写一个字旋转一下，写完一行再另起一行从左到右写，直到写完。解下来后，纸条上的文字消息杂乱无章、无法理解，这就是密文，但将它绕在另一个同等尺寸的棒子上后，就能看到原始的消息。

后来，朱利叶斯·恺撒发明了一种近似于 Atbash 替换字母的方法。当时，没多少人能

够第一时间读懂，这种方法提供了较高的机密性。中世纪，欧洲人不断利用新的方法、新的工具和新的实践优化自己的加密方案。在19世纪晚期，密码学已经被广泛地用作军事上的通信方法。

2. 古典密码（机械阶段）

古典密码的加密方法是以单个字母为作用对象的加密法，一般是文字置换，使用手工或机械变换的方式实现。古典密码系统已经初步体现出近代密码系统的雏形，它比古代加密方法复杂，其变化较小。

图 3-1　Enigma 密码机

随着机械和电子技术的发展，电报和无线通信的出现，加密装置得到了突飞猛进的提高，转子加密机是军事密码学上的一个里程碑，这种加密机是在机器内用不同的转子来替换字母，它提供了很高的复杂性，从而很难攻破。德国的 Enigma 密码机是历史上最著名的加密机，如图 3-1 所示。这种机器有三个转子、一个线路连接板和一个反转转子。在加密开始之前，消息产生者将 Enigma 密码机配置成初始设置，操作员把消息的第一个字母输入加密机，加密机用另一个字母来代替并把这个字母显示给操作员看。Enigma 密码机的加密机制是：通过把转子旋转预定的次数，用另一个不同的字母来代替原来的字母。因此，如果操作员把 T 作为第一个字符敲入机器中，Enigma 密码机可能会把 M 作为密文，操作员就把字母 M 写下来，然后他可以加快转子的速度再输入下一个字符，每加密一个字符操作员就加快转子的速度作为一个新的设置。继续这样下去，直到整个消息被加密。然后，加密的密文通过电波传输，大部分情况是传到潜水艇。这种对每个字母有选择性地替换依赖于转子装置，因此这个过程的关键和秘密的部分（密钥）在于在加密和解密的过程中操作员是怎样加速转子的。两端的操作员需要知道转子的速度增量顺序以使得德国军事单位能够正确地通信。尽管 Enigma 密码机的装置在当时非常复杂，但还是被一组波兰密码学家攻破，从而使得英国知道了德国的进攻计划和军事行动。有人说，Enigma 密码机的破译使第二次世界大战缩短了两年。

3. 现代密码（计算机阶段）

前面介绍的古代加密方法和古典密码，对它们的研究还称不上是一门科学。直到 1949 年香农发表了一篇题为“保密系统的通信理论”的著名论文，该论文首先将信息论引入了密码，从而把已有数千年历史的密码学推向了科学的轨道，奠定了密码学的理论基础。该论文利用数学方法对信息源、密钥源、接收和截获的密文进行了数学描述和定量分析，提出了通用的密钥密码体制模型。需要指出的是，由于受历史的局限，20 世纪 70 年代中期以前的密码学研究基本上是秘密地进行，而且主要应用于军事和政府部门。密码学的真正蓬勃发展和广泛应用是从 20 世纪 70 年代中期开始的。1977 年美国国家标准局颁布了数据加密标准（Data Encryption standard，DES）用于非国家保密机关。该系统完全公开了加密、解密算法。此举突破了早期密码学的信息保密的单一目的，使得密码学得以在商业等民用领域广

泛应用，从而给这门学科以巨大的生命力。

在密码学发展进程中的另一件值得注意的事件是，在 1976 年，美国密码学家迪菲和赫尔曼在一篇题为“密码学的新方向”一文中提出了一个崭新的思想，即不仅加密算法本身可以公开，甚至加密用的密钥也可以公开。但这并不意味着保密程度的降低。因为如果加密密钥和解密密钥不一样。而将解密密钥保密就可以。这就是著名的公钥密码体制。若存在这样的公钥密码体制，就可以将加密密钥像电话簿一样公开，任何用户当他想向其他用户传送一个加密信息时，就可以从这本密钥簿中查到该用户的公开密钥，用它来加密，而接收者能用只有他所具有的解密密钥得到明文。任何第三者不能获得明文。1978 年，美国麻省理工学院的李维斯特、萨英尔和阿德曼提出了 RSA 公钥密码体制，它是第一个成熟的、迄今为止理论上最成功的公钥密码体制。它的安全性是基于数论中的大整数因子分解，该问题是数论中的一个困难问题，至今没有有效的算法，这使得 RSA 公钥密码体制具有较高的保密性。关于 RSA 公钥密码体制将在 3.3 节详细介绍。

按照人们对密码的一般理解，密码是用于将信息加密而不易破译，但在现代密码学中，除了信息保密外，还有另一方面的要求，即信息安全体制还要能抵抗对手的主动攻击。所谓主动攻击指的是攻击者可以在信息通道中注入他自己伪造的消息，以骗取合法接收者的相信。主动攻击还可能篡改信息，也可能冒名顶替，这就产生了现代密码学中的认证体制。该认证体制的目的就是保证用户收到一个信息时，他能验证消息是否来自合法的发送者，同时还能验证该信息是否被篡改。在许多场合中，如电子汇款，能对抗主动攻击的认证体制甚至比信息保密还重要。在密码学的发展过程中，数学和计算机科学至关重要，数学中的许多分支如数论、概率统计、近世代数、信息论、椭圆曲线理论、算法复杂性理论、自动机理论、编码理论等都可以在其中找到各自的位置。密码学形成一门新的学科是受计算机科学蓬勃发展刺激和推动的结果。快速电子计算机和现代数学方法一方面为加密技术提供了新的概念和工具，但另一方面也给破译者提供了有力武器。计算机和电子学时代的到来给密码设计者带来了前所未有的自由，他们可以轻易地摆脱原先用铅笔和纸进行手工设计时易犯的错误，也不用再面对用电子机械方式实现的密码机的高额费用。总之，利用电子计算机可以设计出更为复杂的密码系统。

4. 量子加密

量子加密技术，是利用量子原理，进行密钥的生成、明文的混淆加密、密文的还原解密、密文的通信、反窃听等一系列加密技术。量子加密利用量子力学中测量对粒子物理状态产生不可逆影响的属性，也就是量子不可测量的特性，来确保通信密钥的安全传递。所以它不仅可以解决一次一密的密码本传输问题，还能确保在传输密钥时不会被第三方窃听和复制，是未来加密技术发展的重要方向。第一次真实的量子加密系统，是 1988 年在 IBM 的实验室开发出来的。1995 年，日内瓦大学可以做到相距 23km 完成量子加密通信。2012 年，我国潘建伟院士团队把这个数字推进到了一百千米这个级别。现在这个团队正在尝试在空间轨道上卫星和地面接收站间，实现量子加密信息的传输，距离就已经达到千千米的级别。2016 年中国发射了“墨子”号量子通信卫星，并于 2018 年成功实现了跨洲际的量子保密信息传输。只不过实验中符合条件的光量子态数量实在太少，只有几个到十几个数位，远远不能承载信息的正文，所以到目前为止，量子加密只适合给密钥加密，离大规模应用还有一定距离。

3.1.2 密码系统

密码系统又称为密码体制，是指能完整地解决信息安全中的机密性、数据完整性、认证、身份识别及不可抵赖等问题中的一个或几个的一个系统，其目的是让人们能够使用不安全信道进行安全的通信。

1. 密码系统组成

一个完整的密码系统包括如下五个要素{明文 M，密文 C，密钥 K，加密算法 E，解密算法 D}，示意图如图 3-2 所示。

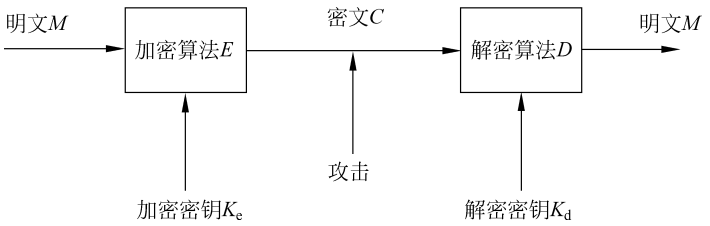


图 3-2 密码系统组成

(1) 明文：是加密输入的原始信息，通常用 m 表示。全体明文的集合称为明文空间，通常用 M 表示。

(2) 密文：是明文经过加密变换后的结果，通常用 c 表示。全体密文的集合称为密文空间，通常用 C 表示。

(3) 密钥：是参与信息变换的参数，通常用 k 表示。全体密钥的集合称为密钥空间，通常用 K 表示。

(4) 加密算法：是将明文变成密文的变换函数，即发送者加密消息时所采用的一组规则，通常用 E 表示。

(5) 解密算法：是将密文变成明文的变换函数，即接收者解密消息时所采用的一组规则，通常用 D 表示。

加密：是将明文 M 用加密算法 E 在加密密钥 K_e 的控制下变换成密文 C 的过程，表示为 $C=E_{K_e}(M)$。

解密：是将密文 C 用解密算法 D 在解密密钥 K_d 的控制下变换成明文 M 的过程，表示为 $M=D_{K_d}(C)$，并要求 $M=D_{K_d}(E_{K_e}(M))$，即用加密算法得到的密文用一定的解密算法总能够恢复成为原始的明文。

【思考】

思考 1. 密码系统哪些要素是必须保密的？哪些要素是可以公开的？

思考 2. 加密/解密算法能否公开？请说明你的理由。

2. 密码系统的设计要求

1883 年 2 月，巴黎 HEC 商学院的德国籍语言学家和教授奥古斯特·柯克霍夫(August Kerckhoffs)在《军事科学报》(Journal of Military Science)上发表了一篇文章，提出了“柯克霍夫原则”，奠定了现代密码学的基础，而他也因此被世人誉为“计算机安全之父”。柯克霍夫原则明确了密码系统的设计要求，显示了密钥在密码系统中的重要性，而这一原则最早被

应用于电报加密。

表述 1：密码系统中的算法即使为密码分析者所知，也无助于用来推导出明文和密文。

表述 2：即使密码系统的任何细节已为人所知，只要密钥没有泄露，它也应该是安全的。

表述 3：密码系统应该就算被所有人知道系统的运作步骤，仍然是安全的。

美国数学家、信息论的创始人克劳德·艾尔伍德·香农(Claude Elwood Shannon)提出的香农公理是对柯克霍夫原则的又一诠释：敌人知道系统。

在密码学中通常假定加密密钥和解密算法是公开的，密码体制的安全性只系于密钥的安全性，这就要求加密算法本身要非常安全。如果提供了无穷的计算资源，依然无法攻破，则称这种密码体制是无条件安全的。除了一次一密之外，无条件安全是不存在的，因此密码系统应尽量满足以下两个条件：

(1) 破译密码的成本超过密文信息的价值；

(2) 破译密码的时间超过密文信息有用的生命周期。

如果满足上面两个条件之一，则密码系统可被认为是安全的。

3. 密码设计原则

1949 年，信息理论的创始人香农发表论文：Common Theory of Secrecy System 证明了如下原则。

1) 混淆

混淆是指明文与密钥以及密文之间的统计关系尽可能复杂化，使破译者无法推导出相互间的依赖关系，从而加强隐蔽性。实现混淆的典型方法：替代。

2) 扩散

扩散是让明文中的每一位(包括密钥中的每一位)直接和间接影响输出密文中的许多位，或者让密文中的每一位受制于输入明文以及密钥中的若干位，以便达到隐蔽明文的统计特性。即明文中任何一点小变动都会使得密文有很大的差异。实现扩散的典型方法：换位。

3.1.3　密码分类

密码除了隐写术以外可以分为古典密码和现代密码两大类，如图 3-3 所示。古典密码一般是以单个字母为作用对象的加密法，现代密码则以明文的二元表示作为基础的加密法。

1. 古典密码

古典密码可细分为替代密码和换位密码。

1) 替代密码

替代密码是指先建立一个替换表，加密时将需要加密的明文依次通过查表，替换为相应的字符，明文字符被逐个替换后，生成无任何意义的字符串，即密文，替代密码的密钥就是其替换表。即明文中的每一个字符(比特、字母、比特组合或字母组合)被映射为另一个字符，简单地说，就是将一个符号替换成另一个符号来形成密文。该操作主要达到非线性变换的目的。

根据密码算法加解密时使用替换表数量的不同，替代密码又可分为单表替代密码和多表替代密码。

① 单表替代密码。

单表替代密码的密码算法加解密时使用一个固定的替换表。单表替代密码又可分为一

般单表替代密码、移位密码、仿射密码、密钥短语密码。

② 多表替代密码。

多表替代密码的密码算法加解密时使用多个替换表。多表替代密码有弗吉尼亚密码、希尔密码、一次一密钥密码、Playfair 密码。

2）换位密码

换位密码是一种早期的加密方法，不是用其他字母来代替已有字母，而是重新排列文本中的字母，类似于拼图游戏，所有的图块都在一个框中，只是排列的位置不同。换位加密法一般是利用几何图形（正方形、矩形），按一个方向填写构造明文，按另一个方向读取形成密文。即明文中字符的位置被重新排序，这是一种线性变换，对它们的基本要求是不丢失信息（即所有操作都是可逆的）。

例如，将明文：Youmustdothatnow 按图 3-4 逐行排列，按列读取得到密文：tuhosaYuttmdnoow。

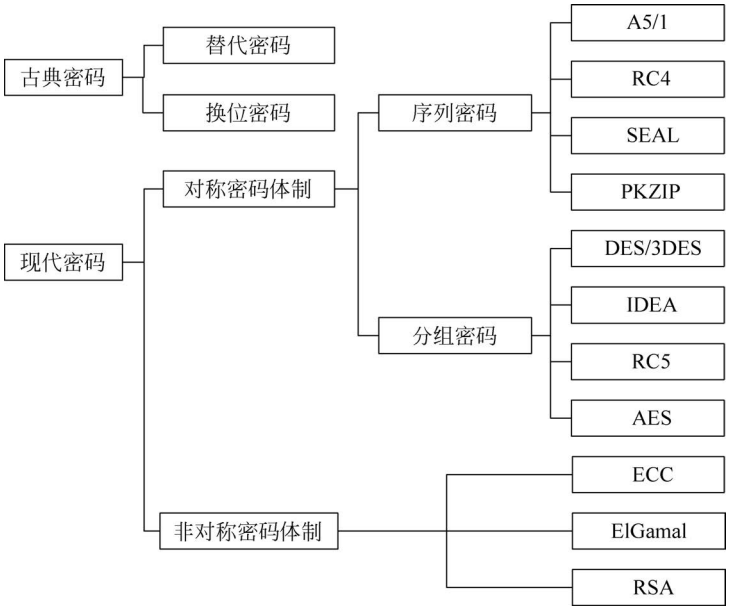


图 3-3 密码的分类

			Y			
		o	u	m		
	u	s	t	d	o	
t	h	a	t	n	o	w

图 3-4 逐行排列

替代和换位是加密算法的基本操作，所有加密算法都是基于这两个操作。

2. 现代密码

按密钥特征对现代密码进行分类，可以分为对称密码算法和非对称密码算法。

1）对称密码算法

对称密码算法也称为对称密码体制、传统密码算法，是指加密和解密使用相同密钥，或加密密钥能够从解密密钥中推算出来的密码算法。在大多数的对称密码算法中，加密密钥和解密密钥是相同的，所以也称这种加密算法为秘密密钥算法或单密钥算法。按明文加密方式分，对称密码算法可以分为流（序列）密码和分组密码。

① 流（序列）密码。

流（序列）密码是将明文流以一个元素（一般是一个字母或一个比特）作为基本的处理单

元，加密过程中是在密钥的控制下，将密钥流（密钥的二进制位）与等长的明文的二进制位进行模2运算输出密文，在解密过程中将密钥流与密文进行逐位模2运算输出明文（模2运算详见附录）。典型的流（序列）密码有A5/1和RC4算法。目前，A5/1算法在很多应用中已经被分组密码所替代，基于软件实现的流密码RC4算法的应用仍较为广泛。

② 分组密码。

分组密码是将明文分为固定比特长度的分组，在密钥的控制下，用固定加密算法对一组一组明文分组进行加密，再输出对应的一组一组密文；最后一组明文长度小于分组长度时，应对最后一组进行填充。一个分组的比特数就称为分组长度。分组密码共有5种工作模式：电码本模式、密文分组链接模式、密文反馈模式、输出反馈模式和计数器模式。典型的分组密码算法有：DES、TDES/3DES（Triple DES，三重数据加密标准）、高级加密标准（Advanced Encryption Standard，AES）、国际数据加密算法（International Data Encryption Algorithm，IDEA）和RC5。其中，DES算法因安全问题，已退出了历史舞台，但其算法思想仍具有研究价值，是密码学研究的入门级算法。TDES/3DES因效率问题，使用领域越来越少，逐步被AES算法取代。

分组密码的加解密速度快、安全性好并得到许多密码芯片的支持，可用于数据加密、数字签名、认证和密钥管理，在计算机通信和信息系统安全领域有广泛的应用。一般而言，分组密码比流（序列）密码的应用范围更广，绝大部分的基于网络的常规加密应用都使用分组密码。

对称密码算法的安全性主要取决于以下两个因素：

① 加密算法必须足够强大，不必为算法保密，仅根据密文就能破译出消息是不可行的。

② 对称密码算法的安全性依赖于密钥的保密，泄露密钥就意味着任何人都可以对发送或接收的消息解密，所以密钥必须保密并保证有足够大的密钥空间，且要求基于密文和加密/解密算法能破译出消息的做法是不可行的。

对称密码算法的密钥长度相对较短，计算量小，加密、解密处理速度快，具有很高的数据吞吐率（硬件加密可达到每秒几百兆字节，软件加密也可以达到每秒兆字节的吞吐率），效率高，适合大数据的加密。密文与明文的长度相同或扩张较小，是目前用于信息加密的主要算法。

由于在使用对称密码算法时，每对收发双方都需要使用唯一密钥。当发送方需要与多人通信时，发送方所拥有的密钥数量将呈几何级数增长，密钥的分发、传输和管理成为了用户负担。密钥是对称密码算法的关键，如何才能把密钥安全地送到接收方，是对称密码算法的突出问题。在使用中客户端不能直接存储对称密码算法的密钥，因为被反编译之后，密钥就泄露了，数据安全性就得不到保障，所以在实际使用中，通信双方在每一次通信中都使用一个一次性密钥，并用公钥对这个密钥进行加密，这样就免去了频繁的密钥维护以及更新过程，即使是密钥被窃取或者发生损坏，那么也只影响一次的通信过程，可以将受攻击的损失降到最低。因此，对称密码算法在分布式网络系统上使用较为困难，主要是体现在密钥传输、管理困难，使用成本较高。

2）非对称密码算法

为解决对称密码算法中如何安全地分发、管理和传输密钥的问题，1976年，两位美国计算机学家Whitfield Diffie和Martin Hellman提出了一种崭新构思，可以在不直接传递密钥的情况下，完成加解密，这就是“Diffie Hellman密钥交换算法”。这个算法启发了其他科学家。人们认识到，加密和解密可以使用不同的规则，只要这两种规则之间存在某种对应关系

即可，这样就避免了直接传递密钥。这种新的加密模式被称为“非对称密码算法”，也叫公开密钥密码体制、双密钥密码体制。非对称密码算法基于数学问题求解的困难性，而不再是基于替代和换位方法。非对称密码算法的设计必须遵循“三公”原则，即公钥公开、密文公开和算法公开。典型的非对称密码算法有椭圆曲线密码(Elliptic Curve Cryptography，ECC)算法、RSA 算法和 ElGamal 算法。

非对称密码算法使用具有配对关系的两个不同密钥，一个可公开，称为公钥；另一个只能被密钥持有人自己秘密保管，且不能基于公钥推导出，称为私钥。用公钥对明文加密后，仅能使用与之配对的私钥解密，才能恢复出明文，反之亦然。因为公钥加密的信息只有私钥解得开，那么只要私钥不泄露，明文就是安全的。

3) 对称密码算法和非对称密码算法的混合使用

非对称密码算法虽然解决了对称密码算法的密钥分发问题，但存在密钥生成和加解密速度较慢，同等安全强度下，需要的密钥位数多、即密钥的长度大等问题，常用于小块数据加密。因此在实际应用中，一般将对称密码算法和非对称密码算法混合使用，即用对称密码算法加密明文，应用非对称密码算法加密对称密码算法的密钥，如应用于军队后勤管理系统、核应急安全数据通信系统的数据加密和汽车防盗等。如图 3-5 所示，当用户 A 想与用户 B 构建通信联系时，可按照如下步骤交换获得对称密钥 K。

图 3-5 基于非对称密码算法的对称密码算法密钥分发

① 用户 A 用非对称密码算法产生一个公私钥对，其中$\{e,n\}$为公钥，$\{d,n\}$为私钥。

② 用户 A 将公钥$\{e,n\}$及其标识符发送给想要构建通信联系的用户 B。

③ 用户 B 生成对称密码算法的密钥 K。

④ 用户 B 使用用户 A 的公钥$\{e,n\}$对密钥 K 进行加密，发送给用户 A。

⑤ 因为只有用户 A 掌握了公钥$\{e,n\}$对应的私钥$\{d,n\}$，故用户 A 将解密获得密钥 K。至此用户 A、用户 B 都获得了对称密码算法的私钥 K 作为会话密钥用于通信。

3.2 典型对称密码算法 AES

3.2.1 AES 算法的数学基础

1. 有限域

有限域亦称伽罗瓦域，是仅含有限个元素的域，它是伽罗瓦于 18 世纪 30 年代研究代数方程根式求解问题时引出的，在近代编码、计算机理论、组合数学等各方面有着广泛的应用。通常用 GF(p^n)表示 p^n 元的有限域。

有限域 GF(2^n)——包含 2^n 个元素。

(1) 加法、减法、乘法、除法都不能脱离该域。

(2) 乘法逆元：对于 GF 中的任意元素 a(除 0 外)，GF 中存在一个元素 a^{-1} 使 $a \cdot a^{-1}=1$，这样的 a^{-1} 被称为乘法逆元。

【思考】

整数集合是不是一个有限域？

答：不是。因为整数集合中不是所有元素均有乘法逆元。实际上，只有元素 1 和−1 有乘法逆元。

2. 既约多项式

既约多项式又被称为不可约多项式，其特点是不能再进行因式分解了。在 AES 算法中选用的既约多项式是 $m(x)=x^8+x^4+x^3+x+1$。该多项式可以被其他既约多项式替代。

关于既约多项式的选择。在参考文献[1]中列举了 30 个既约多项式，上式位列这 30 个既约多项式的第一个。

3. 多项式表示

{57}=01010111 的多项式表示为 $x^6+x^4+x^2+x+1$，其中{}内的数字表示字节。

4. GF(2^8)的运算

1) 加法/减法

有限域 GF(2^8)上的加法/减法即为二进制数的按位异或。

课堂练习：

$$\{57\}+\{83\}=\{D4\}$$

$$(x^6+x^4+x^2+x+1)\oplus(x^7+x+1)=x^7+x^6+x^4+x^2$$

2) 乘法

对于一个 8 位的二进制数 $A=a_7a_6a_5a_4a_3a_2a_1a_0$ 来说，有如下结果：

① $a_7=0$ 时，$\{02\}\cdot A=a_6a_5a_4a_3a_2a_1a_00$；

② $a_7=1$ 时，$\{02\}\cdot A=(a_6a_5a_4a_3a_2a_1a_00)\oplus(00011011)$；

③ $\{03\}\cdot A=A\oplus(\{02\}\cdot A)$。

3) 模运算

模运算定义是：如果 a 是一个整数，n 是一个正整数，定义 $a \bmod n$ 为 a 除以 n 的非负余数。其数学符号为“mod”。例如，$11 \bmod 4=4$，$-11 \bmod 7=3$ 等。模运算就像普通的运算一样，它是可交换、可结合、可分配的，具有如下性质：

$$① (a \bmod n \pm b \bmod n) \bmod n=(a \pm b) \bmod n \tag{3-1}$$

$$② (a \bmod n \times b \bmod n) \bmod n=(a \times b) \bmod n \tag{3-2}$$

$$③ [(a \times b) \bmod n+(a \times c) \bmod n] \bmod n=[a \times (b+c)] \bmod n \tag{3-3}$$

$$④ a^b \bmod n=(a \bmod n)^b \bmod n \tag{3-4}$$

式(3-1)的证明：

令 $a=k_1 \times n+r_1$，$b=k_2 \times n+r_2$

$\therefore a \bmod n=r_1$，$b \bmod n=r_2$

$(a \pm b) \bmod n=((k_1 \pm k_2) \times n+(r_1 \pm r_2)) \bmod n$

$$=(r_1 \pm r_2) \bmod n$$
$$=(a \bmod n \pm b \bmod n) \bmod n$$

式(3-2)的证明：

令 $a=k_1 \times n+r_1, b=k_2 \times n+r_2$

$\therefore a \bmod n=r_1, b \bmod n=r_2$

$$(a \times b) \bmod n=(k_1 \times k_2 \times n^2+(k_1 \times r_2+k_2 \times r_1) \times n+r_1 \times r_2) \bmod n$$
$$=(r_1 \times r_2) \bmod n$$
$$=(a \bmod n \times b \bmod n) \bmod n$$

式(3-3)的证明：

令 $a=k_1 \times n+r_1, b=k_2 \times n+r_2, c=k_3 \times n+r_3$

$\therefore a \bmod n=r_1, b \bmod n=r_2, c \bmod n=r_3$

$$[a \times (b+c)] \bmod n$$
$$=((k_1 \times (k_2+k_3) \times n^2+(k_1 \times (r_2+r_3)+r_1 \times (k_2+k_3)) \times n+ r_1 \times (r_2+r_3)) \bmod n$$
$$=(r_1 \times (r_2+r_3)) \bmod n$$
$$=(r_1 \times r_2+r_1 \times r_3) \bmod n$$
$$=(a \bmod n \times b \bmod n+a \bmod n \times c \bmod n) \bmod n$$
$$=[(a \times b) \bmod n+(a \times c) \bmod n] \bmod n$$

式(3-4)的证明：

设 $a \bmod n=r$

$$\therefore a^b \bmod n=(a \times a \times \cdots \times a) \bmod n$$
$$=(a \bmod n \times a \bmod n \times \cdots \times a \bmod n) \bmod n$$
$$=r^b \bmod n$$
$$=(a \bmod n)^b \bmod n$$

3.2.2 AES 算法概述

1. 产生的背景

在 AES 算法之前，美国广泛使用的是 1972 年由 IBM 公司研发的 DES 算法。由于 DES 算法破译的不断发展，DES 算法的安全性与应用前景面临非常大的挑战。随后推出的 TDES/3DES 是 DES 算法的一个更安全的变形，但由于需要进行三重 DES，速度较慢。因此，美国需要设计一个不用保密的、公开的、免费的分组密码算法，用来保护 21 世纪政府、金融等核心部门的敏感信息，并期望用这个新算法取代渐进没落的 DES 算法，成为新一代数据加密标准。

因此美国对 AES 算法的要求有以下三点：

(1) 安全方面：至少和 3DES 算法一样安全。

(2) 速度方面：比 3DES 算法快。

(3) 成本方面：免费使用。

1997 年 4 月 15 日，NIST 发起征集 AES 算法的活动，并专门成立了 AES 算法工作组。1997 年 9 月 12 日，联邦政府发布了征集 AES 算法候选算法的通知。截至 1998 年 6 月 15

日，NIST 共收到 21 个提交的算法。1998 年 8 月 10 日，NIST 召开第一次 AES 算法候选会议，公布了 15 个候选算法。1999 年 3 月 22 日，NIST 召开第二次 AES 算法候选会议，公开了 15 个 AES 算法候选算法的讨论结果，并从里面选了 5 个算法进一步讨论。2000 年 10 月 2 日，在进一步分析与讨论这 5 个算法后，正式公布由比利时密码学家 Joan Daemen 与 Vicent Rijmen 设计的 Rijndael 算法成为 AES 算法。同时，NIST 发表了一篇 16 页的报告，总结了选择 Rijndael 算法作为 AES 算法的原因：

（1）不管用反馈模式还是用无反馈模式，Rijndael 算法在普遍计算环境的硬件与软件实现上性能一直保持优秀；

（2）Rijndael 算法的密钥建立时间非常短，灵敏性好；

（3）Rijndael 算法的内部循环结构易于并行处理，运算容易，极低的内存需求让它特别适合在存储器有限的环境里使用；

（4）Rijndael 算法能抵抗强力与时间选择攻击。

此外，由于在 AES 算法的选拔过程中，参加者必须提交密码算法的详细规格书、以 ANSIC 和 Java 编写的实现代码以及抗密码破译强度的评估等材料，这就彻底杜绝了隐蔽式安全性。

2. 特点

1）安全性好

AES 算法中，每轮使用不同的常数消除了密钥的对称性，使用了非线性的密钥扩展算法消除了相同密钥的可能性，能够抵抗线性攻击。加密和解密使用不同的变换，从而消除了弱密钥和半弱密钥存在的可能性。因为 AES 算法的密钥长度可变，针对 128/192/256 位的密钥，密钥量分别为 $2^{128}/2^{192}/2^{256}$，足以抵抗穷举搜索的攻击。尽管 AES-128 密钥的弱化版本已经受到了攻击，Niels Ferguson 等在 2000 年实现了对加密 7 轮的 AES-128 的攻击，但迄今为止尚未出现对完整 AES 算法的成功攻击。实践证明 AES 算法能抵抗穷举攻击、线性攻击、差分攻击、相关密钥攻击、插值攻击等。

2）对资源需求小

AES 算法不需要特殊的硬件加解密，所需要的软件资源少，因此非常适合应用在资源紧张的感知设备、智能终端、智能卡等物联网设备中，例如，物联网网络设备路由器、手机等智能终端、信息管理系统、无人机数据链传输、汽车防盗系统、校园卡和公交卡等的数据加密中。图 3-6 和图 3-7 分别给出了某无线路由器安全设置和手机无线局域网络（Wireless Local Area Networks，WLAN）热点上网的信息加密就是采用 AES 算法的实例。

3）执行效率高

由于 AES 算法效率较高，因此适用于对效率有要求的实时数据加密通信。比如在使用虚拟专用网络（Virtual Private Network，VPN）或者代理进行加密通信时，既要保证数据的保密性，又要保证不能有高的延迟，所以通常会使用 AES 算法进行通信加密。例如，ZigBee 技术中，为确保 MAC 帧的完整性、机密性、真实性和一致性，其 MAC 层使用 AES 算法进行加密，并且生成一系列的安全机制。在无人机数据链传输过程中，为了解决军事的数据安全问题，也常采用 AES 算法对其数据进行加解密。

4）适应性强

AES 算法适应性强，广泛适用于不同的 CPU 架构，在不同软件或硬件平台上均易

图 3-6　某无线路由器安全设置

图 3-7　手机 WLAN 热点参数设置

实现。因此在各行各业中被广泛应用，除了逐渐取代 DES 算法在 IPSee、SSL 和 ATM 中的应用外，而且在远程访问服务器、移动通信、卫星通信、财政保密等方面也得到广泛使用。

3.2.3　AES 算法原理

AES 算法是一种典型的对称密码算法、分组密码算法，分组长度为 128 位，而密钥长度可以为 128、192 或 256 位，对应的迭代轮数分别为 10 轮、12 轮或 14 轮，分别被记作 AES-128 算法、AES-192 算法和 AES-256 算法。这里的“位”是指二进制的位数。“轮”是指加密过程中需要循环进行的所有步骤。AES 算法的整个加密过程就是进行若干次轮的循环迭代。AES 算法系列相关参数如表 3-1 所示。

表 3-1　AES 算法相关参数

AES	密钥长度(32 位比特字)	分组长度(32 位比特字)	加 密 轮 数
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

整个加密过程包括初始变换、轮变换、密钥变换三大变换。本书将以 AES-128 算法为例进行说明，算法流程如图 3-8 所示。AES-128 算法每次变换处理的基本单位是一个 4×4 矩阵，该矩阵的每个元素是一个 8 位二进制数(为阅读方便，一般书写为十六进制数)。AES-192 算法和 AES-256 算法的原理与 AES-128 算法一样，只是每次加解密的数据和密钥大小分别为 192 位和 256 位。AES 算法的解密算法与加密算法类似(称为逆加密算法)，主要区别在于轮密钥要逆序使用，四个基本运算都有对应的逆变换，在此不再赘述。

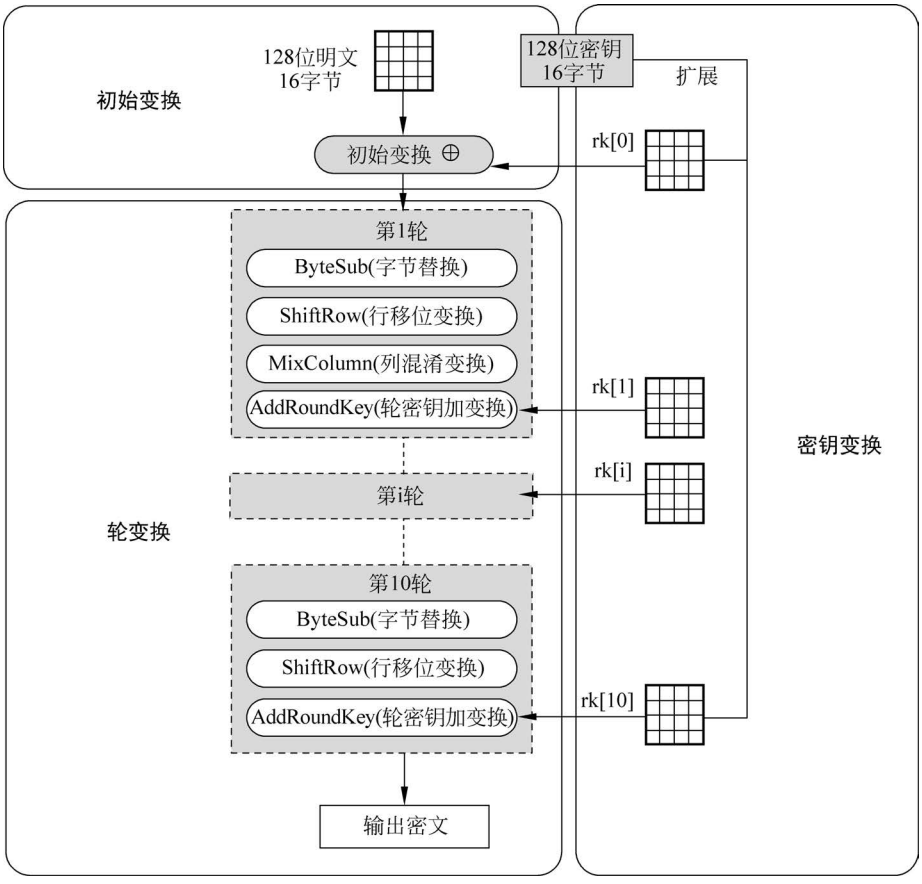


图 3-8　AES-128 算法流程

1. 初始变换

初始变换是将明文转化为明文矩阵，其流程如图 3-9 所示。

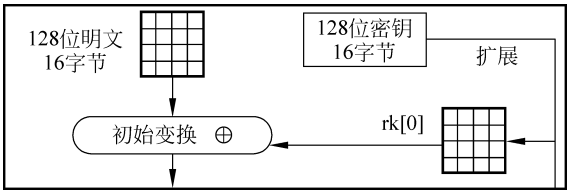


图 3-9　初始变换示意图

1）明文转化为 4×4 矩阵

AES 算法处理的基本单位是字节，因此首先需将输入的 128 位明文和 128 位密钥转换成以字节为基本单位的 4×4 矩阵。

具体变换如图 3-10 所示。

P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}

⇩

P_0	P_4	P_8	P_{12}
P_1	P_5	P_9	P_{13}
P_2	P_6	P_{10}	P_{14}
P_3	P_7	P_{11}	P_{15}

图 3-10　输入信息变为 4×4 矩阵

① 将 128 位明文 P 和 128 位密钥 K 分别按字节分成 16 字节，分别记为 $P=P_0P_1\cdots P_{15}$ 和 $K=K_0K_1\cdots K_{15}$。例如，P 的十六进制表示：ABABCDCDEFEF0101ABABCDCDEFEF0101，前两个字符 AB 对应字节 P_0，最后两个字符则对应字节 P_{15}。

② 将生成的 16 字节按列排序为 4×4 矩阵，称此矩阵为状态矩阵。在算法的每一轮中，状态矩阵的内容将不断发生变化，最后的结果作为密文输出。

在实际应用中，由于明文一般不是以十六进制或二级制表示的。因此，还需对明文进行编码处理。鉴于明文可能是英文、中文、阿拉伯数字等各种形式，一般选择 Unicode 字符集的 UTF-8 编码方式进行编码。

2）矩阵异或

将明文和密钥变换得到的 4×4 矩阵按字节进行异或，输出一个新的 4×4 矩阵。初始变换在 AES 算法中主要起到混淆的作用。

2. 轮变换

轮变换又包括字节替换（ByteSub）、行移位变换（ShiftRow）、列混淆变换（最后一轮无列混淆替换）（MixColumn）和轮密钥加变换（AddRoundKey）四个步骤。

1）字节替换

字节替换又被称为“S 盒变换”，替换规则是：以初始变换输出矩阵的元素为最小处理单位，把字节的高 4 位作为行值，低 4 位作为列值，以该行列值为索引从 S 盒（如表 3-2 所示）的对应位置取出元素作为输出。经过字节替换后的输出仍是一个 4×4 矩阵。

S 盒变换是 AES 算法中唯一的非线性变换，体现在输入位和输出位之间的相关性很低，且输出值不是输入值的线性数学函数，这是保障 AES 算法安全的关键。优点是可查表完成，计算速度快。

表 3-2　S 盒

		y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
x	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	AS	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	SA	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	BO	54	BB	16

【练习】

初始变换中某明文的十六进制数为{EA}，请写出该字节经字节替换后的输出是什么。

2）行移位变换

行移位变换的具体步骤如图 3-11 所示。经过字节替换的输出矩阵 $\boldsymbol{S}$ 的第 0 行不变，第 1 行循环左移 1 字节，第 2 行循环左移 2 字节，第 3 行循环左移 3 字节，形成新的矩阵 $\boldsymbol{S}'$，$\boldsymbol{S}'_{i.j}$ 表示行移位变换输出矩阵 $\boldsymbol{S}'$ 第 i 行、第 j 列元素，$i\in[0,3]$，$j\in[0,3]$。行移位变换属于置换，是一种线性变换，本质上是把数据打乱重排，起到扩散的作用。

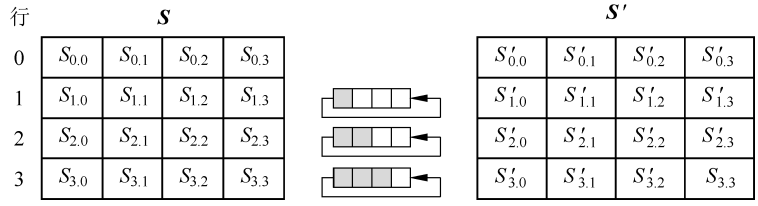


图 3-11　行移位变换示意图

3）列混淆变换

列混淆变换是通过矩阵相乘来实现的，具体方式是：行移位变换的输出矩阵 $\boldsymbol{S}'$ 与固定系数矩阵 $\boldsymbol{C}$ 做乘法后模多项式 x^4+1。计算过程如下：

根据有限域 GF(2^8)上多项式的表示规则，行移位变换输出矩阵 $\boldsymbol{S}'$ 第 j（$j\in[0,3]$）列的 4 个元素（$S'_{0.j}$，$S'_{1.j}$，$S'_{2.j}$，$S'_{3.j}$）可以表示为多项式 $S'_{0.j}+S'_{1.j}\cdot x+S'_{2.j}\cdot x^2+S'_{3.j}\cdot x^3$，它经过列混淆变换后的输出记为（$b'_{0.j}$，$b'_{1.j}$，$b'_{2.j}$，$b'_{3.j}$），表示为多项式 $b_{3.j}\cdot x^3+b_{2.j}\cdot x^2+b_{1.j}\cdot x+b_{0.j}$，由式(3-5)计算获得。符号{}内的数表示字节。

$$b_{3.j}\cdot x^3+b_{2.j}\cdot x^2+b_{1.j}\cdot x+b_{0.j}$$

$$=((S'_{0.j}+S'_{1.j}\cdot x+S'_{2.j}\cdot x^2+S'_{3.j}\cdot x^3)\cdot c(x))\bmod(x^4+1) \tag{3-5}$$

其中，$c(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$。

$$\begin{aligned}&(S'_{0.j}+S'_{1.j}\cdot x+S'_{2.j}\cdot x^2+S'_{3.j}\cdot x^3)\cdot c(x)\\&=(S'_{3.j}\cdot x^3+S'_{2.j}\cdot x^2+S'_{1.j}\cdot x+S'_{0.j})\cdot(\{03\}x^3+\{01\}x^2+\{01\}x+\{02\})\\&=(S'_{3.j}\cdot\{03\})x^6+(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{03\})x^5+\\&\quad(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{03\})x^4+\\&\quad(S'_{3.j}\cdot\{02\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{01\}+S'_{0.j}\cdot\{03\})x^3+\\&\quad(S'_{2.j}\cdot\{02\}+S'_{1.j}\cdot\{01\}+S'_{0.j}\cdot\{01\})x^2+\\&\quad(S'_{1.j}\cdot\{02\}+S'_{0.j}\cdot\{01\})x+S'_{0.j}\cdot\{02\}\end{aligned} \tag{3-6}$$

由于 $x^i\bmod(x^4+1)=x^{i\bmod4}$，所以有式(3-7)～式(3-9)成立。

$$((S'_{3.j}\cdot\{03\})x^6)\bmod(x^4+1)=(S'_{3.j}\cdot\{03\})x^2 \tag{3-7}$$

$$((S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{03\})x^5)\bmod(x^4+1)=(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{03\})x \tag{3-8}$$

$$\begin{aligned}&((S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{03\})x^4)\bmod(x^4+1)\\&=(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{03\})\end{aligned} \tag{3-9}$$

所以

$$\begin{aligned}&b_{3.j}\cdot x^3+b_{2.j}\cdot x^2+b_{1.j}\cdot x+b_{3.j}\\&=((S'_{0.j}+S'_{1.j}\cdot x+S'_{2.j}\cdot x^2+S'_{3.j}\cdot x^3)\cdot c(x))\bmod(x^4+1)\\&=(S'_{3.j}\cdot\{02\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{01\}+S'_{0.j}\cdot\{03\})x^3+(S'_{3.j}\cdot\{03\}+\\&\quad S'_{2.j}\cdot\{02\}+S'_{1.j}\cdot\{01\}+S'_{0.j}\cdot\{01\})x^2+(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{03\}+\\&\quad S'_{1.j}\cdot\{02\}+S'_{0.j}\cdot\{01\})x+\\&\quad(S'_{3.j}\cdot\{01\}+S'_{2.j}\cdot\{01\}+S'_{1.j}\cdot\{03\}+S'_{0.j}\cdot\{02\})\end{aligned} \tag{3-10}$$

把式(3-10)写成矩阵形式，即式(3-11)。

$$\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&03\end{bmatrix}\begin{bmatrix}S_{0.0}&S_{0.1}&S_{0.2}&S_{0.3}\\S_{1.0}&S_{1.1}&S_{1.2}&S_{1.3}\\S_{2.0}&S_{2.1}&S_{2.2}&S_{2.3}\\S_{3.0}&S_{3.1}&S_{3.2}&S_{3.3}\end{bmatrix}=\begin{bmatrix}S'_{0.0}&S'_{0.1}&S'_{0.2}&S'_{0.3}\\S'_{1.0}&S'_{1.1}&S'_{1.2}&S'_{1.3}\\S'_{2.0}&S'_{2.1}&S'_{2.2}&S'_{2.3}\\S'_{3.0}&S'_{3.1}&S'_{3.2}&S'_{3.3}\end{bmatrix} \tag{3-11}$$

所以，系数矩阵 $\boldsymbol{C}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&01&02&03\\03&01&01&03\end{bmatrix}$

上述矩阵乘法和加法都是定义在基于有限域 GF(2^8)上的二元运算上，并不是通常意义上的矩阵乘法和加法，而是模 $m(x)=x^8+x^4+x^3+x+1$ 上的加法和乘法。其中，$m(x)$是AES算法设计者选的有限域 GF(2^8)上的不可约多项式(也称为既约多项式)，是指不能写成两个次数较低的多项式之乘积的多项式。有限域 GF(2^8)上加法计算规则等价于两个字

节的异或。由式(3-11)可知，只需计算系数{01}、{02}和{03}这3个数的有限域GF(2^8)上的乘法，即$(C_{i,j} \cdot S'_{i,j}) \bmod m(x)$的值，其中$C_{i,j}$表示系数矩阵$\boldsymbol{C}$中的元素。计算规则如下：

① 计算乘以{01}。

字节{01}表示成二进制数为00000001，对应的有限域GF(2^8)上的多项式就是1。对于任何$S'_{i,j}$有式(3-12)成立。

$$(\{01\} \cdot S'_{i,j}) \bmod m(x) = S'_{i,j} \tag{3-12}$$

② 计算乘以{02}。

计算$(\{02\} \cdot S'_{i,j}) \bmod m(x)$需两步完成：第一步需计算$\{02\} \cdot S'_{i,j}$，第二步需计算$(\{02\} \cdot S'_{i,j}) \bmod m(x)$。

步骤一：计算$\{02\} \cdot S'_{i,j}$。

由于字节{02}表示成二进制数为00000010，对应的有限域GF(2^8)上的多项式就是x。$\{02\} \cdot S'_{i,j}$相当于字节$S'_{i,j}$对应的多项式乘x，相当于$S'_{i,j}$表示二进制数左移一位，即$S'_{i,j}$对应的多项式的系数不变、次数加1。

步骤二：计算$(\{02\} \cdot S'_{i,j}) \bmod m(x)$。

步骤一计算得到的结果，如果多项式次数小于8，即对应的二进制数最高位为0，相当于$(\{02\} \cdot S'_{i,j}) < m(x)$，所以$(\{02\} \cdot S'_{i,j}) \bmod m(x) = \{02\} \cdot S'_{i,j}$。

步骤一计算得到的结果，如果多项式次数等于8，即对应的二进制数最高位为1，就需要$\bmod m(x)$。此时多项式一定小于$2m(x)$，所以$\bmod m(x)$的余式就相当于$\{02\} \cdot S'_{i,j} - m(x)$的结果。已知有限域GF($2^8$)上的减法也是进行异或运算。由于$\{02\} \cdot S'_{i,j}$只涉及8位二进制数，因此只需与$m(x)$的后8位二进制数00011011(十六进制数就是0x1b)进行异或运算即可。

综上，对于任何$S'_{i,j}$，$(\{02\} \cdot S'_{i,j}) \bmod m(x)$相当于先把字节$S'_{i,j}$表示的二进制数左移1位。移位后，若最高位$a_7 = 1$，则将该二进制数的后8位与{1b}进行异或运算得到结果；若最高位$a_7 = 0$，取后8位就是结果。数学表述如下：

令$S'_{i,j}$表示的二进制数$S'_{i,j} = a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$：

当$a_7 = 0$时，$\{02\} \cdot S'_{i,j} = a_6 a_5 a_4 a_3 a_2 a_1 a_0 0$；

当$a_7 = 1$时，$\{02\} \cdot S'_{i,j} = (a_6 a_5 a_4 a_3 a_2 a_1 a_0 0) \oplus (00011011)$。

③ 计算乘以{03}。

因为在有限域GF(2^8)上，$\{03\} = \{01\} \oplus \{02\}$，所以有式(3-13)成立。

$$(\{03\} \cdot S'_{i,j}) \bmod m(x) = (\{02\} \cdot S'_{i,j} \oplus \{01\} \cdot S'_{i,j}) \bmod m(x) \tag{3-13}$$

【例】 计算下列矩阵经过列混淆后的值。

$$\begin{bmatrix} 87 & F2 & 4D & 97 \\ 6E & 4C & 90 & EC \\ 46 & E7 & 4A & C3 \\ A6 & 8C & D8 & 95 \end{bmatrix}$$

解：

第一步：根据式(3-11)有如下等式成立。

$$\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 03\end{bmatrix}\begin{bmatrix}87 & F2 & 4D & 97\\ 6E & 4C & 90 & EC\\ 46 & E7 & 4A & C3\\ A6 & 8C & D8 & 95\end{bmatrix}=\begin{bmatrix}S'_{0.0} & S'_{0.1} & S'_{0.2} & S'_{0.3}\\ S'_{1.0} & S'_{1.1} & S'_{1.2} & S'_{1.3}\\ S'_{2.0} & S'_{2.1} & S'_{2.2} & S'_{2.3}\\ S'_{3.0} & S'_{3.1} & S'_{3.2} & S'_{3.3}\end{bmatrix}$$

所以 $S'_{0.0}=\{02\}\cdot\{87\}+\{03\}\cdot\{6E\}+\{01\}\cdot\{46\}+\{01\}\cdot\{A6\}$。

第二步：计算$\{02\}\cdot\{87\}$。

① 将$\{87\}$转化为二进制，$\{87\}=(1000\ 0111)$。

② 判断$(1000\ 0111)$最高位的情况。

③ 因为 $a_7=1$，所以$\{02\}\cdot\{87\}=(0000\ 1110)\oplus(0001\ 1011)=(0001\ 0101)$。

第三步：计算$\{03\}\cdot\{6E\}$。

① 将$\{6E\}$ 转化为二进制，$\{6E\}=(0110\ 1110)$。

② 按如下规则计算：$\{03\}\cdot A=A\oplus(\{02\}\cdot A)$。

③ 计算$\{02\}\cdot\{6E\}$。

因为$\{6E\}$二进制数的最高位为 0，所以$\{02\}\cdot\{6E\}=(1101\ 1100)$。

④ 计算$\{03\}\cdot\{6E\}=\{6E\}\oplus(\{02\}\cdot\{6E\})=(1011\ 0010)$。

第四步：继续计算如下。

$$\begin{array}{rl} & \{02\}\cdot\{87\}=(0001\ 0101)\\ & \{03\}\cdot\{6E\}=(1011\ 0010)\\ & \{01\}\cdot\{46\}=(0100\ 0110)\\ \oplus & \{01\}\cdot\{A6\}=(1010\ 0110)\\ \hline & (0100\ 0111)\end{array}$$

所以 $S'_{0.0}=\{02\}\cdot\{87\}+\{03\}\cdot\{6E\}+\{01\}\cdot\{46\}+\{01\}\cdot\{A6\}=\{47\}$。

其余元素按上述计算，最后结果为

$$\begin{bmatrix}47 & 40 & A3 & 4C\\ 37 & D4 & 70 & 9F\\ 94 & E4 & 3A & 42\\ ED & A5 & A6 & BC\end{bmatrix}$$

4）轮密钥加变换

将列混淆变换后的结果与轮密钥进行异或运算，其结果作为后续变换的输入。其中，轮密钥是参与本轮运算的密钥。从图 3-8 可知，除初始密钥外，AES 算法参与运算的每一轮的密钥都是由上一轮的密钥变换而来的。由图 3-12 所示的轮密钥加变换示意图可知，虽然轮密钥加变换非常简单，却能影响矩阵中每一位。

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

列混淆变换输出的4×4矩阵

⊕

AC	19	28	6A
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

本轮密钥4×4矩阵

=

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

轮密钥加变换输出4×4矩阵

图 3-12　轮密钥加变换示意图

3. 密钥变换

密钥变换是以初始密钥的 4×4 矩阵为变换对象，以矩阵列为最小单位生成的，具体包括以下四个步骤：密钥排列、置换、与轮常量异或、生成下一轮密钥的其他列。下文以 AES-128 算法第二轮密码生成为例进行说明。密钥变换的目的是产生 AES 算法参与运算的每一轮密钥。因为 AES-128 算法内部其实不只执行一轮加密，而是一共执行了 11 轮加密，所以 AES 算法会通过一个简单快速的混合操作，根据初始密钥依次生成后面 10 轮的密钥，每一轮的密钥都是依据上一轮生成的，所以每一轮的密钥都是不同的。因此密钥变换又称为密钥扩展。尽管很多人抱怨说这种方式太简单了，经实践和理论证明这其实已经足够安全了。

1）密钥排列

① 按列将输入的 128 位密钥转换为 4×4 矩阵。

② 把初始密钥 $rk_{[0]}$ 最后一列的第一字节放到最后一字节的位置上，其余字节依次向上移动一位，经过密钥排列后的这一列称为排列列。示意图如图 3-13 所示。

2）置换

把排列列经过一个 S 盒替换后，排列列就会被映射为一个崭新的列，称这个崭新的列为置换列。示意图如图 3-14 所示。

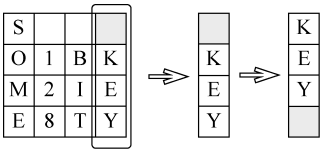


图 3-13　排列列生成示意图

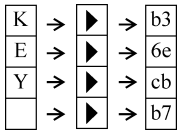


图 3-14　置换列生成示意图

3）与轮常量异或

把置换列和轮常量异或生成第二轮密码的最后一列。轮常量如表 3-3 所示，共四字节。轮常量的主要作用有：一是增加密钥编排中的非线性；二是消除 AES 算法中的对称性。这两种属性都是抵抗某些分组密码攻击所必要的。

表 3-3　轮常量

轮　数	轮　常　量	轮　数	轮　常　量
第 1 轮	01 00 00 00	第 6 轮	20 00 00 00
第 2 轮	02 00 00 00	第 7 轮	40 00 00 00
第 3 轮	04 00 00 00	第 8 轮	80 00 00 00
第 4 轮	08 00 00 00	第 9 轮	1B 00 00 00
第 5 轮	10 00 00 00	第 10 轮	36 00 00 00

4）生成下一轮密钥的其他列

用第二轮密钥的最后一列和初始密钥的第一列异或得到第二轮密钥的第一列；用第二轮密钥的第一列和初始密钥的第二列异或得到第二轮密钥的第二列；用第二轮密钥的第二列和初始密钥第三列异或得到第二轮密钥的第三列。至此，第二轮密钥 4×4 矩阵的四列就全部生成完毕。其过程如图 3-15 所示。$rk_{[1]}$ 表示第二轮密码。

后续轮密钥按上述过程生成，只是在与轮常量异或时轮常量有所不同而已。

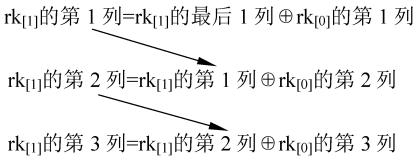


图 3-15　生成下一轮密钥的其他列

课程思政

辩证法中的对立统一规律

世界上任何事物的内部和事物之间都包含矛盾的两个方面，矛盾的双方既对立又统一，事物的运动发展在于自身的矛盾运动。密码学自诞生起就体现了矛盾的对立统一性，密码学作为一门学科包含密码编码学和密码分析学，也就是“攻”与“防”两个对立的方面，这对矛盾不断促进了密码学的发展。世上没有单一性质的、绝对的事物。这一规律也体现于密码学中。

3.2.4　针对 AES 算法的攻击

目前，针对 AES 算法的攻击主要有功耗分析攻击、故障注入攻击、差分-代数分析攻击、不可能差分攻击、零相关线性攻击、子空间路径攻击、混合差分攻击、交换攻击、折回镖攻击和中间相遇攻击等。目前而言，针对 AES 算法的所有已知攻击均不存在比穷尽密钥搜索更快的攻击。

3.3　典型非对称密码算法 RSA

3.3.1　RSA 算法的数学基础

1. 素数

素数又叫质数，是在大于 1 的自然数中只能被 1 和其自身整除的数。每个自然数都可以唯一地分解成有限个素数的乘积，素数因此构成了自然数体系的基石。2300 多年前，古希腊数学家欧几里得在《几何原本》中证明了素数有无穷多个，并提出一些素数可写成“2^p-1”（其中 p 也是素数）的形式。

【梅森素数】

由于这种特殊形式的素数具有独特数学性质，许多著名数学家以及无数数学爱好者对它情有独钟。其中，17 世纪的法国数学家、法兰西科学院奠基人梅森在这方面有过重要贡献。为了纪念梅森，数学界就将“2^p-1”型的素数称为“梅森素数”。梅森素数貌似简单，但当指数 p 值较大时，其素性检验的难度就会很大。正如梅森推测：“一个人，使用一般的验证方法，要检验一个 15 位或 20 位的数字是否为素数，即使终生的时间也是不够的。”享有“数学英雄”美誉的瑞士数学家及物理学家欧拉 1772 年在双目失明的情况下，以顽强毅力靠心算证明了 $2^{31}-1$ 是第 8 个梅森素数；该素数有 10 位，堪称当时世界上已知的最大素数。在“手算笔录年代”，人们历尽艰辛，共计才找到 12 个梅森素数。随着计算机时代的到来，大大加快了梅森素数的探究步伐。1996 年初，美国数学家及程序设

计师沃特曼编制了一个梅森素数计算程序，并把它放在网页上免费使用。这一计算程序就是著名的互联网梅森素数大搜索(Great Internet Mersenne Prime Search，GIMPS)项目，也是全球首个基于互联网的网格计算项目。目前，全球有近 70 万人参与该项目，动用了超过 180 万核中央处理器联网来寻找梅森素数(如图 3-16 所示)——这在数学史上前所未有，在科学史上也极为罕见。

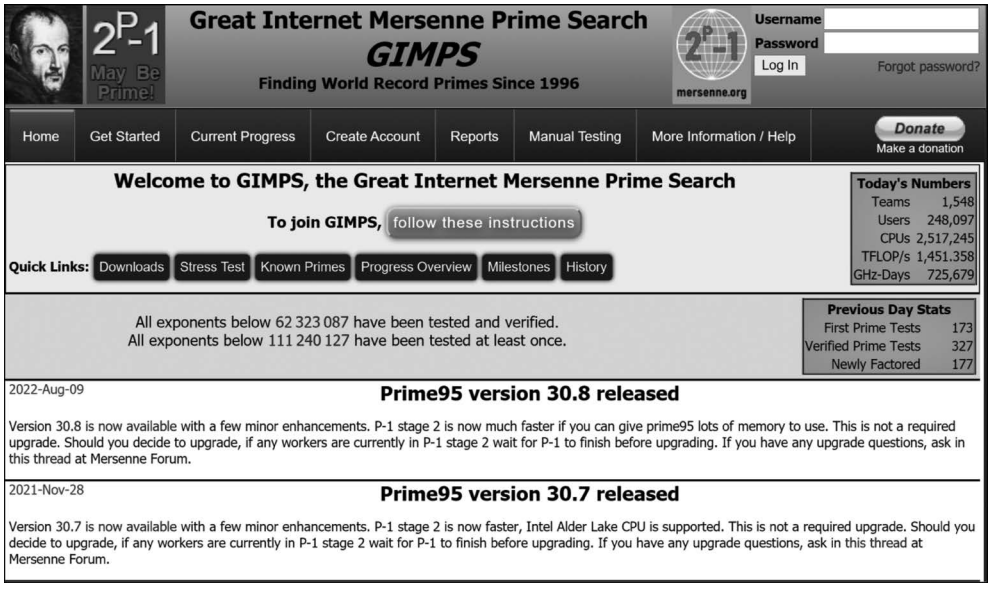


图 3-16　GIMPS 项目

用因式分解法可以证明，若 2^p-1 是素数，则指数 p 也是素数；反之，当 p 是素数时，2^p-1 却未必是素数。前几个较小的梅森数大都是素数，然而梅森数越大，梅森素数也就越难出现。目前仅发现五十几个梅森素数。2019 年，一位名叫帕特里克·罗什的美国人利用 GIMPS 项目，成功发现第 51 个梅森素数 $2^{82589933}-1$；该素数有 24862048 位，是迄今为止人类发现的最大素数。如果用普通字号将它打印下来，其长度将超过 100 千米！是否存在无穷多个梅森素数是未解决的著名难题之一。

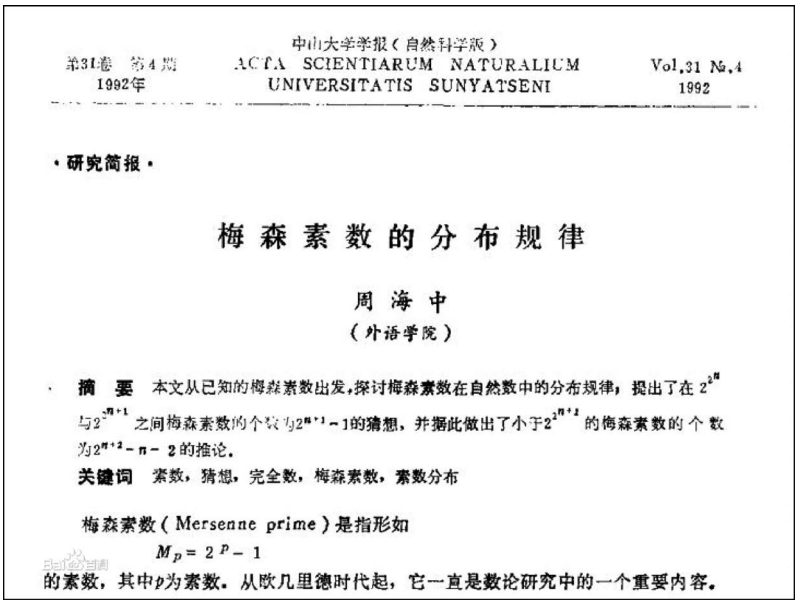
寻找梅森素数最新的意义是：它促进了分布式计算技术的发展。从最新的 17 个梅森素数是在因特网项目中发现这一事实，可以想象到网络的威力。分布式计算技术使得用大量个人计算机去做本来要用超级计算机才能完成的项目成为可能，这是一个前景非常广阔的领域。它的探究还推动了快速傅里叶变换的应用。

梅森素数在实用领域也有用武之地，现在人们已将大素数用于现代密码设计领域。其原理是：将一个很大的数分解成若干素数的乘积非常困难，但将几个素数相乘却相对容易得多。在这种密码设计中，需要使用较大的素数，素数越大，密码被破译的可能性就越小。

课程思政

由于梅森素数的探究需要多种学科和技术的支持，也由于发现新的“大素数”所引起的国际反响，使得对于梅森素数的研究能力已在某种意义上标志着一个国家的科技水平，而

不仅仅是代表数学的研究水平。我国数学家及语言学家周海中于 1992 年在《梅森素数的分布规律》一文中关于梅森素数分布的研究成果被国际上命名为“周氏猜测”，如图 3-17 所示。通过向学生介绍我国数学家的成就，激发学生的爱国热情和民族自豪感，以及学好该课程的信心和动力。



中山大学学报（自然科学版）
第31卷 第4期 1992年
ACTA SCIENTIARUM NATURALIUM UNIVERSITATIS SUNYATSENI
Vol.31 No.4 1992

·研究简报·

梅森素数的分布规律

周海中
（外语学院）

摘 要 本文从已知的梅森素数出发，探讨梅森素数在自然数中的分布规律，提出了在 2^{2^n} 与 $2^{2^{n+1}}$ 之间梅森素数的个数为 $2^{n+1}-1$ 的猜想，并据此做出了小于 $2^{2^{n+2}}$ 的梅森素数的个数为 $2^{n+2}-n-2$ 的推论。

关键词 素数，猜想，完全数，梅森素数，素数分布

梅森素数（Mersenne prime）是指形如

$$M_p = 2^p - 1$$

的素数，其中 p 为素数。从欧几里德时代起，它一直是数论研究中的一个重要内容。

图 3-17 我国数学家及语言学家周海中和“周氏猜测”

2. 互质

互质又被称为互素，若 $N(N \geqslant 2)$ 个整数的最大公因数是 1，则称这 N 个整数互质。互质的主要性质有：

（1）任意两个质数构成互质关系，比如 13 和 61。

（2）一个数是质数，另一个数只要不是该数的倍数，两者就能构成互质关系，比如 3 和 10。

（3）如果两个数之中，较大的那个数是质数，则两者构成互质关系，比如 97 和 57。

（4）1 和任一自然数都是互质关系，比如 1 和 99。

（5）p 是大于 1 的整数，则 p 和 $p-1$ 构成互质关系，即相邻的两个自然数是互质数，比如 57 和 56。

（6）p 是大于 1 的奇数，则 p 和 $p-2$ 构成互质关系，即相邻的两个奇数是互质数，比如 17 和 19，17 和 15。

3. 大素数因子分解理论

已知两个大的质数，计算其乘积是很简单的。但反过来，已知两个大质数的乘积，将其分解为两个质数是非常困难的，即因其计算量非常大，目前还没有算法在较短时间内能有效求解该问题。表 3-4 给出了在现有计算机算力下，完成大整数因子分解需要的时间。但随着量子计算机的发展，这一分解时间将大幅缩短。

表 3-4　完成大数分解需要的时间

整数 n 的十进制位数	因子分解的运算次数	所需计算时间(每微秒一次)
50	1.4×10^{10}	3.9 小时
75	9.0×10^{12}	104 天
100	2.3×10^{15}	74 年
200	1.2×10^{23}	3.8×10^{9} 年
300	1.5×10^{29}	4.0×10^{15} 年
500	1.3×10^{39}	4.2×10^{25} 年

4. 欧拉定理

如果数 x 和数 n 互质，那么 $x^{\phi(n)}=1 \bmod n$。其中，$\phi(n)=(p-1)(q-1)$ 是 n 的欧拉函数值，表示小于 n 的素数的个数，即在小于或等于 n 的正整数中与 n 互质的数的个数，其中，p 和 q 都是素数。

5. gcd(a,b)最大公因数

gcd(a,b)表示两个正整数 a,b 的最大公因数($a>b$ 且 $a \bmod b$ 不为 0)。gcd(a,b)的计算方法主要有欧几里得算法。

6. 欧几里得算法

给定整数 a,b，且 $b>0$，重复使用带余除法，即用每次的余数为除数去除上一次的除数，直至余数为 0。最后一个不为 0 的余数就是 a 和 b 的最大公因数 gcd(a,b)。

【例】 求 gcd{224,34}。

解：

gcd{224,34}余数 20
=gcd{34,20}余数 14
=gcd{20,14}余数 6
=gcd{14,6}余数 2
=gcd{6,2}余数 0
=2
所以 gcd{224,34}=2。

7. 同余关系

若两个自然数对于一个给定的模数有相同的余数，则称这两个数具有同余关系。例如 $7\equiv4 \bmod 3$，即 7 和 4 被 3 除的余数相同，都为 1，则 7 和 4 具有同余关系。

3.3.2　RSA 算法概述

RSA 算法是 1977 年由罗纳德·李维斯特、阿迪·萨莫尔和伦纳德·阿德曼一起提出的。1987 年 7 月首次在美国公布，当时他们三人都在麻省理工学院工作实习。RSA 就是他们三人姓氏开头字母拼在一起组成的。2002 年，RSA 算法的三位发明人因此获得图灵奖。RSA 算法的安全性是基于大数分解的难题，其公钥和私钥是一对大素数的函数，虽然迄今为止还没有从理论上证明破解 RSA 算法的难度等价于分解两个大素数之积，但 RSA 算法能抵抗到目前为止已知的所有密码攻击，已成为公钥密码的国际标准，一

直是最广为使用的非对称密码算法。如表 3-4 所示，这种算法的密钥越长，它就越难破解。根据已经披露的文献，目前被破解的最长 RSA 密钥是 768 位二进制。也就是说，长度超过 768 位的密钥，还无法破解（至少没人公开宣布）。1024 位的 RSA 密钥基本安全，2048 位的密钥极其安全。

只要有网络的地方，就有 RSA 算法。例如，网络服务器端生成公钥和私钥。在浏览器向服务器索取网页时，服务器将公钥存放在网页中发送给用户 A。用户 A 在发送消息给服务器的时候用公钥给明文加密，再将密文发送给服务器。在 A 的发送过程中，如果密文被黑客 B 截获，B 无法破解 A 发送的密文。因为经 RSA 算法公钥加密的密文必须由私钥解密，虽然 B 可以获得服务器分发的公钥，但是经公钥加密的密文不能由公钥解密。服务器收到 A 发送的密文后，用私钥将其解密，将处理结果用私钥加密后再发送给 A。这时就可能存在风险，如果服务器返还给 A 的密文被 B 截获，虽然 B 无法解密 A 发出的密文，但是 B 有公钥，可以解密服务器发出的密文。通过将服务器发出的密文解密，B 可以得到对自己有潜在价值的信息，这样就有可能对 A 产生不利影响。

为了避免服务器返回的密文被 B 截获并用公钥破解，A 也可以主动利用 RSA 算法生成密钥对，将生成的公钥做好起止标记放在明文的某个位置，用服务器分发的公钥给这个新的明文加密后再发送给服务器；服务器收到密文后用自己的私钥解密，按照起止标记提取出 A 分发的公钥，用此公钥给处理结果加密，将密文返回给 A。虽然 B 可以截获此密文但是无法破解，因为此密文不是用服务器的私钥加密。当 A 收到服务器返回的密文后，用自己保存在网页中的临时私钥将密文解密得到明文，这样就保证了通信的安全性。

课程思政

他们的名字无人知晓，他们的功绩永世长存。密码学背后默默付出的英雄们！

目前我们能了解到的密码学进展，都是军方、保密机构允许公开的。在密码学这个特殊的行业，有很多默默工作、不计个人名利的英雄们。他们并不是没有机会像 RSA 算法三位发明人一样获得图灵奖的殊荣，而是出于国家民族安全需要，暂不能将他们的研究成果公开。或许只能等他们的研究成果彻底过时，官方才会公布资料；或许他们永远无法被人知晓。据相关资料记载，事实上，早在李维斯特、萨莫尔和阿德曼三人独立发明 RSA 概念的几年之前，RSA 的理念实际上已被英国政府通信总部（Government Communications Headquarters，GCHQ）的克利福·柯克斯（Cliff Cocks）提出。因为英国 GCHQ 的工作是保密的，甚至在机密的加密技术领域内也绝非广为人知。因此，学术界最终认定 RSA 算法的发明人仍是李维斯特、萨莫尔和阿德曼，因此他们在 2002 年获得图灵奖。这并没有在任何程度上削弱李维斯特、萨莫尔和阿德曼三人成就的意思，只是希望后人们永远记住密码学背后默默奉献的英雄们！他们的名字可能无人知晓，但他们的功绩将永世长存。推荐电影《暗算》和书籍《红军破译科长曹祥仁》。

3.3.3 RSA 算法原理

RSA 算法可分为密钥生成、加密、解密三部分。

1. 密钥生成

在RSA算法中，须生成两个密钥，一个是可以公开的密钥，一般称为“公钥”；另一个是不能公开的密码，一般称为“私钥”。RSA算法密钥生成包括以下五步：

(1) 选取两个互质的大素数 p 和 q，且 p 和 q 均需保密，比如 p 和 q 选择100～200位十进制数或更大的素数。

(2) 按式(3-14)计算其欧拉函数：

$$\phi(n)=(p-1)(q-1) \tag{3-14}$$

$$n=p\times q \tag{3-15}$$

式中，n 表示一个十进制数，n 转换为二进制的位数即为RSA算法密钥的位数，一般取1024位，重要场合取2048位。使用中称“多少位RSA算法”中的“位”就是指密钥的二进制位数。因此，p 和 q 具体的长度是由 n 确定的。

(3) 选一个公开的整数 e，满足 $1<e<\phi(n)$，且 $\gcd\{\phi(n),e\}=1$，实际中 e 一般取65537。

(4) 计算整数 d，满足 $d\times e\equiv 1 \bmod \phi(n)$。其中符号≡表示同余关系，所以有式(3-16)成立。

$$d=[k\phi(n)+1]/e \tag{3-16}$$

式中，k 为整数。式(3-16)具体推导如下。

由同余关系可知，$d\times e\equiv 1 \bmod \phi(n)$ 可表示为 $(d\times e)\bmod\phi(n)=1\bmod\phi(n)$。由于 $\phi(n)$ 为两个大素数的欧拉函数，所以 $(d\times e)\bmod\phi(n)=1\bmod\phi(n)=1$，表示该式 $(d\times e)$ 除以 $\phi(n)$ 的余数为1，所以 $(d\times e)=k\phi(n)+1$，即 $d=[k\phi(n)+1]/e$，且 d 和 e 互质。

(5) 确定 $\{e,n\}$ 为公开密钥，$\{d,n\}$ 为秘密密钥。

2. 加密

(1) 将明文进行分组，使得每个分组对应的十进制数小于 n，即分组长度小于 $\log_2 n$。

(2) 分别对每个明文分组按式(3-17)进行如下加密运算：

$$c_i=m_i^e \bmod n \tag{3-17}$$

式中，m_i 表示第 i 个分组的明文，$m_i<n$(这里假设 m_i 以十进制数表示)；c_i 表示 m_i 加密后的密文。

3. 解密

将密文 c_i 按式(3-18)进行如下计算，得到对应的分组明文 m_i。

$$m_i=c_i^d \bmod n \tag{3-18}$$

将所有明文分组依次排序，即解密出对应的明文。

【例】 在RSA公钥密码体制中，设 $p=7$，$q=13$，$e=5$。当明文 $m=10$ 时，求 d 和相应的密文。

1) 密钥生成

(1) 计算欧拉函数。

$$n=p\times q=7\times 13=91$$

$$\phi(n)=(p-1)(q-1)=6\times 12=72$$

(2) 计算 d。

$d=[k\phi(91)+1]/e=(72k+1)/e=29$，$k$ 为整数。

(3) 确定公钥和私钥。

确定{5,91}为公钥，{29,91}为私钥。

2) 加密

$$c=m^e \bmod n=10^5 \bmod 91=82$$

3) 结果验证

$$m=c^d \bmod n=82^{29} \bmod 91=?$$

其中，82^{29} 是一个非常大的数，已经超出计算机所允许的整数取值范围，那如何才能计算出来呢？这里要运用到 3.2.1 节学习过的模运算性质，通过快速指数法来计算：

$$(a \bmod n \times b \bmod n) \bmod n=(a \times b) \bmod n$$

$$a^b \bmod n=(a \bmod n)^b \bmod n$$

【快速指数法】

$82^{29} \bmod 91$

$=(82\times 82^{28}) \bmod 91$

$=(82 \bmod 91\times 82^{28} \bmod 91) \bmod 91$

$=(82\times(82^2)^{14} \bmod 91) \bmod 91$

$=(82\times 6724^{14} \bmod 91) \bmod 91$

$=(82\times((6724 \bmod 91)^{14}) \bmod 91) \bmod 91$

$=(82\times(81^{14} \bmod 91) \bmod 91$

$=(82\times(81^2)^7 \bmod 91) \bmod 91$

$=(82\times(6561^7 \bmod 91)) \bmod 91$

$=(82\times(6561 \bmod 91)^7 \bmod 91) \bmod 91$

$=(82\times 9^7 \bmod 91) \bmod 91$

$=(82\times 9) \bmod 91$

$=10$

注：当 $a<n$ 时，$a \bmod n=a$，所以 $82 \bmod 91=82$。

由于 RSA 算法的计算量较大，在实际应用中很少用于大块的数据加密，通常在混合密码系统中用于加密会话的密钥，或用于数字签名和身份认证等短消息加密。

3.3.4 针对 RSA 算法的攻击

1. 共模攻击

在存在密钥统一生成的中央机构的情况下，为了节约资源等原因，中央机构往往会选择同一个整数 n，并利用 n 给不同的用户产生各自的公钥-私钥对：(e_1,d_1)，(e_2,d_2)，…，(e_i,d_i)，则第 i 个用户的公钥和私钥分别为(e_i,n)和(d_i,n)。此时，任意一个用户都可以使用自己的公钥-私钥对计算出所有用户公用的 $\phi(n)$，进而计算出任意其他用户的私钥(d_i,n)，从而解密他们的密文。

共模攻击的另一种情况是对于一个外部的攻击者，在不知道任意一个公钥对应私钥的情况下的攻击方法。假如有一个系统内部的广播消息，使用两组或以上的公钥加密，分别发

送给对应的系统用户。外部的攻击者在监听到两个密文(这两个密文加密的消息是相同的)的情况下,可以使用扩展欧几里得算法高效地恢复出加密的消息。假设两个密文分别被公钥(e_1,n)和(e_2,n)加密成如式(3-19)所示的形式:

$$\begin{aligned} c_1 &= m^{e_1} \bmod n \\ c_2 &= m^{e_2} \bmod n \end{aligned} \tag{3-19}$$

此时,假设 e_1 和 e_2 互素,则攻击者可以使用扩展欧几里得算法计算出 $s_1e_1+s_2e_2=1$。然后通过式(3-20)所示过程恢复出明文 m:

$$c_1^{s_1} \cdot c_2^{s_2} \bmod n = m^{s_1e_1+s_2e_2} \bmod n = m \tag{3-20}$$

2. 小指数攻击

小指数攻击存在两种形式,分别是小私钥指数 d 攻击和小公钥指数 e 攻击。

1) 小私钥指数 d 攻击

小私钥指数 d 攻击,即选择一个较小的 d,虽然可以加快解密速度,但是这也使 RSA 加密参数的选择极其容易遭受穷举 d 攻击。即使当选择 d 的长度约为 $\lambda/2$ 时(λ 为 p 和 q 的长度),依然存在某些攻击方法来恢复出 d。

2) 小公钥指数 e 攻击

当公钥中的 e 值过小时,尤其是选择 $e=3$ 时,RSA 加密算法存在更多的攻击点。例如,当使用 e 来加密一个长度小于 $|n|/3$ 的消息时,即根据式(3-16)计算密文 $c=m^e \bmod n$。由于 m 的长度小于 $|n|/3$,因此 c 在数值上小于 n,即在加密过程中没有进行任何模运算。此时消息 m 可以直接通过计算 c 的实数立方根得出。

小公钥指数 e 攻击需要限定加密消息长度小于 $|n|/3$,利用中国剩余定理,该攻击可以将攻击的消息扩展到任意长度。由于共模攻击的存在,中央机构给三个用户分别选择了三组不同的大素数(p_1,q_1)、(p_2,q_2)和(p_3,q_3),根据式(3-15)产生了对应的 n_1、n_2 和 n_3,但是却使用了同样的 $e=3$。运用上述三个不同的公钥$(n_1,3)$、$(n_2,3)$和$(n_3,3)$对消息 m 加密,可以得到三份密文:

$$\begin{aligned} c_1 &= m^3 \bmod n_1 \\ c_2 &= m^3 \bmod n_2 \\ c_3 &= m^3 \bmod n_3 \end{aligned}$$

此时,如果攻击者得到这三份密文 c_1、c_2 和 c_3 及对应的公钥$(n_1,3)$、$(n_2,3)$和$(n_3,3)$,可以利用中国剩余定理解密消息 m。具体过程如下:

令 $n=n_1\times n_2\times n_3$,则可以得到一个密文 c,即

$$\begin{aligned} c &= c_1 \bmod n_1 \\ c &= c_2 \bmod n_2 \\ c &= c_3 \bmod n_3 \\ c &= m^3 \bmod n \end{aligned} \tag{3-21}$$

由于 $m<n$,且 $m^3<n$,因此有 $c=m^3$,即明文 m 可通过计算密文 c 的实数立方根直接获得。因此抵抗小指数攻击最简单的办法就是 e 和 d 都取较大的值。

综上,因为 RSA 算法密钥的产生受素数产生技术的限制,所以也有它的局限性。

(1) 密钥的产生受素数产生技术的限制,因而难以做到一次一密,分组长度太大。

(2) 为保证安全性,n 至少需要 600bit。但这样做的代价是运算量很高,加解密速度比对称密码算法慢几个数量级。随着大整数素因数分解算法的改进和计算机计算能力的提高,对 n 的长度要求在不断增加,因此不利于实现数据格式的标准化。

3.4 国产密码算法

3.4.1 国产密码概述

国产密码算法简称国密算法,是由国家密码管理局认定的拥有自主知识产权的密码算法。2013 年斯诺登事件曝光,美国国家安全局(National Security Agency,NSA)曾与加密技术巨头——RSA 公司签订秘密交易,交易中 NSA 可以获取 RSA 加密算法中的后门,这一事件标志着加密算法的自主性缺失成为国家信息安全的威胁。我国自 2002 年以来就不断推动研发国密算法,并于 2012 年批准 SM2、SM3、SM4 等密码学算法作为行业标准。目前已经公布的国密算法包括 SSF33、SM1、SM2、SM3、SM4、SM7、SM9 以及祖冲之(ZUC)密码等算法,其中 SM1、SM4、SM7、ZUC 密码是对称密码算法;SM2、SM9 是非对称密码算法;SM3 是哈希算法。

3.4.2 祖冲之密码算法

祖冲之密码算法,是一种基于线性返回移位寄存器设计的密码算法,于 2012 年 3 月成为密码行业标准,于 2016 年 10 月成为国家标准,2018 年 4 月以补篇形式进入,由 ISO 及 IEC 联合制定国际标准 ISO/IEC 18033—4,是中国第一个成为国际密码标准的密码算法,广泛应用于无线通信领域,我国 4G 入网检测已要求手机终端全部支持祖冲之密码算法。

祖冲之密码算法共有三层:线性反馈移位寄存器、比特重组和非线性函数。线性反馈移位寄存器定义在素数域上,增加了线性复杂度,提高了算法安全性;比特重组也使攻击变得困难;非线性函数使用 S 盒、异或等运算,降低了能耗和硬件门电路数目。

祖冲之密码算法的软硬件实现是近年来信息安全领域的热点,高通公司研究人员在高通 700MHzHexagonDSP 上实现了祖冲之密码算法,其性能如表 3-5 所示。Lei Wang 等给出了基于现场可编程门阵列(Field Programmable Gate Array,FPGA)的几种优化硬件实现,其性能如表 3-6 所示。可见祖冲之密码算法在软硬件实现上都有优异的性能。

表 3-5 祖冲之密码算法软件实现性能

长度/Byte	128	256	512	1024	1500	2048	4096
速度/bit/轮	0.3989	0.5460	0.6694	0.7547	0.7865	0.8060	0.344

表 3-6 祖冲之密码算法硬件实现性能

序号	频率/MHz	面积/m^2	吞吐量/Mbps	吞吐量/Mbps
1	126	311	2016	6.5
2	108	356	2456	9.7
3	222.4	575	7111	12.3

3.4.3　国密算法与国际加密算法对比

1. SM1 与 AES 算法

SM1 算法是国家密码管理局审批的密码算法，其分组长度和密钥长度都是 128bit，但该算法不公开，仅固化在加密芯片中。SM1 算法安全性和软硬件实现性能与 AES 算法相当，表 3-7 给出 SM1 与 AES 算法的相关对比。

表 3-7　SM1 与 AES 算法对比

项　　目	SM1 算法	AES 算法
计算结构	基于椭圆曲线	Rijndael 结构
存储空间长度	128bit	128/192/256bit
加密解密速度	约几十兆 bps	约 50Mbps

2. SM4 与 DES 算法

SM4 算法是我国自主设计的分组对称密码算法，是无线局域网标准的分组数据算法，密钥长度和分组长度为 128bit。SM4 算法在计算过程中加入了非线性变换，安全性高于 3DES 算法。表 3-8 给出了 SM4 与 DES 算法的相关对比。

表 3-8　SM4 与 DES(3DES)算法对比

项　　目	SM4 算法	DES(3DES)算法
计算轮数	32 轮	16 轮
分组长度	128bit	64bit
密钥长度	128bit	64bit
加密解密速度	约几十兆 bps	约 50Mbps
性能	硬件、软件实现均快	硬件实现较快，软件实现较慢
安全性	较高	较低(3DES 较高)

3. SM2 与 RSA 算法

SM2 算法由国家密码管理局于 2010 年 12 月 17 日发布，全称为椭圆曲线算法。RSA 算法的安全性基于素数分解问题，其复杂度是亚指数级的，SM2 算法基于离散对数问题，目前数学上还找不到亚指数级的算法求解该问题，所以同等安全性要求下 SM2 算法需要的密钥长度更短。SM2 与 RSA 算法对比如表 3-9 所示。

表 3-9　SM2 与 RSA 算法对比

项　　目	SM2 算法	RSA 算法
计算结构	基于椭圆曲线	基于可逆模幂运算
复杂度	指数级	亚指数级
加密解密速度	约几兆 bps	约 0.1Mbps
性能	硬件、软件实现均快	硬件实现较快，软件实现较慢
相同安全性所需公钥位数	较少(160 位的 SM2 与 1024 位的 RSA 安全性相近)	较多

国密算法相比于国际加密算法其安全性和执行速度具有一定的优势。国密算法与国际加密算法相比，每组算法执行占用内存大小相当，作为我国拥有自主知识产权的加密算法，能够在不依赖别国技术的情况下保障信息安全。

本章小结

本章首先从密码学的基本概念出发，介绍密码学的历史、密码系统、密码技术的分类等基本知识。随后，分别介绍了物联网实际应用中的典型对称密码算法 AES 算法和非对称密码算法 RSA 算法的基本原理。最后，在此基础上介绍了几种国产密码算法。

思考与练习

1. 红蓝军通过对称密码通信需要多少个密钥？

2. 简述密码系统的组成。哪些要素是必须保密的？哪些要素是可以公开的？

3. 试分析下例中的哪部分是密钥 K，加密算法 E，解密算法 D。

密文：KCOCUQNFKGT

明文：IAMASOLDIER

4. 对称密码体制和非对称密码体制有什么区别？

5. Alice 生成了一个公钥和一个私钥。不幸的是，她后来把私钥弄丢了。试回答如下问题：

(1) 其他人能发送给她加密信息吗？

(2) 她能解密以前收到的消息吗？

(3) 她能对文档进行数字签名吗？

(4) 其他人能验证她以前签过名的文件吗？

6. 假设在下列算力条件下：

(1) 每台计算机每秒可尝试 10^{20} 个密钥；

(2) 共有 10^{100} 台计算机；

(3) 所有的计算机可运转 10^{20} 年。

如果依然要保证无法通过暴力破解遍历整个密钥空间，则对称密码的密钥需要达到多少 bit？请从下列选项中选出正确的答案。

A. 只要 512bit 就够了

B. 至少需要 1024bit

C. 至少需要 4096bit

D. 至少需要 10000bit

E. 100000bit 也不够

注：这里假设的算力是远远超过实际情况的。相比之下，超级计算机“京”每秒可执行 10^{16} 次浮点运算，宇宙中所有粒子的总数为 10^{87} 个左右，宇宙的年龄为 10^{11} 年左右。

7. AES 算法中 S 盒的作用是什么？若 S2 的输入为 110101，其输出是什么？

8. 说明 RSA 算法中密钥产生和加解密的过程。如何寻找大素数？

9. 柯克霍夫原则是(　　)。

A. 密码系统的任何细节已为人悉知，该密码系统不可用

B. 密码系统的任何细节已为人悉知，该密码系统仍可用

C. 密码系统的任何细节已为人悉知，只要密钥未泄露，它也应该是安全的

D. 密码系统的任何细节已为人悉知，密钥泄露了也是安全的