密码学原理及案例分析

单广荣 齐爱琴 主编

科学出版社

基金项目：甘肃省重点学科计算机科学与技术

密码学原理及案例分析

单广荣　齐爱琴　主编

科学出版社

北京

内 容 简 介

本书内容涵盖了密码学技术、PKI 技术、PMI 技术、身份认证技术、无线安全技术等几个方面，实践项目既包含对密码学原理的理解和运用，又融合当今网络安全的某些主流技术，以适应基础与验证性、综合和设计性两种不同层次的要求。

本书共 6 章，第 1 章介绍密码学技术，第 2 章介绍 PKI 技术，第 3 章介绍 PMI 技术，第 4 章介绍身份认证技术，第 5 章介绍无线安全技术，第 6 章介绍数据备份及恢复技术。

本书可作为计算机类、网络技术类和信息安全类相关专业本科生的辅助教材，也可作为网络安全工程师、网络管理员的参考书以及密码学方面的实践培训教材。

图书在版编目(CIP)数据

密码学原理及案例分析/单广荣，齐爱琴主编. —北京：科学出版社，2017.11
ISBN 978-7-03-053269-5

Ⅰ. ①密… Ⅱ. ①单… ②齐… Ⅲ. ①密码学－研究 Ⅳ. ①TN918.1

中国版本图书馆 CIP 数据核字(2017)第 128532 号

责任编辑：邹 杰 / 责任校对：郭瑞芝
责任印制：吴兆东 / 封面设计：迷底书装

科学出版社出版
北京东黄城根北街 16 号
邮政编码：100717
http://www.sciencep.com

北京京华虎彩印刷有限公司印刷

科学出版社发行 各地新华书店经销

*

2017 年 11 月第 一 版 开本：787×1092 1/16
2018 年 1 月第二次印刷 印张：16 1/2
字数：380 000

定价：58.00 元

(如有印装质量问题，我社负责调换)

前　言

随着计算机科学技术、通信技术、微电子技术的发展，计算机和通信网络的应用进入了人们的日常生活和工作中，出现了电子商务、电子政务、电子金融等必须确保信息安全的网络信息系统，密码技术在解决网络信息安全问题中发挥着重要作用。使用密码技术可以有效地保障信息的机密性，也可以保护信息的完整性和真实性，防止信息被篡改、伪造和假冒等。密码技术是信息安全的基础技术，密码学是信息安全学科建设和信息系统安全工程实践的基础理论之一。

密码学的特点是理论性与实践性都很强，涉及的知识面较广，概念繁多，并且比较抽象，仅靠课堂教学，学生难以理解和掌握。在学习一般性原理和技术的基础上，必须通过一定的案例分析训练，才能真正掌握其内在机理。为了进一步提高学生的综合应用和设计创新能力，西北民族大学数学与计算机科学学院联合西普科技于2011年共同建立了计算机网络与信息安全实验室。实验室不仅满足了学生对各种安全设备的操作培训，还能从计算机网络安全知识及技能的角度出发，培养学生的理论认知及实践能力，能够从原理验证、实训应用、综合分析、自主设计及研究创新等多个层次培养学生的综合素质。

本书结构合理，可读性强，注重应用。既包含了对密码学原理的理解和运用，又融合了当今网络安全的某些主流技术，以适应基础与验证性、综合和设计性两种不同层次的要求。此外，本书增加了无线加密技术的案例分析内容和数据备份及恢复内容。熟练使用这些工具和设备，对学生提高密码学水平、积累网络安全实践经验具有非常重要的意义。

本书由西北民族大学数学与计算机科学学院单广荣、齐爱琴编写，负责全书统筹及策划、撰写并且修改第1～6章，王倩、王燕凤负责全书校对。作者均为从事计算机网络安全教学、科研的一线教师，有丰富的教学实践经验，本书结构严谨、概念准确，内容组织合理、语言使用规范。

在本书写作过程中，得到诸多专家和领导的热情支持与指导，在此一并表示衷心感谢。由于作者水平有限，书中难免存在不足之处，恳请广大读者批评指正。

作　者

2017年6月

目　录

第1章 密 码 学

密码学(Cryptology)作为数学的一个分支，是密码编码学(Cryptography)和密码分析学(Cryptanalysis)的统称。密码学通过加密变换，将可读的信息变换为不可理解的乱码，从而起到保护信息和数据的作用，直接支持机密性、完整性和不可否认性。当前信息安全的主流技术和理论都是基于以算法复杂性理论为特征的现代密码学的。密码学的发展历程大致经历了三个阶段：古代加密方法(手工阶段)、古典密码(机械阶段)和近代密码(计算机阶段)。

密码学中的五元组为{明文、密文、密钥、加密算法、解密算法}，对应的加密方案称为密码体制。明文是作为加密输入的原始信息，通常用 m 来表示，明文空间通常用 M 表示；密文是明文加密变换后的结果，通常用 c 表示，密文空间通常用 C 表示；密钥是参与密码转换的参数，通常用 k 表示，密钥空间通常用 K 表示；加密算法是将明文变换为密文的变换函数，加密过程通常用 E 表示，即 $c = E_k(m)$；解密算法是将密文恢复为明文的变换函数，解密过程通常用 D 表示，即 $m = D_k(c)$。

密码体制是指完成加解密功能的密码方案。近代密码学中所出现的密码体制从原理上可分为两大类，即对称密码体制和非对称密码体制。对称密码体制也称为单密钥密码体制，基本特征是加密密钥与解密密钥相同，根据其对明文的处理方式可分为流密码和分组密码。非对称密码体制也称为公开密钥密码体制、双密码体制，其加密密钥和解密密钥不同，形成一个密码对，用其中一个密钥加密的结果可以用另一个密钥来解密。

古典密码学案例分析主要包括移位密码、乘法密码、仿射密码、Playfair 密码、维吉尼亚密码等；流密码加密案例分析包括 RC4 和 LFSR 等算法的加解密案例分析。对称密码基本加密案例分析主要包括 DES、3DES、IDEA、AES 和 SMS4 等算法的加解密案例分析。对称密码工作模式案例分析主要是针对不同算法采用不同的分组方式和填充模式进行加密案例分析；散列函数案例分析包括 MD5、SHA-1/256 和 HMAC 等算法的加解密案例分析。非对称加密案例分析包括 RSA、ELGAMAL、ECC 等算法的加解密案例分析和密钥生成案例分析；数字签名案例分析包括 RSA-PKCS 签名算法、ELGAMAL 签名算法、DSA 签名算法和 ECC 签名算法等签名算法的案例分析；密码学数学基础案例分析主要针对密码学常用的数学知识进行相应的案例分析，包括大数运算、素性测试、模幂、原根、求逆和二次剩余等。文件加解密案例分析包括对文本、图片、音频、视频等多种格式的文件的加解密案例分析。数据库加解密应用案例分析包括单元级加密和数据库级加密案例分析。基于 SSH 协议的通信安全案例分析包括密码认证方式和密钥认证方式的 SSH 案例分析。基于 GnuPG 的加密及签名案例分析使用 GnuPG 实现文件及邮件的加解密和签名。PGP 在文件系统、邮件系统中的应用则使用 PGP 实现文件及邮件的加解密和签名。

1.1 古 典 密 码

古典密码学主要包括移位密码、乘法密码、仿射密码、Playfair 密码、维吉尼亚密码等。

古典密码体制的一般定义为 $M=C=K=Z_{26}$，其中 M 为明文空间，C 为密文空间，K 为密钥空间，Z_{26} 为 26 个整数(对应 26 个英文字母)组成的空间；要求 26 个字母与模 26 的剩余类集合{0,1,2,…,25}建立一一对应的关系。

1.1.1　移位密码

移位密码在加密实现上就是将 26 个英文字母向后循环移动 k 位，其加解密可分别表示为

$$c = E_k(m) = m + k(\text{mod } 26)$$
$$m = D_k(c) = c - k(\text{mod } 26)$$

其中，m、c、k 是满足 $0 \leqslant m,c,k \leqslant 25$ 的整数。

1.1.2　乘法密码

乘法密码通过对字母等间隔抽取以获得密文，其加解密可分别表示如下

$$c = mk(\text{mod } 26)$$
$$m = ck^{-1}(\text{mod } 26)$$

其中，m、c、k 是满足 $0 \leqslant m,c,k \leqslant 25$，且 $\gcd(k,26)=1$ 的整数。

1.1.3　仿射密码

仿射密码的加密是一个线性变换，将移位密码和乘法密码相结合，其加解密可分别表示为

$$c = E_{a,b}(m) = am + b(\text{mod } 26)$$
$$m = D_{a,b}(m) = a^{-1}(c-b)(\text{mod } 26)$$

其中，a、b 是密钥，是满足 $0 \leqslant a,b \leqslant 25$ 和 $\gcd(a,26)=1$ 的整数，即 a 和 26 互素；a^{-1} 表示 a 的逆元，即 $a^{-1} \cdot a \equiv 1 \bmod 26$。

1.1.4　Playfair 密码

Playfair 是一种人工对称加密技术，由 Charles Wheatstone 在 1854 年发明，得名于其推广者 Lord Playfair。Playfair 密码是一种著名的双字母单表替代密码。实际上 Playfair 密码属于一种多字母替代密码，它将明文中的双字母作为一个单元对待，并将这些单元转换为密文字母组合。Playfair 密码基于一个 5×5 的字母矩阵，该矩阵通过使用一个英文短语或单词串即密钥来构造，去掉密钥中重复的字母得到一个无重复字母的字符串，然后将字母表中剩下的字母依次从左到右、从上往下填入矩阵中。

p	l	a	y	f
i	r	s	d	g
m	c	h	e	b
k	n	o	q	t
u	v	w	x	z

图 1.1　字母矩阵

例如，若密钥为“playfair is a digram cipher”，去除重复字母后，得到有效密钥“playfirsdgmche”，可得字母矩阵如图 1.1 所示。

注意：字母 i、j 占同一个位置。

设明文字母对为(P1,P2)，Playfair 密码的加密算法如下。

(1)若 P1、P2 在同一行，密文 C1、C2 分别是紧靠 P1、P2 右端的字母，其中第一列被看作最后一列的右方(解密时反向)。

(2)若 P1、P2 在同一列，密文 C1、C2 分别是紧靠 P1、P2 下方的字母，其中第一行被看作最后一行的下方(解密时反向)。

(3)若 P1、P2 不在同一行，也不在同一列，则 C1、C2 是由 P1、P2 确定的矩形其他两角的字母，且 C1 和 P1 在同一行，C2 和 P2 在同一行(解密时处理方法相同)。

(4)若 P1=P2，则两个字母间插入一个预先约定的字母，如 q，并用前述方法处理；如 balloon，则以 ba lq lo on 来加密。

(5)若明文字母数为奇数，则在明文尾填充约定字母。

1.1.5 维吉尼亚密码

维吉尼亚(Vigenenre)密码是最著名的多表代换密码，是 16 世纪法国著名密码学家 Blaise de Vigenenre 发明的。维吉尼亚密码使用一个词组作为密钥，密钥中每一个字母用来确定一个代换表，每一个密钥字母被用来加密一个明文字母，第一个密钥字母加密第一个明文字母，第二个密钥字母加密第二个明文字母，等所有密钥字母使用完后，密钥再次循环使用，于是加解密前需先将明/密文按照密钥长度进行分组。密码算法可表示如下：

设密钥 $K=(k_1, k_2, \cdots, k_d)$，明文 $M=(m_1, m_2, \cdots, m_n)$，密文 $C=(c_1, c_2, \cdots, c_n)$；

加密变换为 $c_i = E_{ki}(m_i) = m_i + k_i \pmod{26}$

解密变换为 $m_i = D_{ki}(c_i) = c_i - k_i \pmod{26}$

通常通过查询维吉尼亚表进行加解密。

此处以移位密码为例说明，乘法密码、仿射密码、Playfair 密码和维吉尼亚密码可参照完成。

1. 加解密计算

(1)参照案例分析原理，在“明文”栏输入所要加密的明文，在“密钥”栏输入相应的密钥长度，如图 1.2 所示。

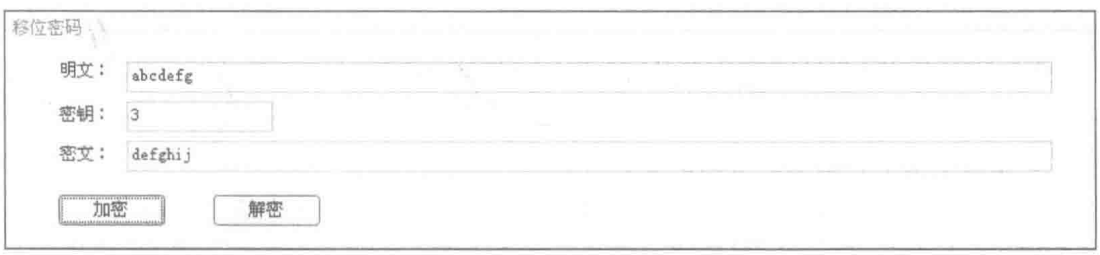

图 1.2 输入明文和密钥长度

(2)单击“加密”按钮，在“明文”文本框内就会出现加密后的密文。

2. 扩展案例分析

(1)单击扩展案例分析下的按钮，进入相应算法的扩展案例分析面板，此处以移位密码扩展案例分析面板，如图 1.3 所示。

图 1.3　移位密码扩展案例分析面板

(2) 在“密钥 k”栏中输入一个 0～25 的整数，如 19，单击“确定”按钮后，系统显示“明文-密文映射表”。

(3) 在“明文”文本框中输入明文(如“Classical Cryptology.”)，并单击“加密”按钮，在“密文”文本框内就会出现加密后的密文。

(4) 解密过程是加密过程的逆过程，在“密文”文本框中输入密文(确保密钥已经正确输入)，单击“解密”按钮即可得到相应的明文。

3. 算法跟踪

1) 加密跟踪

(1) 选择要跟踪的算法即移位密码，在相应的算法计算区域填写明文和密钥。

(2) 单击“跟踪加密”按钮，此时会弹出选择跟踪调试器对话框，对话框中所列出的可选调试器根据系统中所安装的调试器而不同。

(3) 选择“新实例 Microsoft CLR Debugger 2005”或其他调试器并单击“是”按钮，打开调试器窗口，出现选择编码的对话框时选择“(自动检测)”，进入代码跟踪界面。

(4) 选择对应 C#源代码的 xxx.cs 标签页并按下快捷键 F10 开始跟踪，如图 1.4 所示。

(5) 当算法跟踪完毕后，会自动切换回案例分析窗口并显示计算结果。

(6) 查看案例分析面板中的计算结果，然后切换回调试器，单击工具栏中的“停止调试”按钮或按 Shift + F5 快捷键以停止调试。

(7) 关闭调试器，弹出保存对话框，单击“否”按钮不保存。

图 1.4 跟踪过程

2) 解密跟踪

跟踪解密算法时，选择要跟踪的算法，在相应的算法计算区域填写密文和密钥，单击“跟踪解密”按钮会显示解密跟踪过程，具体步骤与加密跟踪步骤类似，在此不再赘述。

1.2 密码学数学基础

大多数运算器只支持小于 64 位的整数运算，无法进行加密算法的运算。为满足加密算法的需要，可通过建立大整数运算库来解决这一问题，通常通过以下两种方式进行处理。

(1) 将大整数当作字符串处理，即将大整数用十进制字符数组表示。这种方式便于理解，但效率较低。

(2) 将大整数当作二进制流进行处理，计算速度快。

大数运算包括素性测试、模幂、原根、求逆、二次剩余等。

1.2.1 素性测试

Monte Carlo 算法和 Las Vegas 算法均为素性测试算法。

1. Monte Carlo 算法

Monte Carlo 算法又称为概率素性检测算法，算法描述如下。

输入：p 为一个正整数。

输出：若 p 为素数，则输出 YES，否则输出 NO。

```
Prime_Test(p)
flag=0;
重复 log2 p 次:
在(1,p-1]区间均匀随机地选取 x;
如果 gcd(x,p)>1 或 x^((p-1)/2) ≠ ±1 mod p, return(NO);
如果 flag=0 且 x^((p-1)/2) ≡ -1 mod p, flag=1;
结束重复;
如果 flag=0, 即在重复中 x^((p-1)/2) ≡ -1 mod p 没有出现过, return(NO);
return(YES)
```

2. Las Vegas 算法

Las Vegas 算法又称为素性证明，算法描述如下。

输入：p 为一个正基数；$q_1, q_2, \cdots, q_k$ 为 $p-1$ 的全体素因子，其中 $k \leqslant \log_2 p$。

输出：若 p 为素数，则输出 YES，否则输出 NO。

```
Prim_ Certify(p,q[k])
    在区间[2,p-1]均匀随机地选取 g
    for(i=1,i++,k)do
        如果 g^((p-1)/q_i) ≡ 1 mod p，输出 NO_DECISION 并终止程序;
        如果 g^(p-1) ≡ 1 mod p，输出 NO 并终止程序;
输出 YES 并终止程序
```

1.2.2 模幂

对于 $b, c < m$，模幂 $b^c \bmod m$ 按照整数幂的通常定义，b 自乘 c 次，但要模 m；模幂算法描述如下。

输入：整数 b、c、m，其中 $b>0$，$c \geqslant 0$，$m>1$。

输出：$b^c \bmod m$。

```
mod_exp(b,c,m)
    if(c=0) return(1);
    if(c mod 2=0) return(mod_exp(b^2 mod m,c/2,m)); //c/2 表示 c 除以 2 取整
     return (b·mod_exp(b^2 mod m,c/2,m))
```

1.2.3 原根

在数论中，特别是整除理论中，原根是一个很重要的概念。

对于两个正整数 $(a,m)=1$，由欧拉定理可知，存在正整数 $d \leqslant m-1$，如欧拉函数 $d=\varnothing(m)$，即小于等于 m 的正整数中与 m 互质的正整数的个数，使得 $a^d \equiv 1(\bmod m)$。

由此，在 $(a,m)=1$ 时，定义 a 对模 m 的指数 $\mathrm{ord}_m(a)$ 为使 $a^d \equiv 1(\bmod m)$ 成立的最小的正整数 d。由前知 $\mathrm{ord}_m(a)$ 一定小于等于 $\varnothing(m)$，若 $\mathrm{ord}_m(a)=\varnothing(m)$，则称 a 是模 m 的原根。

1.2.4 求逆

乘法逆元的定义为：对于$w \in \mathbf{Z}_n$，存在$x \in \mathbf{Z}_n$，使得$wx \equiv 1 \bmod n$，则w是可逆的，称x为w的乘法逆元，记为$x = w^{-1}$，其中$\mathbf{Z}_n$表示小于n的所有非负整数集合。

通常通过扩展欧几里得算法和费马小定理求乘法逆元，此处使用扩展欧几里得算法。

扩展欧几里得算法的定义为：如果有整数f，$\gcd(d,f)=1$，那么d有一个模f的乘法逆元；即对于小于f的正整数d，存在一个小于f的正整数d^{-1}，使得$d \times d^{-1} \equiv 1 \bmod f$。扩展欧几里得算法的具体描述如下。

ExtendedEUCLID(d,f)

(1) $(X_1, X_2, X_3) \leftarrow (1, 0, f)$；$(Y_1, Y_2, Y_3) \leftarrow (1, 0, d)$。

(2) 若 $Y_3 = 0$，返回 $X_3 = \gcd(d,f)$；无逆元。

(3) 若 $Y_3 = 1$，返回 $Y_3 = \gcd(d,f)$；$Y_2 = \mathrm{d} - 1 \bmod f$。

(4) $Q = \lfloor X_3/Y_3 \rfloor$。

(5) $(T_1, T_2, T_3) \leftarrow (X_1 - Q \cdot Y_1, X_2 - Q \cdot Y_2, X_3 - Q \cdot Y_3)$。

(6) $(X_1, X_2, X_3) \leftarrow (Y_1, Y_2, Y_3)$。

(7) $(Y_1, Y_2, Y_3) \leftarrow (T_1, T_2, T_3)$。

(8) 返回(2)。

1.2.5 二次剩余

二次剩余的定义为：a与p互素，p是奇素数，若$x^2 = a \bmod p$，则称a是模p的二次剩余；否则称a是模p的非二次剩余。

二次剩余定理：若p是奇素数，则整数$1, 2, \cdots, p-1$中正好有$(p-1)/2$个是模p的二次剩余，其余的$(p-1)/2$个是非二次剩余。

本案例使用运算器工具完成大数运算、素性测试、模幂、原根、求逆和二次剩余的计算，具体步骤如下。

1) 加、减、乘、除、模、求逆运算

选择进制类型和计算类型，输入要计算的操作数，单击“计算”按钮，显示计算结果，如图1.5所示。

2) 模幂运算

在图1.5所示的界面中，选择进制类型和计算类型，输入要计算的b、e、m值，单击“计算”按钮，显示模幂计算结果。

3) 生成大素数原根

在图1.5的界面中，选择进制类型和计算类型，单击“随机生成”按钮，显示随机生成的大素数以及大素数的原根。

4) 二次剩余判断

在图1.5所示的界面中，选择进制类型和计算类型，输入a、p值，单击“计算”按钮。显示二次剩余的判断结果。

图 1.5　输入界面

5) 素性测试

在图 1.5 所示的界面中，选择进制类型和计算类型，输入待测试的大整数，单击“测试”按钮，显示测试结果。

1.3　流密码加密

流密码(Stream Cipher)也称为序列密码，每次加密处理数据流的一位或一字节，加解密使用相同的密钥，是对称密码算法的一种。1949 年 Shannon 证明只有一次一密密码体制是绝对安全的，为流密码技术的研究提供了强大的支持，一次一密的密码方案是流密码的雏形。流密码的基本思想是利用密钥 K 产生一个密钥流 $k_1k_2\cdots k_n$ 对明文流 $M=m_1m_2\cdots m_n$ 进行如下加密：$C=c_1c_2\cdots c_n=E_{k1}(m_1)E_{k2}(m_2)\cdots E_{kn}(m_n)$。若流密码所使用的是真正随机产生的、与消息流长度相同的密钥流，则此时的流密码就是一次一密的密码体制。

流密码分为同步流密码和自同步流密码两种。同步流密码密钥流的产生独立于明文和密文；自同步流密码密钥流的产生与密钥和已经产生的固定数量的密文字符有关，即是一种有记忆变换的序列密码。

目前常用的流密码有 RC4 密码和基于 LFSR 的流密码。

1.3.1　RC4 流密码算法

RC4 是 1987 年 Ron Rivest 为 RSA 公司设计的一种流密码，是一个面向字节操作、具有密钥长度可变特性的流密码，是目前为数不多的公开的流密码算法。目前的 RC4 至少使用 128 位密钥。RC4 的算法可简单描述为：对于 n 位长的字，有共 $N=2^n$ 个可能的内部置换状态

矢量 $S=S[0], S[1], \cdots, S[N-1]$，这些状态是保密的。密钥流 K 由 S 中的 2^n 个元素按一定方式选出一个元素而生成，每生成一个密钥值，S 中的元素就重新置换一次，自始至终置换后的 S 包含 0～N–1 的所有 n 比特数。

RC4 有两个主要算法：密钥调度算法(KSA)和伪随机数生成算法(PRGA)。KSA 的作用是将一个随机密钥变换成一个初始置换，及相当于初始化状态矢量 S，然后 PRGA 利用 KSA 生成的初始置换生成一个伪随机输出序列。

密钥调度算法的算法描述如下：

```
for i=0 to N-1 do
S[i]=i;
j=0;
for i=0 to N-1 do
j=(j+S[i]+K[i mod L]) mod N;
swap(S[i],S[j]);
```

初始化时，S 中元素的值被设置为 0～N–1，密钥长度为 L 字节，$S[0],\cdots,S[N-1]$对于每个 $S[i]$根据密钥 K 确定的方案，将 $S[i]$置换为 S 中的另一个元素。

伪随机数生成算法的算法描述如下：

```
i=0;
j=0;
while(true)
i=(i+1) mod N;
j=(j+S[i])mod N;
swap(S[i],s[j]);
output k=S[(S[i]+S[j])mod N];
```

PRGA 主要完成密钥流的生成，$S[0],\cdots,S[N-1]$中，对于每个 $S[i]$，根据当前 S 的值，将 $S[i]$与 S 中的另一个元素置换，，当 $S[N-1]$完成置换后，操作再从 $S[0]$开始重复。

加密时将 K 值与下一个明文字节异或，解密时将 K 值与下一密文字节异或。

1.3.2 基于 LFSR 的流密码

线性反馈移位寄存器(Linear Feedback Shift Register, LFSR)是产生密钥流最重要的部件。GF(2)上一个 n 级反馈移位寄存器由 n 个二元存储器与一个反馈函数 $f(a_1,a_2,\cdots,a_n)$组成。每一个二值(0,1)存储器称为反馈移位寄存器的一级，a_i 表示第 i 级存储器的内容。在某一时刻，这些级的内容构成该反馈移位寄存器的一个状态，共有 2^n 种可能的状态，每一状态对应一个 n 维向量$(a_1,a_2,\cdots,a_n)$。

在主时钟确定的周期区间上，每一级存储器 a_i 都将其内容传递给下一级 a_{i-1}，并根据寄存器当前的状态计算 $f(a_1,a_2,\cdots,a_n)$的值作为 a_i 下一时刻的内容，即从一个状态转移到下一状态。

若反馈移位寄存器的反馈函数 $f(a_1,a_2,\cdots,a_n)=c_na_1\oplus c_{n-1}a_2\oplus\cdots\oplus c_1a_n$，其中系数 c_i=0,1，加法为模 2 加，即该反馈函数是 $a_1,a_2,\cdots,a_n$ 的线性函数，则称为线性反馈移位寄存器，用 LFSR 表示，如图 1.6 所示。