第6讲

若干实用的对称密码

我们主要考察如下五种加密算法

1. Triple DES
2. RC5
3. RC6
4. Blowfish
5. IDEA

1 TRIPLE DES

- **DES算法设计的优点是很多的。DES在世界商业界的普遍使用，使得如何加强DES的安全性成为一个十分实际的问题。Quisquater 、Toms Berson等曾建议采用长达768 bits密钥的方案。由于已经证明DES不能成为群，见**

  K.W.Campbell and M.J.Wiener

  Proof that DES is not a group

  In Advances in Cryptology——Crpto'92.

  Springer——Verlag, New York,1993.

于是多重DES，尤其是三重DES还在普遍使用。

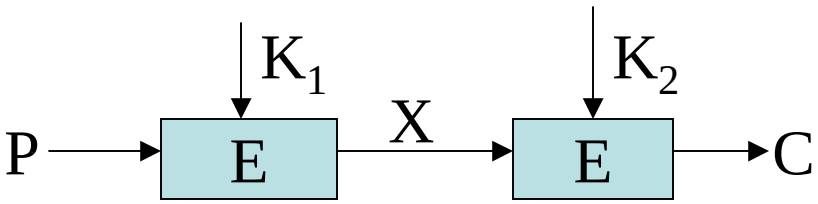
（1）二重DES(Double DES)

- **给定明文P和两个加密密钥k_1和k_2，采用二重DES对P进行加密，有**

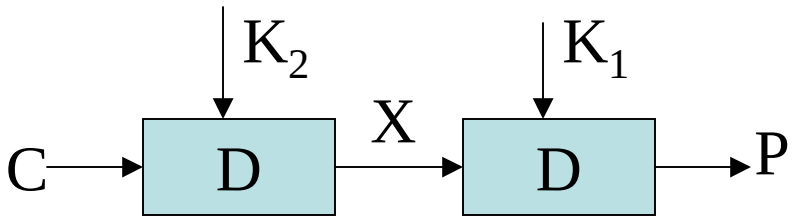
  密文　$C=E_{K2}(E_{K1}(P))$

对C进行解密D，有

  明文　$P=D_{K1}(D_{K2}(C))$



加密图



解密图

对于二重DES的加密，所用密钥的长度为

$56\times2=112$ bits

这样是否真正能增强DES的强度呢？问题在于下式能否成立：

$$E_{K2}(E_{K1}(P)) = E_{K3}(P) \qquad (4.1)$$

DES是一个从集合A到集合A的一个映射。其中：

$A=\{(a_1,a_2,a_3,\ldots\ldots,a_{64}) | a_i \in Z_2=\{0,1\}\}$，　　$|A|=2^{64}$

映射DES事实上可视为对A的一个作用，作用方式为置换。所有可能的置换数为 $(2^{64})!=10^{34738000000}$。

然而，DES对每一个不同的密钥只决定唯一的映射。而密钥数$2^{56}<10^{17}$，(4.1)式一般是不能成立的。已知DES不能构成群！

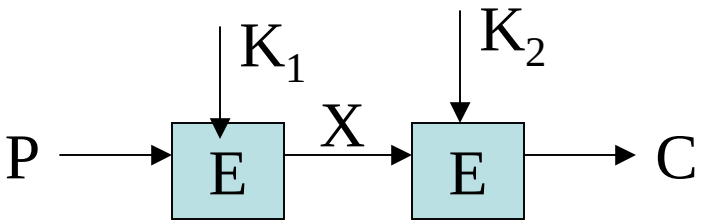
- **关于DES不是群的详细证明可以参见我们开始列出的文献。**

- 注：**二重DES很难抵挡住**中间相遇攻击法（**Meet-in-the-Middle Attack)**

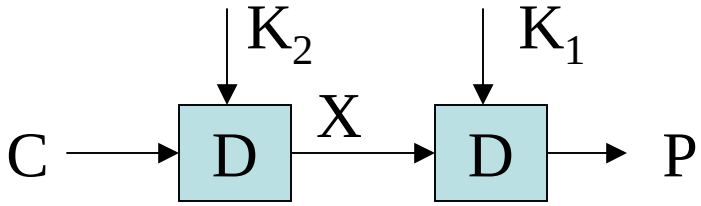
由 $C=E_{K2}(E_{K1}(P))$

从图中可见

$X=E_{K1}(P)=D_{K2}(C)$



加密



解密

若给出一个已知的明密文对（P,C)，我们对2^{56}个所有密钥K_1，分别利用DES对明文P加密，得到一张密钥对应于密文X的一张表；类似地对2^{56}个所有可能的密钥K_2，分别利用DES对密文C解密，得到相应的“明文”X。做成一张X与K_2的对应表。比较两个表就会得到真正使用的密钥对K_1,K_2。

对二重DES的中间相遇攻击的分析

- 已知，给定一个明文P，经二重DES加密有2^{64}个可能的密文。而二重DES所用密钥的长度应是112 bits，所以选择密钥有2^{112}个可能性。于是对给定明文P加密成密文 C,有$2^{112}/2^{64}=2^{48}$种可能 的密钥被选用。于是，对确定的明密文对（P、C），密文不符的大约有2^{48} 个 ，这个数字也对应于中间不相符的密文；然而中间密文的样本空间有2^{64} 个样本，于是中间不相遇的概率为 $2^{48-64}=2^{-16}$。这样，对已知明文-密文对的中间相遇攻击成功的概率为$1-2^{-16}$。
- 攻击用的代价不大于$2^{56} + 2^{56}$ ，也就是数量级为2^{56} 这和单次 DES攻击代价2^{55} 基本差别不大。

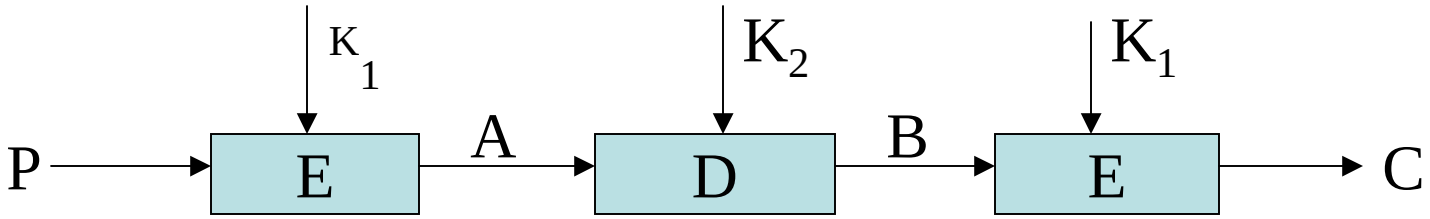
（2）带有双密钥的三重DES
（Triple DES with Two Keys)

- **Tuchman给出双密钥的EDE模式（加密-解密-加密）：**

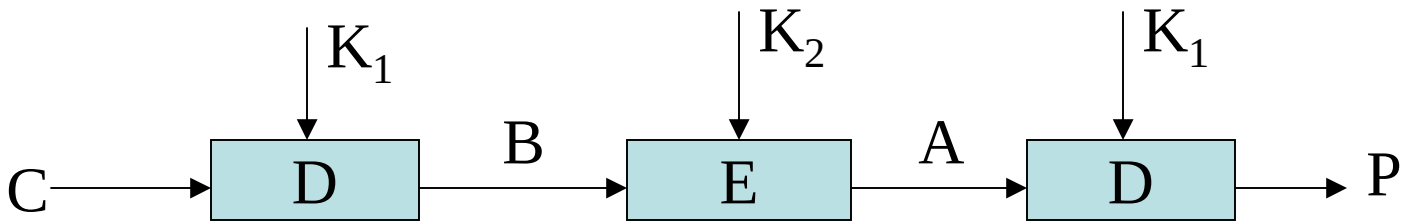
  $C=E_{K1}(D_{K2}(E_{K1}(P)))$ ……对P加密

  $P=D_{K1}(E_{K2}(D_{K1}(C)))$ ……对C解密

  这种替代DES的加密较为流行并且已被采纳用于密钥管理标准（The Key Manager Standards ANSX9.17和ISO8732).



加密图



解密图

对双密钥的三重DES的分析

- 该模式由IBM设计, 可与常规加密算法兼容
- 这种替代DES的加密较为流行并且已被采纳用于密钥管理标准（The Key Manager Standards ANSX9.17和ISO8732).
- 交替使用K_1和K_2可以抵抗中间相遇攻击.如果$C=E_{K2}(E_{K1}(E_{K1}(P)))$,只需要2^{56+2}次加密
- 到目前为止，还没有人给出攻击三重DES的有效方法。对其密钥空间中密钥进行蛮干搜索，那么由于空间太大为$2^{112}=5\times10^{33}$，这实际上是不可行的。若用差分攻击的方法，相对于单一DES来说复杂性以指数形式增长，要超过10^{52}。

- 目前还没有针对两个密钥三重DES的实用攻击方法。但对两个密钥三重DES的攻击有一些设想，以这些设想为基础将来可能设计出更成功的攻击技术。
- **Merkle R. and Hellman,M. “On the security of multiple encryption”. Communication of the ACM, July 1981**
- **Oorschot ,P and Wiener, M. “A Known-plaintext attack on two-key triple encryption” Proceedings, EUROCrypt’90,1990: published by Springer-Verlag**

虽然对上述带双密钥的三重DES到目前为止还没有好的实际攻击办法，但人们还是放心不下，又建议使用三密钥的三重DES，此时密钥总长为**168bits.**

$$C=E_{K3}(D_{K2}(E_{K1}(P)))$$

三密钥的三重DES

- 密钥的有效长度为168位
- 与DES的兼容性可以通过令$K_3=K_2$或$K_1=K_2$得到
- 许多基于Internet的应用里用到：PGP和S/MIME

Triple-DES的四种模型

- **DES-EEE3：三个不同密钥，顺序使用三次加密算法**
- **DES-EDE3：三个不同密钥，依次使用加密-解密-加密算法**
- **DES-EEE2：K1=K3，同上**
- **DES-EDE2：K1=K3，同上**

2 RC5

Ron Rivest 1994设计、1995公开

RC5具有如下的特性：

- **1. 适用于软件或者硬件实现**
- **2. 运算速度快**
- **3.** 能适应于不同字长的程序（**一个字的bit数是RC5的一个参数；不同字长派生出相异的算法**）
- **4.** 加密的轮数可变（**轮数是RC5的第二个参数，这个参数用来调整加密速度和安全性的程度**）
- **5.** 密钥长度是可变的（**密钥长度是RC5的第三个参数**）
- **6.** RC5形式简单，**易于实现，加密强度可调节**
- **7.** 对记忆度要求不高（**使RC5可用于类似Smart Card这类的对记忆度有限定的器件**）
- **8.** 高保密性（**适当选择好参数**）
- **9.** 对数据实行bit循环移位（**增强抗攻击能力**）

对RC5的系统描述：

（1）RC5的参数

RC5实际上是由三个参数决定的一族加密算法。

参数	定义	允许值
w	**字的bit数大小。RC5加密的基本单位为2个字块**	**16,32,64**
r	**轮数**	**0,1, …,255**
b	**密钥字节的长度（8-bit bytes)**	**0,1, …,255**

- **RC5加密明文块的长度为32，64，128 bits。并且对应同样长度的密文。密钥长度为从0到2040 bits。一个特定的RC5表示为**

  RC5-w/r/b

Rivest建议使用的标准RC5为

RC5-32/12/16

（明文分组长度64，加密轮数12，密钥长度128 bits)

（3） RC5的加密

整个加密使用了下述3个基本运算和它们的逆运算：

- **模2^w加法运算，表示为“+”；**
- **逐比特异或运算，表示为“⊕”；**
- **字的循环左移运算：字x循环左移y比特，表示为**

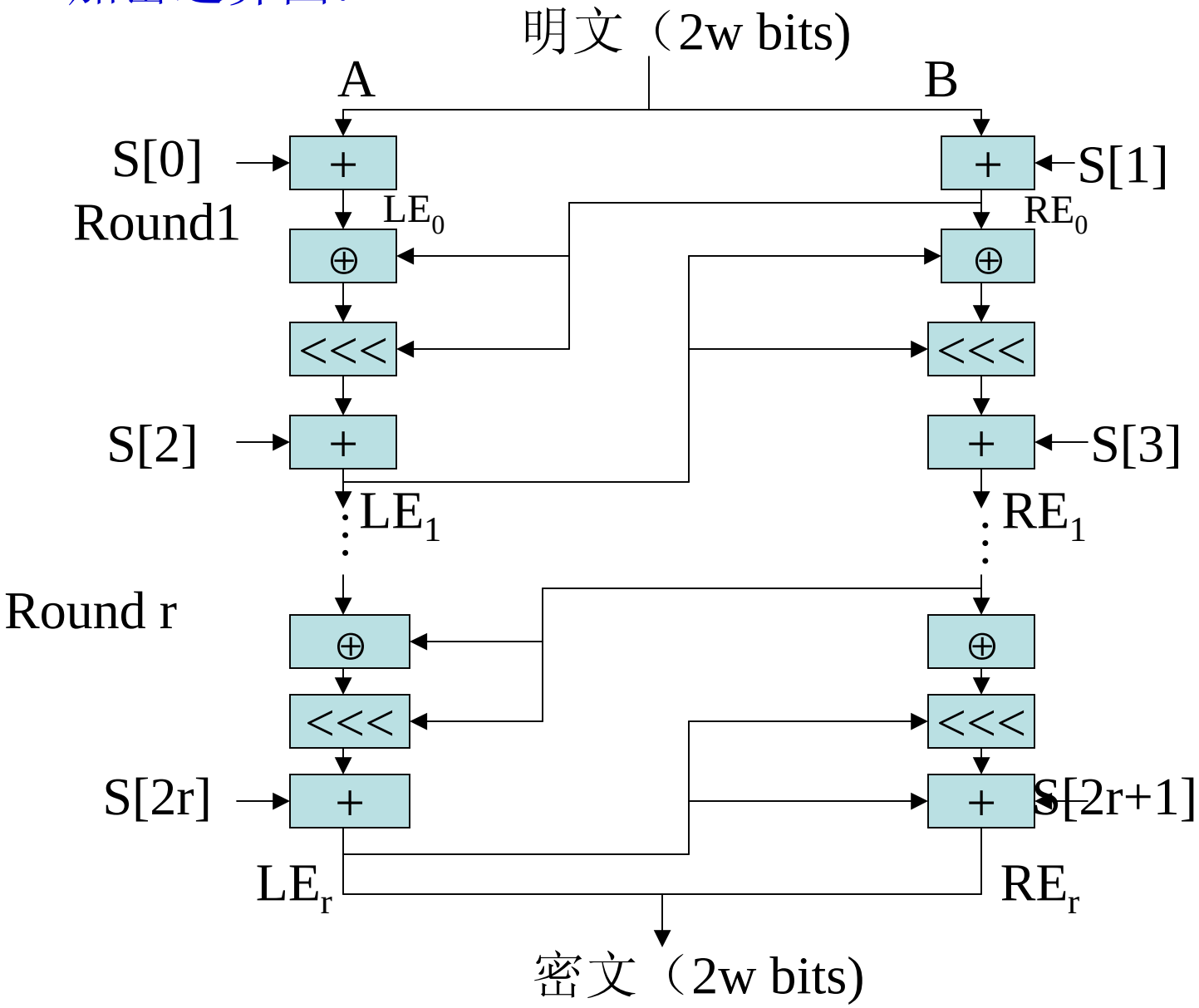
$x<<<y$

它的逆为循环右移y比特，表示为

$x>>>y$

如$(a_0,a_1,a_2,\ldots,a_{n-1})<<<3=(a_3,a_4,\ldots,a_{n-1},a_0,a_1,a_2)$

加密运算图：


明文（2w bits)
A
B
S[0]
+
+
S[1]
Round1
LE0
RE0
⊕
⊕
<<<
<<<
S[2]
+
+
S[3]
LE1
RE1
Round r
⊕
⊕
<<<
<<<
S[2r]
+
+
S[2r+1]
LEr
REr
密文（2w bits)

- 将明文分组为左右A,B；用变量LE_i，RE_i参与
- 运算程序为：

$LE_0=A+S[0]$

$RE_0=B+S[1]$

for i=1 to r do

$LE_i=((LE_{i-1} \oplus RE_{i-1})<<<RE_{i-1})+S[2\times i];$

$RE_i=((RE_{i-1} \oplus LE_i)<<<LE_i)+S[2\times i+1];$

对RC5的攻击请看：

http://grampus.jaist.ac.jp:8080/miyaji-lab/index.html

（4） RC5的解密

对两个1-字变量LD_r和RD_r。用变量LD_i和RD_i从r到1做：

for i=r down to 1 do

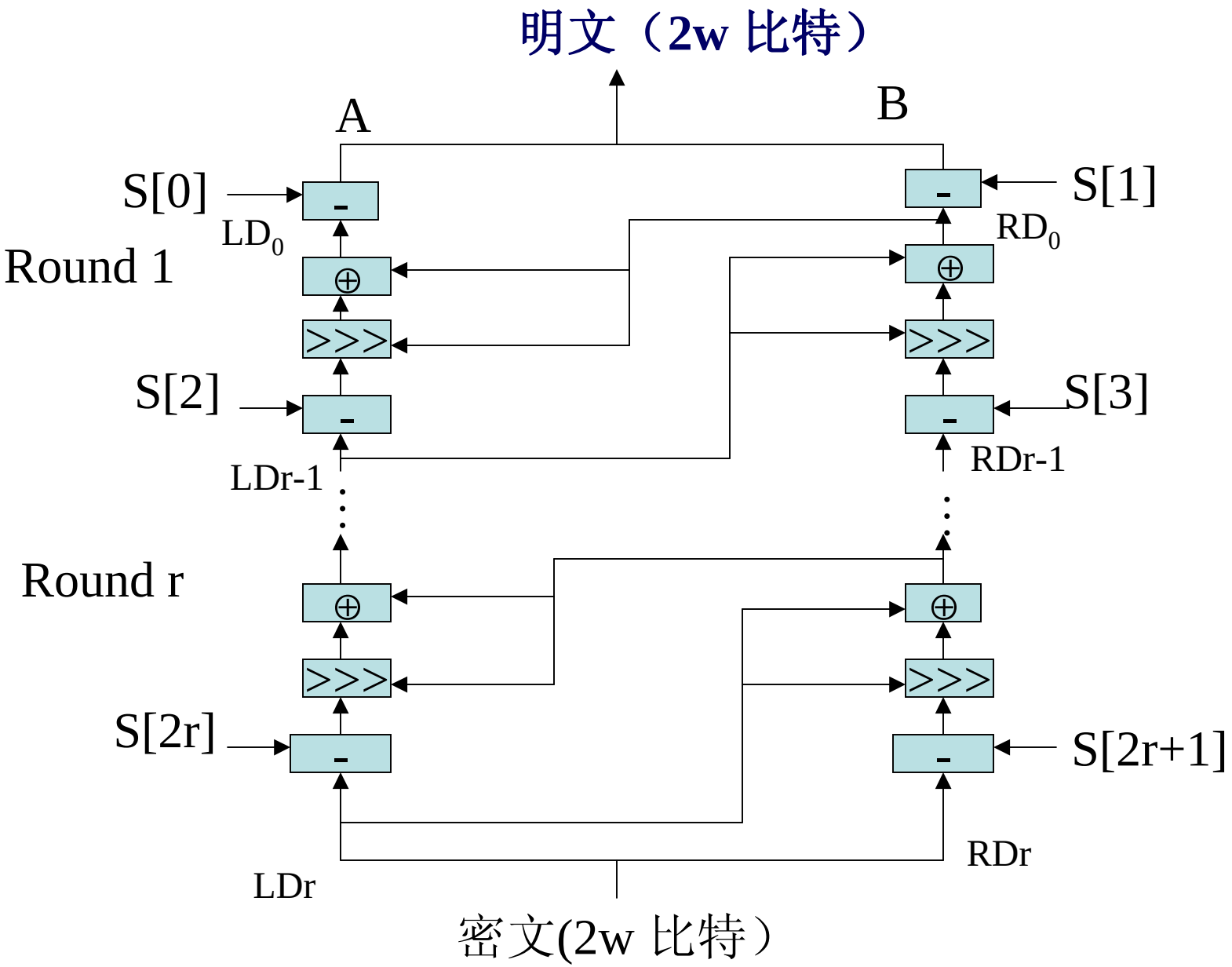
$RD_{i-1}=((RD_i-S[2*i+1]>>>LD_i) \oplus LD_i)$;

$LD_{i-1}=((LD_i-S[2*i]>>>RD_{i-1}) \oplus RD_{i-1})$;

$B=RD_0-S[1]$;

$A=LD_0-S[0]$.

注：请见下页的RC5 解密图！


明文（2w 比特）
A
B
S[0]
S[1]
LD0
RD0
Round 1
S[2]
S[3]
LDr-1
RDr-1
Round r
S[2r]
S[2r+1]
LDr
RDr
密文(2w 比特）

3 RC6分组密码简介

- **企图入选为21世纪加密标准算法AES（没运气！！）。RC6是RC5的进一步改进。像RC5那样，RC6实际上是利用数据的循环移位。**
- **RC5自1995年公布以来，尽管至今为止还没有发现实际攻击的有效手段，然而一些理论攻击的文章先后也分析出RC5的一些弱点。**
- **RC6的加密程序：RC6-w/r/b**

Input： 明文存入四个w-bit寄存器A,B,C,D

轮数r

w-bit轮密钥S[0,1, …,2r+3]

Output： 密文存入寄存器A,B,C,D

Procedure:

B=B+S[0]

D=D+S[1]

for i=1 to r do

{

t=(B×(2B+1))<<<$\log_2 w$

u=(D×(2D+1))<<<$\log_2 w$

A=((A⊕t)<<<u)+S[2i]

C=((C ⊕u)<<<t)+S[2i+1]

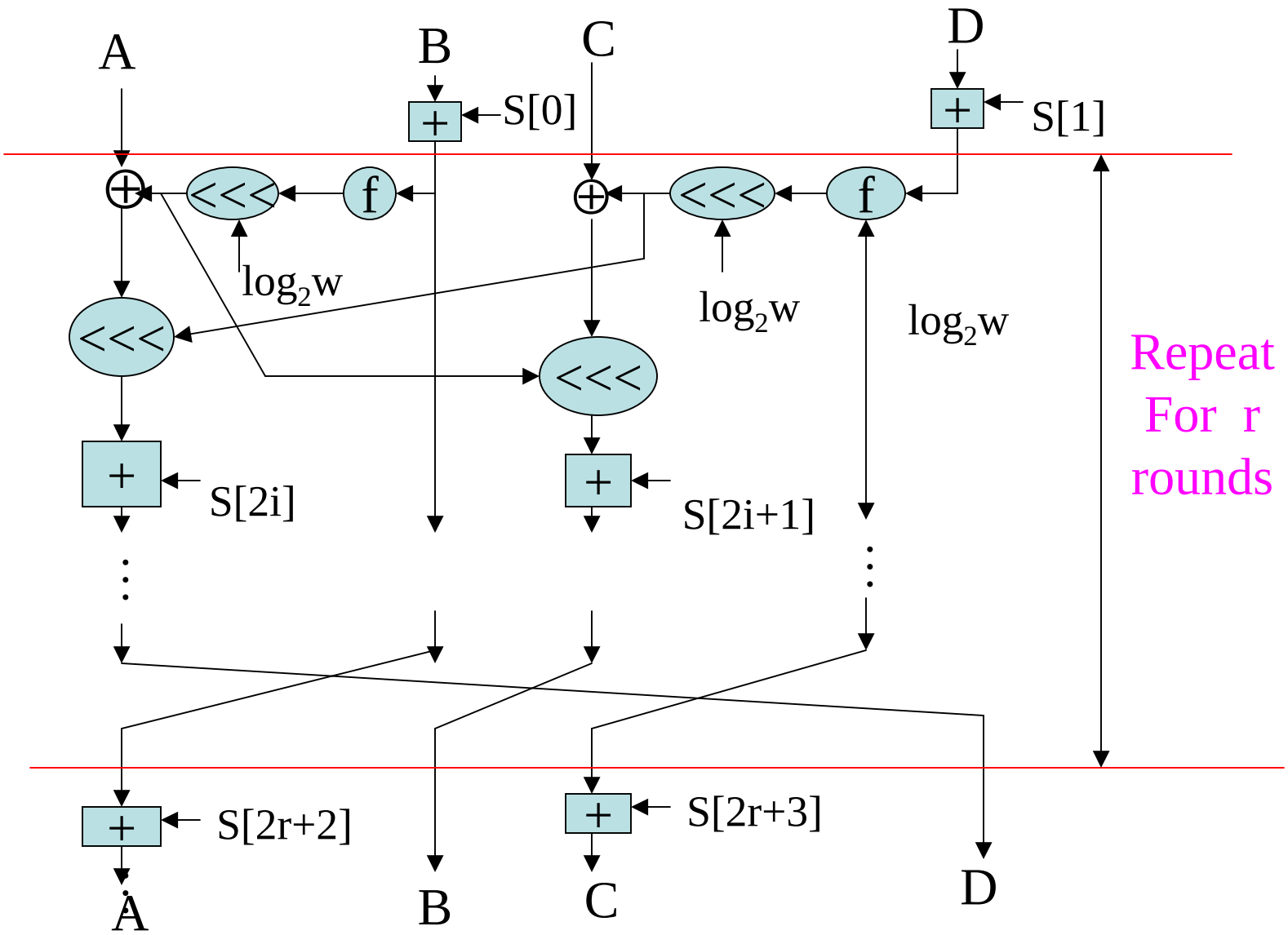
(A,B,C,D)=(B,C,D,A) ……右边寄存器到左边寄存器的并行分配

}

A=A+S[2r+2]

C=C+S[2r+3]

RC6-w/r/b加密图，其中f(x)=x× (2x+1):



RC6 Decryption (for AES)

```
C = C - S[ 43 ]
A = A - S[ 42 ]
for i = 20 downto 1 do
   {
      (A, B, C, D) = (D, A, B, C)
      u = ( D x ( 2D + 1 ) ) <<< 5
      t = ( B x ( 2B + 1 ) ) <<< 5
      C = ( ( C - S[ 2i + 1 ] ) >>> t ) Å u
      A = ( ( A - S[ 2i ] ) >>> u ) Å t
   }
D = D - S[ 1 ]
B = B - S[ 0 ]
```

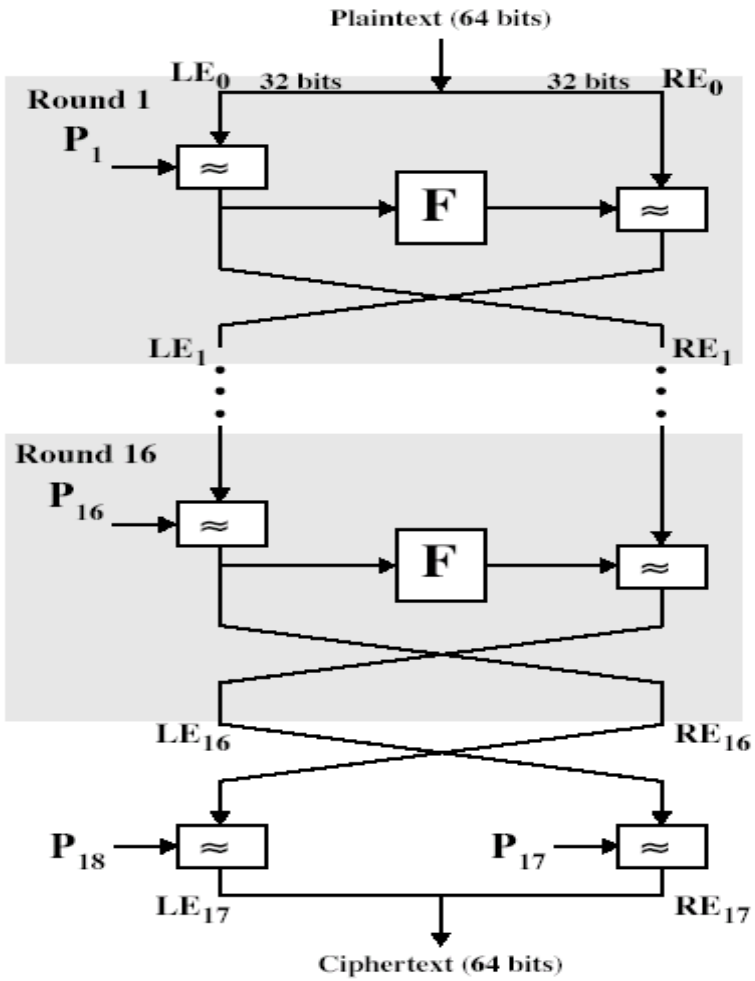
RC6小结

- **RC6 more than meets the requirements for the AES; it is**
  - **simple,**
  - **fast, and**
  - **secure.**

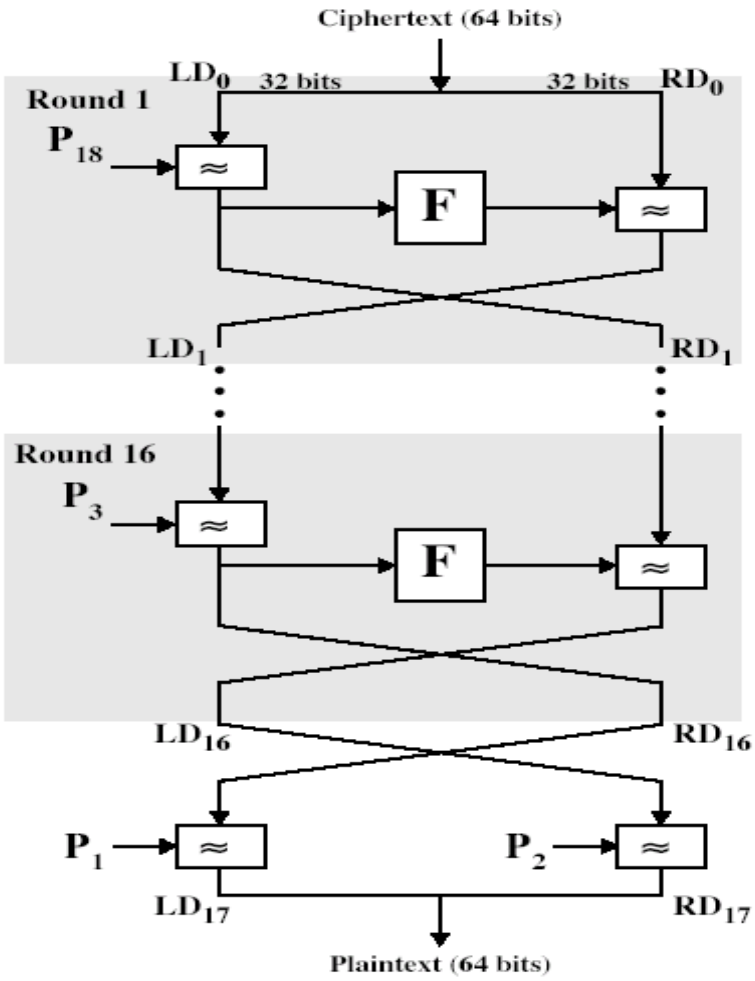
4　Blowfish算法

- 作者为Bruce Schneier[93]
- BLOWFISH算法特点
  - 采用了Feistel结构，16轮
  - 快速：18时钟周期一个字节
  - 紧凑：消耗不到5k内存
  - 简单：结构简单，易于实现和判定算法强度
  - 安全性可变：通过选择不同的密钥长度选择不同的安全级别。从32位到32*14=448位不等
  - 子密钥产生过程复杂，一次性

Blowfish算法的加密与解密

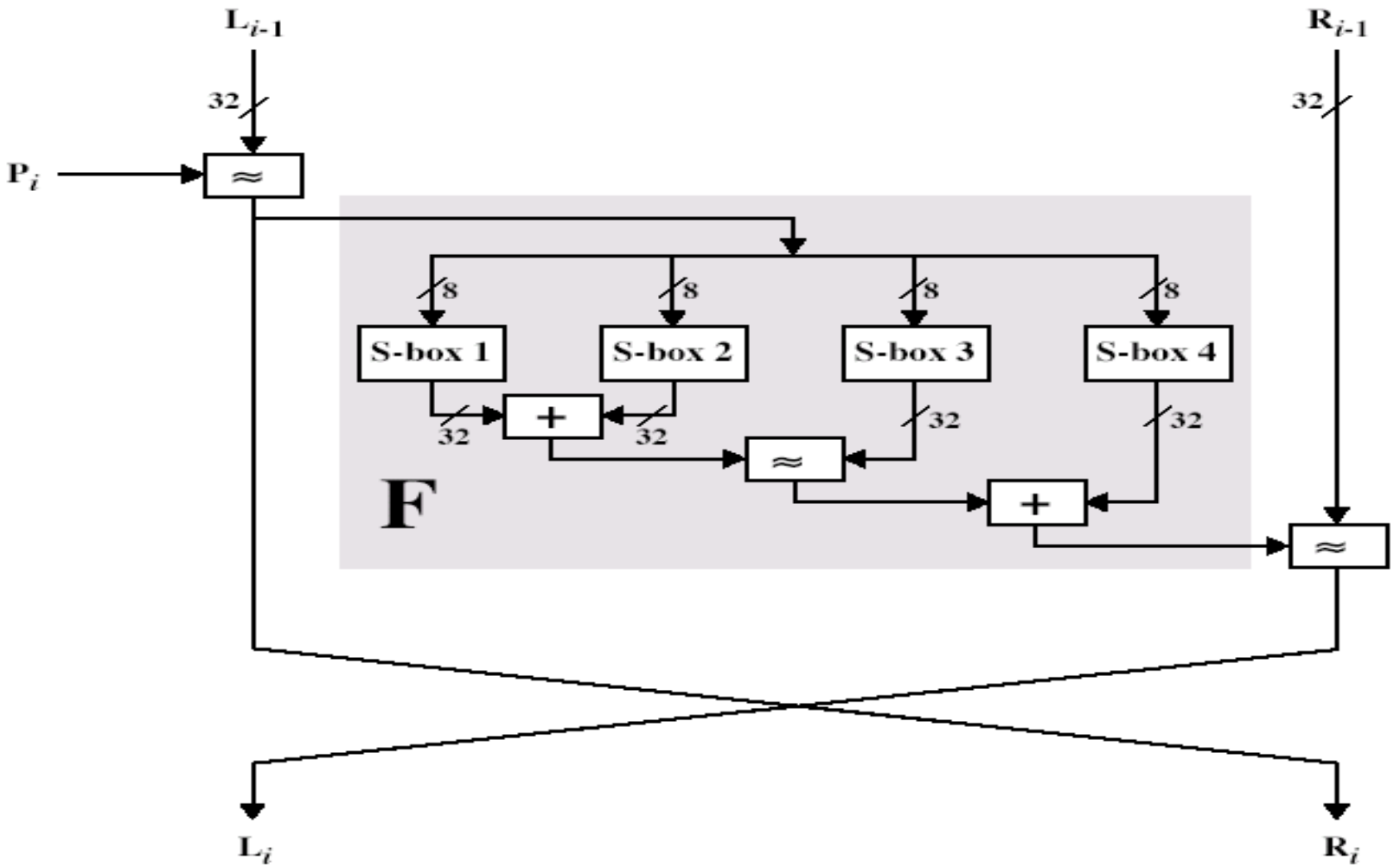


(a) Encryption



(b) Decryption

Blowfish算法的一轮



Blowfish算法讨论

- 使用两个基本运算:模2^{32}加+,按位异或Å
- BLOWFISH算法可能是最难攻破的传统加密算法，因为S-BOX密钥相关
- 算法本身的特点
  - 由于子密钥和S-BOX产生需要执行521个BLOWFISH加密算法，所以不适合于密钥频繁变化的应用场合
  - 子密钥和S-BOX产生可以保存起来
- 与Feistel分组密钥算法不同，每一步的两个部分都参与运算，不是简单的传递
- 密钥变长带来灵活性
- 速度快，在同类算法中相比较是最快的

5 国际数据加密标准IDEA（International Data Encryption Algorithm)算法

- 1990年瑞士联邦技术学院的来学嘉和Massey提出，PES，91年修订，92公布细节
- 设计目标从两个方面考虑
  - 加密强度
  - 易实现性
- 强化了抗差分分析的能力， PGP

IDEA算法特点

- 64位分组,128位密钥
- 运算: XOR Å,模2^{16}(65536)加 + ,模 (2^{16}+1)（65537）· 乘
- 三种运算均不满足分配律与结合律
- 有大量弱密钥
- 难以直接扩展到128位块

IDEA设计思想

- 得到confusion的途径
  - 按位异或
  - 以2^{16}(65536)为模的加法
  - 以$2^{16}+1$ (65537)为模的乘法
  - 互不满足分配律、结合律
- 得到diffusion的途径
  - 乘加(MA)结构
- 实现上的考虑
  - 软件和硬件实现上的考虑

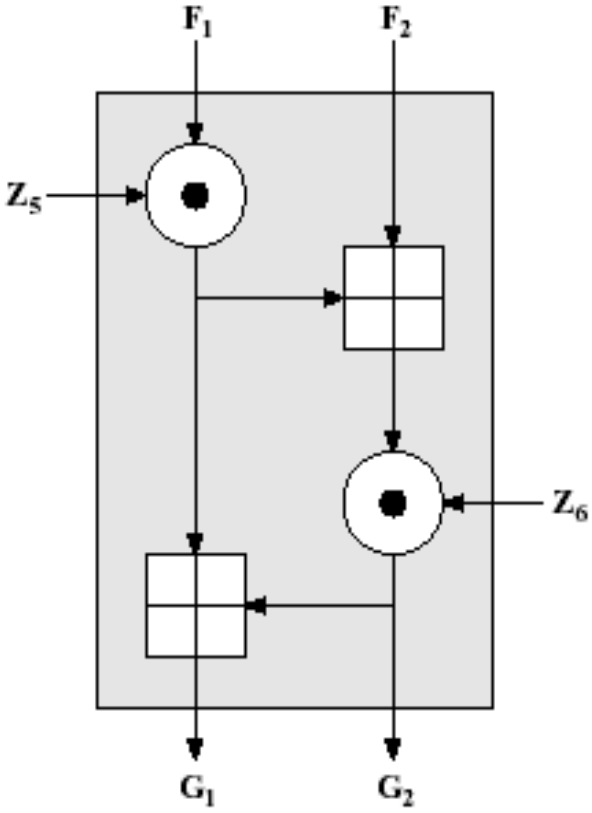


Figure 4.3 Multiplication/addition (MA) Structure

IDEA加密算法

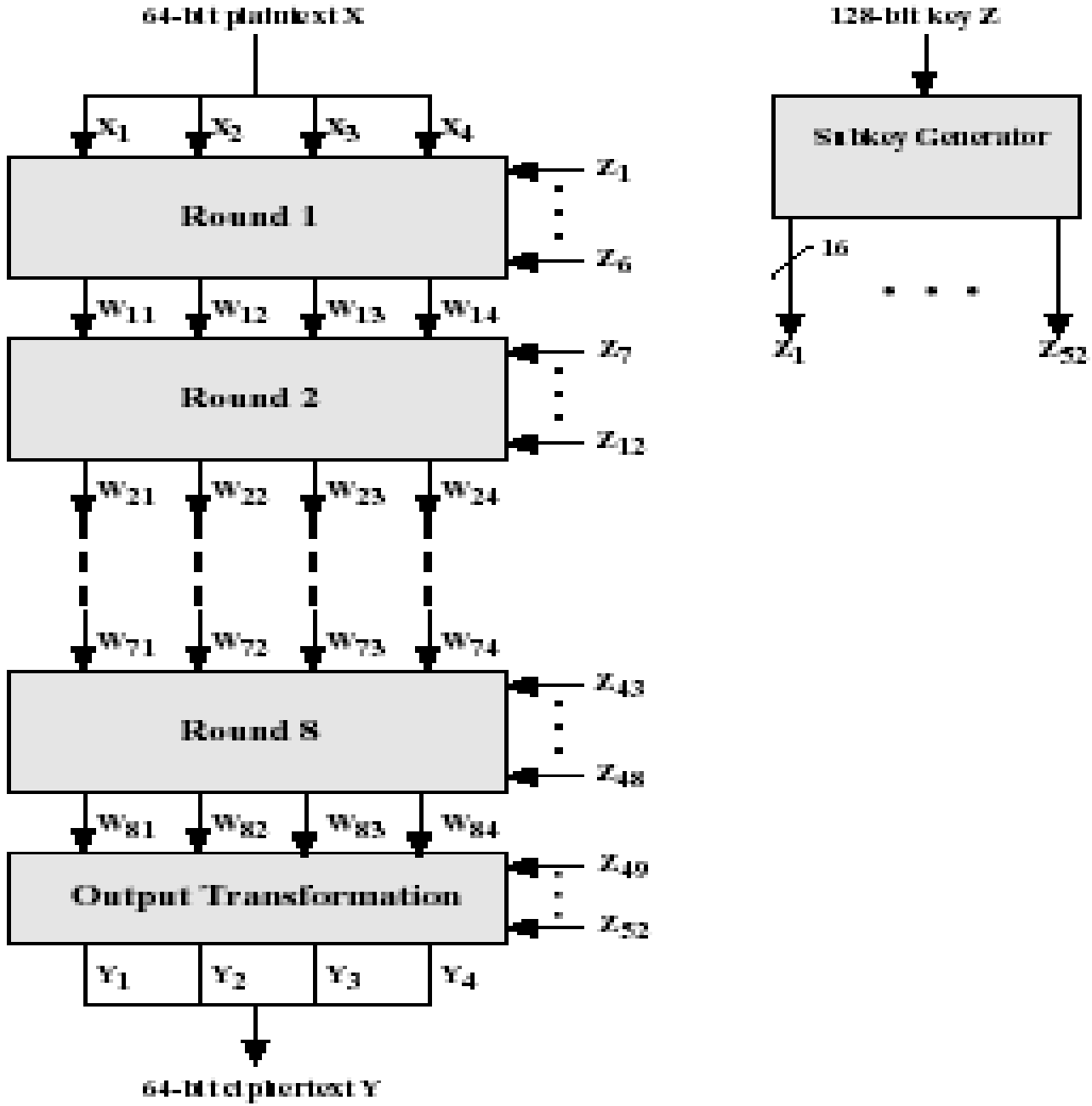


Figure 4.4 Overall IDEA Structure

IDEA
每一轮

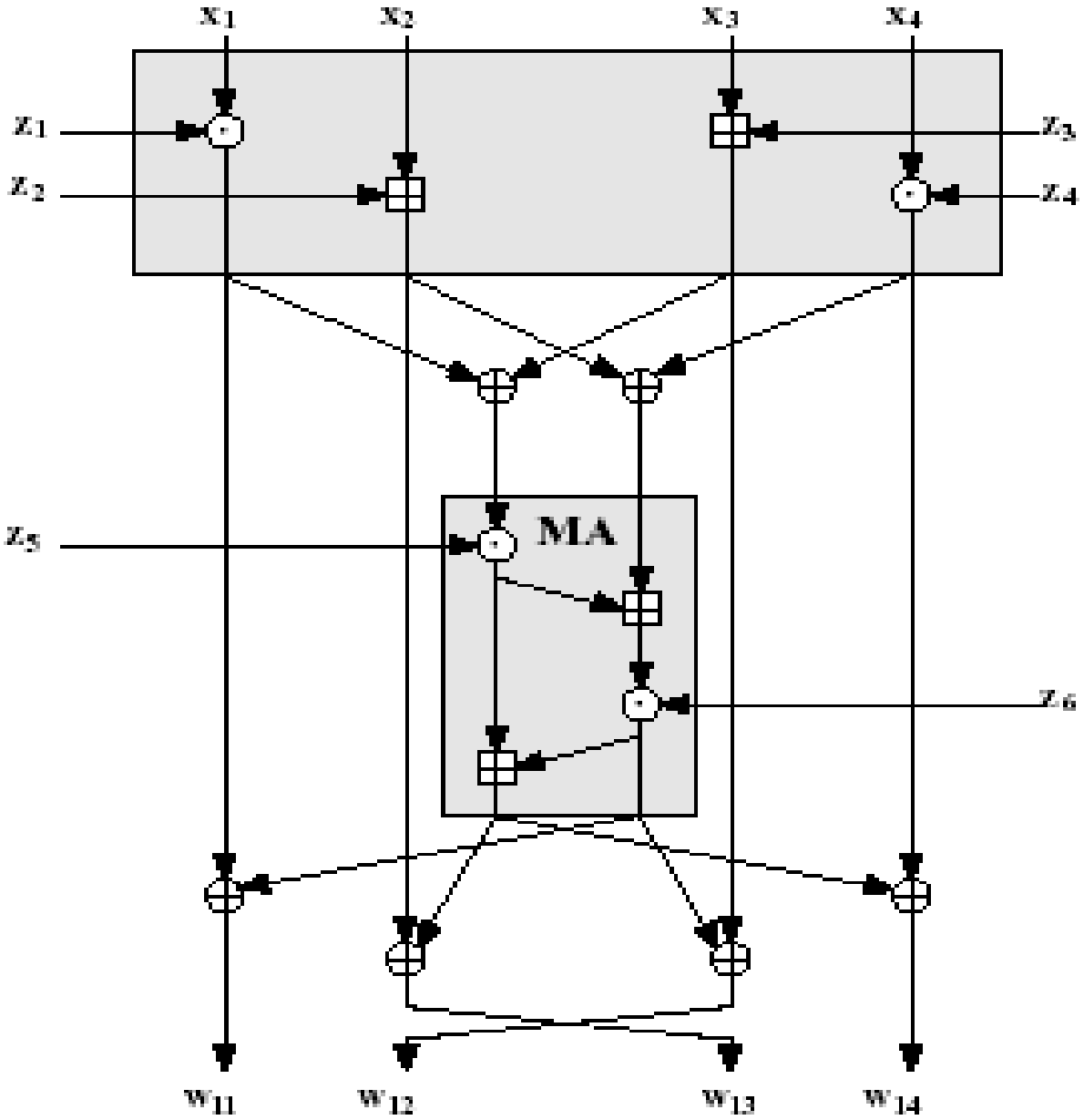
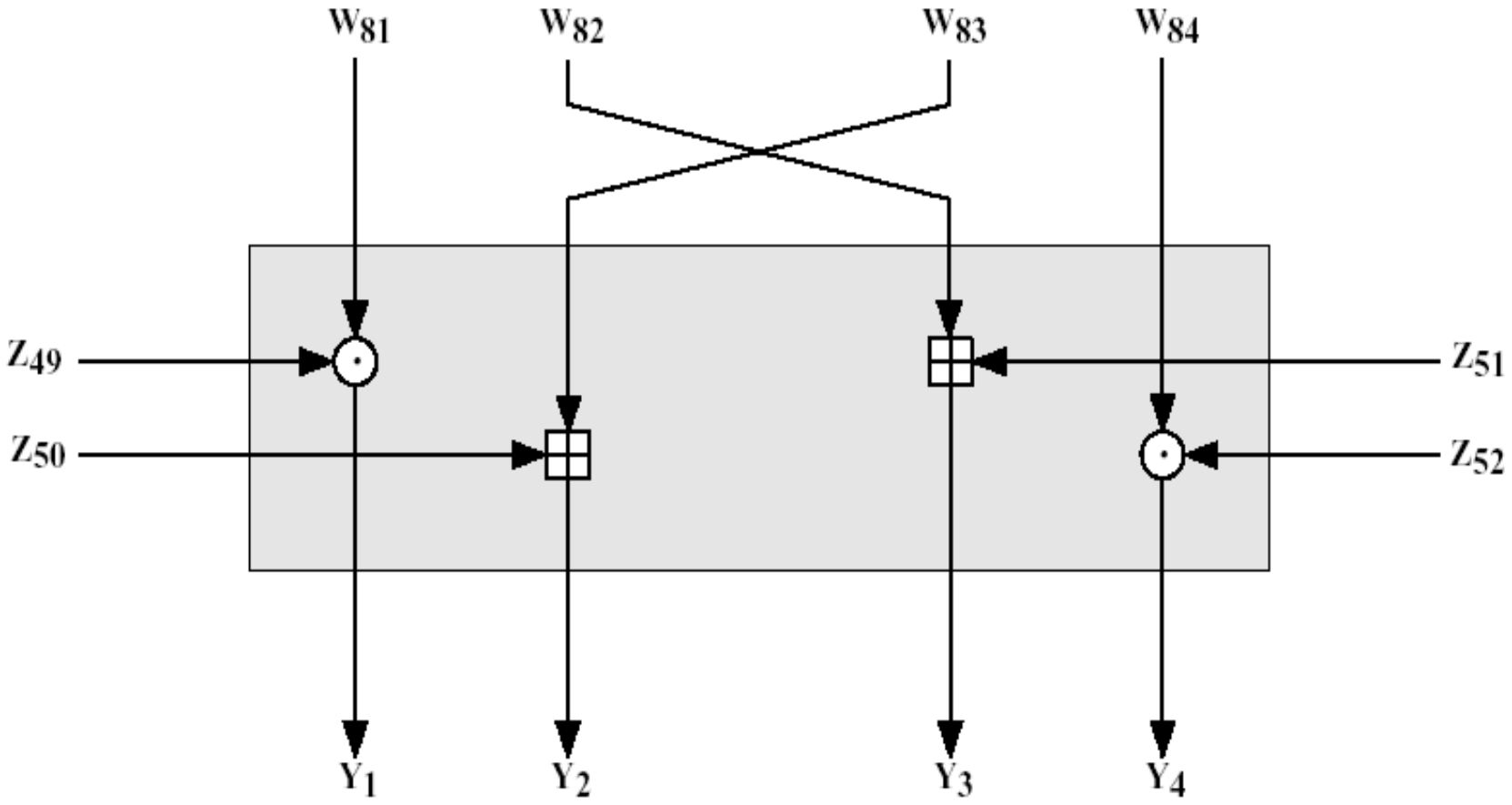


Figure 4.5 Single Round of IDEA (first round)

IDEA输出变换阶段



分组密码的工作模式

分组密码算法是提供数据安全的基本构件，为了将分组密码应用于各种各样的实际应用，NIST定义了5种“工作模式”。从本质上讲，选择工作模式是一项增强密码算法或者使算法适应具体应用的技术。

分组密码的工作模式

- 电码本（ECB）模式
- 密文分组链接（CBC）模式
- 密文反馈（CFB）模式
- 输出反馈（OFB）模式
- 计数器（CTR）模式

其它模式：级连（CM）模式、扩散密码分组链接（PCBC）模式

一、电码本模式

- 明文被分成若干等长的明文分块
- 一次处理一组明文分块，每次用相同的密钥加密
- 适用于数据较少的情况
- 缺点：没有隐藏明文消息的结构，相同的明文分组对应着相同的密文

电码本模式——加密与解密

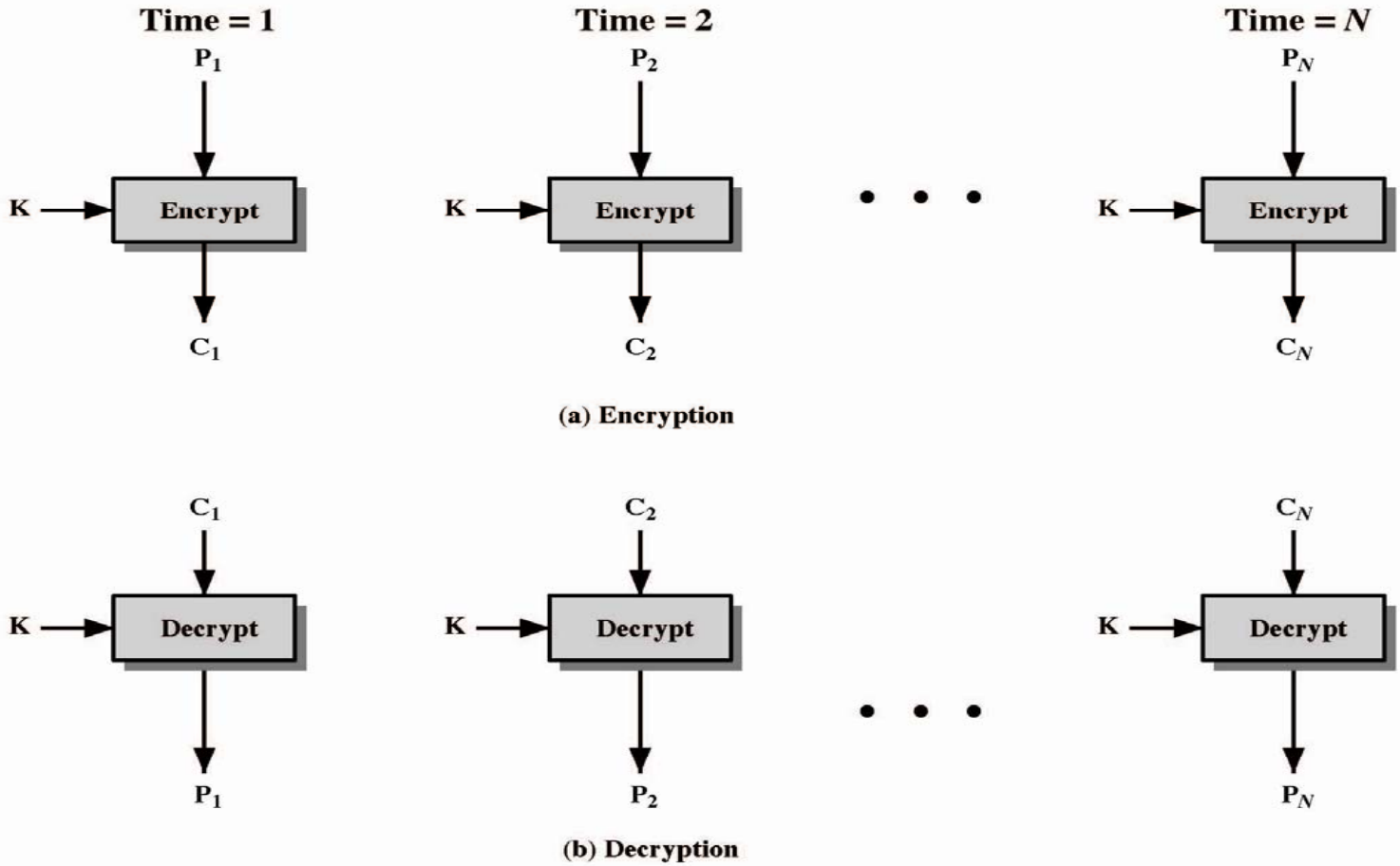


Figure 3.11 Electronic Codebook (ECB) Mode

电码本模式——数学描述

- 加密：$C_i=E_k(P_i)$
- 解密：$P_i=D_k(C_i)$

二、密文分组链接模式

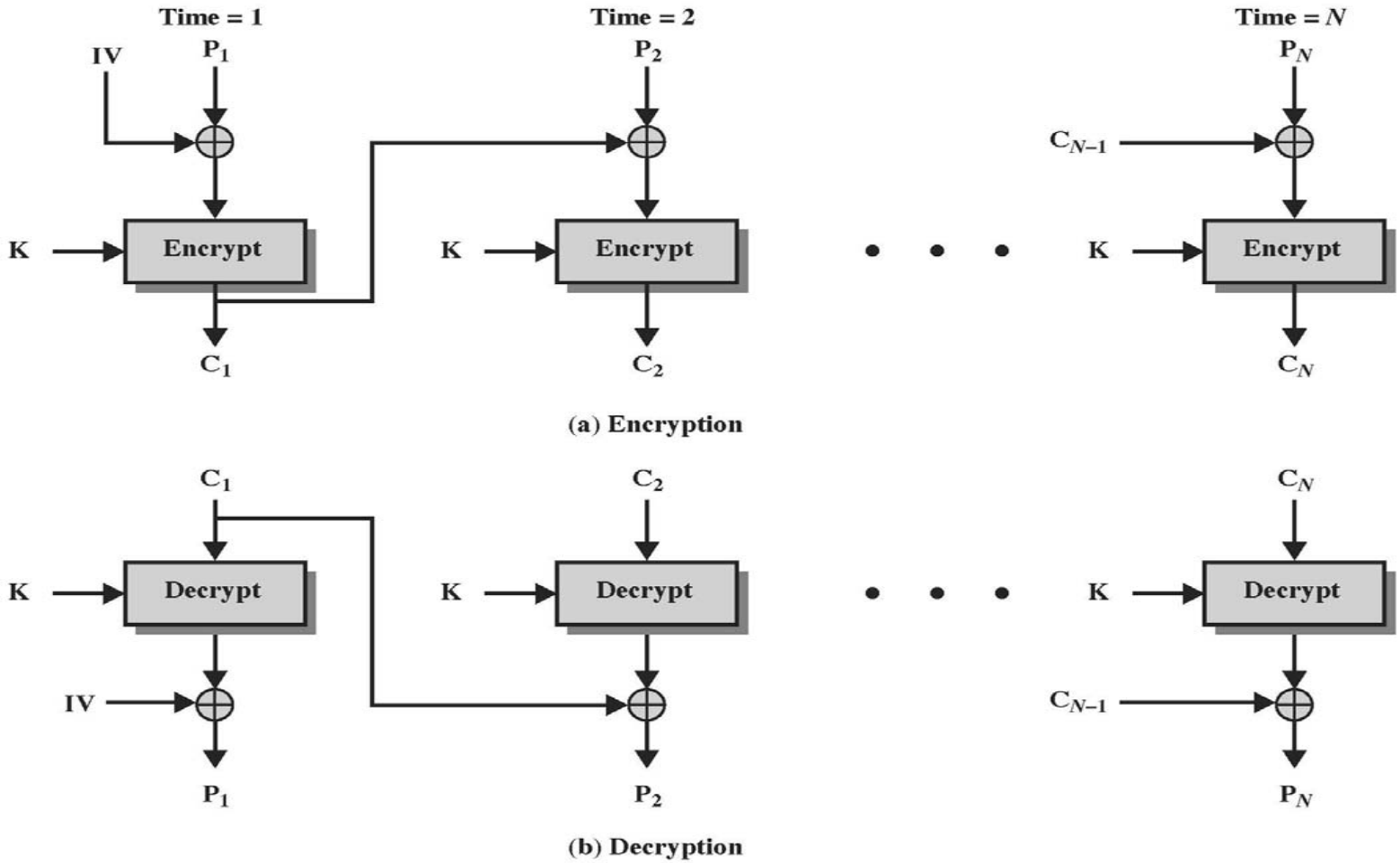


Figure 3.12 Cipher Block Chaining (CBC) Mode

密文分组链接模式

- 加密：$C_i = E_k(P_i \oplus C_{i-1})$
- 解密：$P_i = D_k(C_i) \oplus C_{i-1}$
- 密文分组C_i不仅与对应的明文分组P_i有关，也和此前所有的明文分组$P_1,\ldots,P_{i-1}$有关
- 优点：能够隐蔽明文的数据模式；能够在一定程度上防止分组的重放、插入和删除等攻击
- 缺点：易导致错误传播。任何一个明文或密文分组出错都会导致其后的密文分组出错

三、密文反馈模式

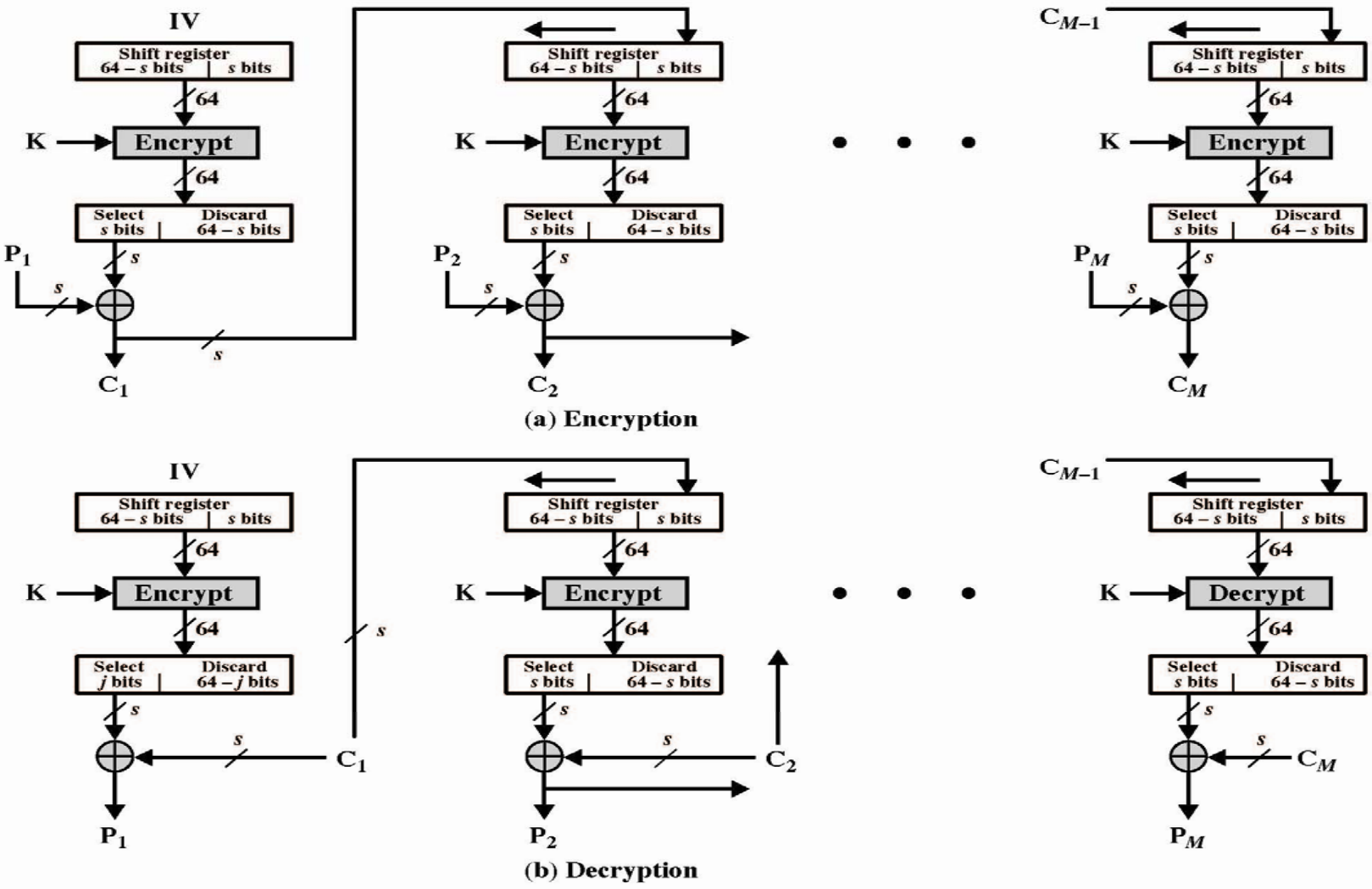


Figure 3.13 *s*-bit Cipher Feedback (CFB) Mode

密文反馈模式

- 实质上是一种自同步流密码
- 可以实现良好的安全性

四、输出反馈模式

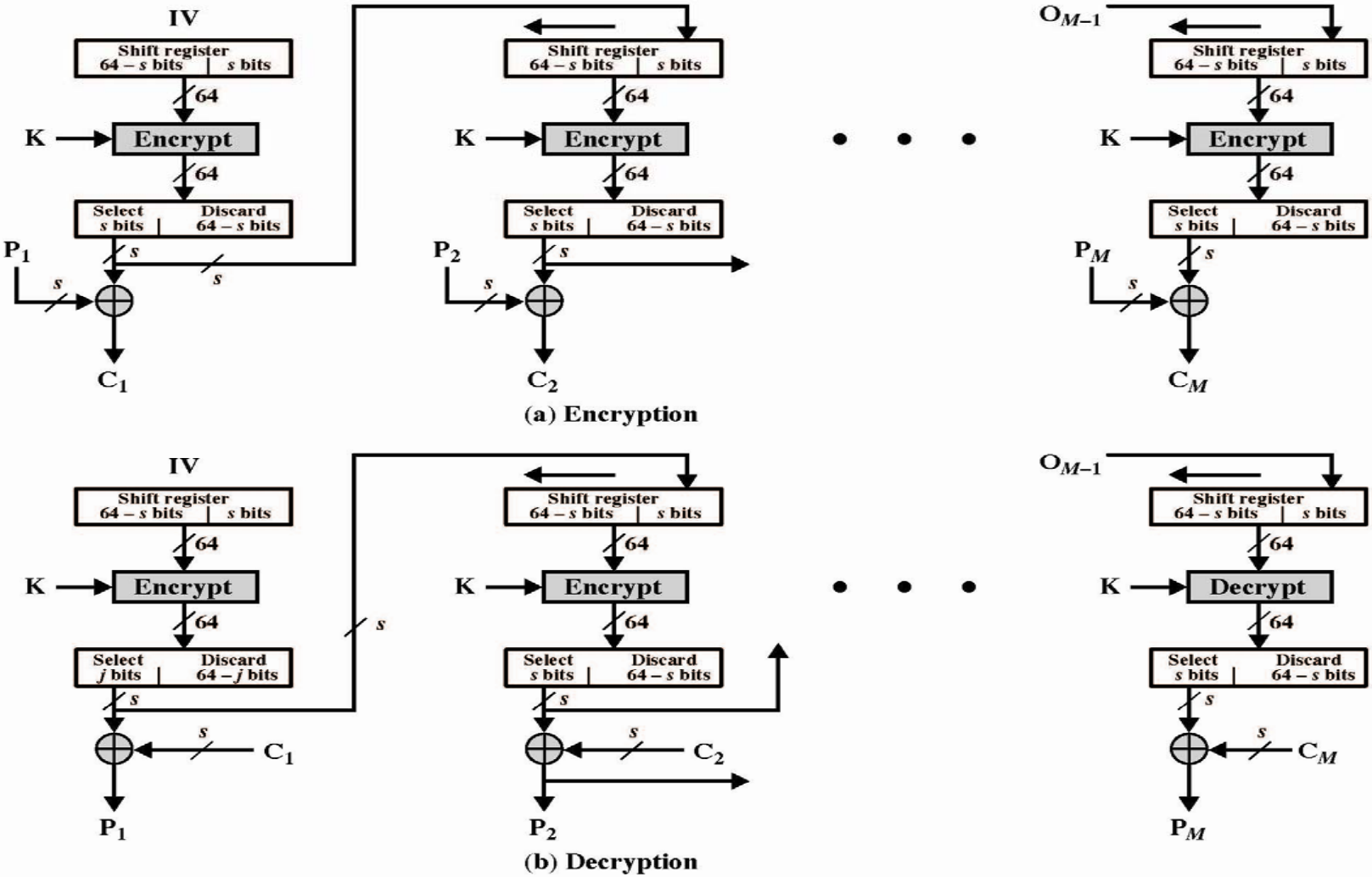


Figure 3.14 *s*-bit Output Feedback (OFB) Mode

输出反馈模式

- 优点：传输过程中在某位上发生的错误不会影响其它位
- 缺点：抗消息流篡改攻击的能力不如密文反馈模式

五、计数器模式

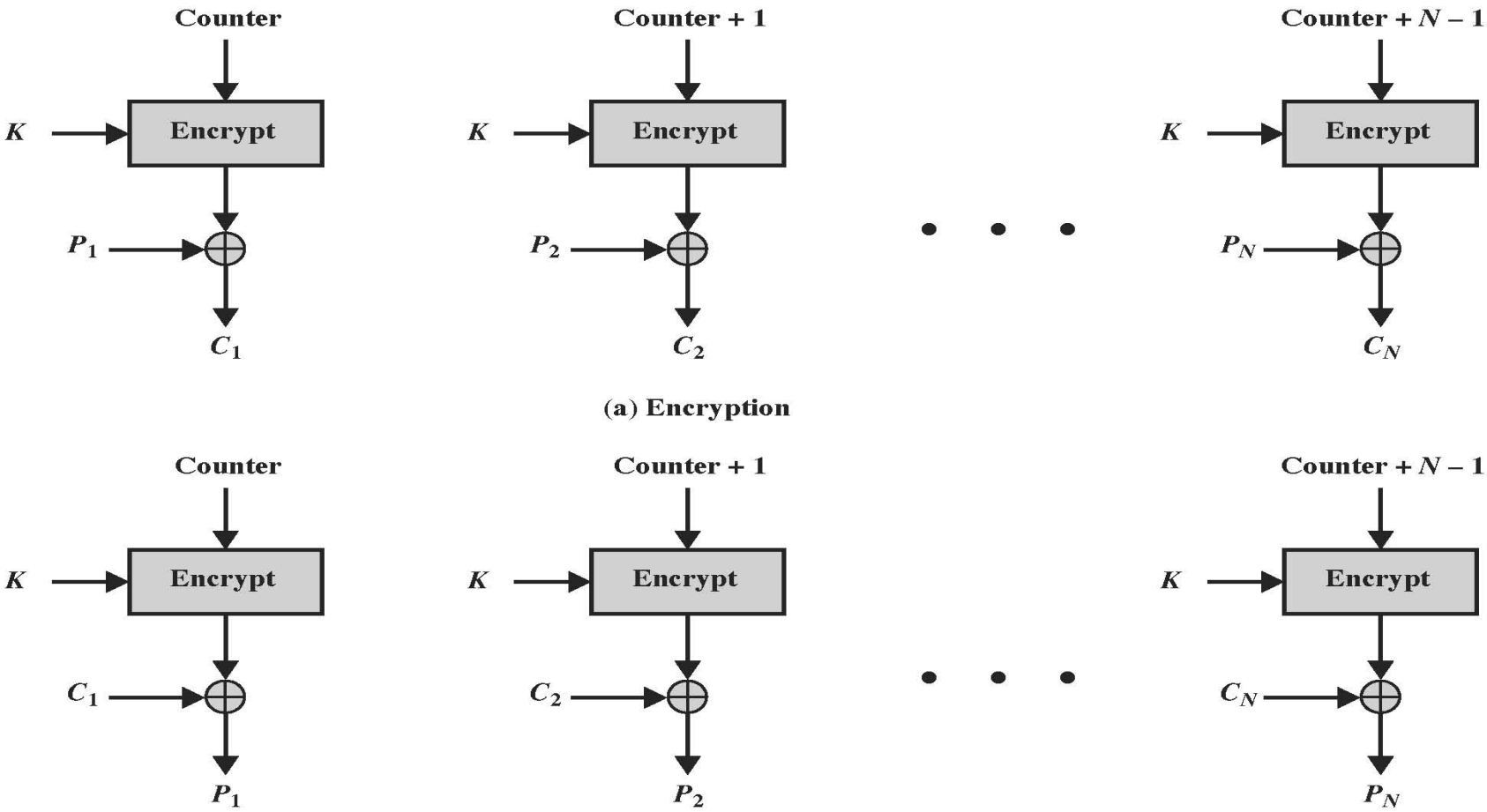


Figure 3.15 Counter (CTR) Mode

计数器模式

- 能够实现并行计算
- 能够证明计数器模式至少和之前的四种模式一样安全