第3章 非线性组件

对于分组密码来说，密文随机性是衡量其是否安全的重要指标。从数学的角度解释，就是将明文作为自变量，将密文作为因变量，因变量越随机越好。如果密码变换是线性函数，则不安全；非线性变换相对安全，但是随着对确定型非线性函数的深入研究，简单的密码变换也不再安全。因此，要使用非确定型非线性函数作为密码变换，例如引入密钥或参数来增强非确定性；同时，若对于确定密钥无法列出该非线性函数的具体表达式，也能够使密码变换足够安全。

为了更好地实现非线性强度，只能在目前计算机可承受的计算能力范围内设计足够好的小规模非线性组件，如常用的S盒就是其中的一种，然后将这些组件通过一定的方式组合，大多数情况下采用某个代数结构上的线性组合，进而形成输入输出规模较大的可逆非线性函数，这样的函数经过多次重复迭代，每次迭代中加入不同密钥，最后形成具体的分组密码方案。

分组密码的非线性组件除了S盒之外，还有一种ARX结构，同样可以提供非线性功能，即模加(Add)、循环移位(Rotation)和异或(Xor)3种运算组合在一起的结构，这种结构在很多分组密码和哈希函数中用到，具有优秀的混淆、扩散作用。目前发展最成熟的非线性组件是S盒，因此，S盒的密码指标和设计方法备受关注[71-85]，本章主要从这两方面进行介绍。

3.1 S盒的密码指标

S盒通常是分组密码和哈希函数中最重要的非线性组件之一，它的密码性能是决定整个密码算法安全性的关键，同时S盒的工作速度与整个算法的混淆速度成正比。对S盒的深入研究不仅有助于得到密码性能更优的非线性函数，进而提升分组密码的设计水平，而且对于密码算法分析也有相当重要的价值。

假设S盒输入为 n 比特，输出为 m 比特，则可以表示为映射

$$S(x)=(f_1(x),f_2(x),\cdots,f_m(x)):F_2^n\rightarrow F_2^m$$

简称 $S(x)$ 是一个 $n\times m$ 的S盒，每一个分量函数 $f_i(x)$ 是 n 比特输入、1比特输出的布尔函数($1\leqslant i\leqslant m$)。当参数 m 和 n 选择得很大时，搜索某些攻击所用的统计特性比较困难；但是，m 和 n 过大会给S盒的设计实现带来困难，如增加算法的存储量、降低算法执

行速度等。从实现设计和安全性评估两个角度折中考虑，目前比较流行的是 4×4、8×8 的 S 盒。

S 盒的设计必须满足几个主要密码指标，通常包括差分均匀度、非线性度、代数次数、项数、代数免疫度、扩散特性和严格雪崩特性。

3.1.1　差分均匀度

差分分析和线性分析的提出给久经考验的 DES 密码以致命一击，尽管这两种攻击方法需要大量的选择明文或已知明文，但至少打破了 DES 牢不可破的梦想，可以在远少于强力攻击的时间内恢复主密钥，而导致分析成功的根本原因在于 DES 的 S 盒密码性能较弱。此后，越来越多的学者开始着力设计“强”的 S 盒，以增强基于 S 盒设计的分组密码的安全性。大量研究表明，S 盒抗差分分析的能力本质上取决于它的差分分布表和差分均匀度。

定义 3-1　称 $2^n\times 2^m$ 阶矩阵 $\boldsymbol{\Lambda}(S)$ 为 $n\times m$ 的 S 盒的差分分布表，如果

$$\boldsymbol{\Lambda}(S)=\begin{pmatrix}\lambda_{00} & \lambda_{01} & \cdots & \lambda_{0(2^m-1)}\\ \lambda_{10} & \lambda_{11} & \cdots & \lambda_{1(2^m-1)}\\ \vdots & \vdots & \ddots & \vdots\\ \lambda_{(2^n-1)0} & \lambda_{(2^n-1)1} & \cdots & \lambda_{(2^n-1)(2^m-1)}\end{pmatrix}$$

其中 $\lambda_{\alpha\beta}=|\{x\in F_2^n\mid S(x)\oplus S(x\oplus\alpha)=\beta\}|$，$\alpha=0,1,\cdots,2^n-1$，$\beta=0,1,\cdots,2^m-1$。

显然，$\lambda_{0\beta}=\begin{cases}2^n(\beta=0)\\ 0(\beta\neq 0)\end{cases}$，$2\mid\lambda_{\alpha\beta}$，并且 $\sum\limits_{\beta=0}^{2^m-1}\lambda_{\alpha\beta}=2^n$。特别地，当 S 为置换时，$\lambda_{\alpha 0}=\begin{cases}2^n(\alpha=0)\\ 0(\alpha\neq 0)\end{cases}$。

差分分析的关键在于利用了 S 盒差分分布表中的部分非零元素 $\lambda_{\alpha\beta}$，如果这些元素值明显大于其他各元素值，则对应的输入输出差分概率较大，即 $p(\alpha\to\beta)=\dfrac{\lambda_{\alpha\beta}}{2^n}$，有利于进行差分分析，为此引入差分均匀度的概念。

定义 3-2　设 $n\times m$ 的 S 盒的差分分布表为 $\boldsymbol{\Lambda}(S)=(\lambda_{\alpha\beta})$，则称

$$\delta(S)=\max\{\lambda_{\alpha\beta}\mid\alpha=1,2,\cdots,2^n-1,\beta=0,1,\cdots,2^m-1\}$$

为 S 盒的差分均匀度。

例 3-1　设 S 盒输入 3 比特，输出 3 比特，如表 3-1 所示。

表 3-1　S 盒的输入输出

输入	000	001	010	011	100	101	110	111
输出	010	101	011	111	110	100	000	001

对于每一个输入差分 α 和输出差分 β，遍历 $0\leqslant x\leqslant 7$，对满足等式 $S(x)\oplus S(x\oplus\alpha)=\beta$ 的 x 计数，即得到 $\lambda_{\alpha\beta}$。以此类推，得到 $\boldsymbol{\Lambda}(S)$ 中的所有元素。S 盒的差分分布表如表 3-2 所示。

表 3-2　S 盒的差分分布表

$\alpha\backslash\beta$	000	001	010	011	100	101	110	111
000	8	0	0	0	0	0	0	0
001	0	2	2	0	2	0	0	2
010	0	2	2	0	0	2	2	0
011	0	0	0	0	2	2	2	2
100	0	2	0	2	2	0	2	0
101	0	0	2	2	0	0	2	2
110	0	0	2	2	2	2	0	0
111	0	2	0	2	0	2	0	2

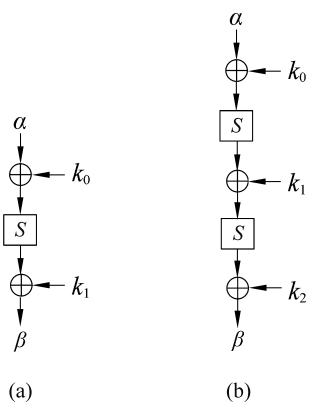
表 3-2 中输入差分为 001 时，输出差分为 001 的概率为 $p(001\rightarrow001)=\frac{2}{8}=\frac{1}{4}$，以此类推。容易看出，这个 S 盒的差分均匀度为 2，输入输出差分的最大概率为$\frac{1}{4}$。

为了抵抗差分分析，S 盒的差分均匀度应当尽可能小。对于 $n\times m$ 的 S 盒，差分均匀度可能达到的最小值为 2^{n-m+1}。但当 $m>n$ 时，S 盒不具备正交性，因此只能寻求 $m<n$ 时 $\delta(S)\geqslant2^{n-m+1}$ 的 S 盒。

定义 3-3　给定 $n\times m$ 的 S 盒，若 $\boldsymbol{\Lambda}(S)$ 的每一行（第一行除外）均有相等数目（一般为 2^{m-1}）的 0 和非零元，且非零元的值均为 2^{n-m+1}，则称 S 为几乎完全非线性（Almost Perfect Nonlinear，APN）置换，此时 $\delta(S)=2^{n-m+1}$。

特别地，当 S 为双射函数时，称 S 为 APN 置换，此时 $\delta(S)=2$，达到最小值。

定理 3-1　对于 $n\times n$ 的 S 盒，当 n 为奇数时有最小差分均匀度 $\delta(S)=2$，当 n 为偶数时通常有 $\delta(S)>2$。



(a)　(b)

图 3-1　采用相同 S 盒的两种迭代加密结构

图 3-1 是采用相同 S 盒的两种迭代加密结构，输入输出差分为 (α,β)。假设 S 盒输入和输出均为 n 比特，差分均匀度为 σ，则这两种结构的差分概率 $p(\alpha,\beta)$ 并不相同。

对于图 3-1(a)的结构，最大差分特征概率由 S 盒的差分均匀度决定，即 $p(\alpha,\beta)\leqslant\sigma/2^n$；对于图 3-1(b)的结构，最大差分特征概率由两个 S 盒的差分均匀度决定，即根据马尔可夫迭代法则，为 $p(\alpha,\beta)\leqslant\sigma^2/2^{2n}$，有可能小于随机函数概率。因此，多轮迭代会减小特定差分路径的概率。

3.1.2　非线性度

非线性度的概念最初由 Pieprzyk 等人在 1988 年提出，它是 S 盒的主要设计指标之一，决定了基于 S 盒设计的密

码算法抵抗线性分析的能力。由于 $n\times m$ 的 S 盒都可以表示成 m 个布尔函数，所以下面先给出布尔函数 f 的非线性度定义。

定义 3-4　令 $f:F_2^n\to F_2$ 是一个 n 元布尔函数，$f(x)$的非线性度如下表示：

$$N_f=\min_{l\in L_n} d_{\mathrm{H}}(f,l)$$

其中，L_n 表示全体 n 元线性和仿射函数之集，$d_{\mathrm{H}}(f,l)$为 f 和 l 之间的海明距离。

定理 3-2　n 元布尔函数 f 的非线性度上界为 $2^{n-1}-2^{(n/2)-1}$，当且仅当 f 为 Bent 函数时达到该上界 $2^{n-1}-2^{(n/2)-1}$。

在定理 3-2 中，n 元 Bent 函数是指和每个线性布尔函数的距离都是 $2^{n-1}-2^{(n/2)-1}$ 的 n 元布尔函数。因为 N_f 为正整数，所以 $n/2$ 必为正整数，这说明 n 为偶数。故 Bent 函数存在的必要条件是 n 为偶数。虽然 Bent 函数的非线性度最佳，但它存在一些缺陷。例如，它不是平衡的(即输出的 0、1 个数不相等)，还有它的代数次数不超过 $n/2$，等等。

定义 3-5　令 $S(x)=(f_1(x),f_2(x),\cdots,f_m(x)):F_2^n\to F_2^m$ 是一个多输出函数，则 $S(x)$的非线性度表示为

$$N_s=\min_{\substack{l(x)\in L_n\\ v\neq 0\in \mathrm{GF}(2)^m}} d_{\mathrm{H}}(v\cdot S(x),l(x))$$

S 盒的非线性度指标可以由线性分布表表示，二者本质相同。非线性度考察满足 $v\cdot S(x)\oplus u\cdot x=1$ 或 $v\cdot S(x)\oplus u\cdot x=0$ 的 x 个数的最小值，线性分布表考察满足 $v\cdot S(x)\oplus u\cdot x=0$ 的 x 的个数，即类似于差分分布表，对于输入 u 和 v 的每组值，遍历 S 盒输入 x，计算线性分布表中的每个值：

$$n_{uv}=|\{x\in \mathrm{GF}(2)^n \mid v\cdot S(x)\oplus u\cdot x=0\}|$$

其中，$u=0,1,\cdots,2^n-1$，$v=0,1,\cdots,2^m-1$，点乘(·)表示两个向量的内积运算，例如：

$$a\cdot b=[a_{n-1}a_{n-2}\cdots a_0]\cdot[b_{n-1}b_{n-2}\cdots b_0]^{\mathrm{T}}=\bigoplus_{0\leqslant i\leqslant n-1} a_ib_i$$

将以上得到的每个值填入一个 $n\times m$ 矩阵，组成 S 盒的线性分布表，即

$$\boldsymbol{L}(S)=\begin{pmatrix} n_{00} & n_{01} & \cdots & n_{0(2^m-1)}\\ n_{10} & n_{11} & \cdots & n_{1(2^m-1)}\\ \vdots & \vdots & \ddots & \vdots\\ n_{(2^n-1)0} & n_{(2^n-1)1} & \cdots & n_{(2^n-1)(2^m-1)}\end{pmatrix}$$

例 3-1 中 S 盒的线性分布表如表 3-3 所示。

表 3-3　例 3-1 中 S 盒的线性分布表

$v\backslash u$	000	001	010	011	100	101	110	111
000	8	4	4	4	4	4	4	4
001	4	6	2	4	6	4	4	6
010	4	6	4	2	2	4	2	4
011	4	4	2	2	4	4	6	2
100	4	2	2	4	4	6	2	4

续表

$v\backslash u$	000	001	010	011	100	101	110	111
101	4	4	4	4	6	2	2	2
110	4	4	6	2	6	6	4	4
111	4	6	4	6	4	6	4	2

表 3-3 中输入掩码为 001 时，输出掩码为 001 的概率为 $p(001\rightarrow001)=\frac{2}{8}=\frac{1}{4}$，以此类推。根据定义 3-5 容易得到上述 S 盒的非线性度为 2。

在随机情形下，输入掩码 $u\in F_2^n$ 对应输出掩码 $v\in F_2^m$ 的概率 p 为 $\frac{1}{2}$。用线性逼近能更好地刻画等式 $v\cdot S(x)=u\cdot x$ 成立的有效性，即 $\left|p-\frac{1}{2}\right|$，通常称为线性逼近优势，$v\cdot S(x)=u\cdot x$ 成立的概率 p 满足 $\left|p-\frac{1}{2}\right|\leqslant\frac{1}{2}-\frac{N_S}{2^n}$。

1993 年，Matsui 提出了适用于分组密码的线性分析方法，该攻击手段本质上基于 S 盒的低非线性度。给定一个 $n\times m$ 的 S 盒，线性分析者计算以下布尔函数值为 1 的自变量 x 的个数：

$$f(x_1,x_2,\cdots,x_n)=(a_0\oplus(\bigoplus_{i=1}^{n}(a_ix_i)))\oplus(\bigoplus_{j=1}^{m}(b_jf_j(x_1,x_2,\cdots,x_n)))$$

其中 $a_0\in F_2^n$，且 $(a_1,a_2,\cdots,a_n)\in F_2^n$，$(b_1,b_2,\cdots,b_m)\in F_2^m$ 的分量均不全为 0，即计算 S 盒所有分量函数的任意非零线性组合的非线性度，使之不偏离 2^{n-1} 太远。

由此可见，为抵抗线性分析，S 盒的非线性度越大越好。利用布尔函数非线性的有关结论可得，$4\times m$ 的 S 盒和 $8\times m$ 的 S 盒的最佳优势分别为 2^{-2} 和 2^{-4}。

3.1.3 代数次数和项数

S 盒的代数次数用于衡量 S 盒的代数非线性程度，代数次数的大小一定程度上反映了 S 盒的线性复杂度，S 盒线性复杂度越高，越难用线性表达式逼近。而项数分布的程度和插值攻击密切相关。

定义 3-6 设 n 元布尔函数 $f:F_2^n\rightarrow F_2$ 的代数正规型为

$$f(x)=a_0\oplus\sum_{\substack{1\leqslant i_1<\cdots<i_k\leqslant n\\1\leqslant k\leqslant n}}a_{i_1i_2\cdots i_k}x_{i_1}x_{i_2}\cdots x_{i_k}$$

其中 $x=(x_1,x_2,\cdots,x_n)$，$a_0,a_{i_1i_2\cdots i_k}\in F_2$。$f(x)$的代数次数 $D(f)$为

$$D(f)=\max\{0\leqslant k\leqslant n\mid a_{i_1i_2\cdots i_k}=1,1\leqslant i_1<\cdots<i_k\leqslant n\}$$

$f(x)$的代数正规型中的 i 次项的个数称为 $f(x)$的 i 次项数，所有 $i(1\leqslant i\leqslant n)$次项数之和称为 $f(x)$ 的项数，即代数项数 $N(f)$为

$$N(f)=|\{a_0=1,a_{i_1i_2\cdots i_k}=1,1\leqslant i_1<\cdots<i_k\leqslant n,1\leqslant k\leqslant n\}|$$

由于 S 盒由若干布尔函数表示，所以关于 S 盒的代数次数作出如下描述。

定义 3-7　设 $n\times m$ 的 S 盒

$$S(x)=(f_1(x),f_2(x),\cdots,f_m(x)):F_2^n\to F_2^m$$

其代数次数定义为

$$D(S)=\min\{D(v\cdot S)\mid v\neq 0,v\in F_2^m\}$$

$$=\min\left\{D\left(\bigoplus_{i=1}^{m}(v_if_i(x))\right)\,\middle|\,v=(v_1,v_2,\cdots,v_m)\neq 0\right\}$$

其中 $D(S)$表示 S 盒的代数次数。特别地，当 $D(S)=k$ 时，称 S 为 k 次 S 盒。

与代数次数定义类似，取项数最小的 $f(x)=v\cdot S(x)$的项数为 S 盒的项数，即

$$N(S)=\min\{N(v\cdot S)\mid v\neq 0,v\in F_2^m\}$$

$$=\min\left\{N\left(\bigoplus_{i=1}^{m}(v_if_i(x))\right)\,\middle|\,v=(v_1,v_2,\cdots,v_m)\neq 0\right\}$$

显然，对于非线性度最优的 $n\times m$ 的 S 盒，其代数次数可能达到的最大值为 $n-1$，而且具有 $D(S)=n-1$ 的 S 盒是存在的。

例 3-2　对于如表 3-1 所示的 S 盒，假设输入为(x_2,x_1,x_0)，输出为(y_2,y_1,y_0)，对应的真值表如表 3-4 所示。对于输出比特 y_0，容易列出其布尔表达式，进一步可写成代数正规型：

$$\begin{aligned}y_0&=f_0(x)\\&=\overline{x_2}\,\overline{x_1}x_0\oplus\overline{x_2}x_1\,\overline{x_0}\oplus\overline{x_2}\,x_1x_0\oplus x_2\,x_1x_0\\&=x_0\oplus x_1\oplus x_1x_0\oplus x_2x_0\oplus x_2x_1\end{aligned}$$

表 3-4　真值表

x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	1	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	1	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	1

同理可得 y_1、y_2 的代数正规型：

$$y_1=1\oplus x_0\oplus x_1x_0\oplus x_2x_1$$

$$y_2=x_0\oplus x_2\oplus x_2x_0\oplus x_2x_1$$

容易推出，将这 3 个代数正规型线性组合之后，代数次数最小仍为 2，所以此 S 盒代数次数为 2。

对于一个好的 S 盒，每个分量布尔函数的代数次数最佳为 $n-1$。若存在分量布尔函数的代数次数太小，则安全性方面相应的积分性质较差，容易受积分分析的攻击。同样，若 $D(S)$很小，且具有 $D(S)$ 次项的各组合布尔函数的项数也很少，则可用较低次的布尔

函数进行快速逼近，因此密码性能安全的S盒要有一定的代数次数和项数。

例 3-3 由例 3-2 可知，该S盒具有代数正规型

$$\begin{cases} y_0 = x_0 \oplus x_1 \oplus x_1x_0 \oplus x_2x_0 \oplus x_2x_1 \\ y_1 = 1 \oplus x_0 \oplus x_1x_0 \oplus x_2x_1 \\ y_2 = x_0 \oplus x_2 \oplus x_2x_0 \oplus x_2x_1 \end{cases}$$

选择S盒的输入 X 满足集合 $V=\{000,001,010,011\}$，对S盒的3个分量布尔函数进行积分，得

$$\int_{X\in V} y_0 = \int_{X\in V} (x_0 \oplus x_1 \oplus x_1x_0 \oplus x_2x_0 \oplus x_2x_1) = 1$$

$$\int_{X\in V} y_1 = \int_{X\in V} (1 \oplus x_0 \oplus x_1x_0 \oplus x_2x_1) = 1$$

$$\int_{X\in V} y_2 = \int_{X\in V} (x_0 \oplus x_2 \oplus x_2x_0 \oplus x_2x_1) = 0$$

即当S盒的输入 X 取遍 V 中所有值时，S盒输出比特中最高位之和为0，其余两位之和为1，表示为 $\bigoplus_{X\in V} S(X)=011$。这种性质可以称为S盒在集合 V 上的积分性质。

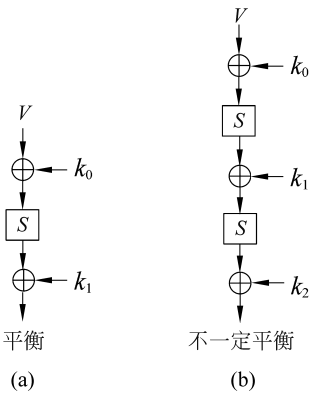


图 3-2 迭代结构的积分性质

基于上述S盒，对图 3-2 的两种结构作积分分析比较：

- 对于图 3-2(a)中的结构，当明文 P 取遍 V 中的值时，输出密文 C 的最高比特之和为0，又称为平衡比特。
- 对于图 3-2(b)中的结构，当明文 P 取遍 V 中的值时，输出密文 C 的任意比特之和可能都不确定。

特殊地，因为此处S盒是一个置换，所以当输入 X 取遍 2^3 个值时，输出值的和必为0。

假设S盒的输入为 n 比特，对于图 3-2(a)的结构，如果S盒的代数次数为 t，则存在明文集合 V 取合适的 2^t 个值，对应的输出存在平衡比特；对于图 3-2(b)的结构，两个S盒迭代后的代数次数可能为 t^2，呈指数级增长，若明文集合 V 仍然取 2^t 个值，则对应的输出不一定有平衡比特。因此，多轮迭代会增加代数次数。

对于一个理想的S盒，除了代数次数之外，每个分量布尔函数的 i 次项数应该接近于 $\binom{n}{i}\Big/2$，若项数太少，有可能提高插值攻击的成功率。

3.1.4 代数免疫度

代数免疫度是针对代数攻击而提出的，与非线性度有一定的关系。如果一个布尔函数 f 的非线性度比较低，那么它的代数免疫度就比较低；如果它的代数免疫度高，它的非

线性度将不会低，然而并不保证非线性度很高。

定义 3-8　设 $f: F_2^n \to F_2$ 是一个 n 元布尔函数，则 $f(x)$的代数免疫度 $\mathrm{AI}_n(f)$是使得 $fg=0$ 或$(f \oplus 1)g=0$ 成立的非零布尔函数 g 的最小代数次数。

定义 3-9　设 $n \times m$ 的 S 盒

$$S(x)=(f_1(x),f_2(x),\cdots,f_m(x)): F_2^n \to F_2^m$$

则 S 盒代数免疫度定义为

$$\begin{aligned}\mathrm{AI}(S)&=\min\{\mathrm{AI}(v \cdot S) \mid v \neq 0, v \in F_2^m\}\\&=\min\left\{\mathrm{AI}\left(\bigoplus_{i=1}^{m}(v_i f_i(x))\right) \middle| v=(v_1,v_2,\cdots,v_m) \neq 0\right\}\end{aligned}$$

例 3-4　设 $f: F_2^3 \to F_2$，$f(x)=x_2x_3 \oplus x_1x_2x_3$，当 $g(x)=x_1$ 时，显然有 $fg=0$，所以布尔函数 f 的代数免疫度为 1。

定理 3-3　$f(x)$是 n 元布尔函数，则 $\mathrm{AI}(f) \leqslant \left\lceil \frac{n}{2} \right\rceil$（$\left\lceil \frac{n}{2} \right\rceil$表示大于$\frac{n}{2}$的最小整数）。

由定理 3-3 可以得到代数免疫度的计算方法。如果存在一个布尔函数 $g(x)$，$g(x) \neq 0$，有 $g(x)f(x)=0$ 或 $g(x)(1 \oplus f(x))=0$，且 $\deg(g)<d$（其中 $\deg(g)$表示 $g(x)$的次数），则有 $\mathrm{AI}(f)<d$；如果布尔函数 $f(x)$不存在次数小于 d 的零化多项式，那么 $f(x)$的代数免疫度为 $\mathrm{AI}(f) \geqslant d$。求解 $\mathrm{AI}(f)$有以下几个步骤：

第一步，令 $d=\left\lceil \frac{n}{2} \right\rceil$，写出次数小于 d 的 n 元布尔函数表达式：

$$\begin{aligned}g(x)=&a_0 \oplus a_1x_1 \oplus \cdots a_nx_n \oplus a_{1,2}x_1x_2 \oplus \cdots \oplus a_{n-1,n}x_{n-1}x_n\\&\oplus \cdots \oplus a_{1,2,\cdots,(d-1)}x_1x_2\cdots x_{d-1}\\&\oplus \cdots \oplus a_{(n-d+2),\cdots,(n-1),n}x_{(n-d+2)}\cdots x_{n-1}x_n\end{aligned}$$

$g(x)$ 的系数个数为 $\sum_{i=0}^{d-1}\binom{n}{i}$。

第二步，对于满足 $f(x)=1$ 的所有 x，有 $g(x)=0$。将 x 代入 $g(x)=0$ 中，得到一个线性方程组，方程的个数为 $w(f)$，未知量个数为 $\sum_{i=0}^{d-1}\binom{n}{i}$。

第三步，判断是否存在次数少于 d 的非零的零化多项式 $g(x)$，归结为求上述齐次线性方程组是否存在非零解。如果上述齐次线性方程组有非零解，则找到布尔函数 $f(x)$的一个次数小于 d 的零化多项式。令 $d=\left\lceil \frac{n}{2} \right\rceil-1$，重复第一步至第三步。如果上述齐次线性方程组只有零解，则 $f(x)$没有次数小于 d 的零化多项式。

第四步，对于满足 $f(x) \oplus 1=1$ 的所有 x，重复第二步和第三步。

第五步，输出 $\mathrm{AI}(f)=d$。

例 3-5　设 $f(x_1,x_2,x_3)=x_1 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3$，则 $d=\left\lceil \frac{3}{2} \right\rceil=2$，即 $g(x)$的代数次数小于 2，其真值表如表 3-5 所示。

表 3-5　f 函数真值表

x_3	x_2	x_1	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

设 $g(x)=a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus a_3x_3$，将使得 $f(x)=1$ 的值 001、011、100、101 分别代入 $g(x)=0$，得

$$\begin{cases} a_0 \oplus a_1 = 0 \\ a_0 \oplus a_1 \oplus a_2 = 0 \\ a_0 \oplus a_3 = 0 \\ a_0 \oplus a_1 \oplus a_3 = 0 \end{cases}$$

以上方程组只有零解，继续将 $f(x) \oplus 1=1$ 的值 000、010、011、111 代入 $g(x)=0$，得

$$\begin{cases} a_0 = 0 \\ a_0 \oplus a_2 = 0 \\ a_0 \oplus a_2 \oplus a_3 = 0 \\ a_0 \oplus a_1 \oplus a_2 \oplus a_3 = 0 \end{cases}$$

以上方程组也只有零解，故 $f(x)$ 的代数免疫度为 2。

对于输入为 n 比特的布尔函数 $f(x)$，代数免疫度最佳为 $\left\lceil \frac{n}{2} \right\rceil$。若对于 $0 \neq v \in F_2^m$，都有 $\mathrm{AI}(v \cdot S)=\left\lceil \frac{n}{2} \right\rceil$，则该 S 盒的代数免疫度达到最优。

3.1.5　扩散特性和严格雪崩特性

扩散特性和严格雪崩特性用于衡量 S 盒输出改变量对于输入改变量的随机性，也是 S 盒设计的重要指标。

定义 3-10　设 $f: F_2^n \to F_2$ 是一个 n 元布尔函数。如果 $f(x \oplus \boldsymbol{\alpha}) \oplus f(x)$ 是一个平衡函数，称 $f(x)$ 关于非零向量 $\boldsymbol{\alpha} \in F_2^n$ 满足扩散准则。如果对所有的向量 $\boldsymbol{\alpha} \in F_2^n$：$1 \leqslant W_{\mathrm{H}}(\boldsymbol{\alpha}) < k$，$f(x)$ 满足扩散准则，称 $f(x)$ 满足 k 次扩散准则。

上述定义中 $W_{\mathrm{H}}(\alpha)$ 指 α 的重量，即 α 的二进制表中 1 的个数。

定义 3-11　设 $n \times m$ 的 S 盒 $S(x)=(f_1(x), f_2(x), \cdots, f_m(x)): F_2^n \to F_2^m$，如果各分量函数 $f_i(x)$ 关于 $\boldsymbol{\alpha}$ 满足扩散准则，就称 S 盒关于元素 $\boldsymbol{\alpha} \in \mathrm{GF}(2)^n$ 满足扩散准则。进

一步，如果 S 盒关于所有 $\boldsymbol{\alpha} \in \mathrm{GF}(2)^n$，$1 \leqslant W_{\mathrm{H}}(\alpha) < k$ 均满足扩散准则，则称 S 盒满足 k 次扩散准则。

特别地，若 $\sum_x (S(x) \oplus S(x \oplus e)) = \left(\sum_x (f_1(x) \oplus f_1(x \oplus e)), \cdots, \sum_x (f_m(x) \oplus f_m(x \oplus e)) \right) = (2^{n-1}, \cdots, 2^{n-1})$ 对任意 $e \in \mathrm{GF}(2)^n$，$W_{\mathrm{H}}(e) = 1$ 均满足扩散准则，则称 S 盒满足严格雪崩准则(Strict Avalanche Criterion，SAC)。

例 3-6　函数 $f(x_1, x_2, x_3) = x_1 x_2 \oplus x_3$ 不满足严格雪崩准则。

因为 $f(x_1 \oplus 1, x_2, x_3) \oplus f(x_1, x_2, x_3) = x_2$ 和 $f(x_1, x_2 \oplus 1, x_3) \oplus f(x_1, x_2, x_3) = x_1$ 是平衡的，但 $f(x_1, x_2, x_3 \oplus 1) \oplus f(x_1, x_2, x_3) = 1$ 不是平衡的，所以函数 f 不满足严格雪崩准则。

例 3-7　函数 $f(x_1, x_2, x_3) = x_1 x_2 \oplus x_2 x_3 \oplus x_1 x_3$ 满足严格雪崩准则。

因为 $f(x_1 \oplus 1, x_2, x_3) \oplus f(x_1, x_2, x_3) = x_2 \oplus x_3$、$f(x_1, x_2 \oplus 1, x_3) \oplus f(x_1, x_2, x_3) = x_1 \oplus x_3$ 和 $f(x_1, x_2, x_3 \oplus 1) \oplus f(x_1, x_2, x_3) = x_2 \oplus x_1$ 都是平衡布尔函数，所以函数 f 满足严格雪崩准则。

定义 3-12　设 $n \times m$ 的 S 盒 $S(x) = (f_1(x), f_2(x), \cdots, f_m(x)) : F_2^n \to F_2^m$，S 盒满足雪崩准则是指改变 S 盒输入的 1 比特，大约有一半输出比特改变。

定义 3-13　设 $n \times m$ 的 S 盒 $S(x) = (f_1(x), f_2(x), \cdots, f_m(x)) : F_2^n \to F_2^m$，S 盒满足严格雪崩准则是指改变输入的 1 比特，每个输出比特改变的概率为 $\frac{1}{2}$。

此定义是对严格雪崩准则的另一种解释，是定义 3-11 的特殊情形。

例 3-8　如图 3-3 所示，哈希函数输出摘要值为 160 比特，输入改变 1 比特时，摘要值随之改变了大约一半。当输入 000 变为 001 时，摘要值改变了 79 比特；当 000 变为 010 时，摘要值改变了 79 比特；当 001 变为 010 时，摘要值改变了 85 比特。

这个哈希函数展示了良好的雪崩效应。

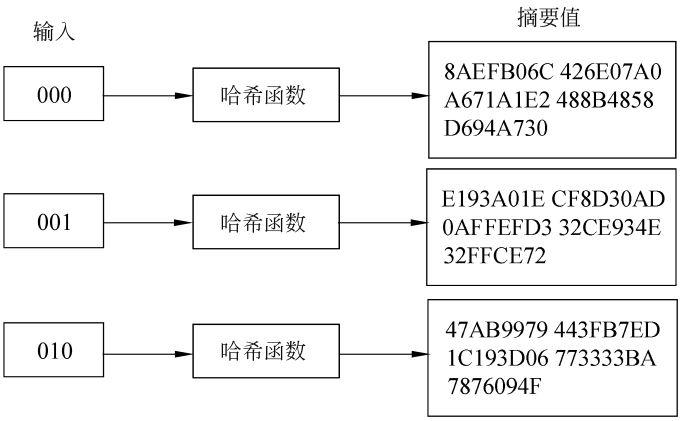


图 3-3　哈希函数的雪崩效应

除了差分均匀度、非线性度、代数次数和项数、代数免疫度、扩散特性与严格雪崩特性这几个主要的密码指标之外，设计 S 盒时还要注意能否满足可逆性、消除不动点等。值得注意的是，S 盒的某些密码指标是一致的，如差分均匀度与非线性度，某些指标有一定的

制约关系，如代数次数与扩散特性，具体设计时需要折中考虑。

3.2 S盒的设计方法

基于3.1节所述S盒的几项主要密码指标，人们提出了许多S盒设计方法。一般地，可以将S盒的设计方法分成三大类：第一类是随机生成，便于密码系统中动态S盒替换；第二类是基于数学函数设计，容易达到密码安全指标；第三类是基于已有组件设计，便于软硬件实现。在轻量级算法设计中由于资源受限，S盒设计倾向于更少的逻辑电路门数和深度，设计这一类S盒时，在保证密码指标的前提下可以通过启发式算法等方法进行逻辑电路门数和深度搜索。

3.2.1 随机生成

利用随机生成方法设计的S盒可以使人们相信没有陷门。其基本步骤是：随机生成一批S盒，再对它们进行测试，选出指标性能较好的进行应用。尽管这是一种完全随机的S盒设计方法，但是用这种方法寻找好的S盒需要花费很多时间，并且要在设计者计算能力允许的条件下进行。

例3-9 DES的S盒一般被认为是随机设计的，1976年美国NSA披露了该S盒的设计原则：

(1) 每个S盒的每一行是整数0～15的一个全排列。

(2) 每个S盒的输出都不是其输入的线性函数或仿射函数。

(3) 改变S盒任意1比特的输入，其输出至少有2比特发生变化。

(4) 对S盒的任意6比特输入x，$S(x)$与$S(x\oplus 001100)$至少有2比特不同。

(5) 对S盒的任意6比特输入x，且$\alpha,\beta\in\{0,1\}$，有$S(x)\neq S(x\oplus 11\alpha\beta 00)$。

(6) 保持S盒的任一位输入不变，其他5位输入任意变化时，所有4比特输出中0与1的总数接近相等。

根据以上原则，设计者给出了8个S盒，以S_1为例，如表3-6所示。S_1包含4行，每一行都是0～15的一个全排列。设输入$B_j=b_1b_2b_3b_4b_5b_6$，将b_1b_6和$b_2b_3b_4b_5$作为二进制数，若b_1b_6和$b_2b_3b_4b_5$对应的十进制数分别为r和$c(0\leqslant r\leqslant 3, 0\leqslant c\leqslant 15)$，则$S_1$中的第$r$行、第$c$列处的十进制数对应的二进制形式就是$S_1$的输出。例如，输入110100，首尾2比特“10”对应2所指示的行，中间4比特“1010”对应10指示的列，S_1输出为2行10列交叉处的数字9，对应二进制形式为1001。

表3-6 DES的S_1

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

由于 S_1 输入 6 比特，输出 4 比特，所以平均每 4 个不同输入对应一个相同的输出，差分均匀度比较大。表 3-7 是 S_1 的差分分布表。

表 3-7　S_1 的差分分布表

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	6	0	2	4	4	0	10	12	4	10	6	2	4
2	0	0	0	8	0	4	4	4	0	6	8	6	12	6	4	2
3	14	4	2	2	10	6	4	2	6	4	4	0	2	2	2	0
4	0	0	0	6	0	10	10	6	0	4	6	4	2	8	6	2
5	4	8	6	2	2	4	4	2	0	4	4	0	12	2	4	6
6	0	4	2	4	8	2	6	2	8	4	4	2	4	2	0	12
7	2	4	10	4	0	4	8	4	2	4	8	2	2	2	4	4
8	0	0	0	12	0	8	8	4	0	6	2	8	8	2	2	4
9	10	2	4	0	2	4	6	0	2	2	8	0	10	0	2	12
10	0	8	6	2	2	8	6	0	6	4	6	0	4	0	2	10
11	2	4	0	10	2	2	4	0	2	6	2	6	6	4	2	12
12	0	0	0	8	0	6	6	0	0	6	6	4	6	6	14	2
13	6	6	4	8	4	8	2	6	0	6	4	6	0	2	0	2
14	0	4	8	8	6	6	4	0	6	6	4	0	0	4	0	8
15	2	0	2	4	4	6	4	2	4	8	2	2	2	6	8	8
16	0	0	0	0	0	0	2	14	0	6	6	12	4	6	8	6
17	6	8	2	4	6	4	8	6	4	0	6	6	0	4	0	0
18	0	8	4	2	6	6	4	6	6	4	2	6	6	0	4	0
19	2	4	4	6	2	0	4	6	2	0	6	8	4	6	4	6
20	0	8	8	0	10	0	4	2	8	2	2	4	4	8	4	0
21	0	4	6	4	2	2	4	10	6	2	0	10	0	4	6	4
22	0	8	10	8	0	2	2	6	10	2	0	2	0	6	2	6
23	4	4	6	0	10	6	0	2	4	4	4	6	6	6	2	0
24	0	6	6	0	8	4	2	2	2	4	6	8	6	6	2	2
25	2	6	2	4	0	8	4	6	10	4	0	4	2	8	4	0
26	0	6	4	0	4	6	6	6	6	2	2	0	4	4	6	8
27	4	4	2	4	10	6	6	4	6	2	2	4	2	2	4	2

续表

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
28	0	10	10	6	6	0	0	12	6	4	0	0	2	4	4	0
29	4	2	4	0	8	0	0	2	10	0	2	6	6	6	14	0
30	0	2	6	0	14	2	0	0	6	4	10	8	2	2	6	2
31	2	4	10	6	2	2	2	8	6	8	0	0	0	4	6	4
32	0	0	0	10	0	12	8	2	0	6	4	4	4	2	0	12
33	0	4	2	4	4	8	10	0	4	4	10	0	4	0	2	8
34	10	4	6	2	2	8	2	2	2	2	6	0	4	0	4	10
35	0	4	4	8	0	2	6	0	6	6	2	10	2	4	0	10
36	12	0	0	2	2	2	2	0	14	14	2	0	2	6	2	4
37	6	4	4	12	4	4	4	10	2	2	2	0	4	2	2	2
38	0	0	4	10	10	10	2	4	0	4	6	4	4	4	2	0
39	10	4	2	0	2	4	2	0	4	8	0	4	8	8	4	4
40	12	2	2	8	2	6	12	0	0	2	6	0	4	0	6	2
41	4	2	2	10	0	2	4	0	0	14	10	2	4	6	0	4
42	4	2	4	6	0	2	8	2	2	14	2	6	2	6	2	2
43	12	2	2	2	4	6	6	2	0	2	6	2	6	0	8	4
44	4	2	2	4	0	2	10	4	2	2	4	8	8	4	2	6
45	6	2	6	2	8	4	4	4	2	4	6	0	8	2	0	6
46	6	6	2	2	0	2	4	6	4	0	6	2	12	2	6	4
47	2	2	2	2	2	6	8	8	2	4	4	6	8	2	4	2
48	0	4	6	0	12	6	2	2	8	2	4	4	6	2	2	4
49	4	8	2	10	2	2	2	2	6	0	0	2	2	4	10	8
50	4	2	6	4	4	2	2	4	6	6	4	8	2	2	8	0
51	4	4	6	2	10	8	4	2	4	0	2	2	4	6	2	4
52	0	8	16	6	2	0	0	12	6	0	0	0	0	8	0	6
53	2	2	4	0	8	0	0	0	14	4	6	8	0	2	14	0
54	2	6	2	2	8	0	2	2	4	2	6	8	6	4	10	0
55	2	2	12	4	2	4	4	10	4	4	2	6	0	2	2	4
56	0	6	2	2	2	0	2	2	4	6	4	4	4	6	10	10
57	6	2	2	4	12	6	4	8	4	0	2	4	2	4	4	0

续表

S_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
58	6	4	6	4	6	8	0	6	2	2	6	2	2	6	4	0
59	2	6	4	0	0	2	4	6	4	6	8	6	4	4	6	2
60	0	10	4	0	12	0	4	2	6	0	4	12	4	4	2	0
61	0	8	6	2	2	6	0	8	4	4	0	4	0	12	4	4
62	4	8	2	2	2	4	4	14	4	2	0	2	0	8	4	4
63	4	8	4	2	4	0	2	4	4	2	4	8	8	6	2	2

显然，S_1 的差分均匀度为 16，根据 3.1 节其他密码指标的定义，可测得非线性度为 18，4 个分量布尔函数的代数次数都为 5，代数项数为[27，33，38，29]，代数免疫度为[4，5，4，5]。

3.2.2　基于数学函数设计

在大多数情况下，基于数学函数设计可以保证 S 盒有较好的密码性能。常用的设计方法有下面几种。

1. 对数函数和指数函数

对数函数和指数函数互为反函数，基于有限域上的这两类函数可以构造 S 盒以及 S 盒的逆，例如 SAFER 系列分组密码使用的 S 盒[9]。

例 3-10　SAFER 系列分组密码的 S 盒设计采用了两个函数，分别是对数函数 $y=\log_a x$ 和指数函数 $y=a^x$，具体如下：

- 对数函数为$\log_{45} x$。当 $x=0$ 时，约定$\log_{45} 0=128$。
- 指数函数为$45^x \bmod 257$。当 $x=128$ 时，约定$45^{128} \bmod 257=0$。

它们的差分均匀度在传统的异或运算下并不好，但是 SAFER 系列分组密码的轮密钥嵌入使用了模 256 加，在差分定义为模 256 减时，它们的差分均匀性还可以接受。指数函数的分量布尔函数的代数次数为[7，7，7，7，7，7，6，6]，代数项数为[160，138，121，130，119，108，46，56]；对数函数的分量布尔函数代数次数都为 7，代数项数为[122，126，113，123，138，110，104，110]。

2. 有限域上的逆映射

许多著名的密码算法采用伽罗瓦域上的逆映射作为 S 盒。例如，SHARK、Square、AES 等密码算法的 S 盒设计都采用了这种方法。

例 3-11　SHARK 的 S 盒基于 GF(2^m)上的映射 $F(x)=x^{-1}$ 设计。差分均匀度为 4，代数次数等于 $m-1$。然而，由于该 S 盒在 GF(2^n)中的表达式太简单，导致了 SHARK 密码对插值攻击的脆弱性。

基于不同代数结构上数学函数的复合运算进行设计可以消除抗插值攻击的脆弱性，例如 AES 分组密码的 S 盒设计。

例 3-12 AES 的 S 盒由以下两个有限域上的变换合成而得。在 GF(2^8)中定义不可约多项式 $m(x)=x^8+x^4+x^3+x+1$。

(1) 对于 $x\neq 00$,在 GF(2^8)上计算 $x\rightarrow x^{-1}$,定义 $00\rightarrow 00$。

(2) 对得到的逆字节进行 GF(2)的仿射变换:

$$\begin{bmatrix} s_7 \\ s_6 \\ s_5 \\ s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_7 \\ y_6 \\ y_5 \\ y_4 \\ y_3 \\ y_2 \\ y_1 \\ y_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

按上述方法构造的 AES 的 S 盒如表 3-8 所示。这种 S 盒的差分均匀度为 4,非线性度为 112,都达到了最优。分量布尔函数的代数次数为 [7,7,7,7,7,7,7,7],代数项数为 [110,112,114,131,136,145,133,132],代数免疫度为[4,4,4,4,4,4,4,4],也都达到了最优。

表 3-8 S 盒的列表描述

S	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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	a5	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	5a	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	b0	54	bb	16

3. 有限域上的幂函数

利用有限域上的特殊函数设计 S 盒引起了许多研究者的关注，基于幂函数 $y=x^{a}$ 的设计应运而生，例如 MISTY 算法中使用的 S 盒[76]。

例 3-13　MISTY 算法一共有两个 S 盒，分别是 7 比特输入输出的 S_7 和 9 比特输入输出的 S_9。设计 S_7 和 S_9 的三大标准如下：

(1) 平均差分/线性概率足够小。

(2) 硬件实现延迟小。

(3) 代数次数尽可能高。

S 盒的设计原则主要为利用 GF(2^7)和 GF(2^9)上的幂函数并进行仿射变换 $A\circ x^{\alpha}\circ B$，α 要满足$(2^i-1,\alpha)=1$，遍历线性变换 A 和部分 B，不影响差分均匀度和线性概率，但是会影响代数次数和硬件实现效率(延迟)。值得注意的是，代数次数与 α 的海明重量一致。

遍历 $S_i(x)=A\circ x^{\alpha}(i=7,9)$的线性变换 A。对于 GF(2^7)上的 S_7，通过测试得到每个布尔函数：

- 代数次数为 4 时，输出比特的逻辑电路深度至少为 21。
- 代数次数为 3 时，输出比特的逻辑电路深度至少为 10。
- 代数次数为 2 时，输出比特的逻辑电路深度至少为 1。

经折中考虑，选择代数次数为 3、代数项数为 13 的 S_7。以下是其输出比特的 7 个布尔表达式：

$$y_0=x_0\oplus x_1x_3\oplus x_0x_3x_4\oplus x_1x_5\oplus x_0x_2x_5\oplus x_4x_5\oplus x_0x_1x_6\oplus x_2x_6\oplus x_0x_5x_6\oplus x_3x_5x_6\oplus 1$$

$$y_1=x_0x_2\oplus x_0x_4\oplus x_3x_4\oplus x_1x_5\oplus x_2x_4x_5\oplus x_6\oplus x_0x_6\oplus x_3x_6\oplus x_2x_3x_6\oplus x_1x_4x_6\oplus x_0x_5x_6\oplus 1$$

$$y_2=x_1x_2\oplus x_0x_2x_3\oplus x_4\oplus x_1x_4\oplus x_0x_1x_4\oplus x_0x_5\oplus x_0x_4x_5\oplus x_3x_4x_5\oplus x_1x_6\oplus x_3x_6\oplus x_0x_3x_6\oplus x_4x_6\oplus x_2x_4x_6$$

$$y_3=x_0\oplus x_1\oplus x_0x_1x_2\oplus x_0x_3\oplus x_2x_4\oplus x_1x_4x_5\oplus x_2x_6\oplus x_1x_3x_6\oplus x_0x_4x_6\oplus x_5x_6\oplus 1$$

$$y_4=x_dx_3\oplus x_0x_4\oplus x_1x_3x_4\oplus x_5\oplus x_2x_5\oplus x_1x_2x_5\oplus x_0x_3x_5\oplus x_1x_6\oplus x_1x_5x_6\oplus x_4x_5x_6\oplus 1$$

$$y_5=x_0\oplus x_1\oplus x_2\oplus x_0x_1x_2\oplus x_0x_3\oplus x_1x_2x_3\oplus x_1x_4\oplus x_0x_2x_4\oplus x_0x_5\oplus x_0x_1x_5\oplus x_3x_5\oplus x_0x_6\oplus x_2x_5x_6$$

$$y_6=x_0x_1\oplus x_3\oplus x_0x_3\oplus x_2x_3x_4\oplus x_0x_5\oplus x_2x_5\oplus x_3x_5\oplus x_1x_3x_5\oplus x_1x_6\oplus x_1x_2x_6\oplus x_0x_3x_6\oplus x_4x_6\oplus x_2x_5x_6$$

S_9 为 GF(2^9)上的线性变换 $A\circ x^{\alpha}$，选择代数次数为 2，最小硬件实现逻辑电路深度为 12。以下是 S_9 的 9 个输出比特表达式：

$$y_0=x_0x_4\oplus x_0x_5\oplus x_1x_5\oplus x_1x_6\oplus x_2x_6\oplus x_2x_7\oplus x_3x_7\oplus x_3x_8\oplus x_4x_8\oplus 1$$

$$y_1 = x_0x_2 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_4x_5 \oplus x_1x_5 \oplus x_0x_6 \oplus x_2x_6 \oplus x_7 \oplus x_0x_8 \oplus x_3x_8 \oplus x_5x_8 \oplus 1$$

$$y_2 = x_0x_1 \oplus x_1x_3 \oplus x_4 \oplus x_0x_4 \oplus x_2x_4 \oplus x_3x_4 \oplus x_4x_5 \oplus x_0x_6 \oplus x_5x_6 \oplus x_1x_7 \oplus x_3x_7 \oplus x_8$$

$$y_3 = x_0 \oplus x_1x_2 \oplus x_2x_4 \oplus x_5 \oplus x_1x_5 \oplus x_3x_5 \oplus x_4x_5 \oplus x_5x_6 \oplus x_1x_7 \oplus x_6x_7 \oplus x_2x_8 \oplus x_4x_8$$

$$y_4 = x_1 \oplus x_0x_3 \oplus x_2x_3 \oplus x_0x_5 \oplus x_3x_5 \oplus x_6 \oplus x_2x_6 \oplus x_4x_6 \oplus x_5x_6 \oplus x_6x_7 \oplus x_2x_8 \oplus x_7x_8$$

$$y_5 = x_2 \oplus x_0x_3 \oplus x_1x_4 \oplus x_3x_4 \oplus x_1x_6 \oplus x_4x_6 \oplus x_7 \oplus x_3x_7 \oplus x_5x_7 \oplus x_6x_7 \oplus x_0x_8 \oplus x_7x_8$$

$$y_6 = x_0x_1 \oplus x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_4x_5 \oplus x_2x_7 \oplus x_5x_7 \oplus x_8 \oplus x_0x_8 \oplus x_4x_8 \oplus x_6x_8 \oplus x_7x_8 \oplus 1$$

$$y_7 = x_1 \oplus x_0x_1 \oplus x_1x_2 \oplus x_2x_3 \oplus x_0x_4 \oplus x_5 \oplus x_1x_6 \oplus x_3x_6 \oplus x_0x_7 \oplus x_4x_7 \oplus x_6x_7 \oplus x_1x_8 \oplus 1$$

$$y_8 = x_0 \oplus x_0x_1 \oplus x_1x_2 \oplus x_4 \oplus x_0x_5 \oplus x_2x_5 \oplus x_3x_6 \oplus x_5x_6 \oplus x_0x_7 \oplus x_0x_8 \oplus x_3x_8 \oplus x_6x_8 \oplus 1$$

上述两个 S 盒的差分均匀度都为 2，属于 APN 置换。然而，由于代数次数太低，采用这两个 S 盒设计的 MISTY1 和 KASUMI 算法都已被理论攻破。因此，基于代数次数太低的 S 盒设计密码算法，若迭代轮数不够则容易受到高阶差分分析或积分分析攻击。目前一般不直接用幂函数作为 S 盒，对它的研究还在继续。

3.2.3 基于已有组件设计

1. 基于已有 S 盒设计

基于一些已知的密码性能良好的 S 盒也可以构造新的 S 盒，例如 Serpent 算法所使用的 S 盒就是基于 DES 的 S 盒构造出来的[77]。通过这种构造方法找到一个各项指标都满足的 S 盒并不容易。小规模 S 盒组合构造的方式有很多，相对容易穷举测试；但是组合生成的大规模 S 盒不一定满足所有设计指标。

例 3-14 Serpent 算法采用 8 个不同的 4 比特 S 盒，需满足以下 3 个准则：

(1) 每个差分特性的最大概率为$\frac{1}{4}$，并且只有 1 比特非零的输入差分将永远不会导致只有 1 比特非零的输出差分。

(2) 每个线性逼近的概率在$\frac{1}{2}\pm\frac{1}{4}$范围内，并且只有 1 比特非零的输入掩码与只有 1 比特非零的输出掩码之间的线性关系概率在$\frac{1}{2}\pm\frac{1}{8}$范围内。

(3) 每个分量布尔函数的非线性次数最大，即达到 3。

S 盒的设计原理借鉴了 RC4 的模式。首先设两个数组，分别是 Serpent[.]和 sbox[32][16]。Serpent[.]包含“sboxesforserpent”这 16 个字符的 ASCII 码低 4 位值；sbox

[32][16]是 32×16 的二维数组，用 8 个 DES 的 S 盒(代码中记为 sbox)对其进行初始化。Swapentries(·,·)的作用是交换值，具体交换如算法 3-1 所示。如果生成的数组满足所需的差分和线性指标，保存该数组并将其作为 S 盒。重复该过程，直到生成 8 个 S 盒。

算法 3-1　Serpent 的 8 个 S 盒生成

```
index := 0
repeat
  currentsbox := index modulo 32;
  for i := 0 to 15 do
    j := sbox[(currentsbox+1) modulo 32][serpent[i]];
    swapentries (sbox[currentsbox][i],sbox[currentsbox][j]);
  if sbox[currentsbox][.] has the desired properties, save it;
  index := index +1;
until 8 S-boxes have been generated
```

为了更好地理解算法 3-1，以下对该代码作简要解释。

符号约定：currentsbox 用 c 表示，index 用 in 表示，sbox 用 s 表示，serpent 用 se 表示，算法 3-1 的流程图如图 3-4 所示。

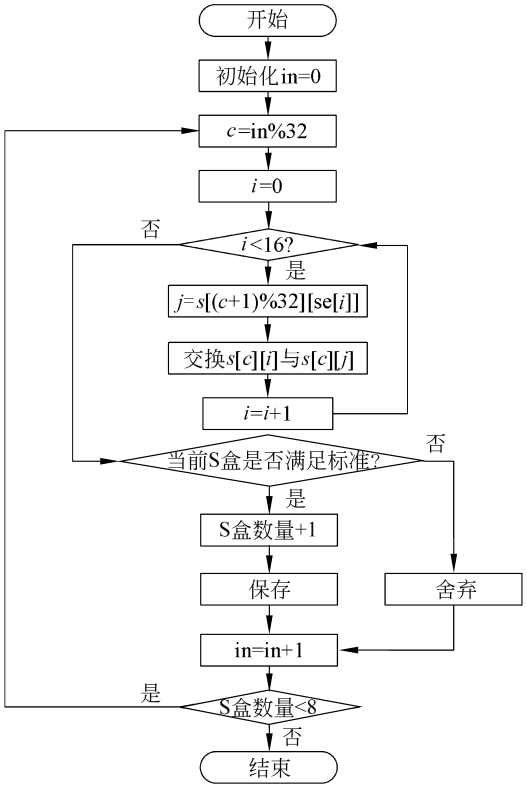


图 3-4　算法 3-1 的流程图

例 3-15　分组密码 Serpent 数组长为 16，其内容为“sboxesforserpent”这 16 个字符的 ASCII 码低 4 位值，如图 3-5 所示。

s	b	o	x	e	s	f	o	r	s	e	r	p	e	n	t
3	2	15	8	5	3	6	15	2	3	5	2	0	5	14	4

图 3-5 “sboxesforserpent”的低 4 位

sbox 数组是由 DES 的 8 个 S 盒初始化而成的。下面以 DES 的 S_1 为例，它对应 sbox[4][16] 共 64 个数。

当 $i=0, c=0$ 时，进入循环。

当 $i=0$ 时，$j=s[1][3]=4$，此时交换 $s[0][0]$ 与 $s[0][4]$的值。

当 $i=1$ 时，$j=s[1][2]=7$，此时交换 $s[0][1]$与 $s[0][7]$的值。

……

一直循环到 $i=15$，此时会产生一个新的 S 盒 sbox[0][.]，如图 3-6 所示。判断该 S 盒是否满足标准，若满足则保存下来。最终生成的 8 个 S 盒如表 3-9 所示。

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

图 3-6 新的 S 盒 sbox[0][.]

表 3-9 生成的 8 个 S 盒

S_1	3	8	15	1	10	6	5	11	14	13	4	2	7	0	9	12
S_2	15	12	2	7	9	0	5	10	1	11	14	8	6	13	3	4
S_3	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
S_4	0	15	11	8	12	9	6	3	13	1	2	4	10	7	5	14
S_5	1	15	8	3	12	0	11	6	2	5	4	10	9	14	7	13
S_6	15	5	2	11	4	10	9	12	0	3	14	8	13	6	7	1
S_7	7	2	12	5	8	4	6	11	14	9	1	15	13	3	10	0
S_8	1	13	15	0	14	8	2	11	7	4	12	10	9	3	5	6

关于这 8 个 S 盒的安全指标测试留作习题。

2. 基于特定结构的 S 盒构造

下面基于经典的分组密码整体结构进行新的 S 盒构造，例如 CRYPTON v0.5 和 CRYPTON v1.0 分别基于 Feistel 结构、SP 结构设计了 S 盒[80]。

例 3-16 分组密码 CRYPTON v0.5 中使用的 S 盒基于 3 轮 Feistel 结构构造，生成的 S_0 和 S_1 互逆，如图 3-7 所示。

其中，s_0、s_1 和 s_2 是 3 个 4×4 的小规模 S 盒，不要求可逆，如表 3-10 所示。

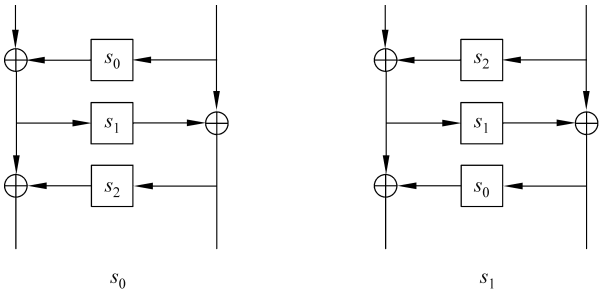


图 3-7　CRYPTON v0.5 的 S 盒

表 3-10　3 个 4×4 的小规模 S 盒

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_0	15	9	6	8	9	9	4	12	6	2	6	10	1	3	5	15
s_1	10	15	4	7	5	2	14	6	9	3	12	8	13	1	11	0
s_2	0	4	8	4	2	15	8	13	1	1	15	7	2	11	14	15

s_0、s_1 和 s_2 的差分均匀度分别为 2、4、4。以 s_0 为例，虽然其不可逆，但是并不影响差分均匀度达到最优 2。由这 3 个小规模 S 盒构成的两个大规模 S 盒 S_0、S_1 差分均匀度都为 8。

例 3-17　CRYPTON v1.0 中使用的 S 盒利用可逆 SP 结构构造，如图 3-8 和表 3-11 所示，其中 s_0、s_1 是两个 4×4 的可逆 S 盒。

输入的 8 比特分两组，分别查 s_0、s_1，输出的 8 比特再进行线性变换，最后经过两个小规模 S 盒的逆输出。显然，新构成的 S 盒是对合的，即 $S=S^{-1}$。

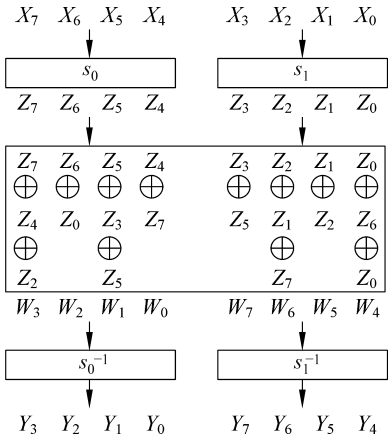


图 3-8　CRYPTON v1.0 的 S 盒

表 3-11　两个 4×4 的小规模 S 盒

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_0	15	14	10	1	11	5	8	13	9	3	2	7	0	6	4	12
s_1	11	10	13	7	8	14	0	5	15	6	3	4	1	9	2	12