密码学原理与实践

(第三版)

第3章 分组密码与高级加密标准

苏 明

[加] Douglas R. Stinson 著

冯登国 等译

概览

- 3．1 引言
- 3．2 代换-置换网络
- 3．3 线性密码分析
  - 3．3．1 堆积引理
  - 3．3．2 S盒的线性逼近
  - 3．3．3 SPN的线性密码分析
- 3．4 差分密码分析

概览

- 3．5 数据加密标准
  - 3．5．1 DES的描述
  - 3．5．2 DES的分析
- 3．6 高级加密标准
  - 3．6．1 AES的描述
  - 3．6．2 AES的分析
- 3．7 工作模式

3．1　引言

- 大多数分组密码是乘积密码，包括一系列**置换**与**代换**操作

- 典型的迭代密码：
  - ✓ 一个轮函数、一个密钥编排方案
  - ✓ 明文的加密经过Nr轮

3．1 引言

- 密钥K----→ $K^1, \ldots, K^{Nr}$ （**密钥编排**方案）
- **轮函数**g: $(K^r, w^{r-1}) \rightarrow w^r$

加密

$$
\begin{aligned}
w^0 &\leftarrow x \\
w^1 &\leftarrow g(w^0, K^1) \\
w^2 &\leftarrow g(w^1, K^2) \\
&\vdots \quad \vdots \quad \vdots \\
w^{\mathrm{Nr}-1} &\leftarrow g(w^{\mathrm{Nr}-2}, K^{\mathrm{Nr}-1}) \\
w^{\mathrm{Nr}} &\leftarrow g(w^{\mathrm{Nr}-1}, K^{\mathrm{Nr}}) \\
y &\leftarrow w^{\mathrm{Nr}}
\end{aligned}
$$

解密

$$
\begin{aligned}
w^{\mathrm{Nr}} &\leftarrow y \\
w^{\mathrm{Nr}-1} &\leftarrow g^{-1}(w^{\mathrm{Nr}}, K^{\mathrm{Nr}}) \\
&\vdots \quad \vdots \quad \vdots \\
w^1 &\leftarrow g^{-1}(w^2, K^2) \\
w^0 &\leftarrow g^{-1}(w^1, K^1) \\
x &\leftarrow w^0
\end{aligned}
$$

3.2 代换-置换网络

- SPN (Substituion Permutation Network)

包含两个变换：

$$\pi_S : \{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell}$$

$$\pi_P : \{1, \cdots, \ell m\} \rightarrow \{1, \cdots, \ell m\}$$

3.2 代换-置换网络

密码体制 3.1　代换–置换网络

设 ℓ, m 和 Nr 都是正整数，$\pi_S : \{0,1\}^{\ell} \to \{0,1\}^{\ell}$ 和 $\pi_P : \{1,\cdots,\ell m\} \to \{1,\cdots,\ell m\}$ 都是置换。设 $\mathcal{P} = \mathcal{C} = \{0,1\}^{\ell m}$，$\mathcal{K} \subseteq (\{0,1\}^{\ell m})^{\mathrm{Nr}+1}$ 是由初始密钥 K 用密钥编排算法生成的所有可能的密钥编排方案之集。对一个密钥编排方案 $(K^1,\cdots,K^{\mathrm{Nr}+1})$，我们使用算法 3.1 来加密明文 x。

算法 3.1　$\mathrm{SPN}(x, \pi_S, \pi_P, (K^1,\cdots,K^{\mathrm{Nr}+1}))$

$w^0 \leftarrow x$

for $r \leftarrow 1$ **to** $\mathrm{Nr}-1$

do
- $u^r \leftarrow w^{r-1} \oplus K^r$
- **for** $i \leftarrow 1$ **to** m
  - **do** $v^r_{<i>} \leftarrow \pi_S(u^r_{<i>})$
- $w^r \leftarrow (v^r_{\pi_P(1)},\cdots,v^r_{\pi_P(\ell m)})$

$u^{\mathrm{Nr}} \leftarrow w^{\mathrm{Nr}-1} \oplus K^{\mathrm{Nr}}$

for $i \leftarrow 1$ **to** m

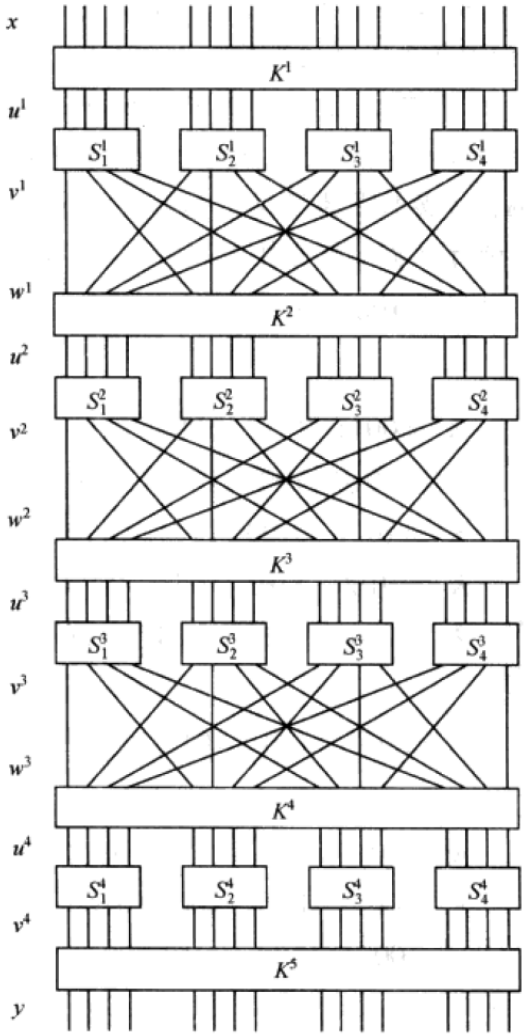
　do $v^{\mathrm{Nr}}_{<i>} \leftarrow \pi_S(u^{\mathrm{Nr}}_{<i>})$

$y \leftarrow v^{\mathrm{Nr}} \oplus K^{\mathrm{Nr}+1}$

output(y)

$$u^r \xrightarrow{SBox} v^r \xrightarrow{\pi\_p} w^r \oplus K^{r+1} \to u^{r+1}$$

3.2 代换-置换网络


x
K¹
u¹
S¹₁ S¹₂ S¹₃ S¹₄
v¹
w¹
K²
u²
S²₁ S²₂ S²₃ S²₄
v²
w²
K³
u³
S³₁ S³₂ S³₃ S³₄
v³
w³
K⁴
u⁴
S⁴₁ S⁴₂ S⁴₃ S⁴₄
v⁴
K⁵
y

3.2 代换-置换网络

- 例子：

$$K^1 = 0011 \quad 1010 \quad 1001 \quad 0100$$
$$K^2 = 1010 \quad 1001 \quad 0100 \quad 1101$$
$$K^3 = 1001 \quad 0100 \quad 1101 \quad 0110$$
$$K^4 = 0100 \quad 1101 \quad 0110 \quad 0011$$
$$K^5 = 1101 \quad 0110 \quad 0011 \quad 1111$$

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_s(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_p(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

$$x = 0010 \quad 0110 \quad 1011 \quad 0111$$

y=?　　计算第一轮w^0 —— $-w^1$

3.2 代换-置换网络

$w^0 = 0010\ \ 0110\ \ 1011\ \ 0111$

$K^1 = 0011\ \ 1010\ \ 1001\ \ 0100$

$u^1 = 0001\ \ 1100\ \ 0010\ \ 0011$

$v^1 = 0100\ \ 0101\ \ 1101\ \ 0001$

$w^1 = 0010\ \ 1110\ \ 0000\ \ 0111$

$K^2 = 1010\ \ 1001\ \ 0100\ \ 1101$

$u^2 = 1000\ \ 0111\ \ 0100\ \ 1010$

$v^2 = 0011\ \ 1000\ \ 0010\ \ 0110$

$w^2 = 0100\ \ 0001\ \ 1011\ \ 1000$

$K^3 = 1001\ \ 0100\ \ 1101\ \ 0110$

$u^3 = 1101\ \ 0101\ \ 0110\ \ 1110$

$v^3 = 1001\ \ 1111\ \ 1011\ \ 0000$

$w^3 = 1110\ \ 0100\ \ 0110\ \ 1110$

$K^4 = 0100\ \ 1101\ \ 0110\ \ 0011$

$u^4 = 1010\ \ 1001\ \ 0000\ \ 1101$

$v^4 = 0110\ \ 1010\ \ 1110\ \ 1001$

$K^5 = 1101\ \ 0110\ \ 0011\ \ 1111$

y=1011 1100 1101 0110

3.2 代换-置换网络

S 盒 $\pi_S : \{0,1\}^\ell \rightarrow \{0,1\}^\ell$

- 存储代价?

  $\ell 2^\ell$

- 类似的：Sbox，see in DES, AES

3．3 线性密码分析

For each of the plaintext-ciphertext pairs, we will begin to decrypt the ciphertext**, using all possible candidate keys** for the **last round** of the cipher. For each candidate key, we compute the values of the relevant state bits involved in the linear relationship, and determine if the above-mentioned linear relationship holds. Whenever it does, we increment a counter corresponding to the particular candidate key. At the end of this process, we hope that the candidate key that has **a frequency** count **furthest from 1/2** times the number of plaintext-ciphertext pairs contains the correct values for these key bits.

3．3 线性密码分析

- 堆积引理： 考虑随机变量组合后的偏差

$$\Pr[\mathbf{X}_i = 0] = p_i \qquad i = 1, 2, \cdots$$

$$\Pr[\mathbf{X}_i \oplus \mathbf{X}_j = 0] = p_i p_j + (1 - p_i)(1 - p_j)$$
$$\Pr[\mathbf{X}_i \oplus \mathbf{X}_j = 1] = p_i (1 - p_j) + (1 - p_i) p_j$$

X_i的偏差定义为$\epsilon_i = p_i - 1/2$, 那么$X_i \oplus X_j$的偏差为？

$$2\epsilon_i\epsilon_j$$

堆积引理

一般的：

引理 3.1(堆积引理)　设 $\mathbf{X}_{i_1},\cdots,\mathbf{X}_{i_k}$ 是独立随机变量，$\epsilon_{i_1,i_2,\cdots,i_k}$ $(i_1<i_2<\cdots<i_k)$ 表示随机变量 $\mathbf{X}_{i_1}\oplus\mathbf{X}_{i_2}\oplus\cdots\oplus\mathbf{X}_{i_k}$ 的偏差，则

$$\epsilon_{i_1,i_2,\cdots,i_k}=2^{k-1}\prod_{j=1}^{k}\epsilon_{i_j}$$

Pf. 归纳证明

堆积引理

推论 3.2 设 $\mathbf{X}_{i_1},\cdots,\mathbf{X}_{i_k}$ 是独立随机变量，$\epsilon_{i_1,i_2,\cdots,i_k}$ 表示随机变量 $\mathbf{X}_{i_1}\oplus\mathbf{X}_{i_2}\oplus\cdots\oplus\mathbf{X}_{i_k}$ 的偏差，若对某 j，有 $\epsilon_{i_j}=0$，则 $\epsilon_{i_1,i_2,\cdots,i_k}=0$。

随着随机变量数目的增多， **偏差越来越小**；
如果存在一个无偏差，则整体无偏差。

S盒的线性逼近

考虑一个S盒： $\pi_S : \{0,1\}^m \to \{0,1\}^n$

- m重输入$X = (x_1, \cdots, x_m)$均匀随机的从$\{0,1\}^m$中选取
- 直觉上，n重输出 $Y = (y_1, \cdots, y_n)$ 与输入有**相关性**

$(y_1, \cdots, y_n) \neq \pi_S(x_1, \cdots, x_m)$， 则 $\Pr[\mathbf{X}_1 = x_1, \cdots, \mathbf{X}_m = x_m, \mathbf{Y}_1 = y_1, \cdots, \mathbf{Y}_n = y_n] = 0$

$(y_1, \cdots, y_n) = \pi_S(x_1, \cdots, x_m)$， 则 $\Pr[\mathbf{X}_1 = x_1, \cdots, \mathbf{X}_m = x_m, \mathbf{Y}_1 = y_1, \cdots, \mathbf{Y}_n = y_n] = 2^{-m}$

S盒的线性逼近

- 线性逼近S盒 $\mathbf{X}_{i_1} \oplus \cdots \oplus \mathbf{X}_{i_k} \oplus \mathbf{Y}_{j_1} \oplus \cdots \oplus \mathbf{Y}_{j_r}$

- 如果这一形式的随机变量具有偏离0的偏差值，线性分析就成为可能

S盒的线性逼近

- 例： $\pi_S : \{0,1\}^4 \rightarrow \{0,1\}^4$

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0
0	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	1	1	1
0	1	1	0	1	0	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	1	1
1	0	0	1	1	0	1	0
1	0	1	0	0	1	1	0
1	0	1	1	1	1	0	0
1	1	0	0	0	1	0	1
1	1	0	1	1	0	0	1
1	1	1	0	0	0	0	0
1	1	1	1	0	1	1	1

$X_1 \oplus X_4 \oplus Y_2$ 的偏差？

0

$X_3 \oplus X_4 \oplus Y_1 \oplus Y_4$ 的偏差？

-3/8

S盒的线性逼近

- 启发我们寻找S盒更好的线性逼近

设 $N_L(a,b)$ 表示满足如下条件的二进制 8 元组 $(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4)$ 的个数：

$$(y_1, y_2, y_3, y_4) = \pi_S(x_1, x_2, x_3, x_4)$$

$$\left(\bigoplus_{i=1}^{4} a_i x_i\right) \oplus \left(\bigoplus_{i=1}^{4} b_i y_i\right) = 0$$

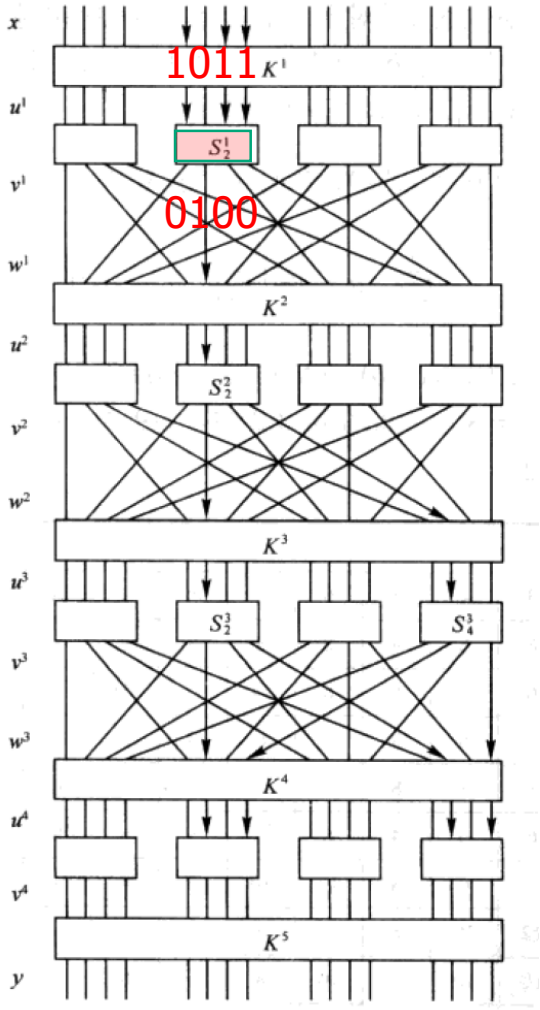
则随机变量 $\left(\bigoplus_{i=1}^{4} a_i \mathbf{X}_i\right) \oplus \left(\bigoplus_{i=1}^{4} b_i \mathbf{Y}_i\right)$ 的偏差公式为：$\epsilon(a,b) = (N_L(a,b) - 8)/16$

S盒的线性逼近

a \ b	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
1	8	8	6	6	8	8	6	14	10	10	8	8	10	10	8	8
2	8	8	6	6	8	8	6	6	8	8	10	10	8	8	2	10
3	8	8	8	8	8	8	8	8	10	2	6	6	10	10	6	6
4	8	10	8	6	6	4	6	8	8	6	8	10	10	4	10	8
5	8	6	6	8	6	8	12	10	6	8	4	10	8	6	6	8
6	8	10	6	12	10	8	8	10	8	6	10	12	6	8	8	6
7	8	6	8	10	10	4	10	8	6	8	10	8	12	10	8	10
8	8	8	8	8	8	8	8	8	6	10	10	6	10	6	6	2
9	8	8	6	6	8	8	6	6	4	8	6	10	8	12	10	6
A	8	12	6	10	4	8	10	6	10	10	8	8	10	10	8	8
B	8	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8
C	8	6	12	6	6	8	10	8	10	8	10	12	8	10	8	6
D	8	10	10	8	6	12	8	10	4	6	10	8	10	8	8	10
E	8	10	10	8	6	4	8	10	6	8	8	6	4	10	6	8
F	8	6	4	6	6	8	10	8	8	6	12	6	6	8	10	8

线性逼近表： $N_L(a,b)$ 的值

SPN的线性密码分析



Active S Box

$$\mathbf{T}_1 = \mathbf{U}_5^1 \oplus \mathbf{U}_7^1 \oplus \mathbf{U}_8^1 \oplus \mathbf{V}_6^1$$

- 在 S_2^1 中，随机变量 $\mathbf{T}_1 = \mathbf{U}_5^1 \oplus \mathbf{U}_7^1 \oplus \mathbf{U}_8^1 \oplus \mathbf{V}_6^1$ 具有偏差 1/4

- 在 S_2^2 中，随机变量 $\mathbf{T}_2 = \mathbf{U}_6^2 \oplus \mathbf{V}_6^2 \oplus \mathbf{V}_8^2$ 具有偏差−1/4
- 在 S_2^3 中，随机变量 $\mathbf{T}_3 = \mathbf{U}_6^3 \oplus \mathbf{V}_6^3 \oplus \mathbf{V}_8^3$ 具有偏差−1/4
- 在 S_4^3 中，随机变量 $\mathbf{T}_4 = \mathbf{U}_{14}^3 \oplus \mathbf{V}_{14}^3 \oplus \mathbf{V}_{16}^3$ 具有偏差−1/4

SPN的线性密码分析

- 假设T_1, T_2, T_3, T_4互相独立，就可用堆积引理来计算偏差

（近似逼近）

$\mathbf{T}_1 \oplus \mathbf{T}_2 \oplus \mathbf{T}_3 \oplus \mathbf{T}_4$具有偏差$2^3(1/4)(-1/4)^3 = -1/32$

$$\mathbf{T}_1 = \mathbf{U}_5^1 \oplus \mathbf{U}_7^1 \oplus \mathbf{U}_8^1 \oplus \mathbf{V}_6^1 = \mathbf{X}_5 \oplus \mathbf{K}_5^1 \oplus \mathbf{X}_7 \oplus \mathbf{K}_7^1 \oplus \mathbf{X}_8 \oplus \mathbf{K}_8^1 \oplus \mathbf{V}_6^1$$
$$\mathbf{T}_2 = \mathbf{U}_6^2 \oplus \mathbf{V}_6^2 \oplus \mathbf{V}_8^2 = \mathbf{V}_6^2 \oplus \mathbf{K}_6^2 \oplus \mathbf{V}_6^2 \oplus \mathbf{V}_8^2$$
$$\mathbf{T}_3 = \mathbf{U}_6^3 \oplus \mathbf{V}_6^3 \oplus \mathbf{V}_8^3 = \mathbf{V}_6^3 \oplus \mathbf{K}_6^3 \oplus \mathbf{V}_6^3 \oplus \mathbf{V}_8^3$$
$$\mathbf{T}_4 = \mathbf{U}_{14}^3 \oplus \mathbf{V}_{14}^3 \oplus \mathbf{V}_{16}^3 = \mathbf{V}_8^2 \oplus \mathbf{K}_{14}^3 \oplus \mathbf{V}_{14}^3 \oplus \mathbf{V}_{16}^3$$

SPN的线性密码分析

$$\mathbf{X}_5 \oplus \mathbf{X}_7 \oplus \mathbf{X}_8 \oplus \mathbf{V}_6^3 \oplus \mathbf{V}_8^3 \oplus \mathbf{V}_{14}^3 \oplus \mathbf{V}_{16}^3$$

$$\oplus \mathbf{K}_5^1 \oplus \mathbf{K}_7^1 \oplus \mathbf{K}_8^1 \oplus \mathbf{K}_6^2 \oplus \mathbf{K}_6^3 \oplus \mathbf{K}_{14}^3$$

代入

$$\mathbf{V}_6^3 = \mathbf{U}_6^4 \oplus \mathbf{K}_6^4$$

$$\mathbf{V}_8^3 = \mathbf{U}_{14}^4 \oplus \mathbf{K}_{14}^4$$

$$\mathbf{V}_{14}^3 = \mathbf{U}_8^4 \oplus \mathbf{K}_8^4$$

$$\mathbf{V}_{16}^3 = \mathbf{U}_{16}^4 \oplus \mathbf{K}_{16}^4$$

SPN的线性密码分析

得到

$$\mathbf{X}_5 \oplus \mathbf{X}_7 \oplus \mathbf{X}_8 \oplus \mathbf{U}_6^4 \oplus \mathbf{U}_8^4 \oplus \mathbf{U}_{14}^4 \oplus \mathbf{U}_{16}^4$$
$$\oplus \mathbf{K}_5^1 \oplus \mathbf{K}_7^1 \oplus \mathbf{K}_8^1 \oplus \mathbf{K}_6^2 \oplus \mathbf{K}_6^3 \oplus \mathbf{K}_{14}^3 \oplus \mathbf{K}_6^4 \oplus \mathbf{K}_8^4 \oplus \mathbf{K}_{14}^4 \oplus \mathbf{K}_{16}^4$$

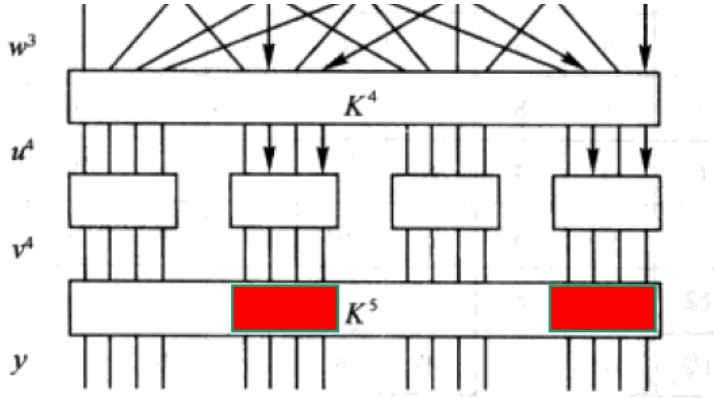
假设固定了K^1, K^2, K^3, K^4密钥

$\mathbf{X}_5 \oplus \mathbf{X}_7 \oplus \mathbf{X}_8 \oplus \mathbf{U}_6^4 \oplus \mathbf{U}_8^4 \oplus \mathbf{U}_{14}^4 \oplus \mathbf{U}_{16}^4$ 具有偏差$\pm 1/32$

SPN的线性密码分析

据此我们可以进行线性密码攻击：

- ✓ 拥有同一密钥K加密的明-密文对
- ✓ 为保证成功率，$T \approx 8000$
- ✓ 将分析获得$K^5_{<2>}, K^5_{<4>}$的8个比特



SPN的线性密码分析

对每一个$(x, y)\in T$及每一个候选子密钥，计算y的一个部分解密并获得$u^4_{<2>}$和$u^4_{<2>}$。

计算 $x_5 \oplus x_7 \oplus x_8 \oplus u^4_6 \oplus u^4_8 \oplus u^4_{14} \oplus u^4_{16}$

真正的候选子密钥对应 的计数器具有接近于$T/2 \pm T/32$之值

线性攻击想要成功，所需的**明-密文对**数目$T \approx c\epsilon^{-2}$. 这里$c \approx 8, \epsilon = 1/32$

SPN的线性密码分析

Matsui, M. "Linear cryptanalysis method for DES cipher". Advances in Cryptology - EUROCRYPT 1993

直观解释

Lemma 2 *Let N be the number of given random plaintexts and p be the probability that equation (1) holds, and assume $|p - 1/2|$ is sufficiently small. Then the success rate of Algorithm 1 is*

$$\int_{-2\sqrt{N}|p-1/2|}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx. \tag{9}$$

Table 2 shows a numerical calculation of expression (9).

N	$\frac{1}{4}\|p-1/2\|^{-2}$	$\frac{1}{2}\|p-1/2\|^{-2}$	$\|p-1/2\|^{-2}$	$2\|p-1/2\|^{-2}$
Success Rate	84.1%	92.1%	97.7%	99.8%

SPN的线性密码分析

算法 3.2　线性攻击$(\mathcal{T}, T, \pi_S^{-1})$

for $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)

　do $\text{Count}[L_1, L_2] \leftarrow 0$

for each $(x, y) \in \mathcal{T}$

do {
　for $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)
　do {
　　$v_{<2>}^4 \leftarrow L_1 \oplus y_{<2>}$
　　$v_{<4>}^4 \leftarrow L_2 \oplus y_{<4>}$
　　$u_{<2>}^4 \leftarrow \pi_S^{-1}(v_{<2>}^4)$
　　$u_{<4>}^4 \leftarrow \pi_S^{-1}(v_{<4>}^4)$
　　$z \leftarrow x_5 \oplus x_7 \oplus x_8 \oplus u_6^4 \oplus u_8^4 \oplus u_{14}^4 \oplus u_{16}^4$
　　if $z = 0$
　　then $\text{Count}[L_1, L_2] \leftarrow \text{Count}[L_1, L_2] + 1$
　}
}

$\max \leftarrow -1$

for $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)

do {
　$\text{Count}[L_1, L_2] \leftarrow |\text{Count}[L_1, L_2] - T/2|$
　if $\text{Count}[L_1, L_2] > \max$
　then {
　　$\max \leftarrow \text{Count}[L_1, L_2]$
　　$\text{max key} \leftarrow (L_1, L_2)$
　}
}

output(maxkey)

SPN的线性密码分析

总结：

- 寻找S盒的最好**线性逼近**
- 分析活动S盒的**变量路径**
- 寻找具有较大偏差的**随机变量组合**
- **穷举轮密钥中的待定密钥比特**，接近偏差

差分密码分析

- 将两个输入的异或与其对应的两个输出的异或比较
- 差分分析是一个选择明文攻击
  - ✓ 攻击者具有大量的4元组(x,x*; y,y*)
  - ✓ $x' = x \oplus x^*$ 是固定的
  - ✓ 应用所有可能的候选密钥对最后一轮进行解密

差分密码分析

定义 3.1 设 $\pi_S : \{0,1\}^m \to \{0,1\}^n$ 是一个 S 盒。考虑长为 m 的有序比特串对 (x, x^*)，我们称 S 盒的输入异或为 $x \oplus x^*$，输出异或为 $\pi_S(x) \oplus \pi_S(x^*)$。注意输出异或是一个 n 长比特串。

对任何 $x' \in \{0,1\}^m$，定义集合 $\Delta(x')$ 为包含所有具有输入异或值 x' 的有序对 (x, x^*)。

$\Delta(x')$ 有多少个元素？

2^m

差分密码分析

差分表 x′=1011

x	x^*	y	y^*	y'
0000	1011	1110	1100	0010
0001	1010	0100	0110	0010
0010	1001	1101	1010	0111
0011	1000	0001	0011	0010
0100	1111	0010	0111	0101
0101	1110	1111	0000	1111
0110	1101	1011	1001	0010
0111	1100	1000	0101	1101
1000	0011	0011	0001	0010
1001	0010	1010	1101	0111
1010	0001	0110	0100	0010
1011	0000	1100	1110	0010
1100	0111	0101	1000	1101
1101	0110	1001	1011	0010
1110	0101	0000	1111	1111
1111	0100	0111	0010	0101

差分密码分析

- 寻找Abnormity

0000	0001	0010	0011	0100	0101	0110	0111
0	0	8	0	0	2	0	2
1000	1001	1010	1011	1100	1101	1110	1111
0	0	0	0	0	2	0	2

定义 $N_D(x', y') = \left|\{(x, x^*) \in \Delta(x') : \pi_S(x) \oplus \pi_S(x^*) = y'\}\right|$

差分密码分析

差分分布表

a'	b'															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

差分密码分析

- 异或扩散率（扩散率）

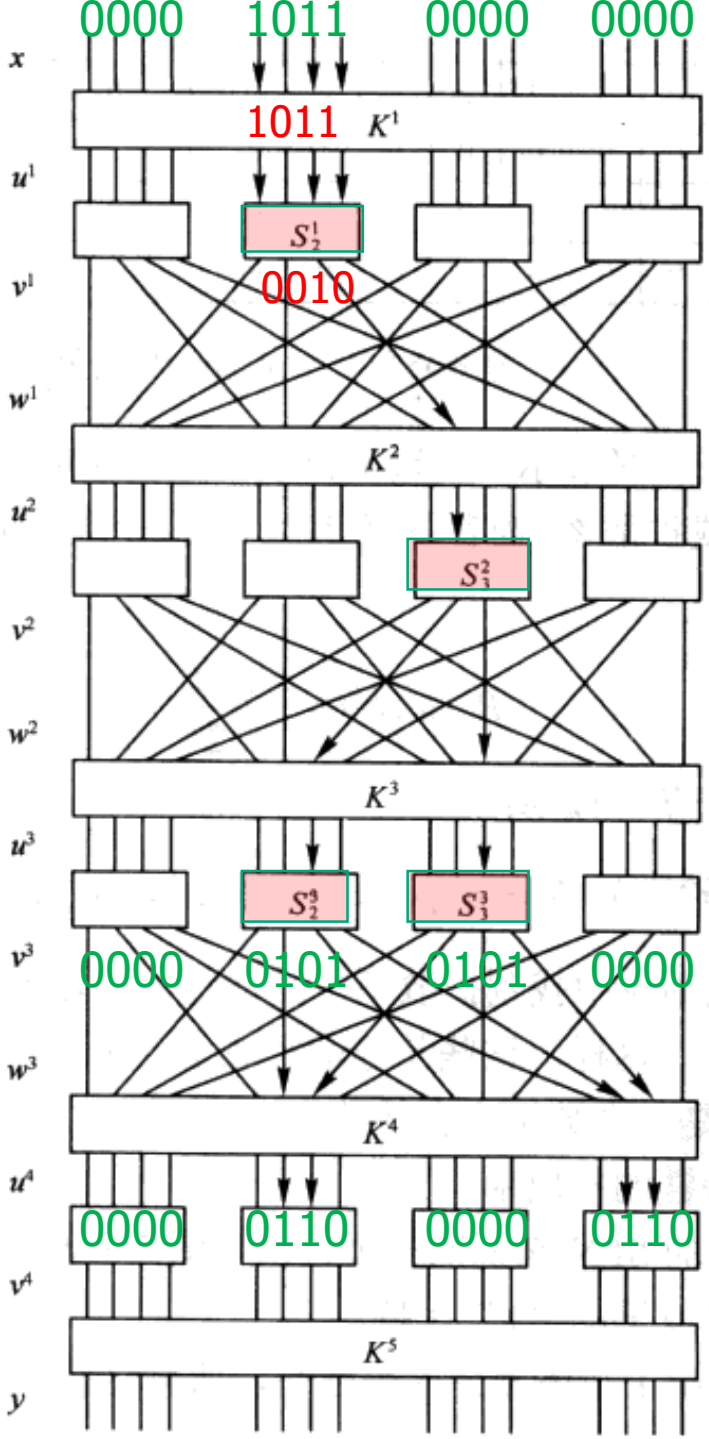
$$R_p(a', b') = \frac{N_D(a', b')}{2^m}$$

$$\Pr[\text{输出异或} = b' \mid \text{输入异或} = a'] = R_p(a', b')$$

由此可以计算一条**差分链的扩散率**（把差分链里的扩散率相乘；逼近）

差分密码分析

$$R_p(0000\,1011\,0000\,0000,\ 0000\,0101\,0101\,0000)$$
$$=\frac{1}{2}\times\left(\frac{3}{8}\right)^3=\frac{27}{1024}$$



$x' = 0000\,1011\,0000\,0000$

- 在 S_2^1 中， $R_p(1011, 0010) = 1/2$
- 在 S_3^2 中， $R_p(0100, 0110) = 3/8$
- 在 S_2^3 中， $R_p(0010, 0101) = 3/8$
- 在 S_3^3 中， $R_p(0010, 0101) = 3/8$

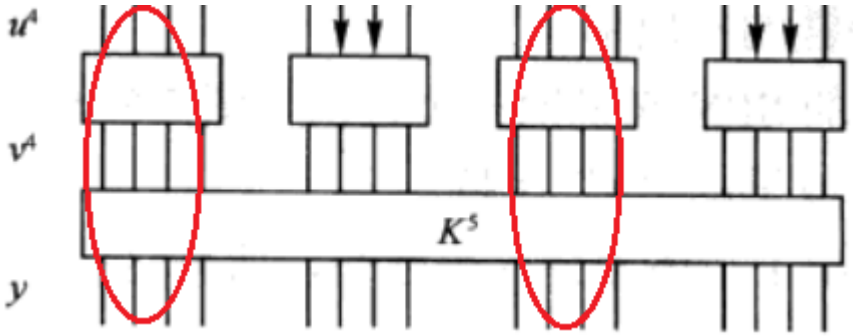
$(u^4)' = 0000\,0110\,0000\,0110$

Pr(*)=27/1024

差分密码分析

- 差分攻击的算法与线性攻击算法类似

T 是 4 元组 (x, x^*, y, y^*) 组成的集合，其中差分值 x' 是固定的。



过滤操作

正确对满足 $(u^4_{<1>})' = (u^4_{<3>})' = 0000$

即 $y_{<1>} = (y_{<1>})^*$ 和 $y_{<3>} = (y_{<3>})^*$

差分密码分析

算法 3.3 差分攻击$(\mathcal{T}, T, \pi_S^{-1})$

for $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)

do $\text{Count}[L_1, L_2] \leftarrow 0$

for each $(x, y, x^*, y^*) \in \mathcal{T}$

do
- **if** $(y_{<1>} = (y_{<1>})^*)$ **and** $(y_{<3>} = (y_{<3>})^*)$
- **then**
  - **for** $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)
  - **do**
    - $v^4_{<2>} \leftarrow L_1 \oplus y_{<2>}$
    - $v^4_{<4>} \leftarrow L_2 \oplus y_{<4>}$
    - $u^4_{<2>} \leftarrow \pi_S^{-1}(v^4_{<2>})$
    - $u^4_{<4>} \leftarrow \pi_S^{-1}(v^4_{<4>})$
    - $(v^4_{<2>})^* \leftarrow L_1 \oplus (y_{<2>})^*$
    - $(v^4_{<4>})^* \leftarrow L_2 \oplus (y_{<4>})^*$
    - $(u^4_{<2>})^* \leftarrow \pi_S^{-1}((v^4_{<2>})^*)$
    - $(u^4_{<4>})^* \leftarrow \pi_S^{-1}((v^4_{<4>})^*)$
    - $(u^4_{<2>})' \leftarrow u^4_{<2>} \oplus (u^4_{<2>})^*$
    - $(u^4_{<4>})' \leftarrow u^4_{<4>} \oplus (u^4_{<4>})^*$
    - **if** $((u^4_{<2>})' = 0110)$ **and** $((u^4_{<4>})' = 0110)$
      - **then** $\text{Count}[L_1, L_2] \leftarrow \text{Count}[L_1, L_2] + 1$

$\text{max} \leftarrow -1$

for $(L_1, L_2) \leftarrow (0,0)$ **to** (F, F)

do
- **if** $\text{Count}[L_1, L_2] > \text{max}$
- **then**
  - $\text{max} \leftarrow \text{Count}[L_1, L_2]$
  - $\text{maxkey} \leftarrow (L_1, L_2)$

output(maxkey)

差分密码分析

- 为了确保选择明文攻击的效率

4元组$(x, x^*; y, y^*)$数量$\boldsymbol{T \approx c\epsilon^{-1}}$

其中c是一个小常数

$$\epsilon^{-1} = \frac{1024}{27} \approx 38$$

$$T: 50 - 100$$

3.5 数据加密标准

- DES(Data Encryption Standard)
  由NIST（美国国家标准局）公开征集
- IBM设计与开发 based on an earlier design by Horst Feistel
- Feistel Structure

DES

- Feistel Structure

每一个状态u^i分成两半 L^i, R^i

轮函数 $g(L^{i-1}, R^{i-1}, K^i) = (L^i, R^i)$

$$L^i = R^{i-1}$$

$$R^i = L^{i-1} \oplus f(R^{i-1}, K^i)$$

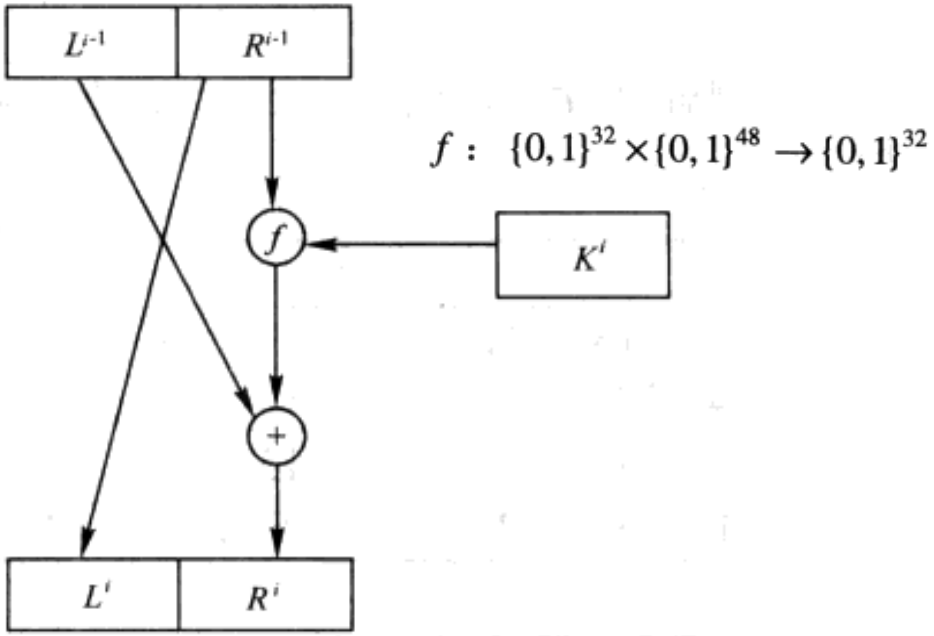
f不需要是单射！ $L^{i-1} = R^i \oplus f(L^i, K^i)$

$$R^{i-1} = L^i$$

DES

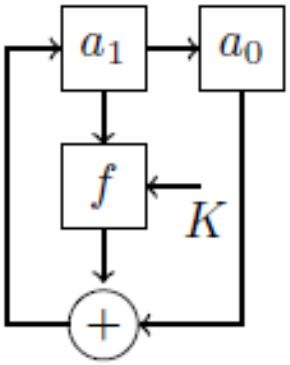
- 密钥编排方案

56位的种子密钥-→$(K^1, K^2, \ldots, K^{16})$



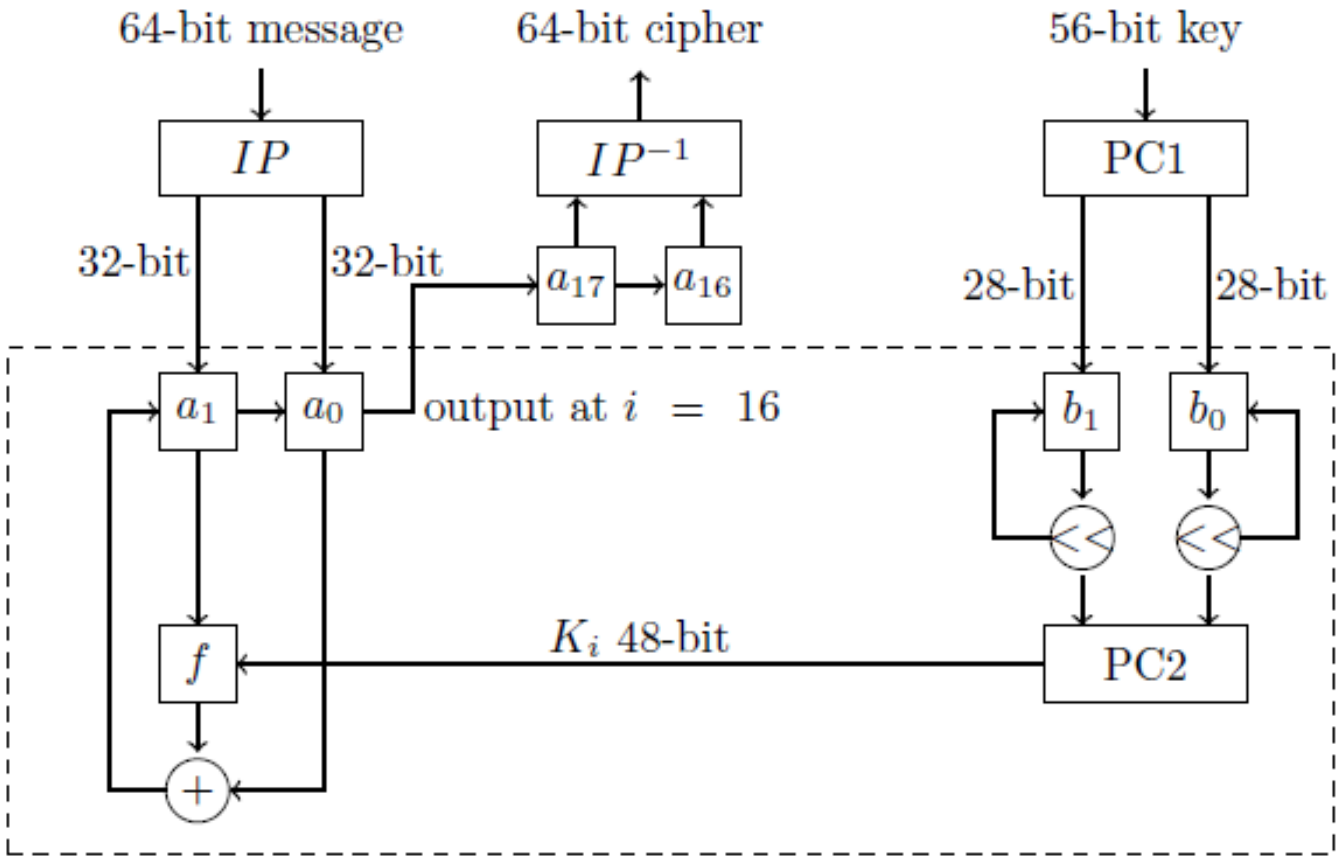
DES

- Data Encryption Standard, NIST 1976



The features of DES
A Feistel cipher with 16 rounds
64-bit block length, each register holds 32-bits
56-bit key
Each round of DES uses a 48-bit subkey,
a subset of the 56-bit key.

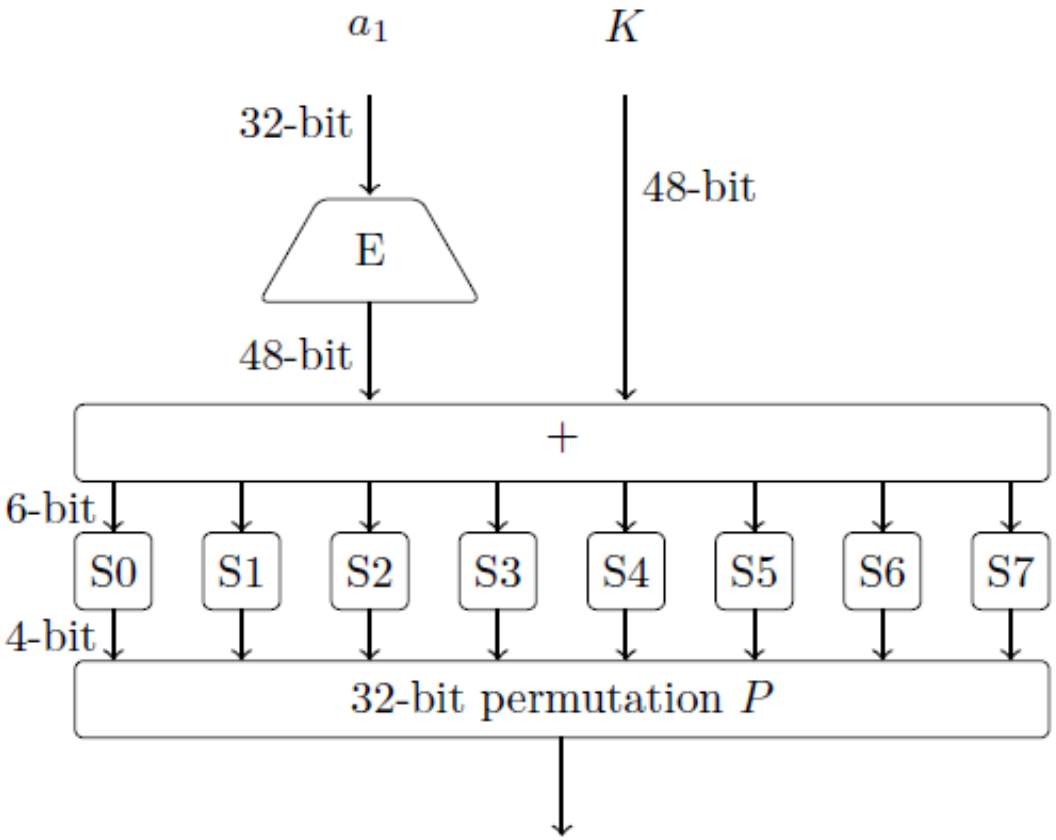
DES


64-bit message
64-bit cipher
56-bit key
IP
IP^{-1}
PC1
32-bit
32-bit
a_{17}
a_{16}
28-bit
28-bit
a_1
a_0
output at $i = 16$
b_1
b_0
f
K_i 48-bit
PC2
+

DES

IP:	initial permutation
IP^{-1}:	the inverse of IP
PC1:	permuted choice 1
PC2:	permuted choice 2
$\ll$:	left shifts for round 1, 2, 9, 16 shift 1, the rest of rounds shift 2.
Update	The update functions for the internal state is $a_{i+2} = f(a_{i+1}, k_{i+1}) + a_i, i = 0, 1, \cdots, 15$ where f is a nonlinear function given by the S-boxes described below, and (a_0, a_1) is image of the plaintext under permutation IP, and the ciphertext is the image of (a_{16}, a_{17}) under the inverse of IP.

DES-Feedback f and S-Boxes



$$f: \{0,1\}^{32} \times \{0,1\}^{48} \rightarrow \{0,1\}^{32}$$

DES-Feedback f and S-Boxes

- Each S-Box is a function with 6-bit input and 4-bit output

c_5c_0	to select row (0, 1, 2, 3), and
$c_4c_3c_2c_1$	to select column $(0, 1, \cdots, 15)$

S-box0															
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

101000→? 13: 1101

DES S-Boxes

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

DES

E()

E 比特选择表					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

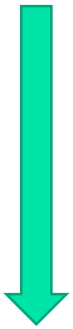
$$\mathbf{E}(A) = (a_{32}, a_1, a_2, a_3, a_4, a_5, a_4, \cdots, a_{31}, a_{32}, a_1)$$

DES

- P()

P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

$$C = (c_1, c_2, \cdots, c_{32})$$

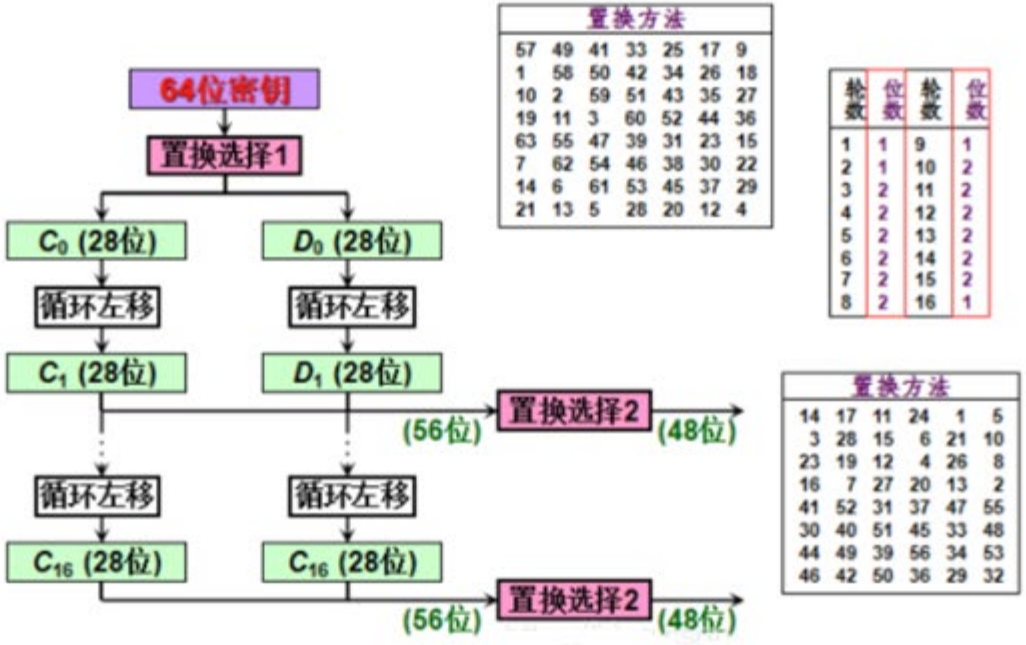


$$(c_{16}, c_7, c_{20}, c_{21}, c_{29}, \cdots, c_{11}, c_4, c_{25})$$

DES

子密钥生成算法

将初始密钥（64位）经过置换表1，生成的序列为（56位），将其分为左半部分C0（28位）与右半部分D0（28位），根据其轮数对应其左移的位数，将这两部分均循环左移，得到C1与D1，然后将其拼接在一起，作为输入（56位）经过置换表2，生成48位的子密钥K1，然后将C1，D1左移进行相同的步骤，生成子密钥K1~K16。



置换方法

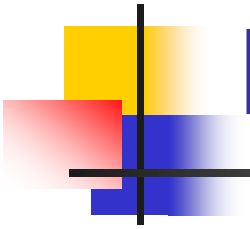
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

轮数	位数	轮数	位数
1	1	9	1
2	1	10	2
3	2	11	2
4	2	12	2
5	2	13	2
6	2	14	2
7	2	15	2
8	2	16	1

置换方法

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

DES-Feedback f and S-Boxes



- Efficient Implementation: hardware and software

- **the encryption and decryption are the same**, except the order of the key schedule is reversed.

DES 分析

- 差分分析　1991　Biham, Shamir

Eli Biham， Adi Shamir， Differential cryptanalysis of DES-like cryptosystems， Journal of Cryptology volume 4, pages 3–72 (1991)

- 线性分析　1994　Matsui

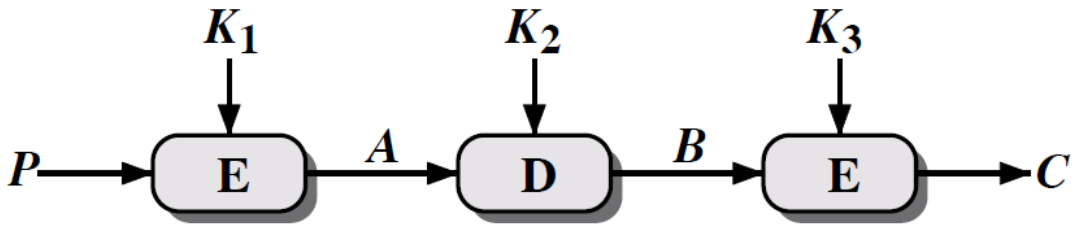
Matsui, M. "Linear cryptanalysis method for DES cipher". Advances in Cryptology - EUROCRYPT 1993.
Matsui, M. "The first experimental cryptanalysis of the data encryption standard". Advances in Cryptology - CRYPTO 1994.

- 硬件破解　（98年 25万美元，密钥搜索机）

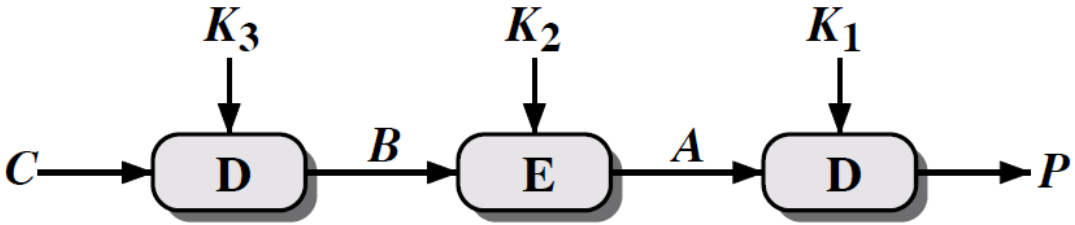
- 有效密钥 56 bits太小 (8位校验位)

3DES

- Enlarge the key Space



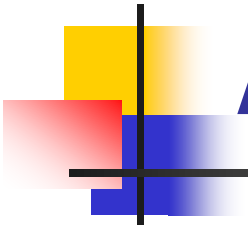
(a) Encryption



(b) Decryption

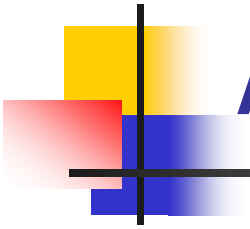
AES

- 1997-2001 NIST 征集标准 (公开、国际性)
- Advanced Encryption Standard
  - ✓ Basic requirements: 128bit分组
  - ✓ Key 支持128，192，256 bits
- 15→5(MARS, RC6, Rijndael, Serpent, Twofish)
- Winner: Daemen, Rijmen



AES

- 安全性 (Primary concern)
- 代价 (计算效率、软硬件实现、智能卡实现)
- 算法与实现特性 (灵活性、简洁性...)



AES

- Advanced Encryption Standard

The length of the input block, the output block and the State is 128 bits.

Three key lengths are available: 128, 192, or 256 bits.

The number of rounds varies from 10 to 14, depending on the key length.

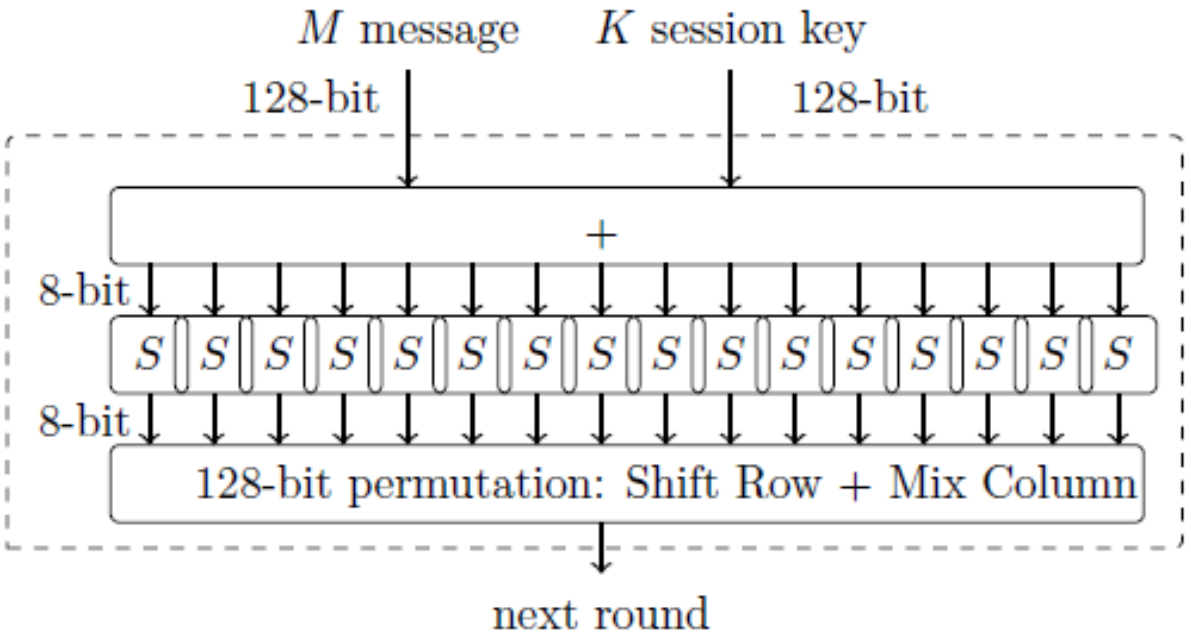
AES

Each round consists of three functions, which are in three "layers" as

(a) 8-bit inverse permutation (sub-byte transform),

(b) 32-bit linear transformation (mix columns operation), and

(c) 128-bit permutation (**Shift** rows operation).



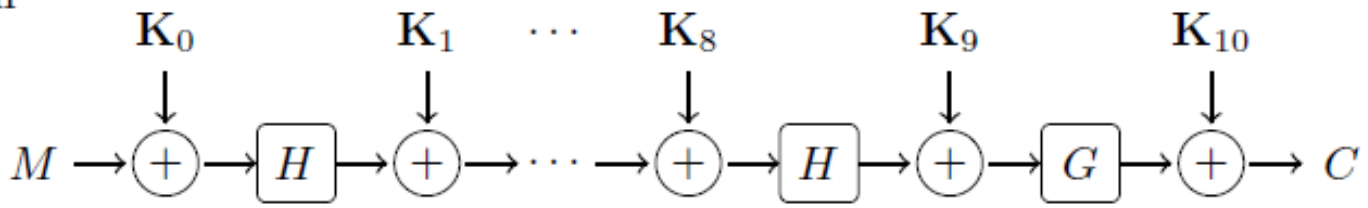
AES

Based on GF(2^8), where $m(x) = x^8 + x^4 + x^3 + x + 1$

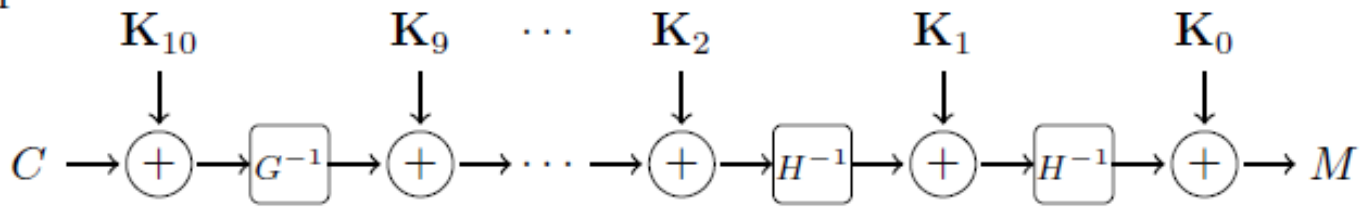
- Add Round Key
- ByteSub Transformation
- ShiftRow
- MixColumn

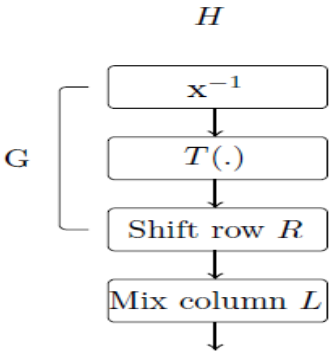
AES-Flow

Encryption



Decryption





SubBytes

AES-Flow

Encryption	Decryption
$M_0 = M + \mathbf{K}_0$, in $M_4(\mathbb{F}_{2^8})$	$C_0 = C + \mathbf{K}_{10}$,
$M_i = H(M_{i-1}) + \mathbf{K}_i, i = 1, 2, \cdots, 9$	$C_1 = G^{-1}(C_0) + \mathbf{K}_9$
$M_{10} = G(M_9) + \mathbf{K}_{10}$	$C_i = H^{-1}(C_{i-1}) + \mathbf{K}_{10-i}, i = 2, 3, \cdots, 10$
The ciphertext is $C = M_{10}$.	The plaintext is $M = C_{10}$.

Key	00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 01, 0b, 0c, 0d, 0e, 0f
Plaintext	00, 11, 22, 33, 44, 55, 66, 77, 88, 99, aa, bb, cc, dd, ee, ff
Ciphertext	69, c4, e0, d8, 6a, 7b, 04, 30, d8, cd, b7, 80, 70, b4, c5, 5a

AES-PseudoCode

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
   byte  state[4,Nb]

   state = in

   AddRoundKey(state, w[0, Nb-1])

   for round = 1 step 1 to Nr-1
      SubBytes(state)
      ShiftRows(state)
      MixColumns(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
   end for

   SubBytes(state)
   ShiftRows(state)
   AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

   out = state
end
```

AES-Details

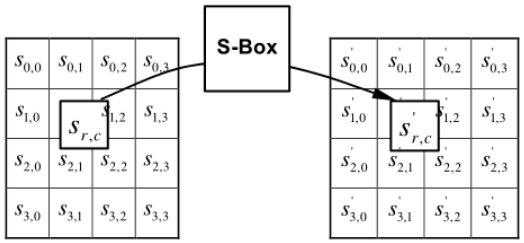
- 明文由16个字节$x_0, x_1, \ldots, x_{15}$组成(over GF(2^8))

x_0	x_4	x_8	x_{12}
x_1	x_5	x_9	x_{13}
x_2	x_6	x_{10}	x_{14}
x_3	x_7	x_{11}	x_{15}

$s_{0,0}$	$s_{0,1}$	$s_{0,2}$	$s_{0,3}$
$s_{1,0}$	$s_{1,1}$	$s_{1,2}$	$s_{1,3}$
$s_{2,0}$	$s_{2,1}$	$s_{2,2}$	$s_{2,3}$
$s_{3,0}$	$s_{3,1}$	$s_{3,2}$	$s_{3,3}$

AES-Details

- S盒具有清晰的代数结构



X	Y															
	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

AES-Details

算法 3.4 SubBytes $(a_7a_6a_5a_4a_3a_2a_1a_0)$

external FieldInv, BinaryToField, FieldToBinary

$z \leftarrow$ BinaryToField $(a_7a_6a_5a_4a_3a_2a_1a_0)$

if $z \neq 0$

  then $z \leftarrow$ FieldInv(z)

$(a_7a_6a_5a_4a_3a_2a_1a_0) \leftarrow$ FieldToBinary(z)

$(c_7c_6c_5c_4c_3c_2c_1c_0) \leftarrow (01100011)$

注　在下面的循环中，所有下标都要经过模 8 约简。

for $i \leftarrow 0$ **to** 7

  do $b_i \leftarrow (a_i + a_{i+4} + a_{i+5} + a_{i+6} + a_{i+7} + c_i) \bmod 2$

return $(b_7b_6b_5b_4b_3b_2b_1b_0)$

教材中的SubBytes逻辑　(Stinson)

AES-Details

SubBytes() transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table (S-box). This S-box, which is invertible, is constructed by composing two transformations:

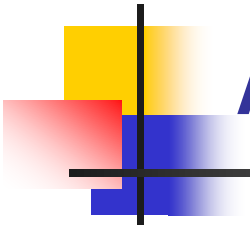
(1) Take the *multiplicative inverse* in the finite field $GF(2^8)$, where the irreducible polynomial is $m(x) = x^8 + x^4 + x^3 + x + 1$; and the element {00} is mapped to itself.

(2) Apply the following *affine transformation* (over GF(2)):

In matrix form, the affine transformation element of the S-box is as follows:

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

for 0<=i<8, where b_i is the ith bit of the byte, and b_i' indicates the corresponding bit to be updated. Note that b_0 is the lowest bit.



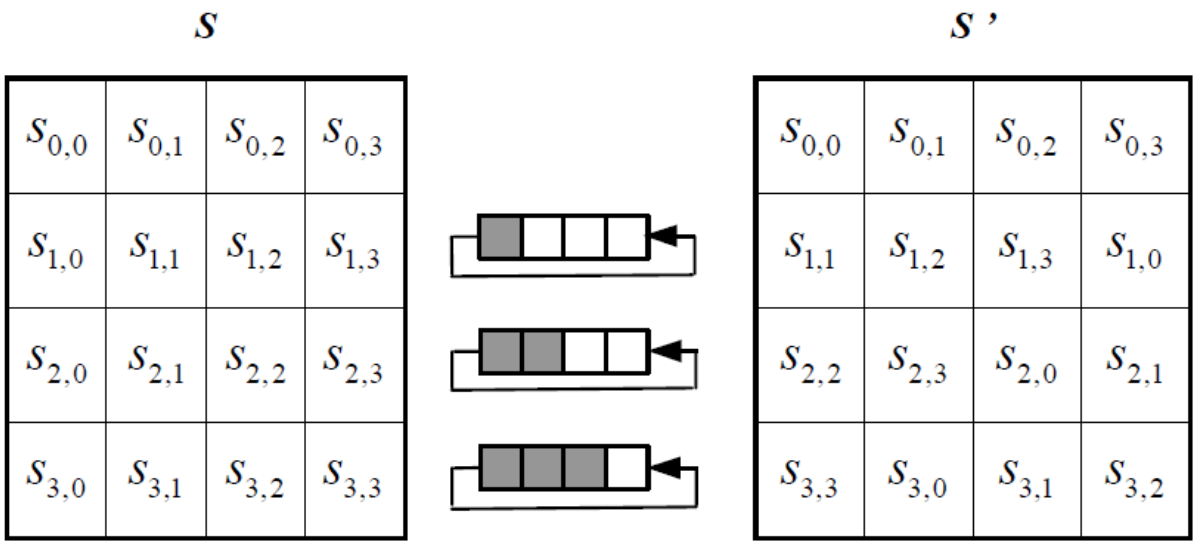
AES-Details

- Question 1: Suppose the input is {01}, what is the output?

- Question 2: Suppose the input is {8d}, what is the output?

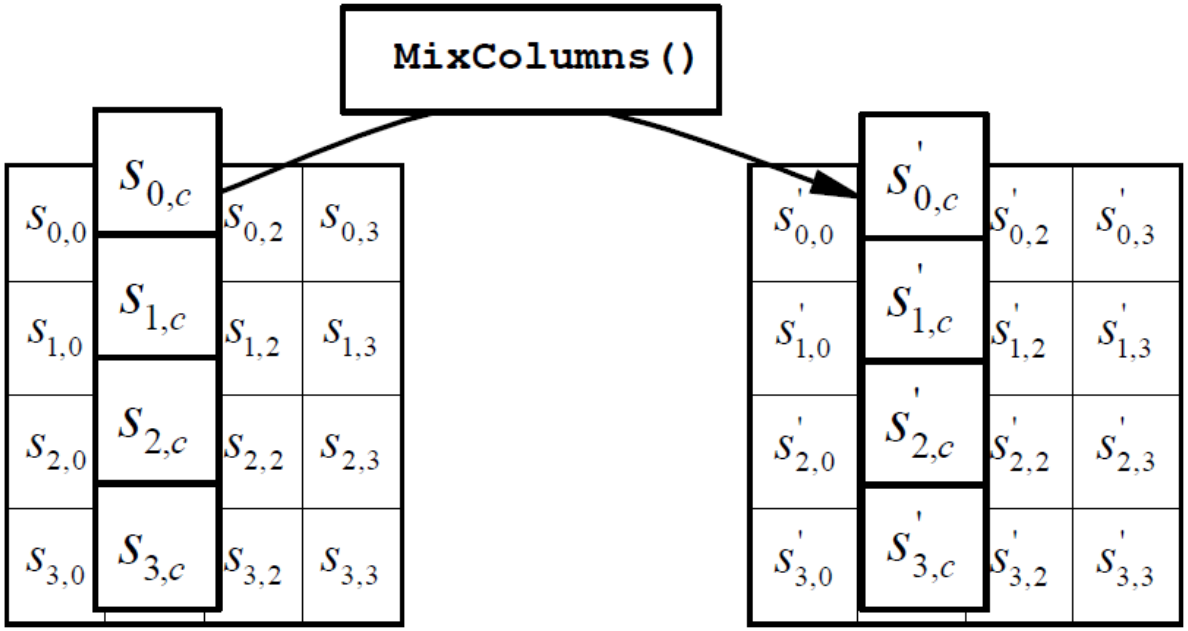
AES-Details

ShiftRows



AES-Details

- MixColumn



$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

AES-Details

- KeyExpansion

128 bit→扩展密钥量(words?)

$$\text{RotWord}(B_0, B_1, B_2, B_3) = (B_1, B_2, B_3, B_0)$$

$$\text{SubWord}\ (B_0, B_1, B_2, B_3) = (B_0', B_1', B_2', B_3')$$

By AES S Box

算法 3.6 KeyExpansion(key)

external RotWord, SubWord

RCon[1] ← 01000000

RCon [2] ← 02000000

RCon [3] ← 04000000

RCon [4] ← 08000000

RCon [5] ← 10000000

RCon [6] ← 20000000

RCon [7] ← 40000000

RCon [8] ← 80000000

RCon [9] ← 1B000000

RCon [10] ← 36000000

for $i \leftarrow 0$ **to** 3

  do $w[i] \leftarrow (\text{key}[4i], \text{key}[4i+1], \text{key}[4i+2], \text{key}[4i+3])$

for $i \leftarrow 4$ **to** 43

do
- temp ← $w[i-1]$
- **if** $i \equiv 0 \pmod 4$
  - **then** temp ← SubWord(RotWord(temp)) ⊕ RCon[$i/4$]
- $w[i] \leftarrow w[i-4] \oplus$ temp

return $(w[0], \cdots, w[43])$

AES的安全性分析

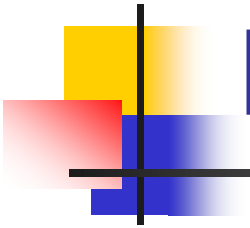
- S盒
- 线性逼近表、差分分布表趋近于均匀分布
- 因此抵抗线性攻击、差分攻击

- 难于找到包含较少活动S盒—**宽轨道策略**

- 目前对AES攻击，没有更好的方法报道

工作模式

- ECB模式 （Electronic CodeBook）
- CBC模式 （Cipher Block Chaining）
- CFB模式 (Cipher feedback)
- OFB模式 （Output feedback ）
- 计数模式 （Counter mode ）
- CCM模式 （ counter with cipher-block chaining MAC ）



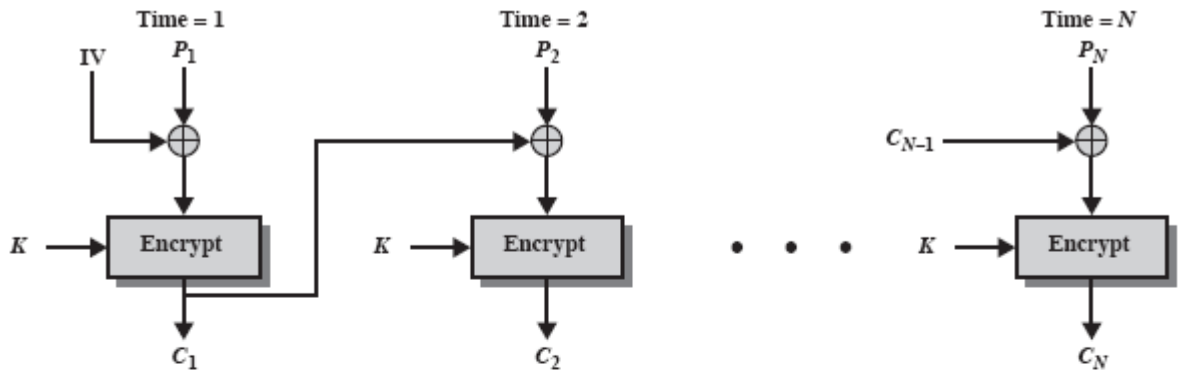
ECB模式

Electronic Code Book Mode

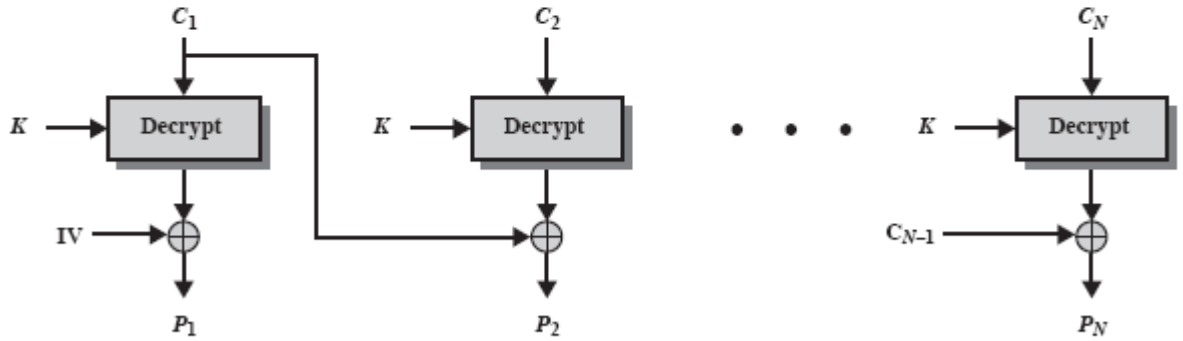
- Weakness?

- 加密相同的明文分组将会产生相同的密文分组

CBC Mode



(a) Encryption



(b) Decryption

Cipher Block Chaining (CBC) Mode

$$\text{Encryption} \quad C_0 = E_K(IV + M_0), \text{and} \quad C_i = E_K(C_{i-1} + M_i), i = 1, \cdots, N-1.$$

$$\text{Decryption} \quad M_0 = D_K(C_0) + IV, \text{and} \quad M_i = D_K(C_i) + C_{i-1}, i = 1, \cdots, N-1.$$

OFB模式

- Block Cipher→Stream
- 同步流密码：密钥流由**迭代**加密一个初始向量IV产生，$z_0 = IV$：密钥流如下：

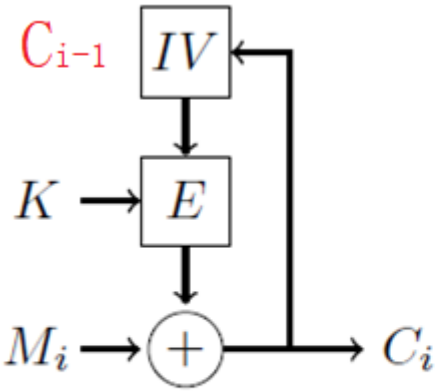
$$z_i = e_K(z_{i-1})$$

- 加密：$y_i = x_i \oplus z_i$
- 解密：$x_i = y_i \oplus z_i$

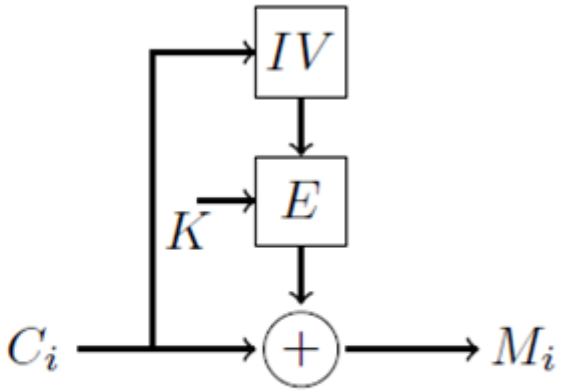
Encryption Modes-CFB

- Cipher Feedback (CFB) Mode:

$$\text{Key stream generation:} \quad K_i = E_K(C_{i-1}), i = 0, \cdots, N-1, C_0 = IV.$$
$$\text{Encryption:} \quad C_i = K_i + M_i = E_K(C_{i-1}) + M_i, i = 0, \cdots, N-1$$



CFB Encryption



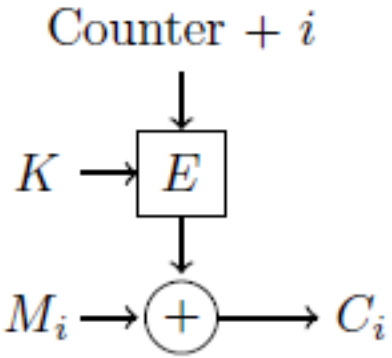
CFB Decryption

Encryption Mode-CTR

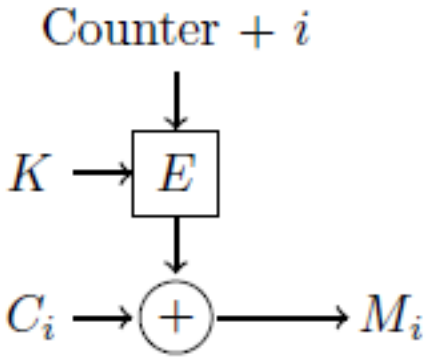
- Counter Mode（不同的构造密钥流方法）

Key stream generation: $K_i = E_K(\text{Counter} + i - 1), i = 0, \cdots, N-1$

Encryption $C_i = K_i + M_i, i = 0, \cdots, N-1$



CTR Encryption



CTR Decryption

密钥流生成，支持并行

CCM模式

- 计数模式（加密）+CBC模式(认证)

- an authenticated encryption algorithm designed to provide both *authentication* and *confidentiality*

- used in the IEEE 802.11i (as CCMP, an encryption algorithm for WPA2), IPSEC and TLS 1.2；available for TLS 1.3